

NFN - Nationales Forschungsnetzwerk

Geometry + Simulation

<http://www.gs.jku.at>



The isogeometric segmentation pipeline

M. Pauley, Dang-Manh Nguyen, D.
Mayer, J. Špeh, O. Weeger, Bert
Jüttler

G+S Report No. 31

June 2015



Der Wissenschaftsfonds.



The isogeometric segmentation pipeline

Michael Pauley, Dang-Manh Nguyen, David Mayer, Jaka Špeh, Oliver Weeger,
Bert Jüttler

Abstract We present a pipeline for the conversion of 3D models into a form suitable for isogeometric analysis (IGA). The input into our pipeline is a boundary represented 3D model, either as a triangulation or as a collection of trimmed non-uniform rational B-spline (NURBS) surfaces. The pipeline consists of three stages: computer aided design (CAD) model reconstruction from a triangulation (if necessary); segmentation of the boundary-represented solid into topological hexahedra; and volume parameterization. The result is a collection of volumetric NURBS patches. In this paper we discuss our methods for the three stages, and demonstrate the suitability of the result for IGA by performing stress simulations with examples of the output.

1 Introduction

In isogeometric analysis (IGA), the space of functions used to approximate the solution of a differential equation is the same space of functions used to represent the geometry. As such, IGA methods work directly with geometry represented by splines and have begun to close the gap between the geometry representations used for analysis and those used for design [1]. An additional benefit of IGA is that it can achieve the same accuracy as the finite element method (FEM) with fewer degrees of freedom (see [4]).

Michael Pauley, Dang-Manh Nguyen, Jaka Špeh, Bert Jüttler, David Mayer
Institute of Applied Geometry, Johannes Kepler University, Altenberger Straße 69, 4040 Linz,
Austria. e-mail: {Michael.Pauley|Jaka.Speh|Bert.Juettler}@jku.at, david_
mayer@aon.at, dmnguyen@cise.ufl.edu
Oliver Weeger
TU Kaiserslautern, Faculty of Mathematics, P.O. Box 3049, 67653 Kaiserslautern, Germany. e-
mail: oliver_weeger@sutd.edu.sg

In order to realize the benefits of IGA when dealing with complex geometries, it is necessary to segment them into IGA-suitable pieces. A rich theory of FEM-suitable mesh generation exists (see e.g., [5, 8, 14, 15, 17, 21, 24]) but these methods aim at a segmentation of a solid into a large number of (approximately uniform) small tetrahedra and hexahedra. The goal of the *isogeometric segmentation pipeline* is to convert a boundary represented model (e.g. something designed in computer aided design (CAD) software) into an IGA-suitable volumetric model. For us this means segmenting the geometry into a small number of *topological hexahedra*, that is, deformed cubes.

Methods for parameterization of solids by a single B-spline or T-spline cube are developed in [13, 20, 25, 26]. A method for multiple patches is given in [22] which requires a predefined segmentation as input. Several papers [9, 10, 19, 23] provide methods for segmentation and parameterization of a solid by B-spline or T-spline cubes, allowing for singularities in the output.

The input into the pipeline can be either a triangulated boundary representation, or a trimmed non-uniform rational B-spline (NURBS) surface boundary representation. A trimmed surface consists of a *master spline* mapping a square into \mathbb{R}^3 , together with a collection of *trimming curves* defining a domain of the surface as a subset of the square. The boundary representation for the pipeline consists of a collection of trimmed surfaces, together with adjacency information. The output of the pipeline is a representation of the solid as a collection of volumetric NURBS. The pipeline consists of the three following stages.

CAD model reconstruction. A triangulated model is converted into a trimmed NURBS surface boundary representation, for input into the remaining stages. This step is not necessary for models that are already represented with trimmed surfaces. It is necessary when only a triangulated solid is available, or when CAD data is available but uses features not supported by the remainder of the pipeline. In the latter case, the CAD model can be triangulated and then reconstructed to produce a solid in a format compatible with the pipeline.

Isogeometric segmentation. A trimmed NURBS surface boundary represented solid is segmented into a collection of boundary represented topological hexahedra. Our segmentation approach, which began development in [6, 11], is based on the edge graph of the solid – the graph defined by the vertices and edges between the trimmed surfaces.

In its present state, our isogeometric segmentation method requires manual interaction in two scenarios. Firstly, it is applicable to the class of contractible solids with 3-vertex-connected edge graphs. This means that a manual preprocessing step is required to ensure that the input satisfies these assumptions. In our ongoing work we are developing methods to automatically ensure these assumptions. Secondly, a post-processing step may be necessary if the segmentation is required to be free of T-joints. So far, we only have a way to eliminate them in special cases. Our belief is that T-joints will not be a barrier to IGA, as contemporary research is developing methods for handling non-matching interfaces ([7]).

Volume parameterization. A collection of boundary represented topological hexahedra is converted to a collection of volumetric splines.

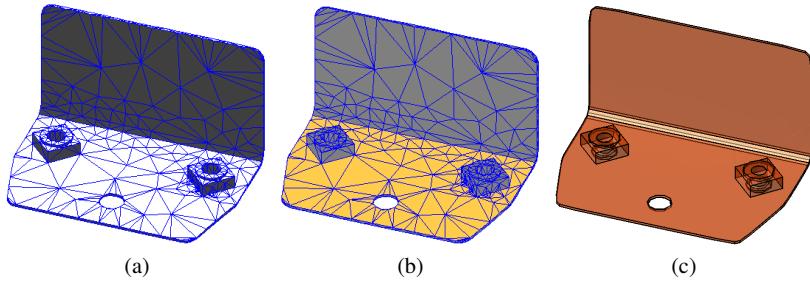


Fig. 1 (a) The triangulated car part. (b) The model is separated into regions. We highlight one here. (c) The CAD model reconstruction (translucent).

Sections 2, 3, 4 of the paper provide details and examples for the above stages of the pipeline. In Section 5, a segmentation produced by our pipeline is used to conduct mechanical simulations, showing the suitability of our results for isogeometric analysis. Section 6 summarizes the state of the pipeline and outlines our ongoing and future work.

