

NFN - Nationales Forschungsnetzwerk

Geometry + Simulation

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



Isogeometric Segmentation via Midpoint Subdivision Suitable Solids

Michael Haberleitner, Bert Jüttler,
Yannick Masson

G+S Report No. 83

April 2019

FWF

Der Wissenschaftsfonds.

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

Isogeometric Segmentation via Midpoint Subdivision Suitable Solids

Michael Haberleitner, Bert Jüttler

Institute of Applied Geometry, Johannes Kepler Universität Linz, Austria

Yannick Masson

Johann Radon Institute for Computational and Applied Mathematics, Linz, Austria

Abstract

The framework of Isogeometric Analysis (IgA) makes frequent use of trivariate NURBS parameterizations (representing topological cuboids) of the computational domain. Several recent publications [1-6] describe methodologies that decompose a given three-dimensional solid in boundary representation into a collection of topological cuboids, or generate trivariate NURBS parameterizations for each of them in a subsequent step. The decomposition can be derived via a segmentation into sufficiently simple “base solids”, for which cuboidal multi-patch representations are readily available. Based on midpoint subdivision, we propose a new class of base solids. In addition, we establish the pre-processing step of face pre-segmentation, which simplifies the splitting operations and improves the shape of the resulting topological cuboids. Finally, we show how to realize the midpoint subdivision by a template mapping approach, which simultaneously generates parameterizations of the base solids as trivariate multi-patch NURBS volumes.

Keywords: Cuboidal decomposition, Volumetric Parameterization, Midpoint Subdivision, Template Mapping

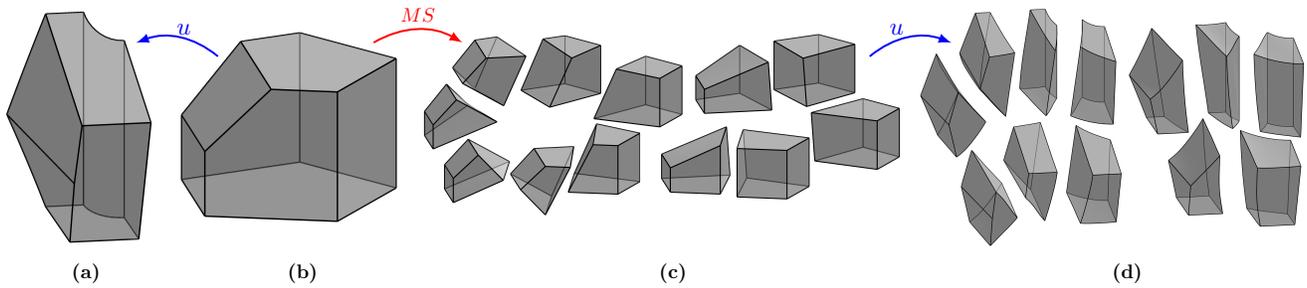


Figure 1: Given a midpoint subdivision suitable solid (a), a corresponding convex template polyhedron (b) is selected and the bijective mapping u is constructed. The template is decomposed into topological cuboids via midpoint subdivision (c). Finally, with the help of the mapping u , the subdivision is transferred to the input solid (d), generating a decomposition into topological cuboids with associated parameterizations as NURBS volumes.

1. Introduction

Since the advent of *Isogeometric Analysis* (IgA) in 2005 [7], there has been a continuous growth of interest in this topic, both from practitioners and the scientific community. IgA combines the main features of *Computer Aided Design* (CAD) and *Finite Element Analysis* (FEA), by reusing the basis functions of the design tools in the analysis step, and prepares the ground to achieve significant performance gains of the overall product development process. The survey article [8], written by two eminent experts in this field, gives an overview of the current state-of-the-art.

Since the availability of volumetric spline models is a prerequisite for IgA, numerous researchers have addressed the construction of such models from various types of input data. One may distinguish between single- and multi-patch parameterizations. The first class is conceptually simpler but has limited flexibility.

Single-patch spline representations for swept volumes and generalized cylinders were constructed in [9, 10]. Harmonic functions are particularly useful, since they can be employed to guarantee injectivity [11] and yield high-quality parameterizations [12]. Given a representation as a tetrahedral mesh, a B-spline volume representing a solid is constructed by first computing parameterizations via harmonic functions and then performing a spline approximation [13].

The limited flexibility of single-patch NURBS parameterizations can be enlarged by employing more advanced spline functions, such as T-splines. Several publications address the computation of parameterizations defined by trivariate T-splines. These

*Corresponding author

Email addresses: michael.haberleitner@jku.at (Michael Haberleitner), bert.juettler@jku.at (Bert Jüttler), yannick.masson@ricam.oeaw.ac.at (Yannick Masson)

combine optimization methods with refinement strategies and techniques from traditional mesh generation [14, 15]. The use of polycube domains allows to deal with objects of non-zero topological genus [16], such as solids defined by Boolean operations [17]. The preservation of boundary features is studied in [18]. A polycube-type construction for cubic polynomial splines over hierarchical T-meshes (PHT-splines) is presented in [19].

Many domains require the use of multi-patch structures, especially if tunnels or voids are present. A geometric modeling framework is established in [6]. Methods based on polycubes apply mesh deformation techniques to a uniform grid [5, 19]. While this gives satisfying results in many cases, it does not preserve the patch structure of a given CAD model. Alternatively, the segmentation into topological cuboids can be constructed by considering pants decomposition of the boundary surface [4], or by recursively applying splitting steps (isogeometric segmentation) that reduce the object’s topological complexity [2, 3, 20]. Once the segmentation has been found, the multi-patch parameterization can be constructed by employing variational techniques and Coons interpolation [1, 21].

Hybrid approaches, which combine an exact representation of the solid’s boundary with a tetrahedral mesh in the interior via intermediate patches defined by offsetting, maintain the compatibility with the initial representation [22].

The present paper contributes to the splitting-based approach to isogeometric segmentation, which is summarized in [23]. The initial framework [2], which is limited to solids with only convex edges, has been extended to contractible solids that may possess non-convex edges [3]. The application to non-contractible solids (with tunnels but without voids) becomes possible after executing a Reeb graph-based preprocessing step [20]. The important problem of intersection-free splitting surface construction has been addressed in [24]. Finally, the construction of NURBS volume parameterizations of the resulting cuboidal solids (which includes a postprocessing step that ensures matching interface parameterizations) has been described in [23]. It should be noted that the parameterization step produces an approximation of the original boundary surfaces, thereby generating a model that is suitable for both design and analysis.

Recall that the recursive splitting steps are applied until the resulting sub-domains belong to a predefined class of base solids. With the help of midpoint subdivision, we propose a new and larger class of base solids. In addition, we establish the preprocessing step of face pre-segmentation, which simplifies the splitting operations and improves the shape of the resulting solids. Finally, we show how to realize the midpoint subdivision by a template mapping approach, which simultaneously produces parameterizations of the base solids as trivariate multi-patch NURBS solids.

These contributions lead to the extended segmentation algorithm, which is described in the next section.

2. The extended segmentation algorithm (ESA)

The first step of converting boundary represented CAD data into volumetric NURBS patches is to subdivide the solid object into a collection of topological cuboids, which can then be parameterized by tensor product NURBS patches in the second

Algorithm ESA: Extended segmentation algorithm

Input : An ESA-suitable solid S .
Output: A decomposition into MS^3 .

```

1 Initialize stack with the solid  $S$ . Apply face presegmentation
  to the faces of  $S$ .
2 while stack is not empty do
3   Take the next solid from the stack.
4   if non-convex edges are present then
5     call CLS to remove at least one of them and put the
     two resulting solids on the stack.
6   else if high-valent vertices are present then
7     call CLS to to remove at least one of them and put
     the two resulting solids on the stack.
8   else if the solid is not a  $MS^3$  then
9     call CLS to generate two solids with fewer vertices
     and put them on the stack.
10  else
11  | output the solid.

```

step. The *isogeometric segmentation pipeline* [2, 3, 23] provides a framework that generates such a subdivision via a recursive algorithm, with the focus on creating a small number of patches. In this article we propose several extensions, which improve the quality of the output shapes.

