Towards an automatic isogeometric analysis suitable trivariate models generation-Application to geometric parametric analysis

Hassan Al-Akhras, Thomas Elguedj, Anthony Gravouil, Michel Rochette

Accepted Manuscript

Towards an automatic Isogeometric Analysis suitable trivariate models generation—Application to geometric parametric analysis

H. Al Akhras, T. Elguedj, A. Gravouil, M. Rochette

PII: S0045-7825(16)31222-1
DOI: http://dx.doi.org/10.1016/j.cma.2016.09.030
Reference: CMA 11142

To appear in: *Comput. Methods Appl. Mech. Engrg.*

Please cite this article as: H. Al Akhras, T. Elguedj, A. Gravouil, M. Rochette, Towards an automatic Isogeometric Analysis suitable trivariate models generation—Application to geometric parametric analysis, *Comput. Methods Appl. Mech. Engrg.* (2016), http://dx.doi.org/10.1016/j.cma.2016.09.030

Towards an Automatic Isogeometric Analysis Suitable Trivariate Models Generation - Application to Geometric Parametric Analysis

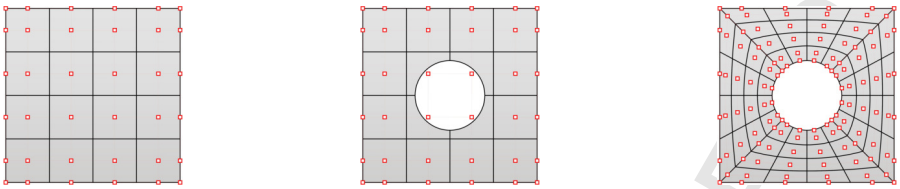
H. AL AKHRAS[a,b], T. ELGUEDJ[a], A. GRAVOUIL[a,c], M. ROCHETTE[b]

[a] *Université de Lyon, CNRS, INSA-Lyon, LaMCoS UMR5259, France*
{*hassan.al-akhras, thomas.elguedj, anthony.gravouil*}*@insa-lyon.fr*
[b] *ANSYS Research & Development, France*
{*hassan.al-akhras, michel.rochette*}*@ansys.com*
[c] *Institut Universitaire de France, France*

Abstract

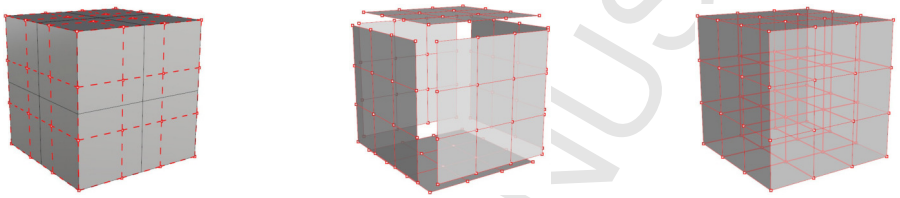
This paper presents an effective method to automatically construct trivariate tensor-product spline models of complicated geometry and arbitrary topology. Our method takes as input a solid model defined by its triangulated boundary surface. Using cuboid decomposition, an initial polycube approximating the input boundary mesh is built. This polycube serves as the parametric domain of the tensor-product spline representation required for isogeometric analysis. The polycube's nodes and arcs decompose the input model's boundary into quadrilateral patches, and these patches form hexahedral domains. Using aligned global parameterization, the nodes are re-positioned and the arcs are re-routed across the surface in a way to achieve low overall patch distortion, and alignment to principal curvature directions and sharp features. The optimization process is based on one of the main contributions of this paper: a novel way to design cross fields with topological (i.e., imposed singularities) and geometrical (i.e., imposed directions) constraints by solving only sparse linear systems. Based on the optimized polycube and parameterization, compatible B-spline boundary surfaces are reconstructed. Finally, the interior volumetric parameterization is computed using Coon's interpolation. In the context of parametric studies based on geometrical parameters, this method can be used to compute the morphing required for reduced order modeling. For different parametric instances with the same topology but different geometries, this method allows to have the same representation: i.e., meshes (or parameterizations) with the same topology. The efficiency and the robustness of the proposed approach are illustrated by several examples.

Keywords: Isogeometric Analysis, Reduced Order Modeling, Trivariate NURBS, Cuboid Decomposition, Cross Field Design, Global Parameterization



(a) (b) (c)

Figure 1: Standard CAD models: Boolean operations. A square plate with the underlying control net (a). A circular hole is cut into the plate by trimming: the underlying control net is still the same (b). Analysis-suitable representation of the plate with a hole (c).



(a) (b) (c)

Figure 2: Standard CAD models: boundary representation method. A cube with the underlying control net (a). The cube appears as a volumetric object but the underlying description is defined as a set of boundary surfaces (b). Analysis-suitable trivariate representation of the cube (c).

1. Introduction

Isogeometric Analysis (IGA), introduced by Hughes et al. [11], is a computational approach that offers the possibility of seamless integration between Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE). This method uses the same type of mathematical representation for both geometry and physical solutions, and thus avoids the costly forth and back transformations between CAD and CAE. There are several candidate technologies available to the IGA framework, of which B-Splines or Non-Uniform Rational B-Splines (NURBS) are the most commonly used since they are the standard technology employed in CAD programs.

Nevertheless, a prerequisite for IGA is the availability of analysis-suitable models. CAD modelers usually use Boolean operations that lead to trimmed models (Figure 1) and employ the Boundary Representation (B-Rep) method for volume models (Figure 2). Aiming at converting B-Rep (potentially trimmed) models into analysis-suitable volume NURBS models, this article focuses on one of the most crucial problems of the conversion process: the generation of a volume parameterization. Surface parameterization would suffice if only the surface geometry is of interest. In many cases, the surface will enclose a volume. The basic problem is to develop a volume (i.e., trivariate) parameterization in such a way that the surface (i.e., bivariate) parameterization is preserved.

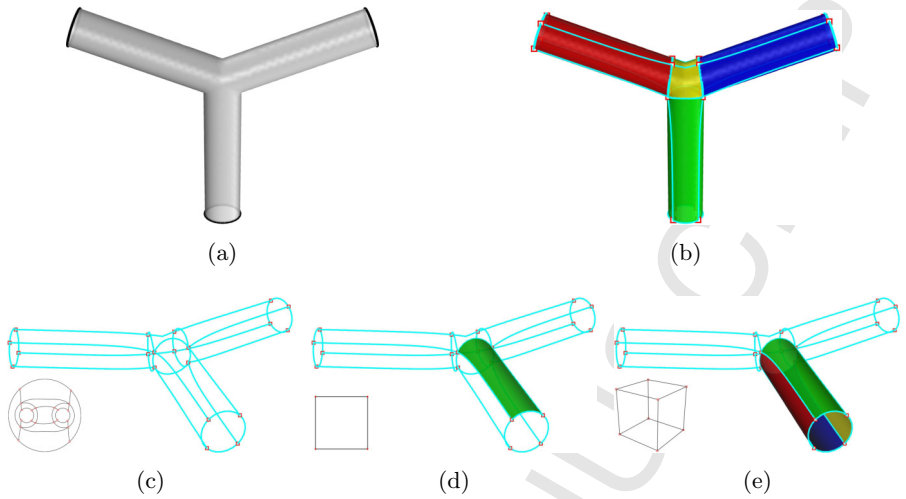


Figure 3: A pants patch (i.e., genus-0 surface with 3 boundaries) (a) and its cuboid decomposition (i.e., polycube's embedding) (b). The polycube structure is equivalent to a quadrilateral layout (c). The polycube's nodes and arcs partition the surface into quadrilateral patches (d). Each quadrilateral patch corresponds to a boundary face of the polycube (e).

A parameterization can be viewed as a mapping from a model embedded in $\mathbb{R}^3$ to a canonical domain. This canonical domain must have the same topology as the model but simplified geometrical features. Due to its tensor-product nature, NURBS fitting demands that the canonical domain (may be composed of a set of sub-domains) keeps structured. In addition, trivariate NURBS require canonical domains embedded in $\mathbb{R}^3$.

A popular shape abstraction method is to use polycubes. A polycube is a set of cube (i.e., hexahedral) sub-domains consistently glued together. The polycube can be used to approximate very roughly the geometry of a model while faithfully replicating its topology. Due to its highly regular and trivariate structure, the polycube is suitable for serving as the canonical domain of the volume parameterization required for trivariate NURBS construction. The polycube consists of nodes and arcs embedded in the model's boundary surface. These nodes and arcs partition the boundary surface into quadrilateral patches. Each patch corresponds to a face of the polycube hexahedral sub-domains (Figure 3). The model's boundary can then be mapped to cover the polycube's boundary seamlessly and consistently. This approach has the advantage of being inherently volumetric in the sense that the surface parameterization (between the model's boundary and the polycube's boundary) can be trivially turned into a volume parameterization. This is done by interpolating the boundary surfaces on each cube sub-domain. However, the quality of the resulting volume parameterization strongly depends on the quality of the polycube's embedding (i.e., the placement of the polycube's nodes and arcs on the model's boundary).

By using polycubes as canonical domains, the problem of finding a volume pa-

rameterization of a solid model is simplified to a problem of finding a surface parameterization between the model's boundary and the polycube's boundary. The isoparametric lines of this parameterization are then extracted and serve to define the quadrilateral control mesh required for tensor-product spline representations.

A recent trend is aligned global parameterizations which adapt the parameterization to the geometry of the surface by fitting its gradient to a smooth cross field interpolating reliable principal curvature directions and geometric features. This field can be seen as a kind of proxy for the parameterization. In other words, each of the required properties for a good parameterization (such as uniformity, orthogonality and singularities) can be redefined in terms of desired properties of the field. Thus the task is shifted from the definition of a surface parameterization to the design of a smooth cross field on the surface. An important property of cross fields is their singularities. A cross field singularity will generate an irregular vertex or a non-quadrilateral polygon in the control mesh.

Cross field design is the generation of a smooth cross field from a set of constraints. These constraints can be topological (i.e., imposed singularities) and/or geometrical (i.e., imposed directions). Topologically, the cross field is constrained by the polycube structure. This means that the cross field must be singular only at the position of irregular nodes of the polycube. Geometrically, the cross field must follow the geometric features of the surface.

The cross field, and hence the aligned global parameterization, are constrained by the polycube structure. Depending on the quality of the polycube's embedding in the surface, the resulting parameterization may contain large distortions or even local non-injectivities due to fold-overs. Based on the gradient of the parameterization's objective functional, the polycube's embedding can be optimized so as to arrive to a local optimum of global embedding quality.

In order to generate a volume parameterization of a given solid model two main issues have to be addressed. The first is the computation of the polycube embedding. The second is the computation of a smooth cross field required for aligned global parameterization. The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 presents the algorithm for polycube construction, cross field design, global aligned parameterization computation, and trivariate splines fitting; Section 4 presents examples and applications of the proposed method.

2. Related Work

2.1. Literature Survey

Since the concept of isogeometric analysis was proposed, many researchers working on computational mechanical and geometric modeling were involved in this field. Works on isogeometric analysis can be classified into four categories: application of IGA to various simulation and analysis problems; error estimation, accuracy and efficiency improvement of IGA framework by refinement operations and parallel computing; application of different spline models

in IGA; constructing analysis-suitable parameterization of computational domain from standard boundary CAD models. The work presented in this paper belongs to the fourth category.

Constructing analysis-suitable parameterization from a given solid model, defined by its boundary triangle mesh or boundary (possibly trimmed) spline surfaces, remains one of the most significant challenges in IGA. For a given object, various computational domains can be constructed with the same shape but with different parameterizations. Cohen et al. [6] studied the parameterization of computational domains in IGA, and showed that the parameterization quality has a great impact on analysis results and efficiency. Pilgerstorfer and Jüttler [21] showed that in IGA the condition number of the stiffness matrix, which is a key factor for the stability of the linear system, depends strongly on the parameterization quality. One basic requirement for computational domains in IGA is that the resulting parameterization should have no self-intersections (i.e., the mapping from the parametric domain to physical domain should be injective). Xu et al. [29] proposed a linear and easy-to-check sufficient condition for the injectivity of a trivariate B-Spline parameterization. In addition, the parameterization should also satisfy these requirements: the isoparametric elements should be as uniform as possible, and the isoparametric structure should be as orthogonal as possible.

