Isogeometric Segmentation. Part II: On the segmentability of contractible solids with non-convex edges

Dang-Manh Nguyen*, Michael Pauley, Bert Jüttler

Institute of Applied Geometry Johannes Kepler University Faculty of Natural Sciences and Engineering, Altenberger Straße 69, A-4040 Linz, Austria

Abstract

Motivated by the discretization problem in isogeometric analysis, we consider the challenge of segmenting a contractible boundary-represented solid into a small number of topological hexahedra. A satisfactory segmentation of a solid must eliminate non-convex edges because they prevent regular parameterizations. Our method works by searching a sufficiently connected edge graph of the solid for a cycle of vertices, called a cutting loop, which can be used to decompose the solid into two new solids with fewer non-convex edges. This can require the addition of auxiliary vertices to the edge graph. We provide theoretical justification for our approach by characterizing the cutting loops that can be used to segment the solid, and proving that the algorithm terminates. We select the cutting loop using a cost function. For this cost function we propose terms which help to select geometrically and combinatorially favorable cutting loops. We demonstrate the effects of these terms using a suite of examples.

Keywords: isogeometric analysis, volume segmentation, isogeometric discretization, edge graph

1. Introduction

Isogeometric analysis (IGA) is an approach to the solution of partial differential equations in which the functions used to approximate solutions are the same as those used to parameterize the geometry [6, 9]. IGA aims to bridge the gap between CAD and analysis communities and their technologies, and in particular, to automate as much as possible the process of translating between CAD and analysis objects.

In the *boundary representation* (BRep), a solid object is represented by a structure consisting of vertices, edges and faces. The faces are trimmed surfaces describing the boundary of the solid. In order to prepare a boundaryrepresented solid for IGA, it is necessary firstly to decompose a solid into blocks that are *topological hexahedra*, i.e., sufficiently smooth (say, continuously differentiable) embeddings of a cube, and secondly to construct a suitable parameterization for each block.

There is much existing work addressing the parameterization question under appropriate conditions. A solid is parameterized by a cube using B-splines in [16, 19] and using T-splines in [23, 24]. In [22], 3D models are parameterized by polycubes. T-spline surfaces or solids are constructed from given quad- or hex-meshes in [18]. The methods of [13] can be used to parameterize one solid by another using harmonic mappings. Swept volumes are parameterized in [1]. The authors of [10] initiated the development of a technique to produce an IGA-suitable decomposition of a boundaryrepresented solid into topological hexahedra. The goal is to produce a small number of topological hexahedra. The method was constrained to solids with only convex edges, that is, for any edge of the solid, the incident faces meet at a convex interior angle at every point of the edge. The existence of a non-convex edge creates significant restrictions on the decomposition. A satisfactory segmentation must cut the solid at this edge, because a single differentiable trivariate patch can only have convex edges. Isogeometric segmentation of solids containing non-convex edges has not previously been considered.

The present paper continues research on IGA-suitable segmentation of a solid into topological hexahedra, by extending the approach of [10] to apply to solids with nonconvex edges. We recursively decompose the edge graph of the solid into smaller subgraphs, by searching for a *cutting loop* in the edge graph. At each step of the decomposition we make sure that the subgraphs have less non-convex edges than the previous graph. Eventually the solid is de-

There is a rich theory for the segmentation of polyhedra into convex polyhedra [2–4]. Techniques have also been developed for parameterizing multiple patches of complex, arbitrary solids [20], relying on a predefined segmentation. In [14, 15] a technique for segmentation and parameterization is developed for triangulated solids, allowing for interior features. The method of [17], also for triangulated solids, uses T-splines and allows for solids of arbitrary genus.

^{*}Corresponding author

Preprint submitted to Elsevier



Figure 1: Early steps in the segmentation of a solid with non-convex edges. See Figure 11 for a complete segmentation into hexahedra and triangular prisms.

composed into new solids with only convex edges. Then, as in [10], these solids can be further decomposed into hexahedra. In Figure 1 we show the first few steps of our segmentation of a model with non-convex edges.