The initial solid, which is required to be contractible (cf. [20]), is represented by its faces (trimmed NURBS surfaces), edges (where adjacent faces meet) and vertices (where neighboring edges meet). Edges and vertices define the edge graph, which is assumed to be planar, 3-vertex-connected and free of double edges. Any solid satisfying these assumption is said to be *ESA-suitable*. Recall from [3] that edges are classified into convex and non-convex ones, see Figure 2.

An *existing vertex* is defined by the intersection of at least three boundary faces. Additionally we consider *auxiliary vertices*, which can be introduced anywhere in the interior of an existing edge of the solid.

Similarly, *existing edges* correspond to non-empty intersections of adjacent boundary faces. In addition, we introduce *auxiliary edges* connecting existing or auxiliary vertices. These edges split faces into two smaller ones. Its endpoints must not belong to the same boundary edge. Auxiliary edges are always non-convex.

It should be noted that the geometric realization of auxiliary edges and vertices requires special care, in order to avoid intersections and singularities. This is discussed in more detail in [25] and [24].

The segmentation procedure of the extended pipeline is summarized in the Extended Segmentation Algorithm ESA. Given a contractible three-dimensional solid S , it recursively performs splitting operations, until the solid is decomposed into a collection of solids that are sufficiently simple and hence suitable for midpoint subdivision. These solids will be referred to as *midpoint subdivision suitable solids* (MS^3). A more precise definition of MS^3 will be provided in section 5.

Compared to the original method (see [23]), the procedure incorporates the following additional features:

Algorithm CLS: Cutting Loop-based Splitting

Input : A contractible three-dimensional solid S .
Output: Two solids S_1 and S_2 .

- 1 Find all possible cutting loops in the edge graph.
 - 2 Rate each loop by a cost function and select the best one.
 - 3 Construct auxiliary vertices and edges on the solid.
 - 4 Use the cutting loop to construct a cutting surface.
 - 5 Split the solid along the cutting surface.
-

- We perform a face pre-segmentation. The faces of the initial solid are subdivided if their shape appears to be unsuitable for the segmentation algorithm. This is described in more detail in Section 3.
- The extended procedure makes systematic use of midpoint subdivision. This type of decomposition, which is available for any solid with only tri-valent vertices, generates one cuboidal sub-domain for each vertex. This contribution is based on two ingredients: First, we need a method to eliminate high-valent vertices, which is described in more detail in Section 4. Second, we need a catalog of MS^3 and a method for actually performing the subdivision into cuboidal sub-domains. The latter method is simultaneously useful for creating the individual trivariate NURBS parameterizations. These two contributions are described in more detail in Sections 5 and 6, respectively.

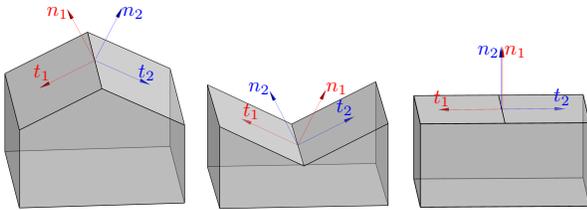


Figure 2: Instances of convex, non-convex and auxiliary edges. Note that the two normal vectors in the right picture actually coincide.

The splitting operations in lines 5, 7 and 9 of ESA are performed by instances of the algorithm for Cutting Loop-based Splitting (CLS, see page 3). First we use Yen’s algorithm [26] to find all available cutting loops in the edge graph, which is possibly extended by auxiliary edges and vertices. Second, the quality of these loops is assessed by evaluating a cost function, which takes combinatorial and geometric aspects into account. Finally, after selecting the best cutting loop according to the cost function, we construct the cutting surface and split the solid, see [24]. The main difference between the three cases (lines 5, 7 and 9) is the choice of the cost function, which is adapted to the specific situation.

The remaining sections provide further details regarding ESA. The first splitting operation (line 5 of the algorithm) is covered by [3] and will not be discussed further.

3. Face pre-segmentation

Clearly, the quality of the segmentation result depends not only on the combinatorial structure of the edge graph, but is strongly influenced by the geometry of the solid. More complex shapes are more challenging than simpler ones and require more sophisticated techniques in order to obtain a satisfactory result. The proposed face pre-segmentation of the initial solid takes this fact into account.

In many cases, the more complicated parts of a solid either correspond to a large number of faces, or to fewer faces with more complex shapes. The latter case could not be dealt with successfully by the original algorithm of [23], since that method was unable to identify suitable segmentations of faces possessing a complex shape. We address this problem by splitting those faces into simpler ones, joined across newly introduced artificial non-convex edges. These edges then trigger splitting operations that simplify the shape of the solid.

The insertion of the artificial edges is applied recursively, and it is repeated until the shape of the face is sufficiently simple. We use the number of vertices (which should not exceed 7) and the total variation of tangent vector’s turning angle to identify faces that need further splitting.

We assume that the shape of the face roughly matches the shape of the associated parameter domain. This is the case if the parameterization has bounded distortion. Consequently, it suffices to analyze the shape of the parameter domain, which is a much simpler task than analyzing the shape of the face itself.

The face is split along one or several straight lines in the parameter domain. We determine the location of these lines with the help of the *medial axis transform* (MAT) of the domain boundary, which is a classical tool for shape analysis and segmentation.

The MAT of a planar domain is formed by the centers of maximal inscribed disks, along with the associated radius information, see [27] and the references cited therein. The local minima of the radius function along the MAT indicate where to split the face. Each of the associated maximal disks touches the boundary of the face in two points, cf. Figure 3. The disk’s diameter connecting them defines the splitting curve.

This simple approach fails in some cases, see Figure 4. In order to address this problem we employ an alternative technique, which is based on the distance function $d(u, v) = \|b(u) - b(v)\|$ of point-pairs on the boundary b of the parameter domain. The potential splitting lines correspond to prominent local minima of this function, with well-separated end points. Typically, this results in several candidates for splitting lines, and we select the shortest one.

4. Elimination of high-valent vertices

In order to arrive at MS^3 , we split high-valent vertices (valency greater than 3) by selecting suitable cutting loops through them. On the one hand, the choice of the cutting loop needs to ensure that the valency of the vertex decreases. On the other hand, it must not increase the valency of any other vertex.

We recall some simple facts about cutting loops. Let v be a vertex with valency n . A valid cutting loop that passes through v splits the solid S into two solids S_1 and S_2 , which both contain

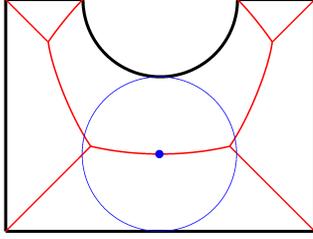


Figure 3: Local minimum of the radius function along the medial axis transform and corresponding maximal disk.

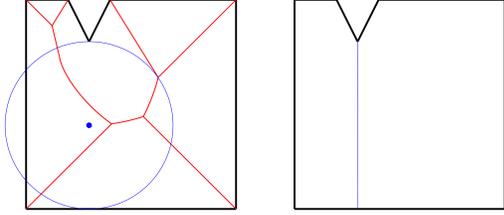


Figure 4: Left: the circle corresponding to the desired cut is not a maximal inscribed circle. The radius function has no local minimum along the MAT. Right: the alternative approach still finds the desired cut.

the vertex v with valencies n_1 and n_2 respectively. The two edges of the cutting loop that contain v can be existing or auxiliary ones, cf. Figure 5.

A careful analysis of the possible configurations gives the following result.

Lemma 1. *The valencies n_1 and n_2 do not exceed the original valency n if the cutting loop contains at most one auxiliary edge through v .*

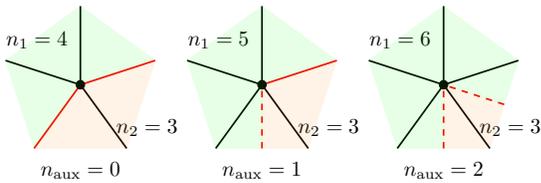


Figure 5: Cutting loops (red) through a vertex with valency 5, with different numbers n_{aux} of auxiliary edges (dashed).

For future reference we state another result.

Lemma 2. *If two faces F_1 and F_2 of an edge graph share two vertices v and w , the edge (v, w) exists and is contained in both faces F_1 and F_2 .*

Proof. We prove this result by contradiction. Let F_1 and F_2 be two faces that share the vertices v and w , but no edge, see Figure 6. We assume that the vertices on both faces are ordered counter-clockwise and denote the paths from v to w and from w to v in face F_1 by \mathcal{P}_1 and \mathcal{Q}_1 , respectively. Similarly, we denote the

paths from w to v and from v to w in face F_2 by \mathcal{P}_2 and \mathcal{Q}_2 , respectively (cf. Figure 6).