The first step toward obtaining a global parameterization is the computation of a canonical domain. A popular shape abstraction method to build canonical domains is to use polycubes [26]. A polycube is a solid composed of cubes. It can be used to approximate very roughly the geometry of an object while faithfully replicating its topology. However, in practice, due to the complexity of shapes, polycubes are usually constructed manually, entailing considerable effort. It is still a challenging problem to automatically create polycubes for high genus geometry and use them in constructing analysis-suitable trivariate splines. For a genus-zero geometry, Zhang et al. [30] proposed a robust and efficient approach to construct injective solid T-splines. They created a parametric mapping between the boundary triangle mesh and the boundary of a unit cube, which is the parameter domain of the solid T-spline. To do this, eight vertices have to be selected by the user, which correspond to the eight corners of the cube. Then twelve curves are found via calculating the shortest path between each pair of the selected vertices. Based on these curves, the input mesh is divided into six sub-meshes, and each one is associated with one face on the unit cube. Then the main work was to map each sub-mesh to a planar unit square using surface parameterization based on harmonic mapping. Based on Morse theory, the work of Zhang et al. [30] has been extended by Wang et al. [27] to models with arbitrary topology. To extract the topology of the input geometry, they used the saddle points of a smooth harmonic scalar field defined over the mesh. Based on these saddle points, a polycube whose topology is equivalent to the input geometry is built and it serves as the parametric domain for the solid T-spline. A polycube mapping is then used to build a one-to-one correspondence between the input triangulation and the polycube's boundary. This method needs the interaction of the user at two stages to construct the poly-

cube domain: to select two extremum constraints which controls the harmonic field, and to choose four seeds points which controls the polycube generation. Choosing these inputs without considering geometric features and symmetry can affect the polycube generation and may yield asymmetric parameterization. Li et al. [15] extended the conventional polycube to a Generalized PolyCube (GPC) [14], which enables the curved cuboid representation of the elementary sub-volumes. This enables the polycube map approach to be applied to more complex objects. After decomposing the entire model into a group of pants patches patches (i.e., genus-0 surfaces with 3 boundaries), they decompose each pants patch into four connected cube-like sub-patches to obtain the polycube. As for the method proposed by Wang et al. [27], this method still needs the interaction of the user to select four seed points which controls the GPC generation. Choosing these inputs without considering geometric features can greatly affect the quality of the generated polycube. Al-Akhras et al. [1] extended the work of Li et al. [15] by proposing a completely automatic pants-to-cuboids decomposition algorithm to generate a polycube. However, there are types of input which are problematic for the proposed method, in particular inputs with sharp features. For such models, the cuboids' polyedges don't align with the sharp features, and in general they don't align with the principal directions of the surface.

By using polycubes as canonical domains, the problem of finding a volume parameterization of a solid model is simplified to a problem of finding a surface parameterization between the model's boundary and the polycube's boundary. In order to convert a triangle mesh into a spline surface, one of the main problems we need to tackle is the extraction of a good quadrilateral control mesh of the surface. This control mesh is said to be good if its edges are orthogonal and aligned with the principal curvature directions of the surface. These two properties make the control mesh optimum in an approximation point of view, and greatly help to reduce unwanted oscillations on the final spline surface built from it. So the main problem in converting a triangle mesh into a spline surface is equivalent to the problem of finding a proper quadrangulation of the surface. Most recent techniques for quadrangulation are based on aligned global parameterizations (Ray et al. [22], Kälberer et al. [12], Bommes et al. [2]). A surface parameterization can be viewed as a mapping from a surface S embedded in $\mathbb{R}^3$ to a canonical domain D embedded in $\mathbb{R}^2$. For topologies of S other than the disk, the domain D must necessarily include discontinuities (e.g. cuts or seams). Methods based on local parameterization often lead to visible breakup of the iso-parametric curves along seams. Global parameterization is of particular interest because it covers the entire surface, and hence can alleviate this problem when the translational and rotational discontinuities in the parameterization are compatible along cuts. Field-alignment techniques adapt the parameterization to the geometry of the surface by fitting the parameterization gradient to a smooth cross field capturing the principal directions and geometric features (e.g. sharp edges and boundaries). Thus the task is shifted from the definition of a surface parameterization to the design of a smooth cross field on the surface.

Many applications require the construction of N-symmetry direction fields

on surfaces, i.e., fields that associate to every point a set of N unary vectors (i.e., directions) forming equal angles between radially consecutive directions. For example, a 2-symmetry direction (line) field can be used to guide texture placement, a 4-symmetry direction (cross) field can be used for quadrangular remeshing (Bommes et al. [2]), and a 6-symmetry direction field can be used for triangular and hexagonal remeshing (Nieser et al. [20]). In quadrilateral remeshing, the cross field singularities play an important role. For instance, a singularity will generate an irregular vertex or a non-quad polygon. A variety of methods were proposed for cross field design. Cross field design is the generation of a smooth cross field from a set of constraints. These constraints can be topological (i.e., imposed singularities) and/or geometrical (i.e., imposed directions). Topologically, the cross field is constrained by the polycube structure. This means that the cross field must be singular only at the position of irregular nodes of the polycube. Geometrically, the cross field must follow the geometric features of the surface. A number of methods rely on manually placed singularities (Ray et al. [24]; Lai et al. [13]; Crane et al. [7]). Such methods allow to control the topological structure of a cross field but the integration of geometrical constraints (such as specified directions) can be difficult [7], may lead to none globally optimal solution [24] and sometimes be in contradiction with the specified topology (in this case the arising of new singularities is inevitable). Other methods compute singularity positions automatically (Ray et al. [23]; Bommes et al. [2]) given a set of sparse geometrical constraints. However, such methods don't give any control over the topological structure of the cross field.

2.2. Contribution

Our work investigates objects with complex geometry and arbitrary topology given in boundary representation. Our goal is to generate a trivariate parameterization with respects to a given solid model defined by its boundary triangle mesh or boundary (possibly trimmed) spline surfaces. In order to have a unified framework for both input types, our method takes as input a solid model defined by its triangulated boundary.

We present a method to compute an optimized aligned global parameterization using an optimization process of the initial polycube's embedding. Such optimization eliminates the effect of the polycube's embedding quality on the final volume parameterization. In addition, it allows the polycube to automatically capture the geometric features of the model (e.g. principal directions, sharp edges, boundaries). The optimization process is based on one of the main contributions of this paper: a novel way to design cross fields with topological (i.e., imposed singularities) and geometrical (i.e., imposed directions) constraints by solving only sparse linear systems.

Our method can be also used to solve one major bottleneck in the context of reduced order modeling for geometric parametric analysis. For different geometrical instances of the same topological model, the issue of having isotopological solutions is addressed by generating the same parameterization for all objects.

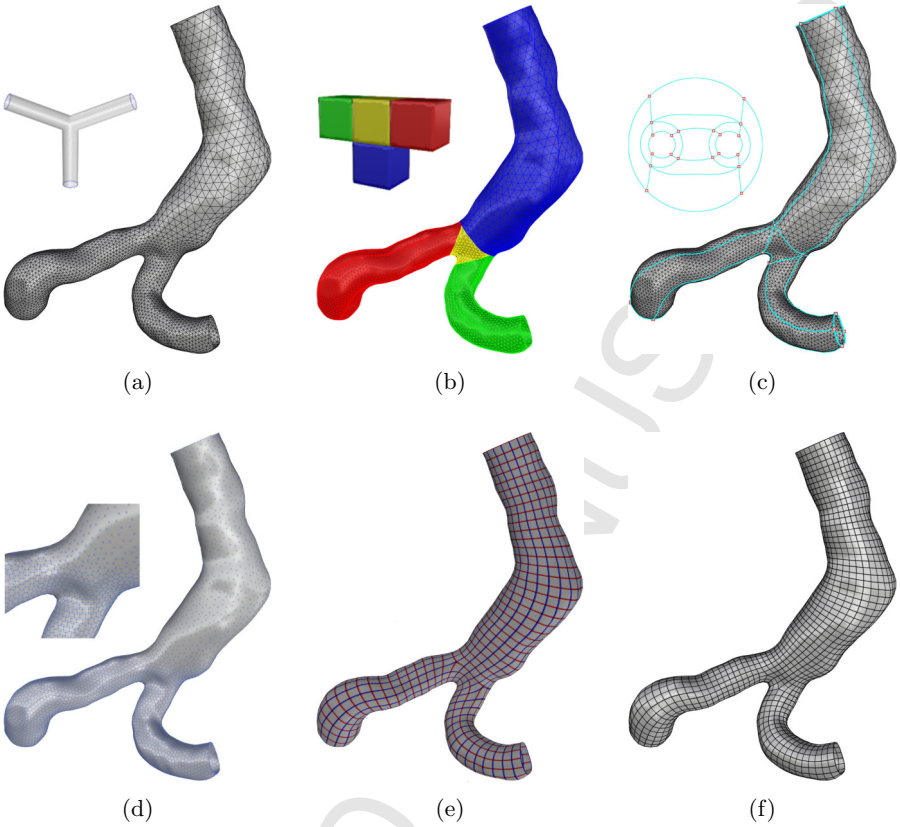


Figure 4: Algorithm overview. The input triangulated boundary surface homeomorphic to a pants patch (a). The polycube generated from cuboid decomposition (b). The embedding of the quad layout induced by the polycube (c). The cross field topologically conformal with the quad layout, and geometrically aligned with the surface principal directions and boundaries (d). The global parameterization aligned with the cross field (e). The control quad mesh extracted from the global aligned parameterization (f).

3. Algorithm

Our method takes as input a solid model defined by its triangulated boundary. The algorithm includes three main steps (Figure 4):

- Using cuboid decomposition, an initial polycube approximating the input boundary mesh is built. The input mesh is decomposed into a set of pants patches, and each pants patch is decomposed into a set of cuboid patches (Section 3.1).

- The polycube's nodes and arcs decompose the input model locally into quadrangular patches, and globally into hexahedral domains. Using aligned global parameterization, the nodes are re-positioned and the arcs are re-routed across

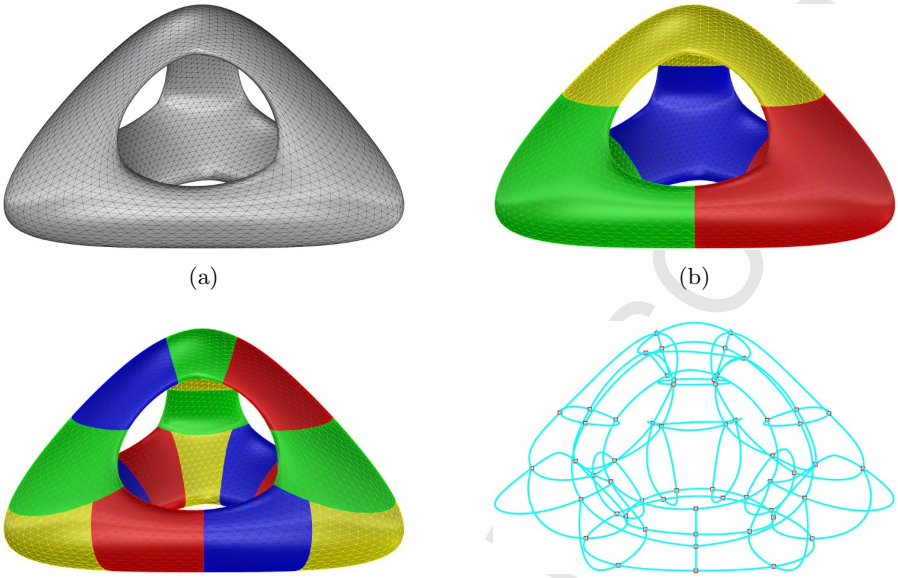


Figure 5: Generalized polycube generation. The input triangulated boundary surface (a). The boundary surface is first decomposed into a set of pants patches: genus-0 surfaces with 3 boundaries (b). Each pants patch is further decomposed into a set of cuboids: boxed regions enclosed by 6 topological disks (c). The embedding of the polycube's nodes and arcs (d).

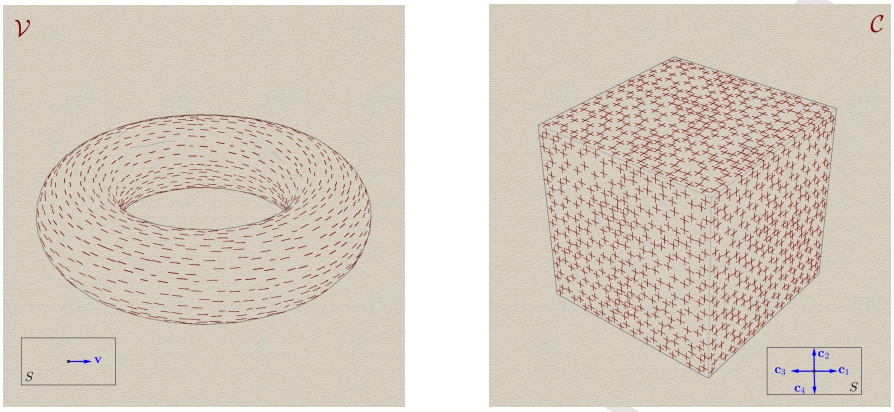
the surface in a way to achieve low overall patch distortion, and alignment to principal curvature directions and sharp features (Sections 3.2 and 3.3).