Non-convex edges make the search for a cutting loop more complex. They impose additional geometric constraints on valid cutting loops. Sometimes it is necessary to add new vertices to the edge graph of the solid. We provide a proof that a cutting loop can be found through any given non-convex edge. We provide a method of choosing among multiple cutting loops, which is a combination of geometric and topological criteria, and depends on choices for several parameters. Our cost function is more complex than that of [10], and we show how our new additions are important for finding reasonable decompositions.

In Section 2 we provide our assumptions about the solid and give some definitions and a way to test for a non-convex edge. We recall the *Isogeometric Segmentation Problem* as stated in [10]. Section 3 covers our approach to solids containing non-convex edges. We give a geometric criterion for the validity of a cutting loop. We provide our algorithm for splitting the edge graph of a solid that contains at least one non-convex edge.

In Section 4 we prove that there always exists a valid cutting loop through a given non-convex edge. As a consequence, combining our algorithm with that from [10] we are capable of reducing a solid to topological hexahedra. Section 5 describes our cost function for selecting among multiple cutting loops. In Section 6 we study several examples, showing how the geometric part of our cost measures the deviation from planarity, and applying several sets of parameters to several examples to explore how the choice of parameters affects the final number of topological hexahedra in the decomposition. In Section 7 we summarize our findings and discuss the outlook.

2. Preliminaries

In this section we describe our assumptions about the input solid and state the *Isogeometric Segmentation Problem*.

2.1. Assumptions

We consider a solid object S given by its boundary representation (BRep). We briefly recall from [10] that the solid object is defined as a collection of vertices, edges and faces. We assume that the edges are represented by NURBS curves, and that the faces are represented as trimmed NURBS patches. Thorough descriptions of NURBS, trimmed NURBS surfaces, and BRep are presented in, e.g., [5].

The edge graph $\mathcal{G}(\mathcal{S})$ of the solid is obtained by restricting the consideration to only the vertices and the edges of the solid.

Consider an edge \mathbf{e} , and its two neighboring faces \mathbf{f}_1 and \mathbf{f}_2 . The normal plane at a point \mathbf{p} of \mathbf{e} intersects \mathbf{f}_1 and \mathbf{f}_2 in two planar curve segments. It also contains the two outward normal vectors \mathbf{n}_1 and \mathbf{n}_2 of \mathbf{f}_1 and \mathbf{f}_2 at \mathbf{p} respectively. Let \mathbf{t}_1 and \mathbf{t}_2 be the two tangent vectors of the two planar curve segments in \mathbf{f}_1 and \mathbf{f}_2 respectively (oriented such that they point away from the edge), see Fig. 2. Recall again from [10] that the edge is said to



Figure 2: Determination of a convex point \mathbf{p} of an edge \mathbf{e} of a solid. The plane normal to \mathbf{e} (indicated by the dashed segments) contains the two vectors \mathbf{t}_1 and \mathbf{t}_2 tangent to the two faces \mathbf{f}_1 and \mathbf{f}_2 at \mathbf{p} , and their two outward normals \mathbf{n}_1 and \mathbf{n}_2 . The convex cone generated by \mathbf{t}_1 and \mathbf{n}_1 intersects that generated by \mathbf{t}_2 and \mathbf{n}_2 only at \mathbf{p} .

be convex at \mathbf{p} if \mathbf{n}_1 and \mathbf{t}_1 do not (trivially or strictly)

separate \mathbf{n}_2 and \mathbf{t}_2 , i.e., the convex cone generated by \mathbf{t}_1 and \mathbf{n}_1 only intersects the convex cone generated by \mathbf{t}_2 and \mathbf{n}_2 at \mathbf{p} . The definition is illustrated in Fig. 2. The following proposition provides an alternative easy-to-check definition.