2 Segmentation of triangulated surfaces and CAD model reconstruction

Consider a complex model for which IGA-based simulation is required. If a collection of trimmed NURBS surfaces is provided describing its boundary, we can segment it into topological hexahedra using the methods in Sections 3 and 4. However, often we only have access to a triangular mesh describing the boundary. This would happen, for example, if a triangulation of a model is available but the original model is not. Furthermore, if a CAD model is of low quality or uses features that are not supported by the segmentation software, it is useful to convert it into a triangle mesh from which a more suitable trimmed surface model can be created. Finally, a more complex CAD model is often obtained by combining several models produced by several users, possibly with different CAD systems. In this situation, using a triangulated CAD model is often the only feasible option.

Sophisticated methods exist for the reverse engineering of CAD data (see [18] and its references). The method we use can afford to be relatively straightforward because (i) we are primarily interested in engineering objects, which tend to have simple faces and sharp edges, compared to general, more free-form designs; (ii) we assume a high quality triangulation is provided as input, not noisy point cloud data; (iii) while more sophisticated methods can reproduce blend surfaces, we do not want them as the later stages of our pipeline are not prepared to deal with them.

- The triangulated boundary is segmented into regions using a *region growing algorithm*. Two triangles are part of the same region if there exists a path between

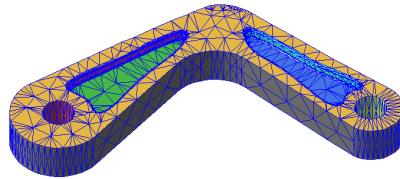


Fig. 2 The region growing algorithm segments the triangulation of the TERRIFIC demonstrator part into 9 regions. Different regions are shaded different colors.

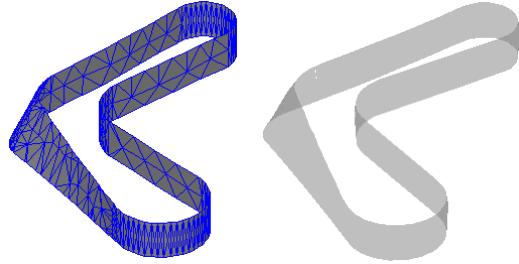


Fig. 3 Example of a NURBS trimmed surface fit to triangulated data.

them which passes only between triangles that meet at an angle close to π . Post-processing can be applied to further segment the large regions with a more strict angle tolerance, or to merge small regions with a less strict angle tolerance. An example is shown in Figure 2.

- For each region, a triangulated domain is constructed whose triangles are in correspondence with the triangles of the region. The points in the interior of the domain, and the positions of holes in the domain, are chosen by a method based on [2, 3].
- A tensor product NURBS master surface is constructed, by fitting the triangulation data using a least-squares approach. Finally, sharp corners in the domains are smoothed out where appropriate. An example of a surface fit to a detected region is shown in Figure 3.

2.1 Examples

We demonstrate the procedure with the following examples.

Car part example. The procedure is applied to a car part consisting of 1220 triangles. Angle tolerance is set to 40° and a secondary region growing phase is applied, with a tolerance of 20° , to those regions with area more than 20% of the average. The result is shown in Figure 1 (a–c).

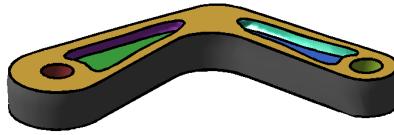


Fig. 4 The TERRIFIC Demonstrator: reconstructed CAD model.

TERRIFIC Demonstrator. The TERRIFIC demonstrator model was created as part of the TERRIFIC European Commission project for testing and demonstrating the project’s research and development. The demonstrator in its original format is a boundary represented CAD model. However, feeding a triangulated version of the model through the CAD model reconstruction step allows us to clean the model up, eliminating the blends which the later steps of our pipeline do not support.

The model has 3672 triangles. The region growing algorithm is used with a cutoff of 12° and regions are merged together if they have area less than 200% of the average. The result is shown in Figure 4.

3 Isogeometric segmentation of boundary represented solids

Here we describe our approach to the problem of segmenting a boundary-represented solid into topological hexahedra. The solid has sharp features, and its faces are described by NURBS-based trimmed surfaces.

The faces, edges and vertices of the solid form a structure called the *edge graph* of the solid (see Figure 5). A graph is 3-vertex-connected if it is connected, it has at least 4 vertices and any 2 vertices can be deleted without disconnecting the graph. We make the following assumptions about the solid:

- the solid is contractible;
- the edge graph is simple and 3-vertex-connected.

Once these assumptions are satisfied, we have an automatic process to obtain a segmentation into topological hexahedra. The process is as follows.

1. Ensure that the solid satisfies the assumptions.
2. For each existing non-convex edge, segment the solid using a cutting loop (Sections 3.1, 3.2) or an extending loop (Section 3.4) containing the edge.
3. If the solid has no non-convex edges, segment it further using cutting loops until we arrive at pre-defined *base solids*.
4. Segment the base solids in a predefined way into topological hexahedra.

Item 1 above is ongoing work: methods based on the concept of a Reeb graph from Morse theory, which has previously been used to good effect in *flow volume decomposition*.

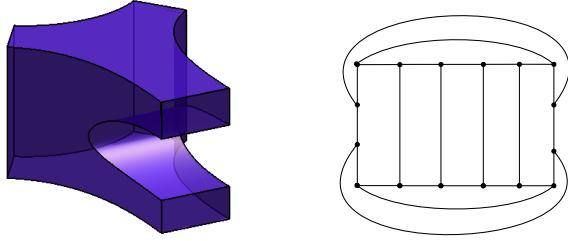


Fig. 5 An example of a contractible solid and a planar representation of its edge graph.