- Based on the optimized polycube and parameterization, compatible B-spline boundary surfaces are reconstructed. The interior volumetric parameterization is computed using Coon's interpolation and the B-spline surfaces as boundary conditions (Section 3.4).

We work with a triangulated connected 2-manifold mesh M with a set of vertices V, edges E and faces F. Its dual mesh M^* has a set of vertices V^*, edges E^* and faces F^*. The dual mesh doesn't have to be explicitly constructed since dual quantities can be stored on the corresponding primal elements. M is also oriented: each face $f \in F$ has coherent normal, and each edge $e \in E$ has an orientation. This allows to define a unique orientation for each dual edge $e^* \in E^*$.

3.1. Polycube Generation

We briefly review our previous work on cuboid decomposition of solid models (for more details, see Al-Akhras et al. [1]). We refer the readers to the book of Hatcher [10] for more details on topology. The starting point of our algorithm is a triangulated mesh M bounding a solid CAD model V. As the boundary



(a) (b)

Figure 6: N-symmetry direction fields. A 1-symmetry direction field (i.e., vector field) $\mathcal{V}$ on a torus: at each point of the surface S, there exists one direction $\mathbf{v}$ (a). A 4-symmetry direction field (i.e, cross field) $\mathcal{C}$ on a cube: at each point of the surface S, there exists one 4-symmetry direction $\mathbf{c}$ which is a set of 4 directions $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$ invariant by a rotation of $\pi/2$ around the normal (b).

of a solid region, $M = \partial V$ is a closed surface and can be of complex geometry and arbitrary topology. We begin by decomposing the triangulated input mesh M into a set of pants patches. Such segmentation decomposes a complicated surface into a set of shapes that have a trivial topology: genus-0 surfaces with 3 boundaries. The Euler characteristics χ for surfaces of most topological types are negative integers. For a pants patch $\chi = -1$, and therefore pants decomposition provides a canonical decomposition scheme for these surfaces. We then decompose each pants patch into a set of cuboid patches. Each cuboid is one boxed region enclosed by 6 disk-like surfaces. The idea is to generate nodes and arcs on each pants patch and decompose it into 4 connected components, each having 8 nodes and 12 arcs like a cuboid (Figure 5).

3.2. Direction Field

In order to obtain a guiding field for the global parameterization (Section 3.3), we design a 4-symmetry direction field (i.e., cross field) $\mathcal{C}$ on the surface. Direction field design is the generation of a smooth direction field from a set of constraints. These constraints can be topological (i.e., imposed singularities) or geometrical (i.e., imposed directions). We strike for a balance between three important properties of direction fields : smoothness, singularity positions, and alignment with local geometry. Topologically, $\mathcal{C}$ must be conformal with the quad layout $\mathcal{L}$ induced by the polycube. This means that $\mathcal{C}$ must be singular only at the position of irregular nodes of $\mathcal{L}$. Geometrically, $\mathcal{C}$ must follow the principal directions and sharp features of the surface.

3.2.1. Definition

A *direction field* defined on a surface S is a tangent unit vector field: at each point of the surface, there exists a direction $\mathbf{w}$ such that $\|\mathbf{w}\| = 1$ and $\mathbf{w} \cdot \mathbf{n} = 0$, where $\mathbf{n}$ is the normal of S. A *N-symmetry direction field* $\mathcal{W}$ is a multivalued direction field: at each point of the surface S, there exists a N-symmetry direction $\mathbf{w}$ which is a set of N directions $\{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ preserved by rotations of $2\pi/N$ around the normal $\mathbf{n}$ of S (Figure 6).

3.2.2. Discretization

The first step of the discretization of a direction field $\mathcal{W}$ on a mesh M is the choice of tangent planes. Storing directions at vertices V requires the definition of tangent planes in terms of extrinsic geometry. Storing directions at faces F is more natural because tangent planes can be defined intrinsically. Hence, in the context on this paper, direction fields will be represented at faces F of the mesh M, or equivalently at vertices V^* of the dual mesh M^*.

The direction field is then sampled by defining one direction on each face. This is done by choosing a local orthonormal frame $(\mathbf{x}, \mathbf{y})$ on each face f. $\mathbf{x}$ is the unit vector along one of the oriented edges e of f, and $\mathbf{y} = \mathbf{n} \times \mathbf{x}$ with $\mathbf{n}$ being the normal of f. The direction $\mathbf{w}$ defined on f can then be expressed in terms of polar coordinates. Since $\mathbf{w}$ has unit norm, it is completely parameterized by the polar angle α it forms with $\mathbf{x}$ (Figure 7a). Angles in adjacent faces can be represented in a common coordinate frame by flattening both faces along their common edge. Given an angle α_i^i in face f_i, it can be expressed in an adjacent face f_j as:

$$\alpha_i^j = \alpha_i^i - \delta_{ij} + \delta_{ji} = \alpha_i^i + \kappa_{ij} \text{ with } \kappa_{ij} = -\delta_{ij} + \delta_{ji} = -\kappa_{ji} \tag{1}$$

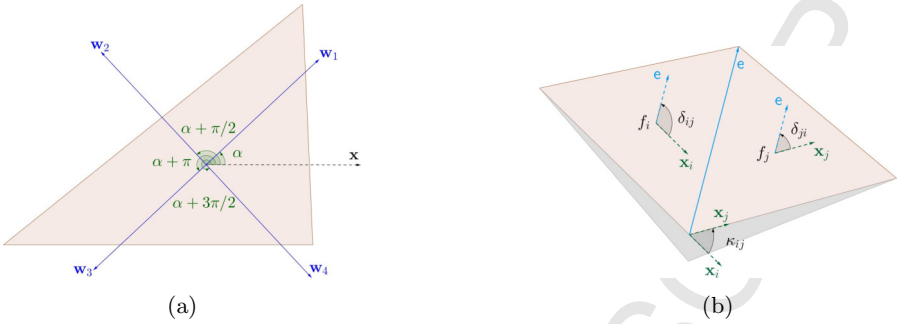
where δ_{ij} and δ_{ji} are the angles between the shared edge e and the reference direction $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively, and $\kappa_{ij} \in]-\pi, \pi]$ represents the angle between reference directions $\mathbf{x}_i$ and $\mathbf{x}_j$ (Figure 7b).

However by representing the direction field only at a set of points, its topology cannot be accessed. In particular, there is no information about the behavior of the direction field between the discretization points. Assume that we want to interpolate a direction $\mathbf{w}$ along an edge $[AB]$ knowing $\mathbf{w}_A$ and $\mathbf{w}_B$. If the direction field is continuous, the angular variation ω along $[AB]$ verifies:

$$\omega_{AB} = \int_A^B d\alpha = \angle(\mathbf{w}_A, \mathbf{w}_B) + (2\pi/N)p_{AB} \tag{2}$$

where $\angle(\mathbf{w}_A, \mathbf{w}_B) \in]-\pi, \pi]$ is the angle between $\mathbf{w}_A$ and $\mathbf{w}_B$, and $p \in \mathbb{Z}$ is an integer. The angular variation ω has infinitely many possible values for all the possible values of p (Figure 8a). This ambiguity can be solved by (Figure 8b):

- specifying the directions at A and B and the integer p. $p \in \mathbb{Z}$ is called the *period jump* [16] and it specifies the number of N^{th} turns the direction $\mathbf{w}$ undergoes when passing from A to B. We will refer to this method, developed by Ray et al. [24], as the *period jump based discretization*. In this case, an



(a) (b)

Figure 7: N-symmetry direction fields representation. A 4-symmetry direction $\mathbf{w}$ on a face f is a set of directions $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4\}$ defined as the image of a reference direction $\mathbf{x}$ by rotations of $\alpha + k(\pi/2), k \in \{0, 1, 2, 3\}$ (a). Angles in adjacent faces f_i and f_j can be expressed in a common coordinate frame using the rotation angle κ_{ij} between reference directions $\mathbf{x}_i$ and $\mathbf{x}_j$ (b).

angle α_i^i defined on face f_i, expressed as α_i^j in an adjacent face f_j, is in general different from α_j^j:

$$\alpha_i^j = \alpha_i^i + \kappa_{ij} + (2\pi/N)p_{ij} \text{ and } \alpha_i^j \neq \alpha_j^j \tag{3}$$

Hence direction angles on all faces and period jumps on all dual edges are required to completely define the discrete direction field.

- specifying one direction at A or B and the real ω. $\omega \in \mathbb{R}$ is called the *connection angle* and it specifies the amount of rotation the direction undergoes when passing from A to B. We will refer to this method, developed by Crane et al. [7], as the *connection based discretization*. In this case, an angle α_i^i defined on face f_i, expressed as α_i^j in an adjacent face f_j, is in general equal to α_j^j:

$$\alpha_i^j = \alpha_i^i + \kappa_{ij} + \omega_{ij} \text{ and } \alpha_i^j = \alpha_j^j \tag{4}$$

Hence a direction angle on one face and connection angles on all dual edges are required to completely define the discrete direction field

For the sake of simplicity, the angle α_i^i defined on face f_i will be simply denoted by α_i. In the context of this paper, we use both discretization methods. Connection based discretization (4) is used for topological design. This is done by finding the proper connection angles on dual edges satisfying the topological constraints. Period jump based discretization (3) is used for geometrical design. This is done by fixing period jumps on all dual edges and then fitting direction angles on primal faces to the geometrical constraints.

3.2.3. Smoothness

A common requirement on direction fields is smoothness. For a mesh M, the curvature of a direction field $\kappa_{\mathcal{W}}$ along a dual edge e^* is the angle difference

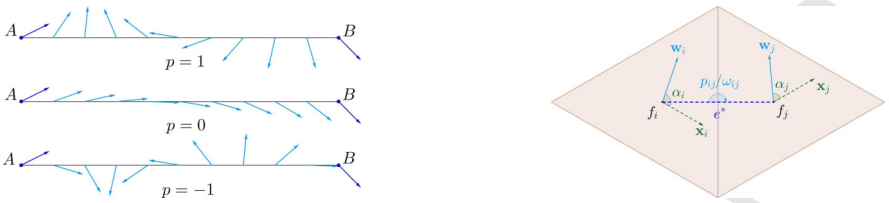


Figure 8: N-symmetry direction fields discretization. Between two directions given at points A and B, different interpolations are possible (a). This ambiguity is solved by specifying angles on both faces and an integer p. p is called the period jump and it specifies the number of N^{th} turns a direction undergoes when traversing a dual edge. This ambiguity can also be solved by specifying an angle on one face and a real ω. ω is called the connection angle and it specifies the amount of rotation a direction undergoes when traversing a dual edge (b).

between neighboring faces represented in a common coordinate frame [24]:

$$\kappa_{\mathcal{W}}(e^*_{ij}) = \omega_{ij} = \alpha_j - [\alpha_i + \kappa_{ij} + (2\pi/N)p_{ij}] \tag{5}$$

where e^*_{ij} is the dual edge oriented from f_i to f_j, $\omega_{ij} \in \mathbb{R}$ is the connection angle, $\alpha_i \in \mathbb{R}$ and $\alpha_j \in \mathbb{R}$ are direction angles in faces f_i and f_j respectively, $\kappa_{ij} \in]-\pi, \pi]$ is the angle between local frames, and $p_{ij} \in \mathbb{Z}$ is the period jump. The smoothness energy of a direction field $E_s(\mathcal{W})$ can be measured as its integrated squared curvature $\kappa_{\mathcal{W}}$ [24]:

$$E_s(\mathcal{W}) = \sum_{e^*_{ij} \in E^*} \left\| \kappa_{\mathcal{W}}(e^*_{ij}) \right\|^2 \tag{6}$$

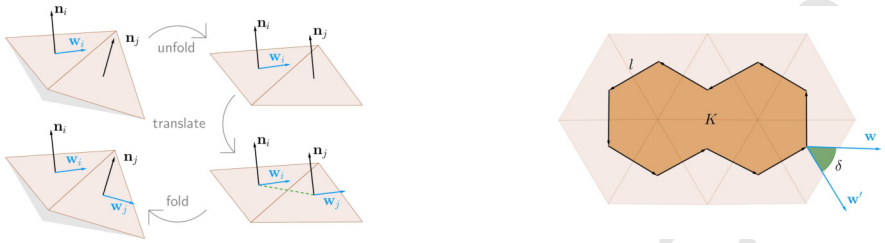
$$= \sum_{e^*_{ij} \in E^*} \|\omega_{ij}\|^2 \tag{7}$$

$$= \sum_{e^*_{ij} \in E^*} \|\alpha_i + \kappa_{ij} + (2\pi/N)p_{ij} - \alpha_j\|^2 \tag{8}$$

Topological design is based on the minimization of energy (7), and geometrical design is based on the minimization of energy (8).