The loop $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$ contains at least one vertex besides v and w . Otherwise, the two faces would share the edge (v, w) , since the edge graph is free of double edges. Analogously we conclude that the loop $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ contains at least one vertex besides v and w . Consequently, removing the vertices v and w disconnects the edge graph. This contradicts the assumption that the edge graph is 3-vertex-connected. \square

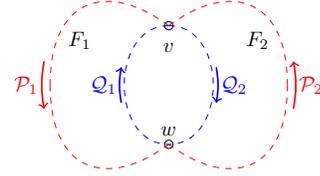


Figure 6: The loops considered in the proof of Lemma 2.

The following theorem shows that there always exists a suitable cutting loop to reduce the valency of a high-valent vertex. The main idea of the proof is to suitably extend the proof of Theorem 1 of [3], which showed that there always exists a valid cutting loop through a given non-convex edge.

Theorem 1. *Any solid S , which fulfills the assumptions of ESA and contains a high-valent vertex v , can be split by CLS into two solids S_1 and S_2 , both containing a copy of v with reduced valency. The splitting does not increase the valency of any other vertex and creates only tri-valent ones.*

Proof. We enumerate the faces that surround v in counter-clockwise order by F_1, \dots, F_n , and the adjacent vertices by v_1, \dots, v_n , with $(v, v_k) = F_k \cap F_{k+1}$, where indices are considered modulo n , see Figure 7. From the n adjacent vertices, we pick two vertices v_i and v_j , which are neither connected by an edge, nor share one of the faces F_1, \dots, F_n . This is always possible, since $n \geq 4$ and the edge-graph is planar.

We show that the path (v_i, v, v_j) can be extended to a valid cutting loop, i.e., any face contains at most one edge of that loop. This loop reduces the valency of v , due to the choice of v_i and v_j . Moreover, Lemma 1 ensures that the valencies of v_i and v_j do not increase. All other vertices are chosen as auxiliary (and hence tri-valent) ones. The construction of this cutting loop requires the analysis of two different cases:

Case 1: v_i and v_j share a face F^* (cf. Figure 7). We connect v_i and v_j by an auxiliary curve in that face, thereby obtaining the desired cutting loop. Lemma 2 implies that F^* is none of the faces F_i, F_{i+1}, F_j or F_{j+1} . Consequently, the cutting loop is valid.

Case 2: v_i and v_j do not share a face (cf. Figure 8). Let F_i^* be the face that shares an edge with F_{i+1} and contains v_i but not v . Similarly, let F_j^* be the face that shares an edge with F_j and contains v_j but not v .

We consider the union of the faces F_{i+1}, \dots, F_j . From its boundary we remove the vertex v and the adjacent edges to obtain the path $q_1 = v_i, \dots, q_\ell = v_j$. The faces, which contain at least one of the vertices q_1, \dots, q_ℓ but not v form a 2-vertex connected sub-graph, whose dual graph is connected. The shortest

path (with respect to the number of edges) from F_i^* to F_j^* in the dual graph indicates how to obtain the cutting loop.

The shortest path in the dual graph corresponds to a chain of faces $H_1 = F_i^*, \dots, H_m = F_j^*$ of the original graph, such that adjacent faces H_k and H_{k+1} share an edge. On each edge $H_k \cap H_{k+1}$ we create an auxiliary vertex w_k for $k = 1, \dots, m-1$ and connect consecutive ones by auxiliary curves in the corresponding faces. Finally we close the cutting loop by connecting v_i to w_1 and v_j to w_{m-1} by auxiliary curves in H_1 and H_m respectively.

Since, by construction, none of the faces H_1, \dots, H_m contains v , they are different from F_i, F_{i+1}, F_j and F_{j+1} . Thus, the cutting loop will be valid. \square

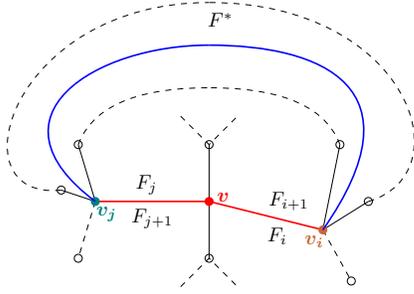


Figure 7: Theorem 1, case 1: v_i and v_j are contained in the face F^* . We connect them by an auxiliary curve.

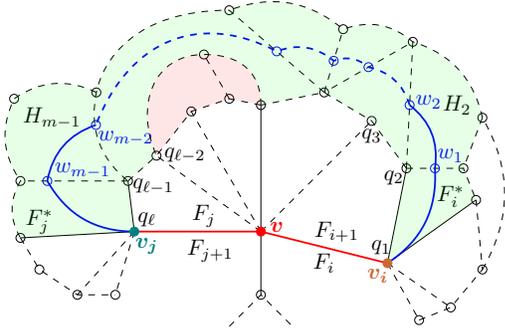


Figure 8: Theorem 1, case 2: v_i and v_j do not share a face. We construct a path in the chain of adjacent faces (green and red) connecting v_i and v_j . The green faces contain the shortest path, with respect to the number of edges.

Note that while Theorem 1 guarantees the existence of a valency reducing cutting loop, it is not used directly to construct it during the execution of CLS. Instead, we use a suitable cost function, which leads to the selection of a cutting loop that splits the solid into two solids, with fewer high-valent vertices, according to a lexicographic ordering.

5. Midpoint subdivision suitable solids (MS^3)

A solid needs to fulfill several requirements to be considered suitable for midpoint subdivision. The primary condition is that the solid must only contain tri-valent vertices. This is due to the

fact that midpoint subdivision splits a solid S with n vertices into n solids, which each contain exactly one vertex of the original solid. The solids will be topological cuboids if and only if the valency of any vertex of S is equal to three.

Based on the duality between planar triangulations and planar graphs with only tri-valent vertices, we generated the edge graphs of all MS^3 with the help of `plantri` [28]. Table 1 reports the number of topologically different edge graphs for MS^3 with at most 12 faces, where we restricted ourselves to solids with at most 6-sided faces. Each of these edge graphs possesses a geometric realization as a convex polyhedron, according to Steinitz's theorem (cf. [29]), which will serve as the template for the midpoint subdivision, see next section. Figures 9 and 10 visualize the edge graphs and possible realizations of the corresponding templates for MS^3 with at most 9 faces.

# faces	4	5	6	7	8	9	10	11	12
# templates	1	1	2	5	10	15	30	44	77

Table 1: Number of different template MS^3 with respect to the number of faces.

So far we took solely combinatorial properties of MS^3 into account. Since the quality of the output shapes depends strongly on the geometry of the input solid, we also employ two simple geometric criteria:

Since all generated sub-domains that share one common face of the original solid meet in a common edge, small interior angles will be present if the original solid has faces with many edges. Therefore we restrict ourselves to solids with at most 6-sided faces. Similarly, in order to prevent the occurrence of small angles at the central vertex, we restrict ourselves to solids with at most 12 faces.

6. Midpoint subdivision

Midpoint subdivision was already investigated in [30] for hexahedral mesh generation. In that publication, the main focus was put on the compatibility of the generated hexahedral sub-meshes across the interfaces, which was achieved via integer programming. In contrast to this earlier work, we focus on the generation of a single NURBS volume parameterization for each of the generated cuboidal sub-domains.

6.1. Midpoint subdivision of template MS^3

We consider a convex polyhedron, which represents one of the template MS^3 , see Figure 10. For every vertex of the polyhedron, we construct a topological cuboid, by introducing new vertices, edges and faces.

First, we choose the midpoints of every edge and every face, and the center of the polyhedron. Second, we create straight edges that connect each face-midpoint to the surrounding edge-midpoints, and the center to all face-midpoints. Third, we create a bilinear cutting surface for each quadrilateral loop of newly created edges, and we subdivide the original planar faces of the polyhedron accordingly.