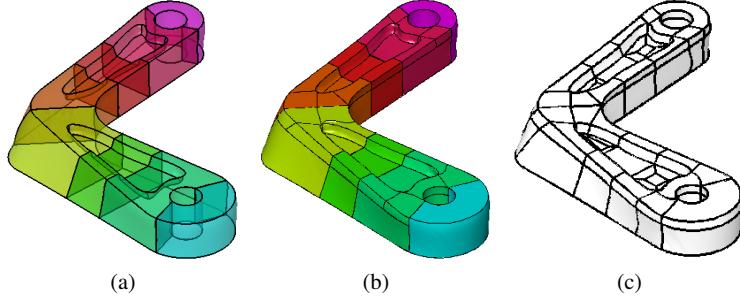


Fig. 6 (a) The TERRIFIC Demonstrator, preprocessed to cut it into contractible pieces with 3-vertex-connected edge graphs. (b) Segmentation into topological hexahedra. (c) The solid is further segmented to eliminate T-joints.

position ([16]), show promise for dealing with non-contractible solids. Items 2–4 are explained below in Sections 3.1–3.4.

Figure 6 shows an example: this stage of the pipeline is applied to the TERRIFIC demonstrator. The demonstrator is first preprocessed to cut it into contractible pieces and artificial edges are added to make its graph 3-vertex-connected. Then these pieces are segmented into hexahedra using the methods of cutting loops and extending loops described in the following sections.

3.1 Theory for contractible solids with only convex edges

An edge e of the solid is *convex* at a point p if the two incident faces at e meet at a convex interior angle at p . The edge e is called *convex* if it is convex at every point, and non-convex otherwise. Non-convex edges have a significant impact on the options for segmentation. In this section we restrict to solids with no non-convex edges; we extend our approach to the more general case in Section 3.2.

Our approach is based on the concept of a *cutting loop*; a cycle in the edge graph consisting of existing edges as well as *auxiliary edges* which can be created between

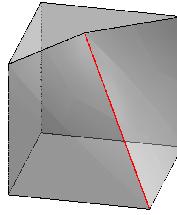


Fig. 7 A solid with a non-convex edge. The edge graph is isomorphic to that of a cube. However, the non-convex edge prevents C^1 volume parameterization by a cube. The segmentation algorithm eliminates non-convex edges as its first priority so that the resulting topological hexahedra have only convex edges.

any two vertices on the same face. A cutting loop is *valid* if it can be used as the boundary of a surface with the following properties:

- the surface is contained in the solid;
- the surface cuts the solid into exactly 2 pieces;
- both pieces satisfy our original assumptions, i.e., they are contractible and the edge graphs are simple and 3-vertex-connected.

For solids with only convex edges, validity is essentially a combinatorial condition, i.e., the structure of the graph matters but the geometry of the trimmed surfaces does not. This allows us to guarantee the existence of a valid cutting loop.

Theorem 1 ([6]). *Assume that all edges are convex. If the edge graph of the solid is not that of a tetrahedron, then a valid cutting loop exists.*

3.2 Theory for contractible solids with non-convex edges

Non-convex edges create additional complications. They must be eliminated during the segmentation process, because they prevent smooth parameterization (see Figure 7). Furthermore, they create new validity criteria for cutting loops, which can sometimes require *auxiliary vertices* (new vertices created along the existing edges of the edge graph) to guarantee the existence (See [11, Section 3]).

Theorem 2 ([11]). *Given a non-convex edge of the solid, there exists a valid cutting loop containing that edge and possibly containing auxiliary vertices, such that cutting the solid with this loop eliminates the non-convex edge.*

This enables an extension of the algorithm outlined in Section 3.1. Cutting loops are used to cut the solid to remove all the non-convex edges. Once the solid has been segmented into pieces with only convex edges, the method of Section 3.1 can be applied.

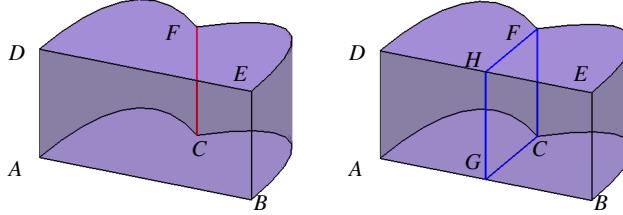


Fig. 8 Left: a solid with a non-convex edge CF . Without using auxiliary vertices, there is no valid cutting loop to cut CF , but it becomes possible with the addition of the vertices G and H .

3.3 Description of the algorithm

Theorems 1 and 2 mean that the following algorithm successfully segments the solid into topological hexahedra, given its edge graph structure:

- search for a valid cutting loop;
- use the cutting loop to construct a surface which cuts the solid into two pieces. The geometric aspects of this step are difficult. Firstly, any auxiliary edges of the loop must be turned into real edges by constructing a curve cutting the face. This problem is dealt with in the paper [12]. Secondly, a trimmed surface must be constructed having the cutting loop as its boundary;
- repeat the procedure for the two pieces until we arrive at *base solids*: a class of solids which have predefined segmentations into topological hexahedra. Our base solids include tetrahedra, hexahedra and prisms.

Each of the base solids is further split into topological hexahedra using pre-defined templates. More details of the algorithm are provided in [6, 11].

Typically, many valid cutting loops exist so a decision procedure is needed to pick one. We decide based on a cost which is a combination of the following components:

Combinatorial component. Aiming for a small number of hexes, our cost includes a component based on the graph theoretic structure:

- the number of edges of the cutting loop (e.g. preferring loops of length 4 since they more quickly lead to topological hexahedra).
- the number of edges of the newly created faces when a face is split by an auxiliary edge (e.g. preferring to split a 6-sided face into two 4-sided faces vs. a 5- and a 3-sided face).

Geometric component. Aiming for hexes with good shape, we include measurements of the planarity of the cutting loop.

A more detailed discussion, with examples, is provided in [11, Sections 5–6].

3.4 Surface extension

The cutting loop approach can sometimes miss elegant segmentations that could be created by extending existing surfaces. For example, [11] provided an example where using the cutting loop method alone results in a segmentation with 31 topological hexahedra. Furthermore, they do not have ideal shape. Extending existing faces would allow a segmentation with only 4 hexes and higher quality shape (see Figure 9). Surface extension can be done by modifying cutting loops into *extending loops* (see Figure 10). In contrast to cutting loops, extending loops result in the construction of two different faces for the two new solids.