3.2.4. Topological Design

On a smooth surface S, a *connection* Ω describes how a direction rotates locally when moved an infinitesimal distance along a given direction. Once this local information is provided, by integrating these infinitesimal rotations along a curve, a direction can be mapped between distinct tangent planes. This process is called *parallel transport*. For a mesh M, a connection can be represented as an angle ω on each dual edge e^*. This angle represents the integrated rotation that a direction undergoes when passing from one face to its neighbor. Once a connection Ω is defined, there is a straightforward way to construct a direction field $\mathcal{W}$ on the mesh M. Namely, starting with an arbitrary face and an initial



(a) (b)

Figure 9: Discrete connections. Transport using Levi-Civita connection can be described as follows: unfold the faces isometrically into the plane, translate the vector along the shared dual edge, and then fold the faces back into their initial configuration (a). In general, a vector parallel transported around a loop l will not end up where it started. The angle between the initial direction $\mathbf{w}$ and the transported direction $\mathbf{w}'$ is called the holonomy δ. Every connection has an associated curvature K. If l bounds a region then the curvature K over this region equals the holonomy δ around the loop (b).

direction, this direction is transported to adjacent faces until the entire mesh is covered.

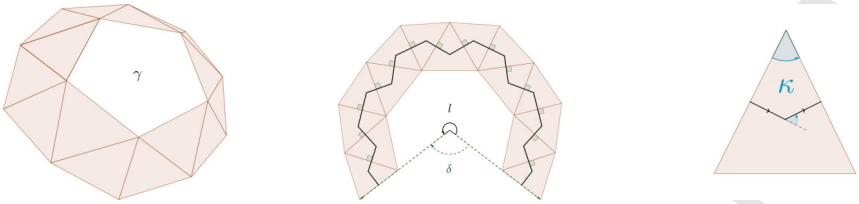
The simplest connection is obtained by setting connection angles to zero on all dual edges: this is the Levi-Civita connection (Figure 9a). In general for a curved surface S, a direction parallel transported around a loop l by a connection will not return to its original orientation. The difference in angle between the initial and final directions is called the *holonomy* δ of the connection around the loop l. Every connection has also an associated *curvature* K. In particular if the loop l bounds a region of the surface S, then the curvature K of the connection over this region equals its holonomy δ around the loop l (Figure 9b).

A cycle γ is a sequence of primal faces that form a loop (Figure 10a). This loop is defined as the sequence of dual edges orthogonal to primal edges shared by adjacent faces of γ after unfolding the cycle isometrically to the plane (Figure 10b). From this definition, the discrete geodesic curvature κ_γ at a face equals the signed angle between corresponding primal edges (Figure 10c). In this case, the holonomy δ_γ of the Levi-Civita connection around the cycle γ can be defined as [19]:

$$\delta_\gamma = -\sum_{e^* \in \gamma} \kappa_\gamma(e^*) \Rightarrow \delta_\gamma = -\kappa_\gamma \tag{9}$$

where κ_γ is the total discrete geodesic curvature of the cycle γ.

Because of holonomy, depending on which connection is used, the direction field might not be consistently defined. A direction field is considered to be consistent only if directions are mapped to themselves modulo $2\pi/N$ by parallel transport. For instance, the holonomy of the Levi-Civita connection equals the Gaussian curvature, so in general a direction transported around a closed loop is not mapped back to itself. As a consequence, transport from one point to another will depend on the choice of path. Crane et al. [7] showed that transport via trivial connections, i.e., connections with globally vanishing holonomy,



(a) (b) (c)

Figure 10: Cycles holonomy. A cycle γ is a sequence of primal faces (a). By unfolding γ isometrically to a plane, a loop l is defined as the sequence of dual edges orthogonal to primal edges shared by adjacent faces of γ (b). Within a face, l turns an angle κ, computed as the oriented angle between corresponding adjacent primal edges (c).

is path-independent. However zero holonomy everywhere implies zero curvature everywhere. According to the Gauss-Bonnet theorem, the total Gaussian curvature of a surface is equal to $2\pi\chi$, where χ is the Euler characteristic of the surface. To overcome this problem, the total curvature is distributed over the mesh in a way that doesn't interfere with parallel transport. In particular, curvature will be concentrated at boundaries or/and at a set of vertices called *singularities* in increment of $2\pi/N$. In this case, for a given cycle γ, a direction might do an arbitrary number of N^{th} turns before coming back to its starting point. This quantity is called the turning number T_γ of the direction field $\mathcal{W}$ along the cycle γ [24]:

$$2\pi T_\gamma = \sum_{e^* \in \gamma} \kappa_{\mathcal{W}}(e^*) - \sum_{e^* \in \gamma} \kappa_\gamma(e^*) \tag{10}$$

where $\kappa_{\mathcal{W}}$ is the curvature of the direction field $\mathcal{W}$, and κ_γ is the geodesic curvature of the cycle γ.

By substituting the curvature $\kappa_{\mathcal{W}}$ of a direction field along a dual edge by its connection angle ω (5), and the total geodesic curvature κ_γ of a cycle by its holonomy δ_γ (9), the turning number T_γ of the direction field $\mathcal{W}$ along the cycle γ can also be expressed as:

$$2\pi T_\gamma = \sum_{e^* \in \gamma} \omega(e^*) + \delta_\gamma \Rightarrow \sum_{e^* \in \gamma} \omega(e^*) = -\delta_\gamma + 2\pi T_\gamma \tag{11}$$

In terms of connections, this states that the trivial connection angles around a given cycle should cancel the holonomy found with Levi-Civita, and potentially add a certain amount of curvature in order to respect the Gauss-Bonnet theorem. Ray et al. [24] showed that the topology of a direction field is entirely defined by the turning numbers of the direction field along $2g$ homology generator cycles, $\max(b-1,0)$ boundary cycles, and s cycles around singularities. We need to fix all topological degrees of freedom in order to restrict to cross fields topologically compatible with the quad layout $\mathcal{L}$. The turning numbers around singularities, homology generators and boundary cycles are determined using the method described by Campen and Kobbelt [4]. Crane et al. [7] presented an algorithm

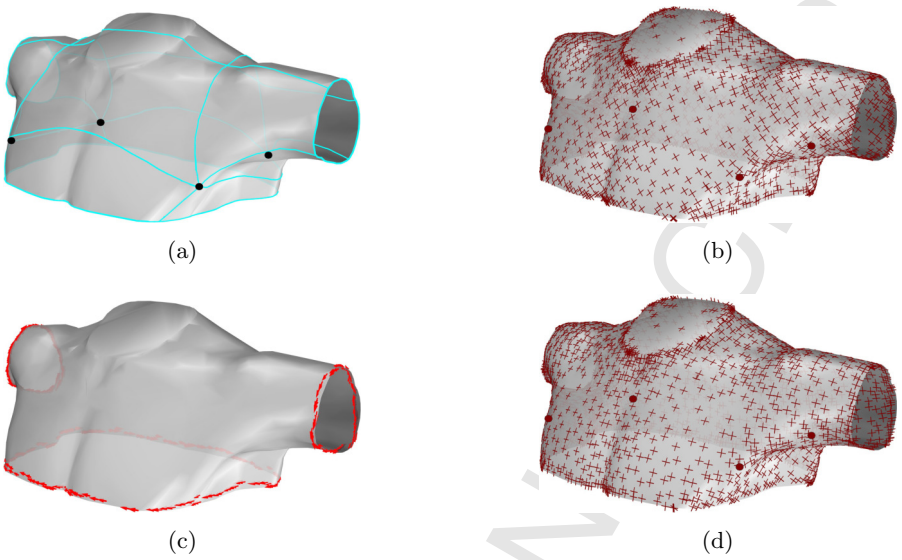


Figure 11: N-symmetry direction fields topological design. The TORSO model along with its quad layout $\mathcal{L}$ obtained from polycube decomposition. The quad layout singular nodes are depicted by circles (a). The cross field $\mathcal{C}$ obtained by setting the initial direction angle to a random value. The cross field singular vertices (depicted by circles) are conformal with the quad layout $\mathcal{L}$ singular nodes (b). The directional constraints on boundary faces corresponding to the direction of boundary edges (c). The cross field obtained by setting the initial direction angle to the mean value of all directional constraint angles expressed in a common face. Locally, the cross field directions don't respect the directional constraints on each face (d).

that, given the above turning numbers, determines the appropriate trivial connection by minimizing energy (7). Finally, we still need to define one direction to construct the direction field. To obtain this direction, all directional constraints are transported from their respective faces to a common face f_0. The initial direction angle α_0 is defined to be the mean of all directional constraint angles expressed in f_0 (Figure 11).

3.2.5. Geometrical Design

Within this topologically fixed space, we now strive to find a smooth direction field that interpolates sparse directional constraints, corresponding to reliable principal directions, feature curves directions, or user specified intents. In other terms, given a mesh M and a subset of faces $F^c \subset F$ with constrained directions $\{\alpha^c\}$, we search for the smoothest interpolating direction field $\mathcal{W}$ restricted to a given topology.

In the general case ($\mathcal{W}$'s topology is free), as presented by Bommes et al. [2], the smoothest interpolating direction field $\mathcal{W}$ is computed by finding integer valued period jumps $\{p\}$ on dual edges and real valued direction angles $\{\alpha\}$ on faces minimizing the direction field smoothness energy $E_s(\mathcal{W})$ subject to directional constraints $\{\alpha^c\}$ on the set of faces F^c. In our case, the direction

field $\mathcal{W}$ is restricted to a given topology so the integer valued period jumps $\{p\}$ are fixed, and the real valued angles $\{\alpha\}$ are the only variables in the constrained optimization problem. For the mesh M, let $|E|$ be the number of its edges and $|F|$ the number of its faces. Then the energy $E_s(\mathcal{W})$ can be expressed as:

$$E_s(\mathcal{W}) = \sum_{e^*_{ij} \in E^*} \|\alpha_i + \kappa_{ij} + (2\pi/N)p_{ij} - \alpha_j\|^2 = \|\mathbf{A}\boldsymbol{\alpha} - \mathbf{b}\|^2 \qquad (12)$$

where $\mathbf{A} \in \mathbb{R}^{|E|\times|F|}$, $\mathbf{b} \in \mathbb{R}^{|E|}$, and $\boldsymbol{\alpha} \in \mathbb{R}^{|F|}$ is the vector of unknowns corresponding to direction angles on faces. For the k^{th} dual edge e^*_{ij} oriented from face f_i to face f_j, $\mathbf{A}(k,i) = 1$, $\mathbf{A}(k,j) = -1$, and $\mathbf{b}(k) = -[\kappa_{ij} + (2\pi/N)p_{ij}]$. The constrained minimization problem can then be formulated as follows:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{|F|}} \|\mathbf{A}\boldsymbol{\alpha} - \mathbf{b}\|^2 \text{ subject to } \alpha_i = \alpha_i^c \ , \ \forall f_i \in F^c \qquad (13)$$

For a given face $f^c \in F^c$, it is not inherently clear by which of $\mathcal{W}$'s N directions the directional constraint is to be interpolated. This choice has an important effect on the fixed values of the period jumps. Assume that for a given face f_i, the direction angle is given by α_i, and the period jumps are given by $\{p_{ij}\}$. If the angle α_i is rotated by a multiple of $2\pi/N$ around the normal (i.e., $\bar{\alpha}_i = \alpha_i + k \cdot 2\pi/N$), in order to preserve the direction field topology, this change must be compensated by updating the affected period jumps (i.e, $\bar{p}_{ij} = p_{ij} - k$). Using a modified version of the technique for reducing the search space [2], we proceed as follows to compute the fixed values of period jumps accommodating the given set of directional constraints (Figure 12):

1. We construct a forest of Dijkstra trees of the dual mesh, where each constrained face in F_c is the root of a separate tree such that no tree connects constrained faces [2].

2. In each constrained face f_i^c, we normalize the direction angle α_i and the constraint angle α_i^c to $]-\pi/N; \pi/N]$. This choice is justified by the fact that a direction field is invariant by rotations of $2\pi/N$ around the normal. We then find the integer $k \in \mathbb{Z}$ minimizing the absolute difference $|\ \alpha_i + k \cdot 2\pi/N - \alpha_i^c\ |$. Finally, the interpolating direction in face f_i^c is set to be $\bar{\alpha}_i = \alpha_i + k \cdot 2\pi/N$.