Proposition 1. The edge \mathbf{e} is convex at \mathbf{p} if and only if

$$\langle \mathbf{n}_1 + \mathbf{n}_2, \mathbf{t}_1 + \mathbf{t}_2 \rangle < 0. \tag{1}$$

Proof. If \mathbf{t}_1 and \mathbf{t}_2 are collinear, the conclusion is affirmed as either $\mathbf{n}_1 + \mathbf{n}_2$ or $\mathbf{t}_1 + \mathbf{t}_2$ will vanish. Otherwise, the two vectors $\mathbf{n}_1 + \mathbf{n}_2$ and $\mathbf{t}_1 + \mathbf{t}_2$ are collinear. It is obvious that \mathbf{n}_1 and \mathbf{n}_2 do not separate \mathbf{t}_1 and \mathbf{t}_2 only if (1) holds. \Box

An edge of (the edge graph of) the solid is called a *convex edge* if every point on the edge is convex. Otherwise, it is called a *non-convex edge*.

In this work, we consider the *class of solids* that satisfy the following assumptions.

- (A1) Each solid is contractible. That is, it is topologically equivalent to (or more precisely homotopic to) a point.
- (A2) Each pair of neighboring edges of a solid meet each other at an interior angle that is not 0 or 2π . Furthermore, at any point of any edge, the two incident faces meet at an angle that is not 0 or 2π .¹
- (A3) The edge graph of each solid is 3-vertex-connected. That is, the edge graph has at least 4 vertices and it remains connected under the removal of any 2 vertices.²

Within this class of solids, we consider the following volume segmentation problems.

2.2. Volume segmentation problems

First we recall the *Isogeometric Segmentation Problem* stated in [10].

Problem 1. Given a solid object S (represented as a CAD model), find a collection of mutually disjoint topological hexahedra \mathcal{H}_i (i = 1, ..., n) whose union represents S. The shape of the topological hexahedra need not to be uniform, and the hexahedra are not required to meet face-to-face, thereby allowing T-joints. However, the number n of topological hexahedra should be relatively small.

In order to partially tackle Problem 1, the authors of [10] considered the following problem.

Problem 2. Solve Problem 1 where the solid object S is associated with an edge graph that has no non-convex edges and satisfies Assumptions (A1) and (A3).

We herein consider the following problem.

Problem 3. Consider a solid object S which satisfies the assumptions (A1-3). Partition S (represented as a CAD model) into a collection of solid objects S_i , i = 1, ..., n, each of which belongs to the class of solids defined by Assumptions (A1-3) and has no non-convex edges.

Problem 2 and 3 can be viewed as two steps of a total process for solving Problem 1 for the class of solids defined via Assumptions (A1-3).

3. Approach: the solid splitting algorithm

In order to tackle Problem 3, we extend the solid splitting algorithm developed in [10] to the case where the solid has non-convex edges. In detail, we split the solid into two new solids whose edge graphs each have a smaller number of non-convex edges than the original solid. We repeat this splitting step until the edge graphs of the resulting solids have no non-convex edges.

In order to present the extended algorithm, we shall generalize the concepts of auxiliary edges, cutting loops, and cutting faces in the paper [10] to the more general setting.

3.1. Definitions

Definition 1. An *auxiliary vertex* is an additional vertex in the edge graph that is an inner point of an existing edge.

An *auxiliary edge* is an additional edge in the edge graph that connects two non-adjacent (existing or auxiliary) vertices on a face.

A cutting loop is a simple closed loop of at least three (existing or auxiliary) edges of the edge graph such that every pair of non-neighboring vertices of the loop do not share a face, and no two consecutive edges are on the same face.

A cutting surface associated to a cutting loop is a multi-sided surface patch that admits the edges of the cutting loop as its boundary. It is a newly created surface patch inside the solid. The surface patch is said to be *well-defined* if it is differentiable and it separates the solid into two smaller solids.

An example of a cutting loop with auxiliary vertices and auxiliary edges is provided in Figure 3.

We note that in comparison with [10], a cutting loop defined in Definition 1 is, according to [10], already capable of splitting the 3-vertex-connected edge graph of the solid into 3-vertex-connected edge graphs. More importantly, a cutting loop can possess auxiliary vertices. Figure 3 shows an example of why auxiliary vertices can be necessary. Valid cutting loops through any of the non-convex

¹We do permit two faces meeting at an angle of π . This includes faces meeting in a G^1 smooth way.