Extending loops can be defined as cutting loops with additional structure: each edge of the loop carries data of whether it extends one of the two incident faces, and if so, which one. The validity of an extending loop can be determined from the way faces meet at vertices and edges. Specifically,

- *it is only possible to extend a surface through a given edge e if e is everywhere concave.* This is because, if e is convex at any point, then the surface would be extended to outside the solid;
- at a vertex v , consider the set of unit tangent vectors pointing into the solid. This set is a region of the unit sphere, whose boundary is a spherical polygon. See Figure 11 for an example. Three faces A, B, C of a solid meet at a vertex v ; the resulting spherical polygon has three corresponding great arcs $\tilde{A}, \tilde{B}, \tilde{C}$. *The geometry of this polygon determines the options available for extending a surface through v .* In the example of Figure 11, face B can be extended through v , resulting in face A being cut. This is because extending arc \tilde{B} into the interior of the spherical polygon, it eventually intersects arc \tilde{A} . A similar situation arises when extending face C .

A slightly different situation occurs for extending face A through v . If arc \tilde{A} is extended into the interior of the polygon, it does not intersect the interior of any of the other arcs, but instead closes up into a great circle. As a consequence, face A can be extended through v in such a way that the vertex v can be deleted from one of the two solids resulting from the cut. The question of whether two faces, meeting at a vertex, can be extended simultaneously into a single face, can also be answered using the geometry of the spherical polygon.

Compare these facts with [11, Proposition 2], where it is shown that the *spherical visibility problem* for the spherical polygon constructed above plays an important role in determining the validity of a cutting loop.

Two possible results of cutting the solid of Figure 11 (left) are shown in Figure 12.

The selection of an extending loop is similar to one for a cutting loop. A cost is assigned based on combinatorial and geometric properties. Minor complications arise from the fact that vertices can be deleted, and the new faces in the two resulting solids can have a different number of vertices.

There remains the question of the geometric realization of the extended surface, i.e., how do we construct the NURBS-based trimmed surface structure for a face which includes the existing face(s) and the newly constructed part? This question is

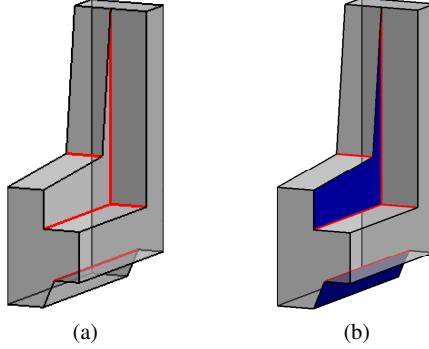


Fig. 9 Half-chair example. The solid in (a) was segmented into 31 topological hexahedra in [6]. However, a segmentation into just 4 topological hexahedra can be produced by extending the faces marked in (b) and merging them together as the first step. This motivates our ongoing work on automatic surface extension as a step in the segmentation procedure.

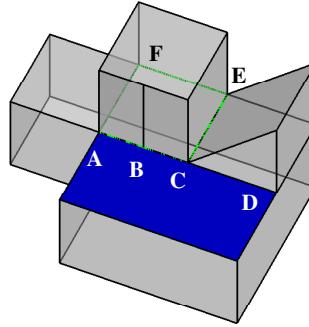


Fig. 10 Example of a solid with some surfaces that can be extended. Loop **A-B-C-E-F** is an extending loop which can be used to extend the highlighted face and cut the solid. An extending loop cannot contain the path **A-B-C-D** due to the geometry at vertex C.

still open; our current attitude is that it is better to only construct the new part, represent the extended face as a union of trimmed surfaces, and merge them together only at the time of volume fitting. This avoids creating an extra source of approximation.

Examples

We demonstrate the segmentation process for several models. For these models, a preprocessing step is applied to ensure that the model is contractible and the edge graph is simple and 3-vertex connected. This pre-processing step is ad-hoc as we have not developed a general method.

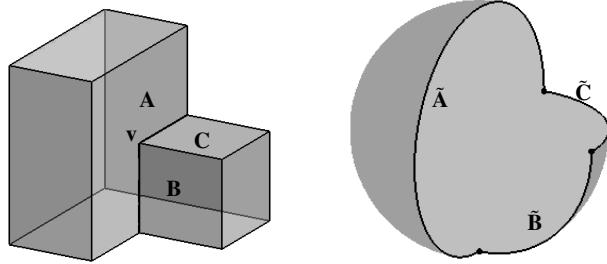


Fig. 11 In the solid (left), three faces $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are incident to a vertex \mathbf{v} . The unit vectors at \mathbf{v} pointing into the solid define a filled spherical polygon with arcs $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}$ containing the tangent vectors pointing along the faces $\mathbf{A}, \mathbf{B}, \mathbf{C}$. The geometry of the spherical polygon determines the available options for extending these faces through \mathbf{v} .

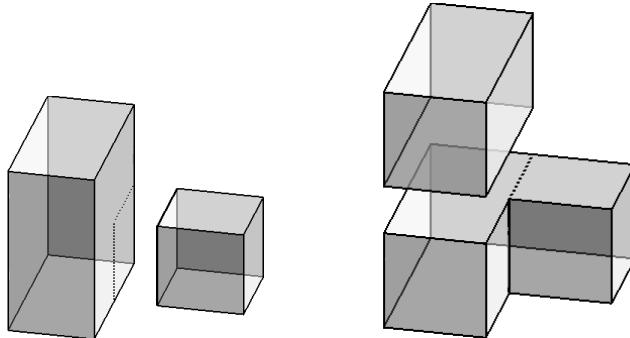


Fig. 12 Two possible results of cutting the solid of Figure 11 (left) by extending surfaces.

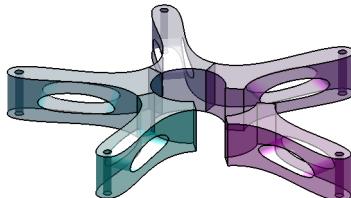


Fig. 13 The chair stand example. This solid has a 5-way symmetry, so one-fifth of it is extracted for isogeometric segmentation (see Figure 14).

Chair stand piece. The chair stand piece (Figure 13) was created to test various preprocessing strategies and demonstrate the algorithm for solids with convex edges. The result for one preprocessing strategy is shown in Figure 14. A second preprocessing strategy is shown in Figure 16 where the resulting topological hexahedra are volume parameterized using the techniques in Section 4.