3. Starting at constrained faces $\{f^c\}$ with interpolating directions $\{\bar{\alpha}\}$, we reconstruct the direction field using the connection angles and following the forest of Dijkstra trees. We note that at this point, the direction field is unchanged : it was only discretized differently so its period jumps could agree with the directional constraints.

4. The fixed period jumps $\{p\}$ are defined to be the integers minimizing the direction field curvature (5) on each dual edge.

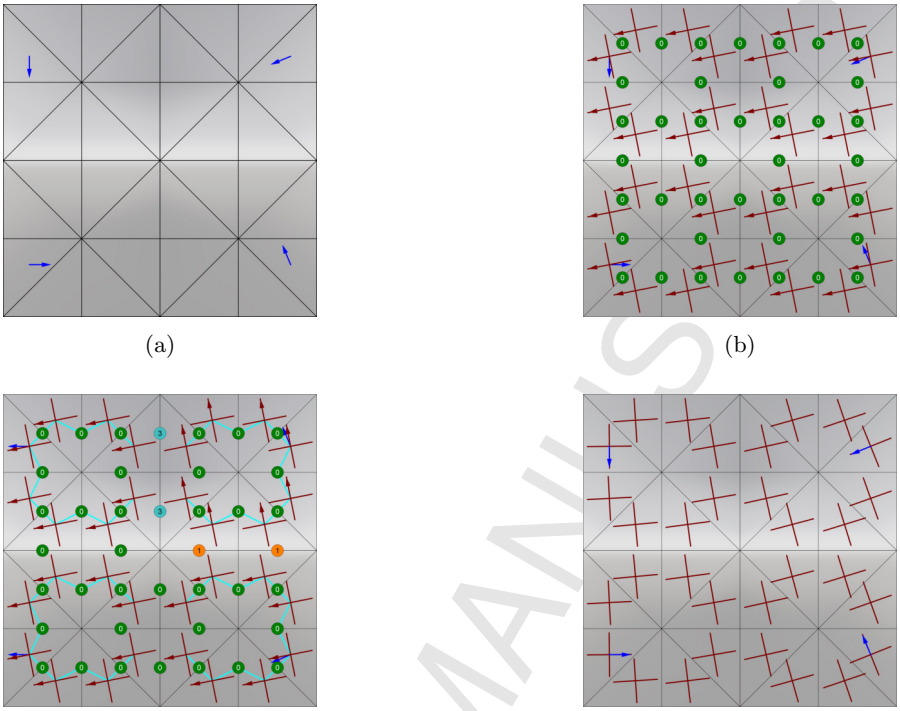


Figure 12: N-symmetry direction fields directional constraints interpolation. We are searching for the smoothest cross field with no singularities and interpolating 4 directional constraints (a). The cross field satisfying topological constraints and constructed using the mean value of the directional constraints. The primary directions are indicated by an arrow on each face, and the period jumps are indicated by a value on each edge (b). The reconstructed cross field starting at constrained faces and following the forest of Dijkstra trees (c). The cross field satisfying topological and geometrical constraints (d).

The geometrical direction field is the solution of a symmetric linear system obtained by setting the gradient of the energy (13) to zero (Figure 13):

$$\frac{\partial E_s(\mathcal{W})}{\partial \boldsymbol{\alpha}} = \mathbf{0} \Rightarrow \mathbf{A}^t\mathbf{A} \cdot \boldsymbol{\alpha} = \mathbf{A}^t\mathbf{b} \tag{14}$$

Hard constraints can be difficult to handle because in some cases additional singularities should arise to exactly satisfy all the constraints. To prevent undesired singularities and have some freedom around imposed singularities, the constrained minimization problem (13) can be modified to integrate soft directional constraints:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{|F|}} (1-\lambda)\left\|\mathbf{A}\boldsymbol{\alpha}-\mathbf{b}\right\|^2 + \lambda\left\|\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}\right\|^2 \text{ subject to } \alpha_i = \alpha_i^c \ , \ \forall f_i \in F^c \tag{15}$$

where $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{|F|}$ is the set of soft constraints on faces, and the parameter $\lambda \in [0,1]$

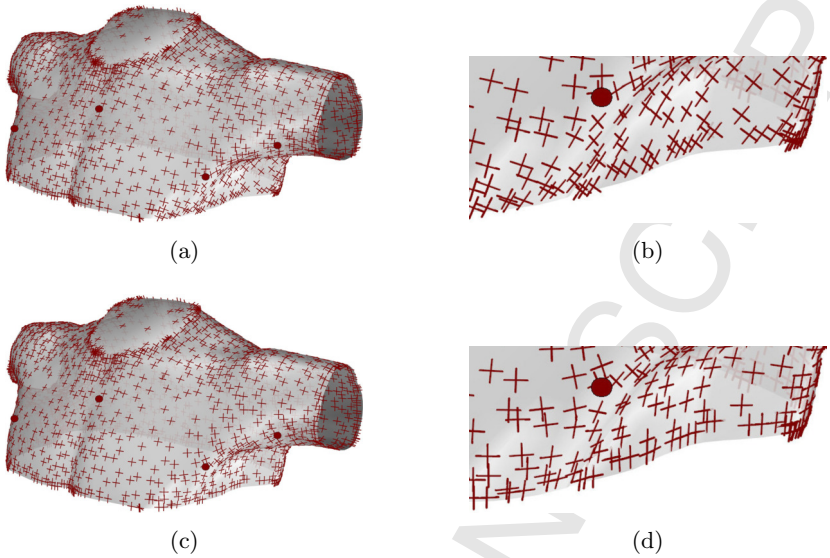


Figure 13: N-symmetry direction fields geometrical design. The topological cross field doesn't satisfy all directional constraints on boundary faces corresponding to boundary edges (a,b). The geometrical cross field satisfies all directional constraints while preserving the same topology (c,d).

is used to control the trade off between smoothness and fitting.

3.3. Aligned Global Parameterization

Every mesh M can be cut to a topological disk M_c, by removing a set of seam edges from the mesh. In this case, a global parameterization (Figure 15) is a piecewise linear map from the mesh $M_c \in \mathbb{R}^3$ to a topological disk domain $D \in \mathbb{R}^2$. Since the parameterization should be piecewise linear, it is sufficient to assign a (u, v) parameter value to each face corner of the mesh. The parameterization should be locally oriented according to the guiding field (Section 3.2). This implies that the gradients ∇u and ∇v of the piecewise linear scalar fields u and v should follow the cross field directions on each face.

3.3.1. Seamless Parameterization

We are interested in global parameterizations where the transition functions across seems are not arbitrary but of a very restricted class: rigid transformations with a rotation angle of some multiple of $\pi/2$. In addition for quadrangulation, it is necessary that the transitional part of the transitions are integral. In this case, we talk about integral seamless parameterization. We follow the method of Bommes et al. [2] to compute such parameterization:

- A cut is a connected graph G of mesh edges, such that $M \setminus G$ is topologically equivalent to a disk $D \in \mathbb{R}^2$. The appropriate cut graph is computed in

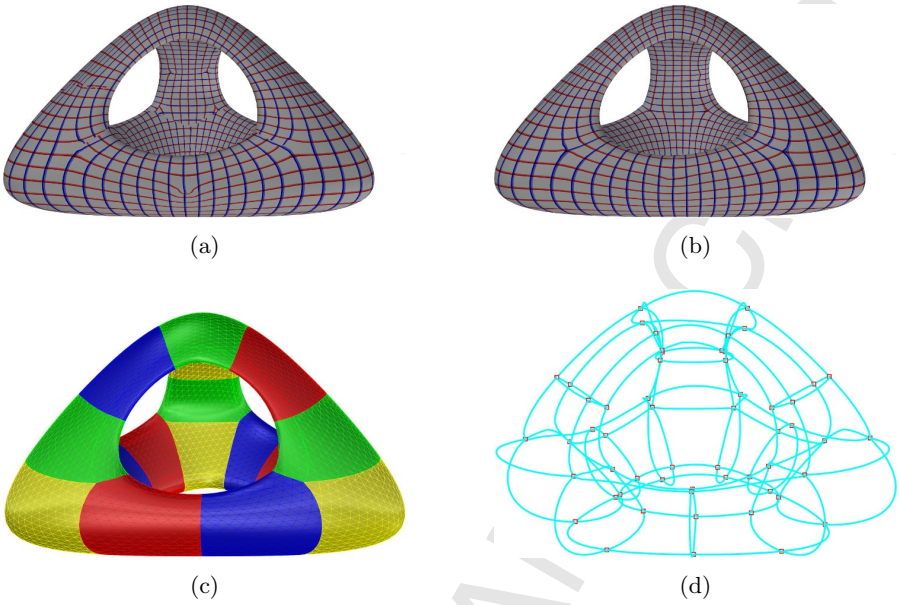


Figure 14: Generalized polycube optimization. The initial parameterization induced by the initial polycube (a). The optimized parameterization by relocating the polycube's nodes (b). The optimized cuboid patches (c). The embedding of the optimized polycube's nodes and arcs (d).

two steps. First we start from a random face and grow a topological disk by constructing a spanning tree of the dual mesh. Thus the primal of all non spanning tree edges is already a cut graph G which transforms M into a topological disk D. The size of this cut graph can be significantly reduced by iteratively removing all open paths. In the second step, paths connecting the cross field singularities to the cut graph are added.

- The cross-field is made consistent: the angles representing the field are changed so that the period jumps across all non-cut edges are equal to zero. It is possible to achieve this since the cut passes through all singularities. As the period jumps are all equal to zero at non-cut edges, if we arbitrarily label one of the directions of the field on a face f as $\mathbf{u}_f$, the label can be consistently propagated to all other faces. The $\pi/2$ rotated vectors of the cross-field are labeled $\mathbf{v}_f$. The vectors $\mathbf{u}_f$ and $\mathbf{v}_f$ are the target values for the gradients of the parametric coordinates ∇u and ∇v in the face f.

- The parameterization is computed as a solution to the constrained minimization problem:

$$\min_{(u,v)} \sum_{f \in F} (\|\nabla u - h\mathbf{u}_f\|^2 + \|\nabla v - h\mathbf{v}_f\|^2) A_f \tag{16}$$

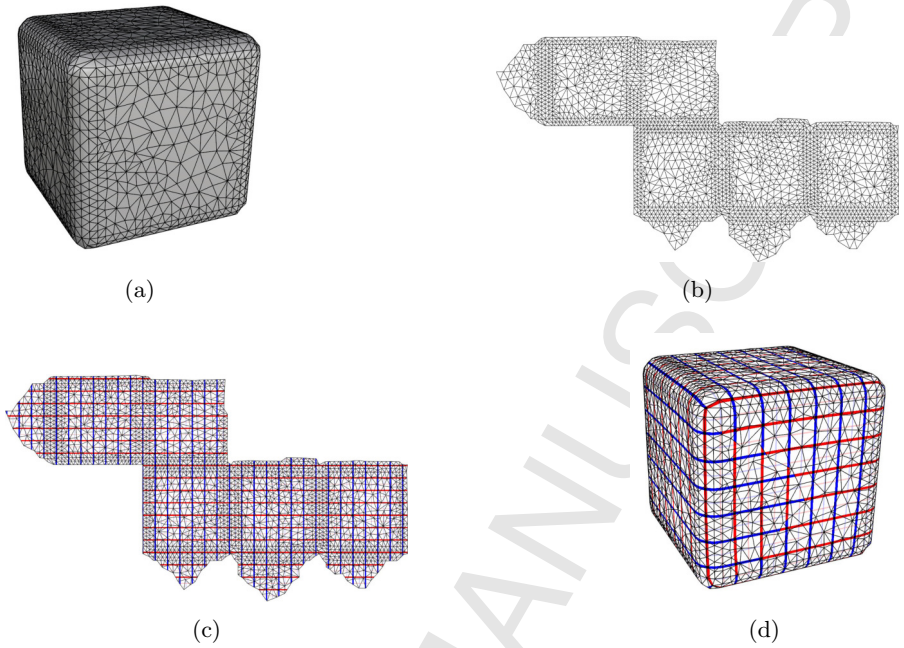


Figure 15: Global parameterization. A triangle mesh (a) and its parameterization (b). The intersection between the Cartesian Grid and the parameterization (c) is inversely mapped onto the mesh in order to obtain a quad mesh (d).

where A_f is the area of the face f, and h is a scale factor that sets the correspondence between the length scale of the parametric domain and the surface. The constraints imposed on (u, v) correspond to transitions across seams. We want the match across seams to be the same as for the guiding cross-field: if the $\mathbf{u}_f$ direction across a seam is transformed to a $\mathbf{v}_f$ direction, then the parametric directions are transformed in the same way. More precisely, if two faces f_i and f_j share a cut edge e_{ij}, with parametric positions of endpoint corners $\mathbf{p}_1^i$ and $\mathbf{p}_2^i$ on one side of the cut, and $\mathbf{p}_1^j$ and $\mathbf{p}_2^j$ on the other side, these are related by:

$$\mathbf{p}_1^j = R_e \mathbf{p}_1^i + \mathbf{t}_e \text{ and } \mathbf{p}_2^j = R_e \mathbf{p}_2^i + \mathbf{t}_e \tag{17}$$

where R_e is a $(\pi/2)p_{ij}$ rotation defined by the period jump p_{ij} of the cross-field on the edge e_{ij}, and $\mathbf{t}_e$ is an unknown translation.