There are several options for choosing edge midpoints, face midpoints, and the center of the polyhedron. Since the template MS^3 are all convex, the center of mass is usually a good choice

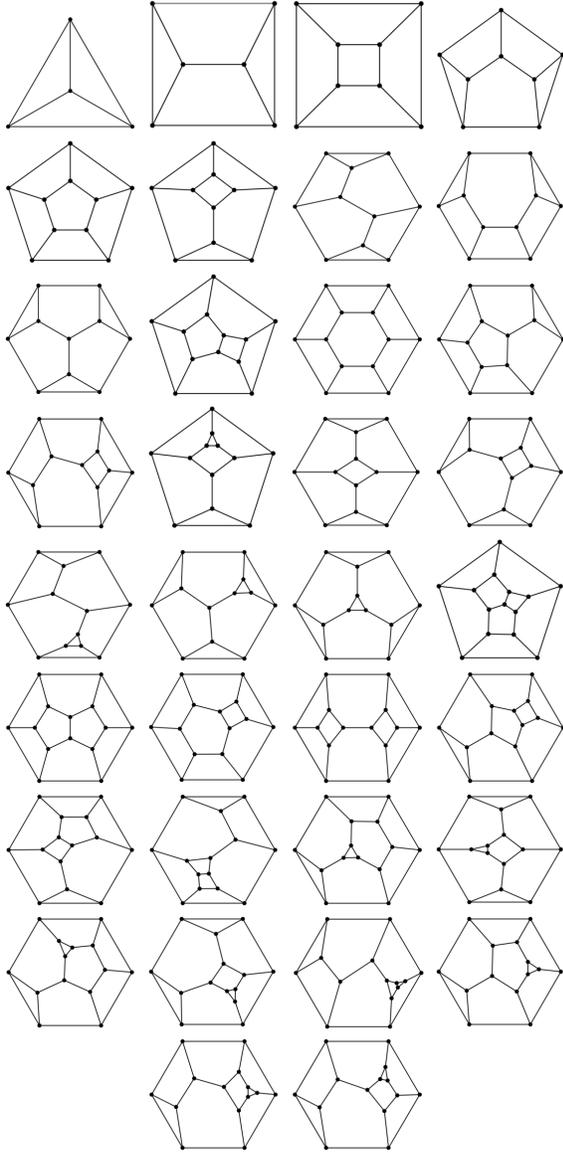


Figure 9: Edge graphs of MS^3 with at most 9 faces.

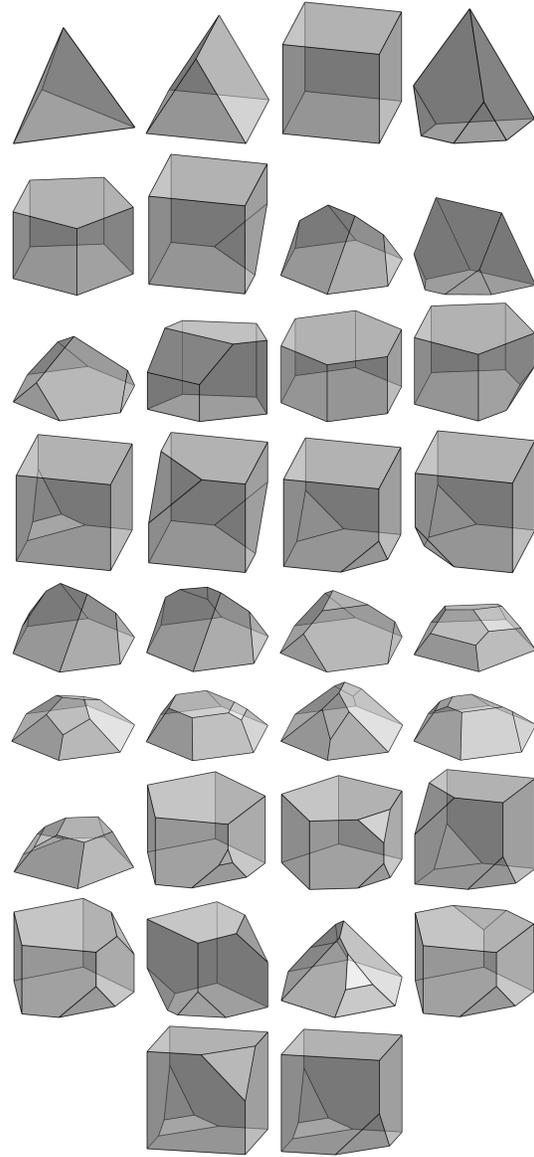


Figure 10: Template MS^3 with at most 9 faces.

in all cases. Figure 11 depicts the midpoint subdivision of a particular instance of a template MS^3 .

6.2. The multi-patch structure

Midpoint Subdivision decomposes a given d -dimensional domain Ω , which is topologically equivalent (i.e., diffeomorphic) to a polygon for $d = 2$ and to a template MS^3 for $d = 3$, into a collection of parameterizable sub-domains. The number of vertices, all of which have the same valency d , is denoted by n_v .

In order to prepare the exposition in the next section, we introduce a multi-patch structure that allows us to describe the entire domain in terms of the smaller sub-domains. For $d = 2$, it is visualized by the part of Figure 12, which is within the dashed polygonal boundary.

The midpoint subdivision is realized by constructing a multi-

patch parameterization

$$p_i : [0, 1]^d \rightarrow \Omega, \quad i = 1, \dots, n_v,$$

with $\Omega = \bigcup_{i=1}^{n_v} \Omega_i$ and $\Omega_i^\circ \cap \Omega_j^\circ = \emptyset$ if $i \neq j$, where $\Omega_i = p_i([0, 1]^d)$.

The boundary facets of the n_v individual parameter domains $[0, 1]^d$ are split into two groups:

- The first d facets are used to parameterize the boundary of Ω . We use the symbols

$$b_{i,j} : [0, 1]^{d-1} \rightarrow \partial\Omega \cap F^{(i,j)}, \quad j = 1, \dots, d, \text{ with}$$

$$b_{i,j}(\xi_1, \dots, \xi_{j-1}, \xi_{j+1}, \dots, \xi_d) = p_i(\xi_1, \dots, \xi_{j-1}, 0, \xi_{j+1}, \dots, \xi_d),$$

to represent the d restrictions of the parameterization p_i to these facets. Here, the symbols $F^{(i,j)}$ denote the d faces of Ω that contain the i -th vertex v_i .

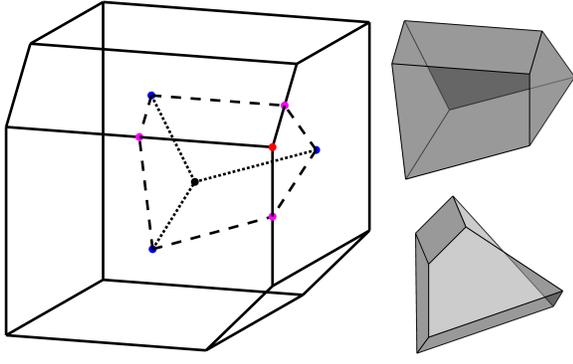


Figure 11: Left: Visualization of the midpoint subdivision (shown for only one vertex). The face midpoints (blue) are connected to all surrounding edge midpoints (purple) and to the solid's center (black). Right: The one constructed sub solid from two different perspectives.

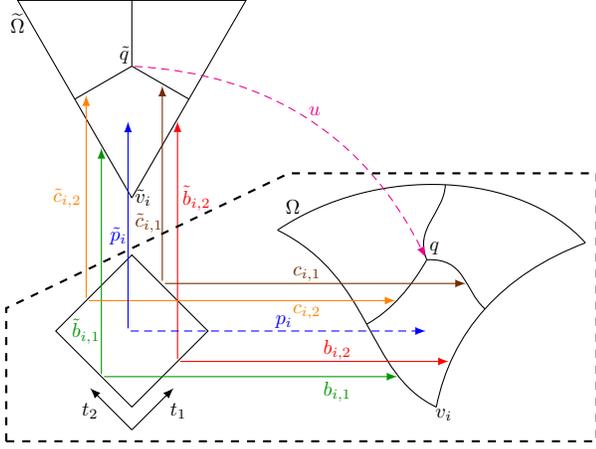


Figure 12: Multi-Patch structure (shown within the dashed poly-gon) and template-mapping problem for $d = 2$.