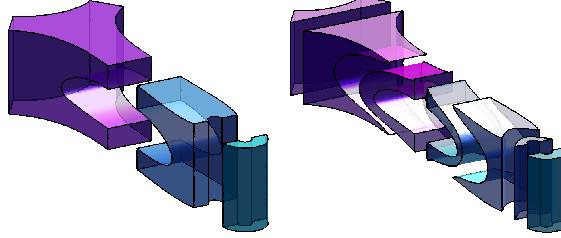


Fig. 14 Left: the chair stand piece, preprocessed to cut it into contractible pieces. Right: the result of the segmentation algorithm.

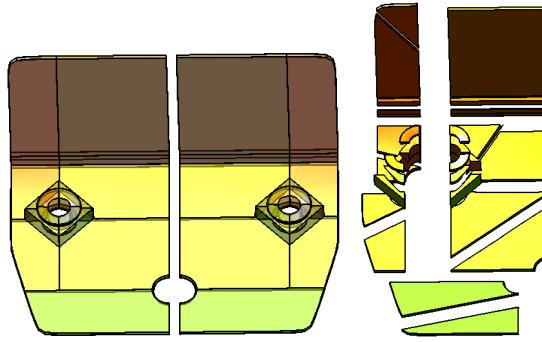


Fig. 15 Left: the car part, divided into two along its plane of symmetry and preprocessed to cut it into contractible pieces. Right: segmentation of one half of the Demonstrator into topological hexahedra.

TERRIFIC Demonstrator The preprocessed TERRIFIC Demonstrator is shown in Figure 6 (a). The solid is segmented into topological hexahedra using the surface extension and cutting loop methods described in the previous sections. The result is shown in Figure 6 (b). The result of the removal of T-joints is shown in Figure 6 (c).

Car part The segmentation procedure is applied to the CAD model reconstruction of the car part (See Section 2 and Figure 1 (a)–(c)). The preprocessed solid (Figure 15, left) has a plane of symmetry. For one half, the resulting contractible pieces are segmented into topological hexahedra (Figure 15, right). Data courtesy of Engineering Center Steyr.

3.5 T-joint removal

For some simulation methods, it may be necessary to post-process the model to eliminate T-joints. We do not have a general method for this. A special-purpose

method was developed which suffices for our segmentation of the TERRIFIC demonstrator. During the segmentation process, we keep track of adjacency information for the solids. When the segmentation of a solid results in the creation of a T-joint, new edges must be added to the adjacent solid(s) which must themselves be segmented. This segmentation may create new T-joints, resulting in a propagation of segmentations through the solid. The result for the TERRIFIC Demonstrator is shown in Figure 6 (c). Ongoing research on advanced IGA techniques aim to handle meshes with T-joints directly (see Section 1).

4 Hexahedron parameterization

The aim is to convert a boundary represented topological hexahedron into a volume parameterization. Besides the possibility of quite distorted hexes, difficulty arises from the fact that the boundary faces are trimmed surfaces, since the domain of such a surface is not a rectangle in general.

A single topological hexahedron is parameterized by the following procedure.

1. Sample each of the 4 edges of a boundary face at parameters such that the distance between consecutive points is approximately constant. This is necessary because the curves from adjacent boundaries do not generally have the same parameterization;
2. Apply 2-dimensional Coon's patches to the sampled points, to produce a grid of points in the domain;
3. Evaluate the grid points, to produce a grid on the 3D surface;
4. Applying the above steps for each of the 6 faces of the topological hexahedron, we now have a grid of sampled boundary points. Grid points agree along the edges because of the way the edges are sampled in Step 1. Apply 3D Coon's patches to the boundary points to produce a grid of interior points. (It may be possible to skip this step if the topological hexahedron is not too heavily distorted);
5. Fit a tensor product B-spline to the sampled points using a least squares method.

The fitting method could be easily modified to include more complex quality measures.

Applying the procedure to a collection of topological hexahedra produces a multi-patch domain (i.e., a domain composed of multiple volume-parameterized topological hexahedra) suitable for isogeometric analysis (see Section 5). If a segmentation of a solid into topological hexahedra has no T-joints, then we can also ensure watertightness of the collection of patches by the following procedure. (The collection is not automatically watertight because the construction of cutting faces during the segmentation, as well as the fitting of the volume patch, can cause errors in the corresponding control points of incident patches.)

- (i) Find all the incident faces between patches (the easiest way to do this is to hold on to the adjacency information from the segmentation procedure);

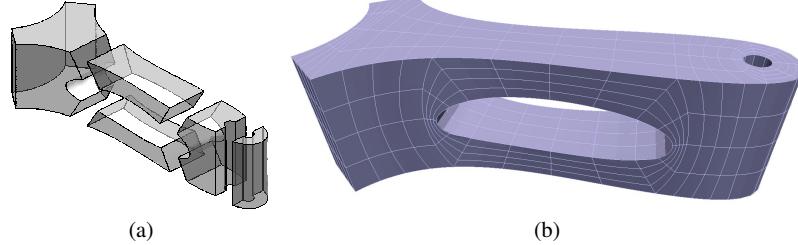


Fig. 16 (a) the chair stand piece, preprocessed to cut it into contractible pieces, using a different strategy from Figure 14. (b) representation of the chair stand as a watertight collection of tensor-product NURBS volumes.

- (ii) Wherever two or more patches meet (more than two patches can meet at vertices or edges), the corresponding control points of all such patches are moved to the average.

This simple adjustment is made possible by the following fact: If two topological hexahedra meet face-to-face in a quadrilateral patch, then this patch is represented by identical trimmed NURBS surfaces for both hexahedra. This is guaranteed by our construction of the splitting surface, cf. Section 3.3. The steps 1-3 of our parameterization procedure therefore give identical parameterization of the matching boundary faces in both patches. Thus, the only possible deviations between the control points are due to the approximation of the inner points in step 5. They can safely be expected to be relatively minor and can be dealt with by the averaging step (ii).