3.3.2. Arcs Embedding Optimization

We have to ensure for each arc of the quad layout $\mathcal{L}$ that the two incident nodes ($\mathbf{n}_1$ and $\mathbf{n}_2$) lie on a common isoparametric curve. In other words, we want to change the parameterization so that a parametric line starting at $\mathbf{n}_1$ passes through $\mathbf{n}_2$. If a mesh can be parametrized without seams, the requirement of

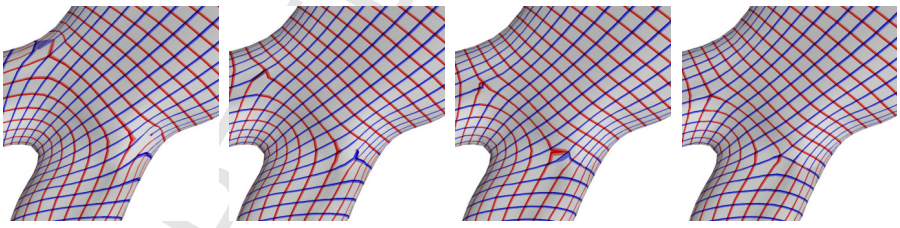
nodes alignment easily translates into a constraint on parameterization: both nodes should be on the same parametric line, i.e., share the same u or v value. For parameterizations with cuts, the situation is more complicated. A parametric line on the surface undergoes a jump to a different point and direction in the parametric space when it crosses a cut. While the rotation is entirely determined by the cross field to which the parameterization is aligned, the positional jump depends on the parameterization itself. The resulting constraint will depend not only on the pair of nodes ($\mathbf{n}_1$, $\mathbf{n}_2$), but also on the variable translational parts $\{\mathbf{t}_e\}$ at the cut edges we cross. In the general case, consider a path crossing cut edges e_i, $i = 0, ..., m$ between nodes $\mathbf{n}_1$ and $\mathbf{n}_2$. Assuming the final direction of the path is $\lambda(u, v) \in \{u, v\}$, and that $\mathbf{p}_1(u_1, v_1)$ and $\mathbf{p}_2(u_2, v_2)$ are the parametric positions of $\mathbf{n}_1$ and $\mathbf{n}_2$ respectively, then the complete constraint has the form [18]:

$$\left[\left(\prod_{i=0}^{m} R_{m-i}\right)\mathbf{p}_1 + \sum_{i=0}^{m-1}\left(\prod_{j=0}^{m-i-1} R_{m-j}\right)\mathbf{t}_i + \mathbf{t}_m\right]_{\lambda(u,v)} = \left[\mathbf{p}_2\right]_{\lambda(u,v)} \tag{18}$$

for either $\lambda(u, v) = u$ or $\lambda(u, v) = v$, where this subscript means taking the u or v coordinate.

3.3.3. Nodes Embedding Optimization

After computing a global parameterization, the quad layout's nodes are relocated based on the gradient of the parameterization's objective functional with respect to their positions. This is done iteratively until a local optimum of global embedding quality is reached. Campen and Kobbelt [4] described a method to perform this repositioning using an easy-to-implement estimator of the objective functional's true gradient (Figures 14 and 16).



(a) (b) (c) (d)

Figure 16: The evolution of the aligned global parameterization after 0 (a), 5 (b), 10 (c), and 15 (d) iterations. In the beginning severe distortions and inversions are present which successively vanish in the course of the optimization.

3.4. Trivariate Splines

We briefly review our previous work on trivariate splines fitting (for more details, see Al-Akhras et al. [1]). Using the quad mesh extracted from the optimized aligned global parameterization, a structured grid of points is generated on

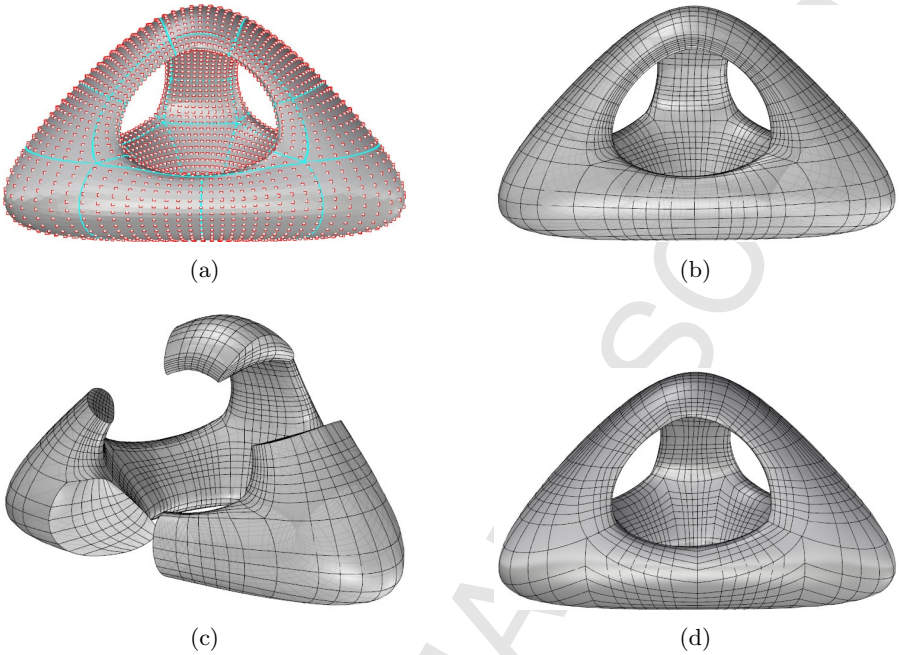


Figure 17: Trivariate splines fitting. The structured point grid extracted from the optimized aligned global parameterization (a). The reconstructed B-spline boundary surfaces (b). The reconstructed B-spline volume (c). The reconstructed B-spline boundary surfaces using local parameterization based on harmonic mapping. Notice the tangential discontinuity of the isoparametric lines across different patches (d).

each patch, and then used to fit the boundary B-spline surfaces [17]. For each cuboid, the volumetric parameterization is obtained using the reconstructed B-spline surfaces as boundary conditions. Keeping the boundary control points fixed, the interior control points of the B-spline solid are computed using Coons' interpolation [28]. The positions of the interior control points are then adjusted by minimizing a Laplacien based energy (Figure 17).

4. Results

The proposed method is implemented in VB.NET and C++. The software *Rhinoceros 5* is used for visualisation, and *RhinoCommon* for NURBS manipulations [25]. In addition, for solving the various sparse linear systems, we use the SuperLU [9], SuiteSparseQR [8], CHOLMOD [5], and CoMISo [3] solvers. All examples in this paper are performed on a 3.1 GHz Intel Xeon CPU with 64 GB Ram.

4.1. Analysis-Suitable Models

Figure 18 illustrates the obtained results for models with high genus, Figure 19 for models with sharp features, and Figures 20-21 for models with boundaries.

Cubic B-Splines were used in all examples.

4.2. Geometric Parametric Analysis

Many modern numerical models of real-life physical phenomenons pose challenges when used in numerical simulations, due to complexity and large size. As a consequence, novel methods are required in order to tackle not only non-linear problems but also large scale and parametric problems. This is the principal motivation behind Reduced Order Models (ROMs).

ROMs are computationally inexpensive mathematical representations that offer the potential for near real-time analysis. In the context of parametric studies, their construction requires accumulating a certain number of system responses to different input excitations. The accumulated system responses are called snapshots. The ROM is constructed by using a compression method aiming at computing a basis spanning the snapshots.

Nevertheless, a good prerequisite for ROM construction (in order to avoid a projection step) is the availability of solution vectors with the same dimension associated to all snapshots. In other words, their construction requires accumulating a certain number of simulations on isotopological meshes (i.e., meshes with the same number of nodes and connectivity) while modifying the input parameters. However for parametric studies based on geometrical parameters, isotopological meshes can be difficult to obtain when using general mesh morphing techniques in the case of a large variation of input parameters.

For models with different geometries but same topology, a potential solution to generate isotopological snapshots is the use of analysis-suitable isogeometric parameterization with the same canonical domain. In addition, isogeometric meshes are based on the parameterization of the model and can handle a large amount of distortions, hence can adapt to large variation of input parameters.

As an application on the plate model (Figure 19) with 6 geometrical parameters (Figure 22a), a simple analysis has been performed by fixing some edges and applying a normal pressure on others (Figure 22b). The ROM enables the construction of a numerical chart where the results of the simulation (in this case the total displacement) corresponding to different input parameters can be accessed in real time (Figures 22c-f). The choice of the snapshots and the precision of the obtained ROM are beyond the scope of this paper.

4.3. Statistical Shape Analysis

Statistical shape analysis is an analysis of the geometrical properties of some set of shapes by statistical methods. It has applications in various fields including medical imaging. For instance, it could be used to quantify differences between normal and pathological vessel, bone or organ shapes. One of the main methods used is principal component analysis. Such techniques require the creation of a proper isotopological shape model for all members of the population. An isogeometric representation over all shapes can be a useful tool. This technique is going to be illustrated on a database of abdominal aortas that present an aneurysm.

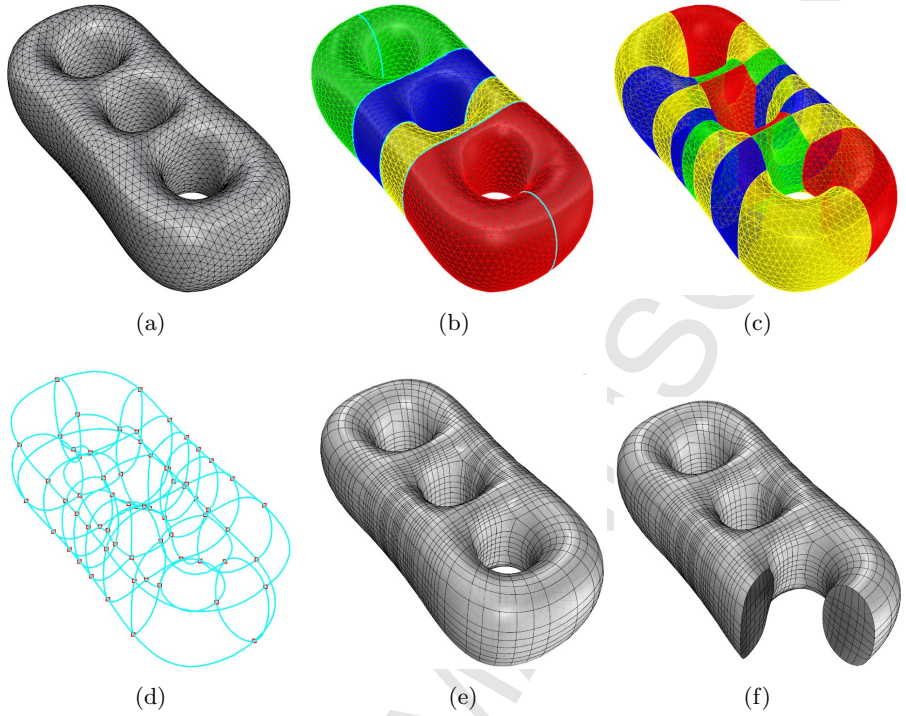


Figure 18: Block model : input triangulated boundary mesh (a); pants decomposition (b); cuboid decomposition (c); optimized polycube embedding (d); B-spline boundary surfaces (e); output B-spline volume (f).

Each input triangle mesh is homeomorphic to a pants patch. Using the cuboid decomposition algorithm, a polycube approximating the input mesh is generated. Then the parameterization is optimized until a local optimum of global embedding quality of the polycube is reached. The control quadrilateral mesh is finally extracted from the optimized global aligned parameterization. Figure 4 illustrates the complete algorithm, and Figure 16 illustrates the aligned global parameterization during the course of the optimization of the polycube's embedding.