- The remaining d facets are used to parameterize the interior interfaces. We use the symbols

$$c_{i,j} : [0, 1]^{d-1} \rightarrow \Omega, \quad j = 1, \dots, d, \text{ with}$$

$$c_{i,j}(\xi_1, \dots, \xi_{j-1}, \xi_{j+1}, \dots, \xi_d) = p_i(\xi_1, \dots, \xi_{j-1}, 1, \xi_{j+1}, \dots, \xi_d).$$

to represent the d restrictions of the parameterization p_i to these facets. The interfaces are joined pairwise with C^0 -smoothness. More precisely, if the edge (v_i, v_i') exists in the solid Ω , there are indices j and j' and a permutation π of the $d - 1$ parameters such that

$$c_{i,j} = c_{i',j'} \circ \pi. \quad (1)$$

Note that the multi-patch parameterization satisfies

$$p_i(0, \dots, 0) = v_i, \quad \text{and} \quad p_i(1, \dots, 1) = q,$$

where q is the midpoint of the domain Ω . The multi-patch structure of a template MS^3 is readily available. We will transfer it to a topologically equivalent general MS^3 .

6.3. Midpoint subdivision of general MS^3

For subdividing a general MS^3 , the simple techniques described in Section 6.1 will not suffice. Instead we consider a topologically equivalent template MS^3 and its associated multi-patch structure, and construct a regular mapping from the template to the solid, which is used to transfer the subdivision.

In our implementation, we pre-computed one realization of each edge graph by a convex polyhedron and used it as a template MS^3 . These realizations are shown in Figure 10. Clearly, it would be possible to create more than one realization per edge-graph and to select the most suitable one (which minimizes the overall deformation of the mapping that transforms the template into the target) when transferring the subdivision. So far, our implementation uses only one geometric realization per MS^3 .

A well-established approach to construct the required mapping from a template to a target is the free-form deformation [31], where a box surrounding the template is mapped into a free-form volume around the target. This regular mapping transforms the boundary of the template into the boundary of the target. Its construction is reasonably simple to implement and produces satisfying results if the shape of the target is relatively similar to the shape of the template. Otherwise, we need a more sophisticated approach, which is based on a systematic use of the multi-patch structure presented in the previous section.

We denote the given domain, which is assumed to be a MS^3 , by Ω , and the associated template by $\tilde{\Omega}$. The multi-patch parameterizations of the solid and of the template will be denoted by p_i and \tilde{p}_i , respectively. Similarly, we will use the tilde to distinguish between the MS^3 and its template.

In order to construct a mapping from the template to the target we consider the following *template mapping problem*, which is visualized in Figure 12:

We consider the given template domain $\tilde{\Omega}$ with the multi-patch parameterization \tilde{p}_i and the boundary parameterizations $\tilde{b}_{i,j}$, for $i = 1, \dots, n_v$ and $j = 1, \dots, d$. In addition, we are given the target domain Ω with associated boundary parameterizations $b_{i,j}$, where the mapping

$$\beta(\tilde{x}) = b_{i,j} \circ \tilde{b}_{i,j}^{-1}(\tilde{x}), \quad \text{if } \tilde{x} \in \text{codomain}(\tilde{b}_{i,j})$$

is assumed to be continuous and bijective. For these given data, we construct a multi-patch parameterization

$$p_i, \quad i = 1, \dots, n_v,$$

of the target domain Ω , which then defines the template mapping

$$u : \tilde{\Omega} \rightarrow \Omega$$

$$u(\tilde{x}) = p_i \circ \tilde{p}_i^{-1}(\tilde{x}), \quad \text{if } \tilde{x} \in \text{codomain}(\tilde{p}_i). \quad (2)$$

In order to make this abstract problem accessible to a numerical solution, we first introduce a multi-patch spline space. It consists of tensor-product spline functions, defined on a union of d -dimensional unit cubes, which are C^0 -smooth across the interfaces between these cubes (see [32]), thereby guaranteeing that (1) is satisfied by construction. Summing up, the multi-patch parameterization takes the form

$$p : [0, 1]^d \times \{1, \dots, n_v\} \rightarrow \Omega$$

$$p = \sum_{k \in \mathcal{K}} d_k N_k, \quad (3)$$

where N_k are multi-patch B-splines and \mathcal{K} their index set.

The individual parameterizations are then obtained as restrictions to the patches $[0, 1]^d \times \{i\}$,

$$p_i(\xi) = p(\xi, i) = \sum_{k \in \mathcal{K}^i} d_k N_k^i(\xi), \quad \xi \in [0, 1]^d, \quad (4)$$

with tensor-product B-splines $N_k^i(\xi) = N_k(\xi, i)$ and index sets \mathcal{K}^i .

We obtain the parameterization (3) by first choosing the degrees and knot vectors of the basis functions N_k and then computing the unknown coefficients d_k . The latter are found by minimizing the functional

$$F(\mathbf{d}) = P(\mathbf{d}) + \lambda Q(\mathbf{d}), \quad (5)$$

where $\mathbf{d} = (d_k)_{k \in \mathcal{K}}$, and $\lambda \in \mathbb{R}^+$. This functional consists of two parts, whose relative influence is controlled by the positive weight λ . With $U_j = [0, 1]^{j-1} \times \{0\} \times [0, 1]^{d-j}$ we define the penalty term

$$P(\mathbf{d}) = \sum_{i=1}^{n_v} \sum_{j=1}^d \int_{U_j} \|b_{i,j}(\bar{x}) - p_i(\bar{x})\|^2 d\bar{x},$$

which measures the accuracy of the boundary representation. Additionally we consider a quality measure $Q(\mathbf{d})$, that keeps the parameterization p regular in the interior of Ω . Several possibilities are listed in [33] for the planar case, which admit extensions to the three-dimensional situation.

By minimizing the functional (5) we obtain the desired coefficients d_k and thus the parameterization p . Together with equations (2) and (4) we finally obtain the mapping u .

Note that the construction of suitable boundary parameterizations $b_{i,j}$ for $d = 3$, which are needed as input for the template mapping problem, is a non-trivial problem itself. We obtain these parameterizations by solving template mapping problems for $d = 2$ (one for each facet).

7. Segmentation examples

Example 1. We consider a regular dodecahedron, see Figure 13 (left). It is a template MS^3 with many rotational symmetries and therefore perfectly suited for midpoint subdivision. Figure 14 shows the result of the segmentation algorithm described in [23], i.e., without face pre-segmentation and midpoint subdivision. The result is valid, but not ideal for parameterization. In contrast, the use of midpoint subdivision leads to 20 symmetrical topological cuboids with planar boundary faces. One of those identical pieces is visualized in Figure 13 (right).

Example 2. We apply the ESA to a cube with two orthogonal slots on opposite sides. Although the shape of this object is not very complex, an automatic segmentation is non-trivial, due to the non-convex shape of the object. Figure 15 shows the result of the original segmentation algorithm without face pre-segmentation and without the use of midpoint subdivision. The object is decomposed into three topological cuboids and two prisms with three-sided base surfaces. However, the shapes of those output solids are again not ideally suited for constructing a volumetric parameterization.

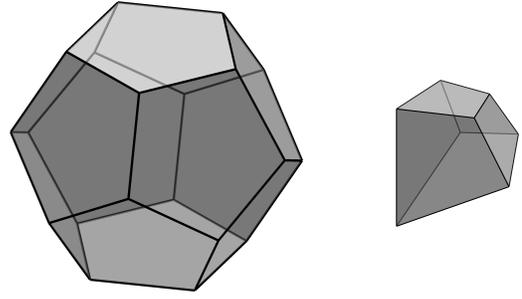


Figure 13: A regular dodecahedron and one of the 20 topological cuboids obtained via midpoint subdivision.

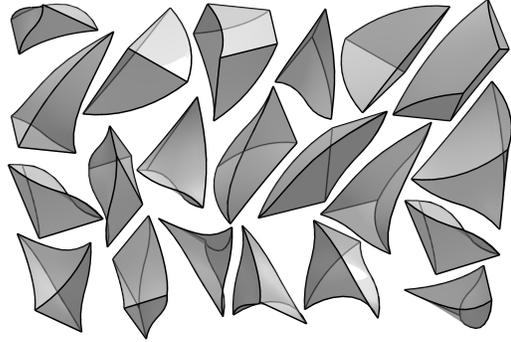


Figure 14: Base solids of the segmentation of the dodecahedron via the algorithm described in [23]. The computed cutting surfaces are all non-planar, some of them being highly curved, and therefore the resulting shapes are hard to parameterize.