5 Simulation

As applications of the segmentation presented in previous sections, we use the TERRIFIC Demonstrator and the chair stand in mechanical simulations, solving the linear elasticity partial differential equation using an isogeometric finite element discretization. In Section 5.1 we briefly present the linear elasticity problem. In Section 5.2 we describe the method for solution on a multi-patch domain. Results are presented in Sections 5.3–5.4.

5.1 The linear elasticity problem

The linear elasticity problem on a domain $\Omega \subset \mathbb{R}^3$ means finding a solution $\mathbf{u} : \Omega \rightarrow \mathbb{R}^3$ of the following boundary value problem:

$$\begin{aligned} -\operatorname{div} \boldsymbol{\sigma}(\mathbf{u}) &= \mathbf{b} && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_u, \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} &= \mathbf{t} && \text{on } \Gamma_n, \end{aligned} \tag{1}$$

where $\Gamma_u \subset \delta\Omega$ is the Dirichlet boundary, $\Gamma_n \subset \delta\Omega$ the Neumann boundary of the domain, the Cauchy stress $\boldsymbol{\sigma}$ is expressed by the linear St. Venant-Kirchhoff constitutive law,

$$\boldsymbol{\sigma}(\mathbf{u}) = 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) + \lambda \operatorname{trace} \boldsymbol{\varepsilon}(\mathbf{u}) \mathbf{I}$$

using Lamé constants $\mu, \lambda > 0$ and the linear strain tensor

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u}^T + \nabla \mathbf{u}).$$

For a finite element discretization the weak form of (1) is required, i.e. seeking $\mathbf{u} \in H^1(\Omega)$, $\mathbf{u}|_{\Gamma_u} = \mathbf{g}$:

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v}) : \boldsymbol{\sigma}(\mathbf{u}) d\mathbf{x} = \int_{\Omega} \mathbf{v}^T \mathbf{b} d\mathbf{x} + \int_{\Gamma_n} \mathbf{v}^T \mathbf{t} d\Gamma \quad \forall \mathbf{v} \in H_0^1(\Omega).$$

Using the 4th-order elasticity tensor \mathbb{C} ,

$$\mathbb{C}_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{kj}), \quad i, j, k, l = 1, 2, 3,$$

the equation can be rewritten as:

$$\int_{\Omega} \nabla \mathbf{v} : \mathbb{C} : \nabla \mathbf{u} d\mathbf{x} = \int_{\Omega} \mathbf{v}^T \mathbf{b} d\mathbf{x} + \int_{\Gamma_n} \mathbf{v}^T \mathbf{t} d\Gamma \quad \forall \mathbf{v} \in H_0^1(\Omega). \tag{2}$$

5.2 Isogeometric treatment with multi-patch domain

The basis of an isogeometric method is a spline representation of the domain Ω and also the numerical solution field \mathbf{u}^h [1]. This means that there exists a global mapping

$$\mathbf{g} : \Omega_0 \rightarrow \Omega, \quad \mathbf{x} = \mathbf{g}(\xi) = \sum_{i=1}^n N_i^p(\xi) \mathbf{c}_i, \tag{3}$$

which parameterizes the whole domain as a B-Spline or NURBS volume, using tensor product basis functions N_i^p of degree p , which are defined on the parameter domain Ω_0 using knot vectors $\Xi \in \mathbb{R}^{n+p+1}$, and control points $\mathbf{c}_i \in \mathbb{R}^3$. Then \mathbf{u}^h is discretized as the push-forward of these spline functions onto the physical domain Ω and a set of displacement control points $\mathbf{d}_i \in \mathbb{R}^3$:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^n N_i^p(\mathbf{g}^{-1}(\mathbf{x})) \mathbf{d}_i. \tag{4}$$

Finally substitution of (4) into (2) and numerical evaluation of the integrals leads to the well-known matrix-vector form of the finite element discretization of the elasticity problem:

$$\mathbf{K}\mathbf{d} = \mathbf{f}, \quad (5)$$

where $N = 3n$ is the dimension of the system, \mathbf{d} is the vector of unknown control point displacements, \mathbf{K} the stiffness matrix and \mathbf{f} the external force vector.

We have to deal with multi-patch partitionings of the initial domain Ω into b subdomains (patches):

$$\overline{\Omega} = \bigcup_{i=1}^b \overline{\Omega^i}, \quad \Omega^i \cap \Omega^j = \emptyset.$$

If two patches Ω_i and Ω_j are adjacent, i.e.

$$\overline{\Omega^i} \cap \overline{\Omega^j} = \Gamma^{ij} \neq \emptyset,$$

we call Γ^{ij} the interface between Ω^i and Ω^j . Thus we can discretize the solutions $\mathbf{u}^i := \mathbf{u}|_{\Omega^i}$ independently, but we have to enforce the conditions $\mathbf{u}^i = \mathbf{u}^j$ on Γ^{ij} . As we deal with conforming parameterizations, i.e. $\mathbf{g}_i|_{\Gamma_{ij}} \equiv \mathbf{g}_j|_{\Gamma_{ij}} \forall \Gamma_{ij}$, this continuity constraint simply means corresponding displacement control points have to be equal, i.e.

$$\mathbf{d}_k^i = \mathbf{d}_l^j \quad \forall \mathbf{d}_k^i \in \mathbf{u}^{hi}|_{\Gamma_{ij}}, \mathbf{d}_l^j \in \mathbf{u}^{hj}|_{\Gamma_{ij}} \text{ with } \mathbf{c}_k^i = \mathbf{c}_l^j. \quad (6)$$

For the enforcement of these constraints basically two possibilities exist, either adding each constraint as additional equations in a Lagrangian multiplier approach, or eliminating degrees of freedom from the system.