For the subsequent principal component analysis using singular value decomposition, all aortas must have the same representation, i.e., meshes with the same number of nodes and connectivity. This can be implicitly achieved by mapping all the input meshes to the same canonical domain. In addition since each decomposition is optimized to fit the given input geometry, not only the output meshes will have the same topology, but they have the optimal geometry representing each shape. Figures 23a-f illustrates isotopological meshes for different patients, and Figure 23g illustrates the morphing between two aortas with different morphologies.

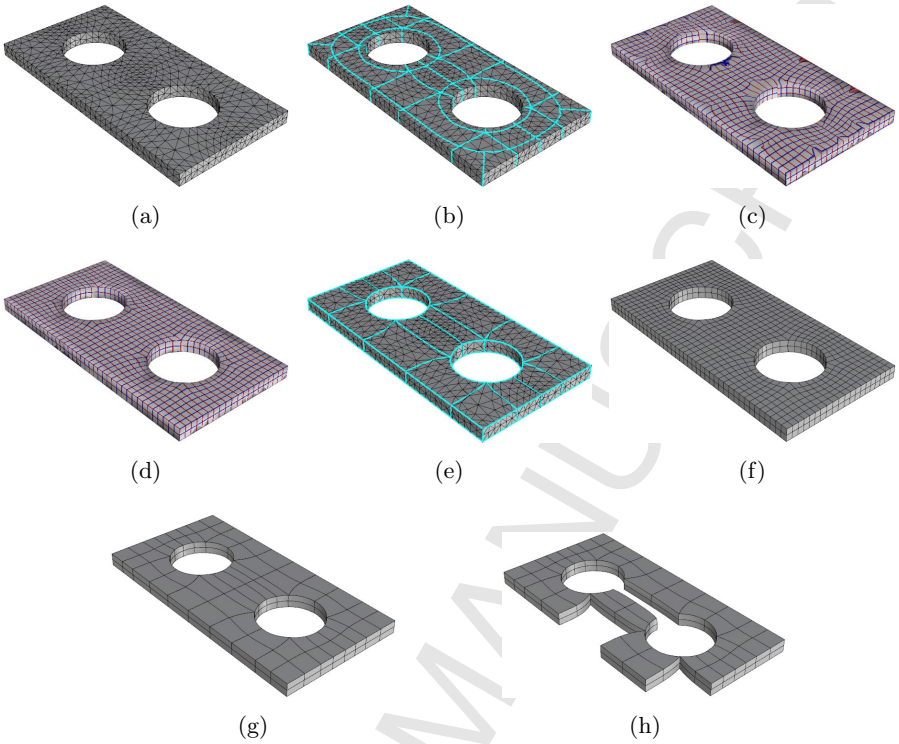


Figure 19: Plate model : input triangulated boundary mesh (a); initial polycube embedding (b); initial aligned global parameterization (c); optimized aligned global parameterization (d); optimized polycube embedding (e); extracted quad mesh (f); B-spline boundary surfaces (g); output B-spline volume (h).

5. Conclusion

In this paper, we proposed a framework to automatically compute a volumetric parameterization for a solid model defined by its triangulated boundary. Based on the input triangulated boundary, we first decompose the solid model into cuboids in two steps: pants decomposition and cuboid decomposition. The set of cuboids compose a generalized polycube approximating the input model. Due to its regular structure, the polycube is suitable for serving as the parametric domain of the tensor-product spline representation required for isogeometric analysis. Using aligned global parameterization, the polycube is optimized in a way to achieve low overall patch distortion, and alignment to principal curvature directions and sharp features. The core of the optimization algorithm is a new way to design cross fields on surface which satisfy exactly a given set of singularities while fitting as closely as possible a given set of directional constraints. The surface spline representation for the model is then constructed as an approximation of the inverse of the computed parameterization. Finally, the volume spline representation is computed using the surface spline as boundary

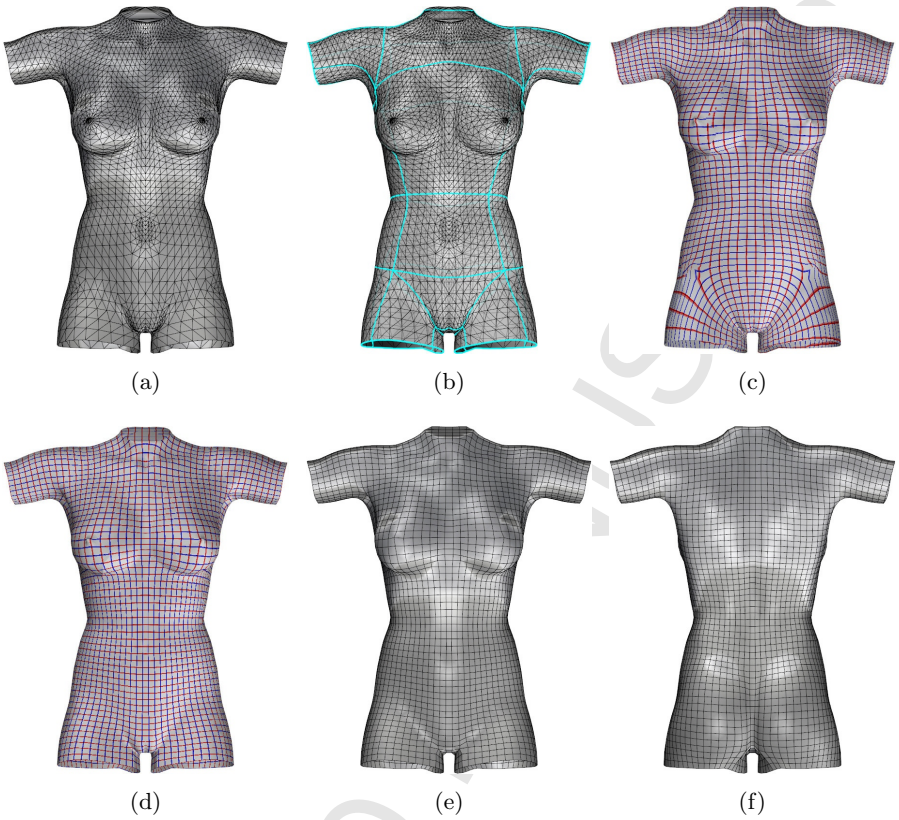


Figure 20: Bust model : input triangulated boundary mesh (a); initial polycube embedding (b); initial aligned global parameterization (c); optimized aligned global parameterization (d); extracted quad mesh (e-f).

condition.

A given surface admits infinitely many pants decomposition. In general, not all pants decomposition results are suitable for the following cuboid decomposition algorithm. The pants-to-cuboids algorithm is very robust and still going to generate the corresponding cuboid decomposition even for very distorted pants patches. However, the extracted volume parameterization is greatly affected by the quality of the initial pants decomposition and might contain very distorted elements. In practice satisfying results were obtained if the pants decomposition is guided by the different geometric criteria (shortest length, symmetry, minima rule).

Pants patches can be geometrically assimilated to three branches meeting together. The cuboid decomposition algorithm generates only T-shaped configurations. However other configurations might be more optimal in some cases in order to capture the geometrical features of the model. In order to select the

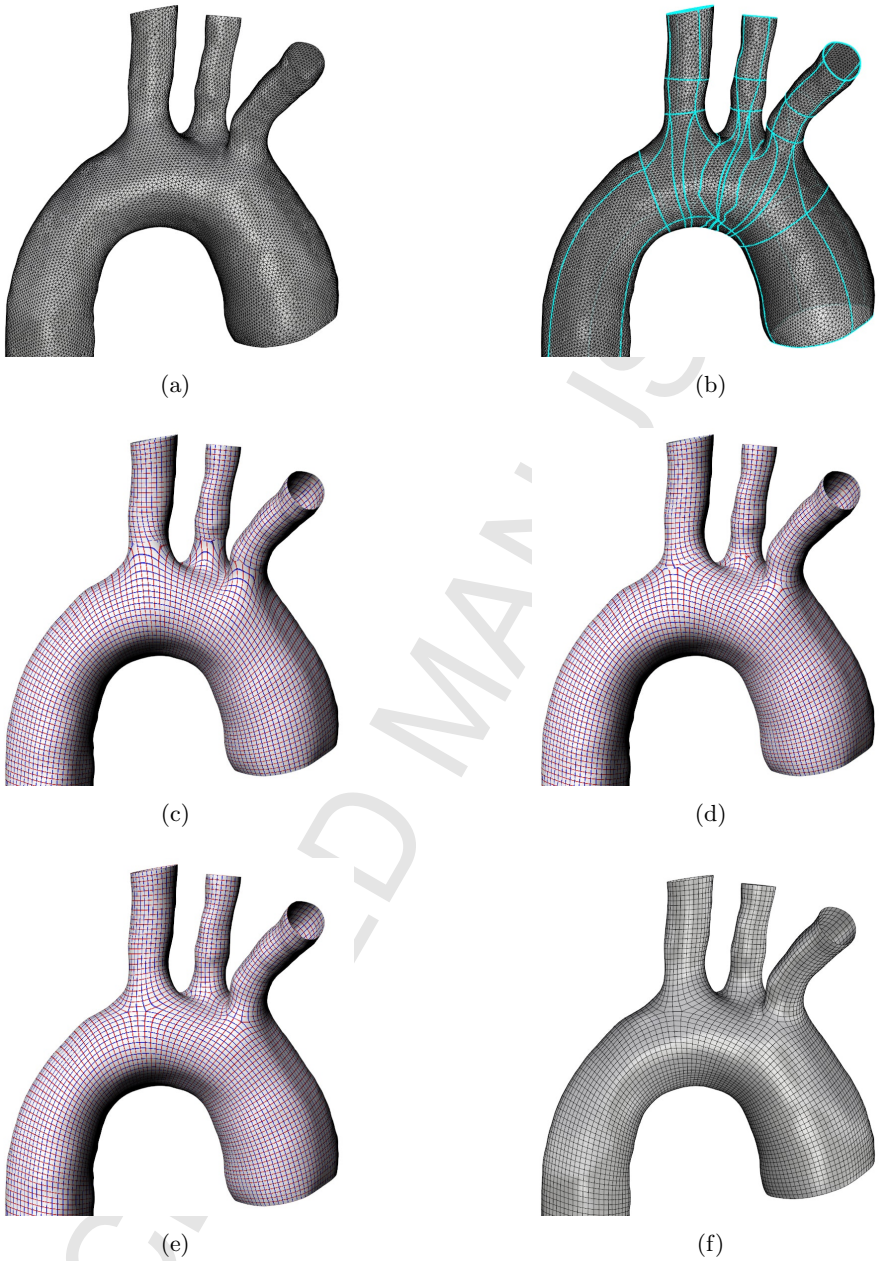


Figure 21: Aortic-cross model: boundary triangle mesh (a); initial polycube embedding (b); initial (c), intermediate (d) and optimized (e) aligned global parameterization; extracted quad mesh (f).