The result of the extended algorithm is shown in Figure 16. Due to the use of face pre-segmentation, the first few cuts are very intuitive, since they use planar cutting surfaces that split the object into four symmetrical convex polyhedra with only trivalent vertices. In the last step, those MS^3 are further decomposed into topological cuboids with bi-linear faces by midpoint subdivision. The final decomposition is visualized in Figure 17.

Example 3. The next object is a non-convex solid, which does not contain a non-convex edge. The segmentation algorithm of [23] uses only one cut that involves the use of an implicit guiding surface [24] due to the non-convex shape of the object. Since both resulting sub-domains, which are depicted in Figure 18, are also strongly non-convex, it is very challenging to represent them as trivariate NURBS patches.

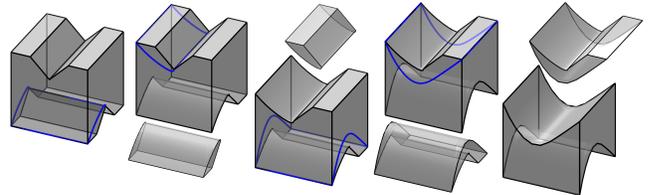


Figure 15: Segmentation of the bi-slotted cube via the method described in [23].

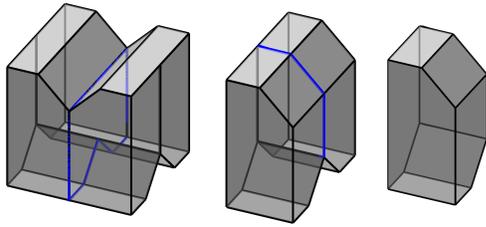


Figure 16: Segmentation of the bi-slotted cube with pre-segmented faces. Only one half/quarter is shown due to symmetry reasons.

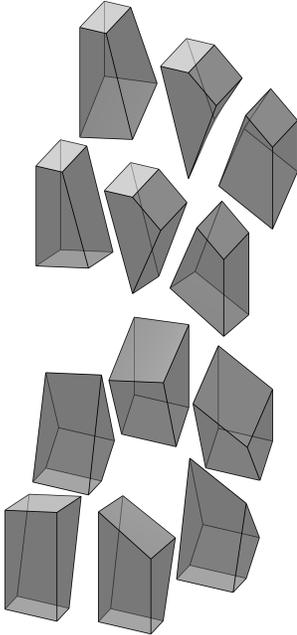


Figure 17: Midpoint subdivision of one resulting MS^3 after the face pre-segmentation.

Since this solid is already a MS^3 , one can directly use midpoint subdivision, without any previous segmentation steps. This requires the use of a hexagonal prism as template MS^3 (see Figure 10, third row, third column). Although the 12 resulting topological cuboids (cf. Figure 19) are all valid, four of them contain edges with a fairly small interior angle, which may cause problems in the subsequent parameterization step.

The reason for the occurrence of these small angles is the highly non-convex shape of the object. The face pre-segmentation step resolves this issue, fostering a split by a horizontal plane in the middle of the solid (cf. Figure 20). The subsequent use of midpoint subdivision leads to two symmetrical sets of 10 topological cuboids, which are visualized in Figure 21. The midpoint subdivision requires again a suitable template MS^3 (see Figure 10, second row, first column). Clearly, the quality of this decomposition improves upon the previous results.

Example 4. The shape of the considered solid resembles the head of an hammer. We cut the hammer along a vertical plane

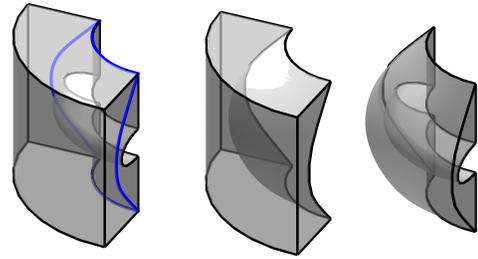


Figure 18: Solid is split into two topological cuboids using the methods of [23] and [24]. The resulting shapes are not ideal due to the highly curved cutting surface.

into two symmetrical pieces, thereby eliminating the hole for the handle in order to make the object contractible. The object is depicted in Figure 22 (top-left).

The extended segmentation algorithm (ESA) first splits off the tip of the hammer, see Figure 22. The resulting piece contains a vertex with valency 5. Therefore it is segmented further into four smaller pieces (topological prisms with three-sided base surfaces) by using cutting loops that pass through the high-valent vertex.

In the next step the remainder of the hammer is split into two MS^3 . The bottom piece is a convex polyhedron with 10 trivalent vertices. Therefore the midpoint subdivision algorithm can be applied directly to decompose this piece into 10 topological cuboids with only planar and bi-linear faces.

While decomposing the bottom piece is easy, the midpoint subdivision of the non-convex center piece is quite interesting. Based on the segmentation of a template polyhedron (see Figure 10, sixth row, second column) it is segmented into 14 topological cuboids, which are depicted in Figure 23. However, this results in four sub-domains that contain an edge with a very small interior angle.

A face pre-segmentation of the input solid resolves this issue. After the first two splits into the tip, center and bottom piece, the center piece is then subdivided by a plane into two smaller MS^3 , which are shown in Figure 24. Finally those two solids can be further decomposed by midpoint subdivision with the use of template MS^3 (see Figure 10, third row, second column). The results are shown in Figures 1 and 25. The latter picture shows the final multi-patch parameterization by NURBS volumes.

Example 5. Finally we combine the segmentation into contractible solids (see [20]¹) with the ESA. Given a solid with several tunnels, we first perform cuts that connect them to the nearest boundary, see Figure 26 (left). After the face pre-segmentation step, the original non-convex edge is eliminated by cutting through it (cf. line 5 of ESA). The two resulting solids, which are depicted in Figure 26 (right), are further decomposed into MS^3 by continuing the steps of the ESA. This is visualized in Figures 27 and 28.

¹In the examples presented in [20], the tunnels were eliminated by cutting the solid with suitably chosen planes. Here we use only segments of these planes as cutting surfaces. This keeps the original solid connected, while the other approach would already subdivide the solid into simpler pieces.

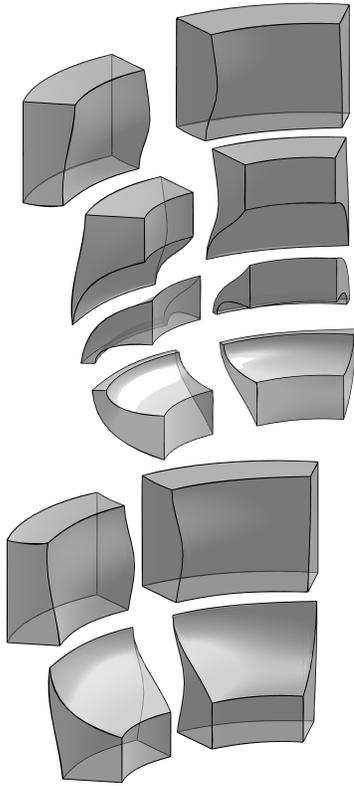


Figure 19: Midpoint subdivision with the use of a template MS^3 .

8. Conclusion and outlook

A representation of the computational domain as a bi- or trivariate spline model is a prerequisite for IgA. However, the standard within the CAD technology is to use boundary representations instead. A conversion into a format suitable for IgA is a challenging task, especially when dealing with volumetric domains. The semi-automatic subdivision steps described in [23] contribute to an isogeometric segmentation algorithm, where a given solid object in boundary representation is transformed into a collection of base solids and finally into topological cuboids. A subsequent parameterization step leads to the desired IgA-suitable domains. We presented several modifications of the ex-

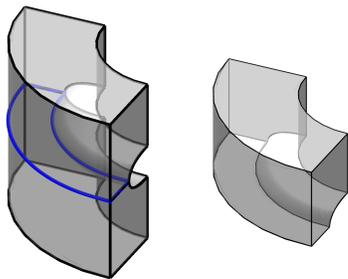


Figure 20: Solid is split into two symmetrical halves after the face pre-segmentation step.