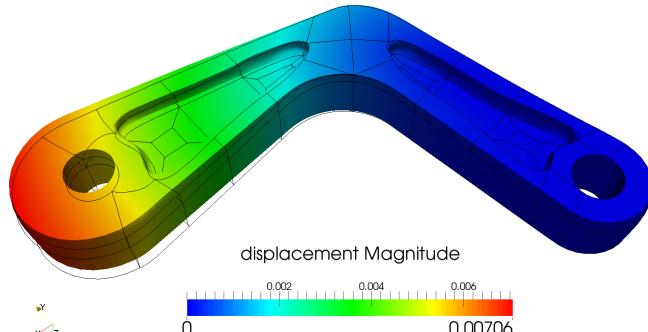
5.3 TERRIFIC Demonstrator

The isogeometric volume parameterization consists of 70 patches of cubic B-Spline volumes, with a total of 24,010 control points and 72,030 degrees of freedom (DOFs). Coupling of displacement control points on 132 interfaces has to be enforced. When Lagrangian multipliers are used for coupling, the isogeometric finite element discretization of the model has 90,048 DOFs, otherwise when elimination is used, it has only 53,925 DOFs.

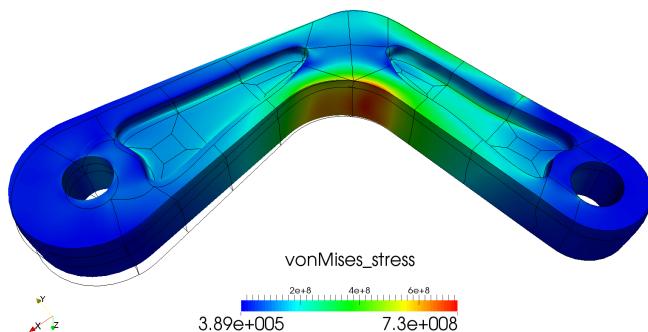
As material parameters of linear elasticity we chose

$$\lambda = 54.0 \text{ GPa}, \quad \mu = 27.8 \text{ GPa}.$$

As boundary conditions we take a clamping of the right bore hole (see Figure 17) by a zero Dirichlet condition at Γ_u and a surface traction on the left bore hole as a Neumann boundary condition on Γ_n :



(a)



(b)

Fig. 17 Linear elasticity simulation of TERRIFIC Demonstrator. Displaced part, colored by (a) norm of displacement in [m] and (b) von Mises stress in [Pa]

$$\begin{aligned} \mathbf{u} &= \mathbf{0} \text{ m} \quad \text{on } \Gamma_u, \\ \mathbf{t} &= (60.0, -42.0, 0.0)^T \text{ MPa} \quad \text{on } \Gamma_n. \end{aligned}$$

Running the simulation, we get a maximal norm of displacement of 7.06 mm and internal energy of 70.1 J. The displaced object can be seen in Figure 17. Comparison with a commercial finite element software shows a very good correspondence of results. Using ANSYS® with a mesh of 20,855 quadratic tetrahedral elements (type SOLID187) and 36,181 nodes resp. 108,543 DOFs, we have a maximum displacement of 7.00 mm and energy 70.2 J. It is also worth remarking that the stress distribution obtained from IGA in Figure 17(b) looks very smooth.

5.4 Chair stand

The chair stand model consists of 12 cubic B-Spline volumes with a total of 3,969 control points and 11,907 DOFs. Using coupling of patches with the Lagrangian multiplier approach we have a total of 14,231 DOFs.

For a realistic simulation of the chair stand we have added an inner cylinder connecting the stand with the seating and a small cylinder connecting to the wheels made from aluminum, and assumed the main parts of the stand are made from plastic (Polyamid P6). Thus the material parameters are:

$$\begin{aligned}\lambda_{Al} &= 54.0 \text{ GPa}, & \mu_{Al} &= 27.8 \text{ GPa}, \\ \lambda_{P6} &= 3.19 \text{ GPa}, & \mu_{P6} &= 0.899 \text{ GPa}.\end{aligned}$$

Assuming a person weighing 81.5 kg sitting on the chair, this would result in a vertical force of 0.10 MPa acting as Neumann boundary condition on the inner aluminum cylinder. Furthermore symmetry boundary conditions are applied on displacements, since we only model one fifth of the total chair stand. As the smaller cylinder connects to the wheel, which we assume as a frictionless link to the ground, no vertical displacement is possible there.

Running the simulation, we get a small vertical displacement of the inner cylinder of 0.82 mm and a maximal von Mises stress of 1.6 MPa. The total internal energy is 64.2 mJ. The deformed stand is visualized in Figure 18. We also validate our results by a comparison to commercial finite element software ANSYS®. Automated mesh generation provides a discretization using a mix of 13,437 quadratic hexahedral and tetrahedral elements (types SOLID186 and SOLID187) with 22,518 nodes and 57,554 DOFs. The simulation results in an evaluated z-displacement of 0.81 mm, a von Mises stress of 1.6 MPa and a total energy of 62.8 mJ.

6 Conclusions and further work

We created a pipeline, including algorithms and software together with a rigorous mathematical treatment, for the conversion of 3D models into an IGA-suitable form. The first step of the pipeline is the reconstruction, if necessary, of a CAD-like boundary representation from triangulated data. Then the boundary represented solid is segmented into a collection of topological hexahedra. Finally, volumetric NURBS patches are constructed for the hexahedra.

We used outputs of the pipeline to conduct simulations based on the linear elasticity problem. Results had good agreement with commercial finite element software, demonstrating the suitability of the segmentation and parameterization for simulation.

While the theory and software developed forms the core of the pipeline, there are several improvements already under development:

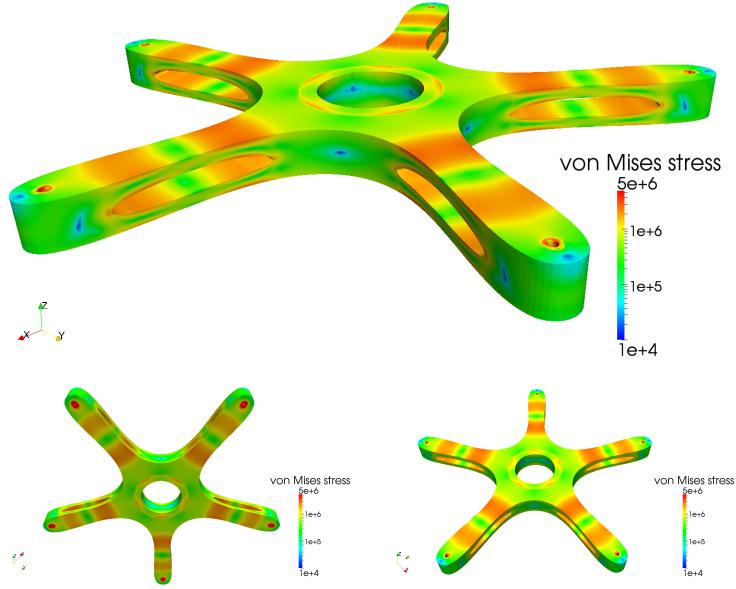


Fig. 18 Linear elasticity simulation of chair stand. Different views of displaced part, colored by von Mises stress in [Pa]