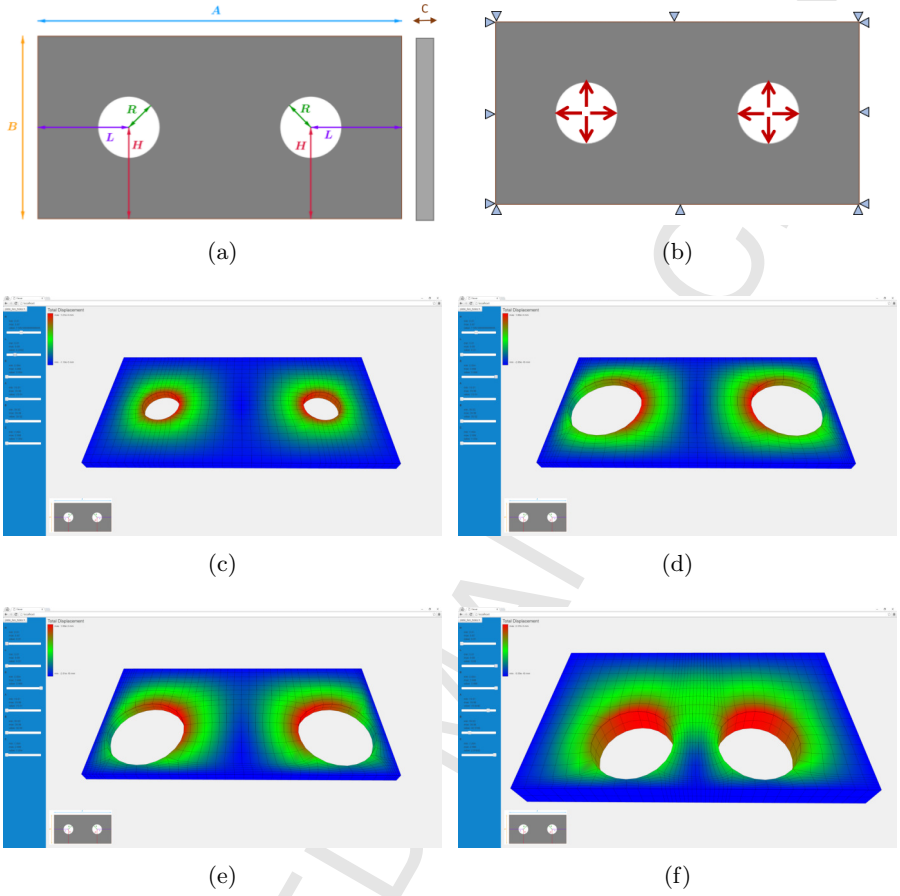


Figure 22: Parametric plate model : the parametric model with 6 geometrical parameters (a); a simple linear analysis is performed by fixing the plate's edges and applying normal pressure of the holes' edges (b); the ROM enables the construction of a numerical chart where the results of the simulation (in this case the total displacement) corresponding to different input parameters can be accessed in real time (c-f).

most suitable configuration in a general way, a potential solution is to extract the skeleton of the model and select the cuboids' configuration in a skeleton-aware manner.

However despite the limitations mentioned above, the proposed method has a great potential in the context of reduced order modeling for geometric parametric studies. Even if the initial polycube decomposition had to be made semi-automatically or manually, it has to be done only once on a generic model. Then this decomposition can be transferred in a very approximate manner to all geometric instances. Finally using the optimized parameterization (based on the proposed cross field design algorithm), the initial approximate polycube

can be optimized to fit the geometry of each instance. This implicitly defines a morphing between the parameterizations of the different parametric instances. In addition, each parameterization is guaranteed to be optimal for a given geometric instance.

6. Acknowledgments

We gratefully acknowledge the French National Research Association (ANRT), and ANSYS France for their support. The "TETRA", "TORSO" and "BLOCK" models are courtesy of the AIM@ShapeProject. We thank Aline Brunon (LaMCoS) for her help in providing the "Bust" model, and Daniel Silva Soto (University of Sheffield) for his help in providing the "Aortic Cross" model. The "Aorta" triangle meshes are obtained from preliminary analysis of preoperative CT scans (using the software EndoSize, Therenva). This analysis is conducted by the vascular surgery department of the CHU (Centre Hospitalier Universitaire, Rennes, France) and the LTSI (INSERM U1099, Université de Rennes 1, France) as part of a research protocol on EndoVascular Aneurysm Repair (EVAR) interventions. The study protocol was approved by the CHU's ethics committee and patient consents were obtained before including their anonymous data into the database.

References

[1] Al-Akhras, H., Elguedj, T., Gravouil, A., Rochette, M., 2016. Isogeometric analysis-suitable trivariate nurbs models from standard b-rep models. Computer Methods in Applied Mechanics and Engineering 307, 256–274.

[2] Bommes, D., Zimmer, H., Kobbelt, L., 2009. Mixed-integer quadrangulation. ACM Transactions on Graphics 28 (3), 77:1–77:10.

[3] Bommes, D., Zimmer, H., Kobbelt, L., 2012. Practical mixed-integer optimization for geometry processing. Proceedings of the 7th International Conference on Curves and Surfaces, 193–206.

[4] Campen, M., Kobbelt, L., 2014. Quad layout embedding via aligned parameterization. Computer Graphics Forum 33 (8), 69–81.

[5] Chen, Y., Davis, T., Hager, W., Rajamanickam, S., 2008. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Transactions on Mathematical Software 35 (3), 22:1–22:14.

[6] Cohen, E., Martin, T., Kirby, R., Lyche, T., Riesenfeld, R., 2010. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 199, 334–356.

[7] Crane, K., Desbrun, M., Schröder, P., 2010. Trivial connections on discrete surfaces. Computer Graphics Forum (SGP) 29 (5), 1525–1533.

[8] Davis, T., 2008. A multithreaded multifrontal sparse qr factorization. University of Florida.

[9] Demmel, J., Eisenstat, S., Gilbert, J., Li, X., Liu, J., 1999. A supernodal approach to sparse partial pivoting. SIAM J. Matrix Analysis and Applications 20 (3), 720–755.

[10] Hatcher, A., 2001. Algebraic Topology.

[11] Hughes, T., Cottrell, J., Bazilev, Y., 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering 195 (39-41), 4135–4195.

[12] Kälberer, F., Nieser, M., Polthier, K., 2007. Quadcover - surface parameterization using branched coverings. Computer Graphics Forum 26 (3), 375–384.

[13] Lai, Y.-K., Jin, M., Xie, X., He, Y., Palacios, J., Zhang, E., Hu, S.-M., Gu, X., 2010. Metric driven rosy field design and remeshing. IEEE Transactions on Visualization and Computer Graphics 16 (1), 95–108.

[14] Li, B., Li, X., Wang, K., Qin, H., 2010. Generalized polycube trivariate splines. In: Proceedings of the 2010 Shape Modeling International Conference. SMI '10. pp. 261–265.

[15] Li, B., Li, X., Wang, K., Qin, H., 2012. Surface mesh to volumetric spline conversion with generalized poly-cubes. IEEE Tansactions On Visualization And Computer Graphics 19 (9), 1539–1551.

[16] Li, W.-C., Valletand, B., Ray, N., Lévy, B., 2006. Representing higher-order singularities in vector field on piecewise linear surfaces. IEEE Tansactions On Visualization And Computer Graphics 12 (5), 1315–1322.

[17] Ma, W., Kruth, J., 1995. Parameterization of randomly measured points for least squares fitting of b-spline curves and surfaces. Computer-Aided Design 27 (9), 663–675.

[18] Myles, A., Pietroni, N., Kovacs, D., Zorin, D., 2010. Feature-aligned t-meshes. ACM Transactions on Graphics 29 (4), 117:1–117:11.

[19] Myles, A., Zorin, D., 2012. Global parameterization by incremental flattening. ACM Transactions on Graphics 31 (4), 109:1–109:11.

[20] Nieser, M., Palacios, J., Polthier, K., Zhang, E., 2010. Hexagonal global parameterization of arbitrary surfaces. ACM SIGGRAPH ASIA 2010 Sketches, 5:1–5:2.

[21] Pilgerstorfer, E., Jüttler, B., 2014. Bounding the influence of domain parameterization and knot spacing on numerical stability in isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 268, 589–613.

[22] Ray, N., Li, W. C., Lévy, B., Sheffer, A., Alliez, P., 2006. Periodic global parameterization. ACM Transactions on Graphics 25 (4), 1460–1485.

[23] Ray, N., Vallet, B., Alonso, L., Lévy, B., 2009. Geometry aware direction field processing. ACM Transactions on Graphics 29 (1), 1:1–1:11.

[24] Ray, N., Vallet, B., Li, W. C., Lévy, B., 2008. N-symmetry direction field design. ACM Transactions on Graphics 27 (2), 10:1–10:13.

[25] RhinoCommon, 2013. http://wiki.mcneel.com/developer/rhinocommon.

[26] Tarini, M., Hormann, K., Cignoni, P., Montan, C., 2004. Polycube-maps. ACM Transactions of Graphics 23 (3), 853–860.

[27] Wang, W., Zhang, Y., Liu, L., Hughes, T. J. R., 2013. Trivariate solid t-spline construction from boundary triangulations with arbitrary genus topology. Computer-Aided Design 45, 351–360.

[28] Wang, X., Gian, X., 2014. An optimization approach for constructing trivariate b-spline solids. Computer-Aided Design 46, 179–191.

[29] Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2013. Optimal analysis-aware parameterization of computational domain in 3d isogeometric analysis. Computer-Aided Design 45, 812–821.

[30] Zhang, Y., Wang, W., Hughes, T. J. R., 2012. Solid t-spline construction from boundary representations for genus-zero geometry. Computer Methods in Applied Mechanics and Engineering 201.

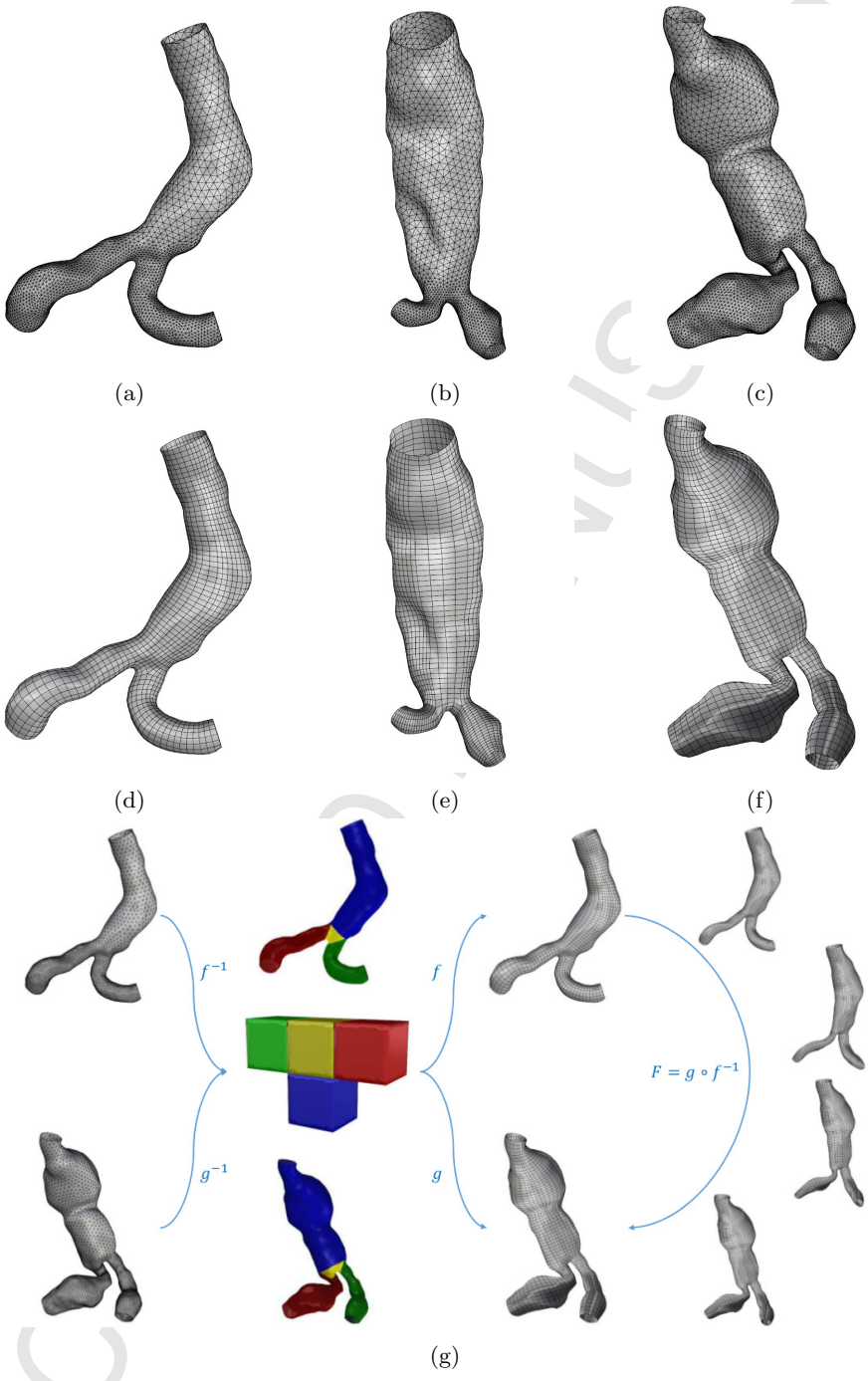


Figure 23: Aorta model : input triangle meshes (a-c); output isotopological quadrangular meshes (d-f); morphing between two aortas with different morphologies (g).

- HIGHLIGHTS
- We construct a volumetric NURBS parameterization from standard CAD models.
- The solid region is first decomposed into a set of cuboids using pants decomposition.
- The set of cuboids compose a polycube approximating very roughly the geometry of the model while faithfully replicating its topology.
- An optimized global parameterization (aligned to a cross field) is computed between the solid's boundary and the polycube's boundary.
- Based on this mapping, surface and volume parameterizations are generated.