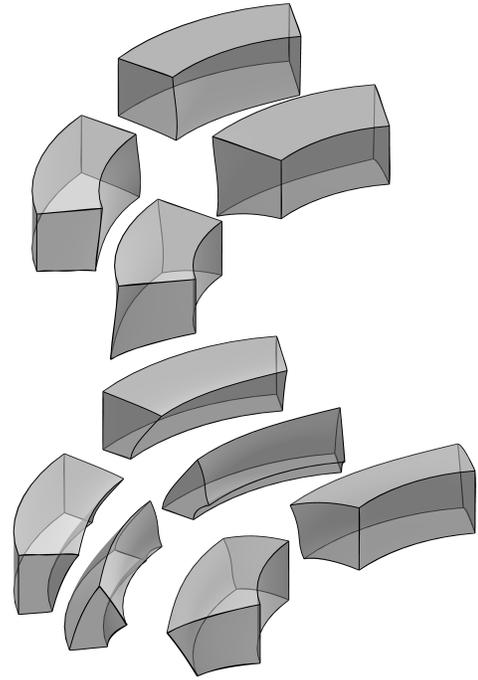


Figure 21: Midpoint subdivision after the face pre-segmentation.

isting isogeometric segmentation algorithm, which improved the shapes of the constructed sub-domains and simultaneously lead to parameterized volumetric domains.

First, the face pre-segmentation step introduces artificial edges in carefully chosen positions of the original solid, thereby increasing the flexibility of the available cutting loops. As a result, the need for strongly curved cutting surfaces is reduced substantially. We obtain better-shaped solids and improve the performance of the overall algorithm, especially by the reduced complexity of the cutting surface construction.

Second, we use midpoint subdivision to decompose a given domain into a collection of topological cuboids in one step, taking existing symmetries into account. However, this method can only be applied to solids with certain properties. Midpoint subdivision returns a collection of topological cuboids, if and only if every vertex of the input solid possesses valency three. We showed that the segmentation algorithm is capable of subdividing an input solid into a collection of midpoint subdivision suitable solids. While performing midpoint subdivision on a convex polyhedron is straightforward, its application to more general (possibly non-convex) domains requires special techniques. We proposed a template mapping approach to resolve this issue. As a by-product, we directly obtain the parameterization of the base solids as volumetric multi-patch spline domains.

Despite these improvements, there are still limitations of the isogeometric segmentation algorithm. First, due to the exponential growth of the number of possible cutting loops with the topological size of the input solid, our method is restricted to solids of rather limited combinatorial size. Second, although the constructed solids will be a valid decomposition of the input domain, their interfaces are not guaranteed to possess matching parameterizations. Thus new challenges arise when performing numeri-

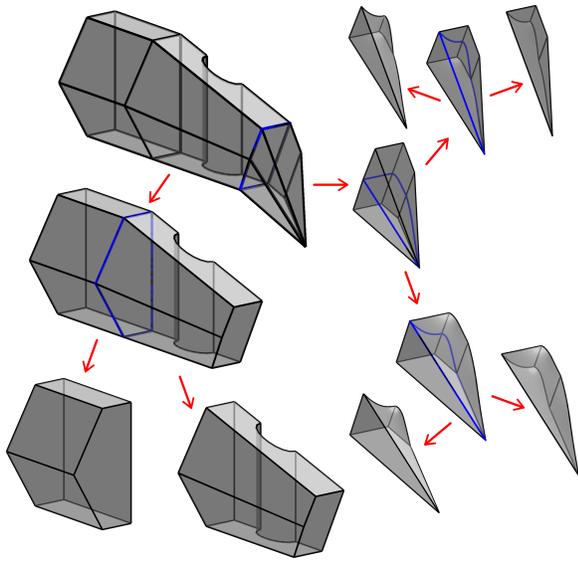


Figure 22: Segmentation of the hammer into MS^3 . The tip contains a high-valent vertex and is segmented into four topological prisms. The remainder is split into two MS^3 .

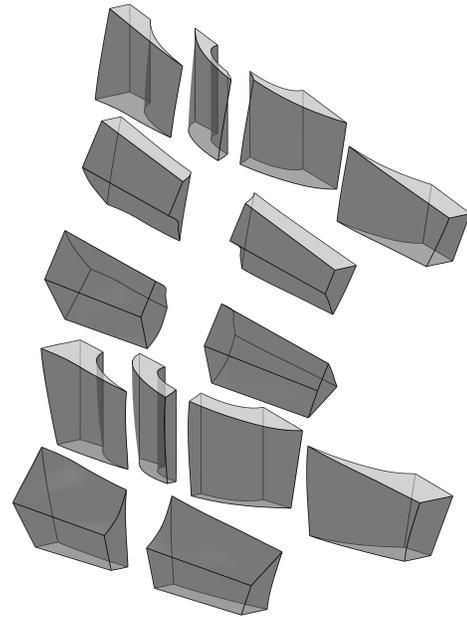


Figure 23: Midpoint subdivision of the center piece of the hammer.

cal simulations. These can be dealt with by employing discontinuous Galerkin methods [34] or similar approaches for weak coupling [35] of sub-domains in IgA. Alternatively, one may post-process the solids to generate matching interfaces. Finally, the segmentation and parameterization process relies on the choice of certain user-defined parameters, which influence the quality of the output. We are currently exploring how to use techniques of machine learning to automatize this process.

Acknowledgments. Supported by the Austrian Science Fund (FWF), NFN S117 “Geometry + Simulation”, and by the European Research Council, ERC Advanced Grant Change (GA no. 694515).

References

- [1] G. Xu, B. Mourrain, R. Duval, A. Galligo, Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications, *Comput. Aided Design* 45 (2) (2013) 395–404.
- [2] B. Jüttler, M. Kapl, D.-M. Nguyen, Q. Pan, M. Pauley, Isogeometric segmentation: The case of contractible solids without non-convex edges, *Comput. Aided Design* 57 (2014) 74 – 90.
- [3] D.-M. Nguyen, M. Pauley, B. Jüttler, Isogeometric segmentation: Part II: On the segmentability of contractible solids with non-convex edges, *Graphical Models* 76 (5) (2014) 426 – 439.
- [4] H. A. Akhras, T. Elguedj, A. Gravouil, M. Rochette, Isogeometric analysis-suitable trivariate NURBS models from standard B-Rep models, *Comput. Methods Appl. Mech. Engrg.* 307 (2016) 256 – 274.
- [5] 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.* 316 (2017) 623–645.
- [6] F. Massarwi, G. Elber, A B-spline based framework for volumetric object modeling, *Comput. Aided Design* 78 (2016) 36 – 47.
- [7] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement, *Comput. Methods Appl. Mech. Engrg.* 194 (2005) 4135–4195.

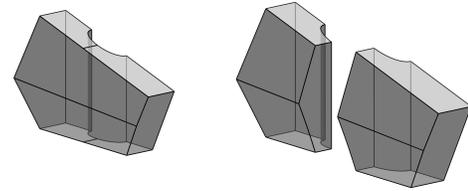


Figure 24: Left: The middle piece after the face pre-segmentation. Right: The splitting of the piece along the new auxiliary edges.

- [8] T. Hughes, G. Sangalli, Mathematics of Isogeometric Analysis: A Conspectus, in: E. Stein, R. de Borst, T. Hughes (Eds.), *Encycl. of Comp. Mechanics Second Edition*, Wiley, 2018, pp. 1–40, Part 2 Fundamentals.
- [9] M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, A.-V. Vuong, Swept Volume Parameterization for Isogeometric Analysis, in: E. Hancock, R. Martin (Eds.), *The Mathematics of Surfaces XIII*, Vol. 5654 of Lecture Notes in Computer Science, Springer, 2009, pp. 19–44.
- [10] T. Martin, E. Cohen, R. Kirby, Volumetric parameterization and trivariate B-spline fitting using harmonic functions, *Comput. Aided Geom. Design* 26 (6) (2009) 648–664.
- [11] T. Nguyen, B. Jüttler, Parameterization of Contractible Domains Using Sequences of Harmonic Maps, in: *International Conference on Curves and Surfaces*, Springer, 2010, pp. 501–514.
- [12] G. Xu, B. Mourrain, R. Duval, A. Galligo, Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method, *J. Comput. Phys.* 252 (2013) 275–289.
- [13] H. Lin, S. Jin, Q. Hu, Z. Liu, Constructing B-spline solids from tetrahedral meshes for isogeometric analysis, *Comput. Aided Geom. Design* 35-36 (2015) 109–120.
- [14] J. López, M. Brovka, J. Escobar, R. Montenegro, G. Socorro, Spline parameterization method for 2D and 3D geometries based on T-mesh optimization, *Comput. Methods Appl. Mech. Engrg.* 322 (2017) 460–482.
- [15] Y. Zhang, W. Wang, T. Hughes, Solid T-spline construction from boundary representations for genus-zero geometry, *Comput. Methods Appl. Mech. Engrg.* 249-252 (2012) 185–197.
- [16] W. Wang, Y. Zhang, L. Liu, T. J. Hughes, Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology, *Comput.*

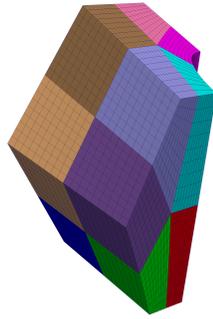


Figure 25: The final volumetric multi-patch parameterization of the front half of the center piece.