- improved geometric construction of the cutting curves and surfaces, with theoretical guarantees;
- addressing solids which are non-contractible or do not have 3-vertex-connected edge graphs without human interaction;
- improving the quality of isogeometric segmentation, in terms of the number and shape of the resulting topological hexahedra, using the surface extension methods described in Section 3.4.

Potential future developments include the use of splines allowing local refinement, such as LR- or THB-splines, and improving the reconstruction of cylindrical surfaces using periodic splines.

Acknowledgments

This work has been funded by the European Commission projects TERRIFIC (Grant Agreement 284981), EXAMPLE (Grant Agreement 324340) and ITN INSIST (Grant Agreement 289361), and the Austrian Science Fund (FWF) project Geometry and Simulation (Project Number S 117). We thank Martin Schifko at En-

gineering Center Steyr and Stefan Boschert at Siemens Corporate Technology for some of the input data used in our work.

References

1. J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, 2009.
2. M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Design*, 14(3):231–250, 1997.
3. M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.
4. D. Großmann, B. Jüttler, H. Schlusnus, J. Barner, and A.-V. Vuong. Isogeometric simulation of turbine blades for aircraft engines. *Computer Aided Geometric Design*, 29(7):519–531, 2012. Geometric Modeling and Processing 2012.
5. S. Han, J. Xia, and Y. He. Constructing hexahedral shell meshes via volumetric polycube maps. *Computer-Aided Design*, 43(10):1222–1233, 2011.
6. B. Jüttler, M. Kapl, D.-M. Nguyen, Q. Pan, and M. Pauley. Isogeometric segmentation: The case of contractible solids without non-convex edges. *Computer-Aided Design*, 57:74–90, 2014.
7. U. Langer and I. Toulopoulos. Analysis of multipatch discontinuous Galerkin IgA approximations to elliptic boundary value problems. Technical Report 1408.0182, arxiv.org, 2014.
8. Y. Lu, R. Gadhi, and T. J. Tautges. Feature based hex meshing methodology: feature recognition and volume decomposition. *Computer-Aided Design*, 33(3):221–232, 2001.
9. T. Martin and E. Cohen. Volumetric parameterization of complex objects by respecting multiple materials. *Computers & Graphics*, 34(3):187–197, 2010. Shape Modelling International (SMI) Conference 2010.
10. T. Martin, E. Cohen, and R. M. Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Comput. Aided Geom. Design*, 26(6):648–664, 2009.
11. D.-M. Nguyen, M. Pauley, and B. Jüttler. Isogeometric segmentation. Part II: On the segmentability of contractible solids with non-convex edges. *Graphical Models*, 76(5):426–439, 2014. Geometric Modeling and Processing 2014.
12. D.-M. Nguyen, M. Pauley, and B. Jüttler. Isogeometric Segmentation: Construction of auxiliary curves. *Computer-Aided Design*, 2015. Proceedings of GDSPM 2015, conditionally accepted.
13. T. Nguyen and B. Jüttler. Parameterization of contractible domains using sequences of harmonic maps. In *Curves and surfaces*, volume 6920 of *Lecture Notes in Comput. Sci.*, pages 501–514. Springer, Heidelberg, 2012.
14. M. Nieser, U. Reitebuch, and K. Polthier. CubeCover – parameterization of 3D volumes. *Comput. Graph. Forum*, 30(5):1397–1406, 2011.
15. A. Sheffer, M. Etzion, and M. Bercovier. Hexahedral mesh generation using the embedded Voronoi graph. In *In Proceedings of the 7th International Meshing Roundtable*, pages 347–364, 1999.
16. B. Strothoff, M. Schifko, and B. Jüttler. Horizontal decomposition of triangulated solids for the simulation of dip-coating processes. *Computer-Aided Design*, 43(12):1891–1901, 2011.
17. T. J. Tautges, T. Blacker, and S. A. Mitchell. The whisker weaving algorithm: a connectivity-based method for constructing all-hexahedral finite element meshes. *Int. J. Numer. Meth. Engrg.*, 39(19):3327–3349, 1996.
18. T. Varady and R. Martin. Reverse engineering. In G. Farin, J. Hoschek, and M.-S. Kim, editors, *Handbook of computer aided geometric design*, pages 651–681. North-Holland, Amsterdam, 2002.

19. W. Wang, Y. Zhang, L. Liu, and T.J.R. Hughes. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design*, 45(2):351 – 360, 2013.
20. X. Wang and X. Qian. An optimization approach for constructing trivariate B -spline solids. *Comput.-Aided Des.*, 46:179–191, 2014.
21. D. White, L. Mingwu, S. E. Benzley, and G. D. Sjaardema. Automated hexahedral mesh generation by virtual decomposition. In *Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories*, pages 165–176, 1995.
22. G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Comput.-Aided Des.*, 45(2):395–404, 2013.
23. W. Yu, K. Zhang, S. Wan, and X. Li. Optimizing polycube domain construction for hexahedral remeshing. *Comput.-Aided Des.*, 46:58–68, 2014.
24. Y. Zhang and C. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. In *Computer Methods in Applied Mechanics and Engineering*, 2006.
25. Y. Zhang, W. Wang, and T. J. R. Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Comput. Methods Appl. Mech. Engrg.*, 249/252:185–197, 2012.
26. Y. Zhang, W. Wang, and T. J. R. Hughes. Conformal solid T-spline construction from boundary T-spline representations. *Computational Mechanics*, 51(6):1051–1059, 2013.