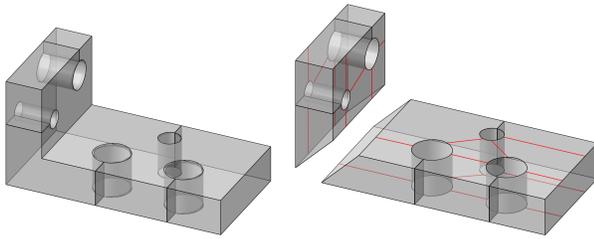


Figure 26: Left: The solid containing tunnels is contractible after the pre-processing step. Right: After the face pre-segmentation the original non-convex edge is eliminated. The artificial edges are shown in red.

- Aided Design 45 (2) (2013) 351 – 360.
- [17] L. Liu, Y. Zhang, T. Hughes, M. Scott, T. Sederberg, Volumetric T-spline construction using Boolean operations, *Engineering with Computers* 30 (4) (2013) 425–439.
- [18] L. Liu, Y. Zhang, Y. Liu, W. Wang, Feature-preserving T-mesh construction using skeleton-based polycubes, *Comput. Aided Design* 58 (2015) 162–172.
- [19] C. Chan, C. Anitescu, T. Rabczuk, Volumetric parametrization from a level set boundary representation with PHT-splines, *Comput. Aided Design* 82 (2017) 29–41.
- [20] B. Strodthoff, B. Jüttler, Automatic decomposition of 3D solids into contractible pieces using Reeb graphs, *Comput. Aided Design* 90 (2017) 157–167.
- [21] G. Xu, B. Mourrain, X. Wu, L. Chen, K.-C. Hui, Efficient construction of multi-block volumetric spline parameterization by discrete mask method, *J. Comput. Appl. Math.* 290 (2015) 589–597.
- [22] T. Martin, E. Cohen, R. Kirby, Mixed-element volume completion from NURBS surfaces, *Computers & Graphics* 36 (5) (2012) 548–554.
- [23] M. Pauley, et al., The Isogeometric Segmentation Pipeline, in: B. Jüttler, B. Simeon (Eds.), *Isogeometric Analysis and Applications 2014*, Springer, 2014, pp. 51–72.
- [24] M. Haberleitner, B. Jüttler, Isogeometric segmentation: Construction of cutting surfaces, *Comput. Aided Design* 90 (2017) 168–179.
- [25] D.-M. Nguyen, M. Pauley, B. Jüttler, Isogeometric segmentation: Construction of auxiliary curves, *Comput. Aided Design* 70 (2016) 89 – 99.
- [26] J. Y. Yen, Finding the k shortest loopless paths in a network, *Management Science* 17 (11) (1971) 712–716.
- [27] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, M. Rabl, Medial Axis Computation for Planar Free-form shapes, *Comput. Aided Design* 41 (2009) 339–349.
- [28] G. Brinkmann, B. McKay, plantri and fullgen, <https://users.cecs.anu.edu.au/~bdm/plantri/>, version 5.0, accessed: 2018-01-30.
- [29] A. Schulz, Drawing 3-Polytopes with Good Vertex Resolution, *J. Graph Algorithms Appl.* 15 (1) (2011) 33–52.
- [30] T. Li, R. McKeag, C. Armstrong, Hexahedral meshing using midpoint subdivision and integer programming, *Comput. Methods Appl. Mech. Engrg.* 124 (1-2) (1995) 171–193.
- [31] T. W. Sederberg, S. R. Parry, Free-form Deformation of Solid Geometric Models, *SIGGRAPH Comput. Graph.* 20 (4) (1986) 151–160.
- [32] F. Buchegger, B. Jüttler, A. Mantzaflaris, Adaptively refined multi-patch B-splines with enhanced smoothness, *Applied Mathematics and Computation* 272, part 1 (2016) 159–172.
- [33] A. Falini, J. Špeh, B. Jüttler, Planar domain parameterization with THB-splines, *Comput. Aided Geom. Design* 35-36 (2015) 95–108.
- [34] C. Hofer, U. Langer, I. Touloupoulos, Discontinuous Galerkin isogeometric analysis of elliptic diffusion problems on segmentations with gaps, *SIAM J. Sci. Comput.* 38 (6) (2016) A3430–A3460.
- [35] M. Ruess, D. Schillinger, A. I. Oezcan, E. Rank, Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries, *Comput. Methods Appl. Mech. Engrg.* 269 (2014) 46–71.

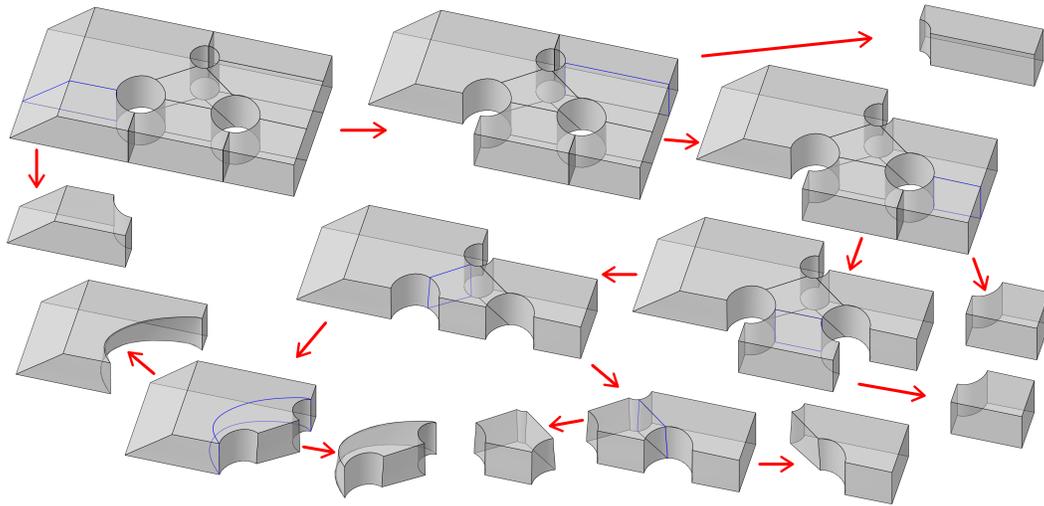


Figure 27: The final decomposition of the bottom part into MS^3 .

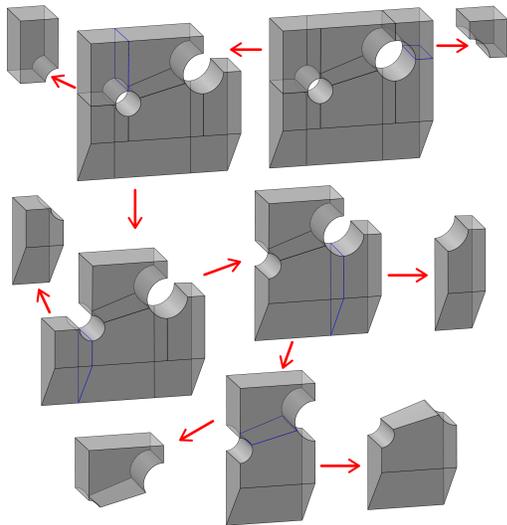


Figure 28: The final decomposition of the side part into MS^3 .