 $^{^{2}}$ From [10, Lemma 2], assuming no face contains the same vertex twice, the edge graph is 3-connected if and only if there are no two vertices which share two faces but are not neighbors in at least one of those faces.



Figure 3: Left: A solid with an edge graph that has three non-convex edges: AD, BD, and CD. Any edge loop that contains any of the non-convex edges and possesses only existing vertices can not qualify as a cutting loop. Right: The cutting loop ADFE that contains the non-convex edge AD. Note that F is an auxiliary vertex; DF and FE are auxiliary edges.

edges can only be constructed with the use of auxiliary vertices.

We again generalize the concept of valid cutting loop in [10].

Definition 2. A cutting loop is said to be *valid* for the edge graph \mathcal{G} of a given solid if: (D1) there exists an associated well-defined cutting surface splitting the given solid into two smaller solids that again satisfy Assumptions (A1-3); (D2) all edges of the cutting loop become convex edges in the two smaller solids.

In the following part of this section, we shall characterize a valid cutting loop in terms of the way it behaves around the vertices. Consider a vertex \mathbf{v} of a cutting loop \mathcal{L} . The closure of the set of all unit vectors pointing from \mathbf{v} into the interior of the solid forms a filled spherical polygon P (See Figure 4); that is, the boundary ∂P is a simple curve in the unit sphere S^2 comprised of great arcs. By great arcs we mean arcs of great circles.



Figure 4: Left: an example solid with a potential cutting loop and a selected vertex \mathbf{v} of the loop. Right: the set of unit vectors at \mathbf{v} pointing into the solid define a spherical polygon P. The edges of the cutting loop define an arc with endpoints on the boundary of P. Proposition 2 characterizes a valid cutting loop by the angles between the arc and the boundary of P.

The following definition will be used in Proposition 2 to characterize valid cutting loops using vertex-wise criteria.

Definition 3. A cutting loop is said to be *valid at a vertex* **v** if there exists a great arc ending at the two unit tangent vectors \mathbf{z}_1 and \mathbf{z}_2 of the two edges of the cutting loop incident to **v** (oriented such that they point away from **v**) satisfying the following two conditions:

- (C1) its interior is entirely contained in the interior of P,
- (C2) it splits interior angles of the spherical polygon P at \mathbf{z}_1 and \mathbf{z}_2 into angles which are strictly between 0 and π .

In Figure 4, v is a valid vertex of the (green) cutting loop.

3.2. Criteria for a valid cutting loop

The following proposition converts the criteria of the validity of a cutting loop into the vertex-wise criteria of Definition 3.

Proposition 2. The cutting loop \mathcal{L} is valid if and only if it is valid at every vertex.

We prove necessary and sufficient conditions separately. Intuitively, we can treat the ball shown in Figure 4 as an infinitesimal neighborhood of the vertex \mathbf{v} . To prove necessity, assume that the cutting loop \mathcal{L} is valid; therefore a surface exists with boundary \mathcal{L} that cuts the solid in two; intuitively, it must also cut an infinitesimal neighbourhood of \mathbf{v} in two; this implies (C1). The surface must also cut non-convex edges into convex ones. Examining this condition at \mathbf{v} gives (C2).

To prove the sufficient condition we must choose an appropriate tangent plane at every point along the loop. Conditions (C1) and (C2) enable us to choose it at the corners; then it boils down to interpolating the tangent plane along the edges.

Proof. Necessary condition. Assume that the cutting loop is valid and consider a well-defined cutting surface associated with \mathcal{L} . Let T be the tangent plane of the cutting surface at a vertex **v** of the cutting loop. Note that \mathbf{z}_1 and \mathbf{z}_2 are not identical due to Assumption (A2). The two vectors \mathbf{z}_1 and \mathbf{z}_2 belong to the plane T and divide T into two disjoint sets: the convex cone generated by \mathbf{z}_1 and \mathbf{z}_2 and its complement in T if \mathbf{z}_1 and \mathbf{z}_2 are not collinear; or the two half-planes defined by \mathbf{z}_1 and \mathbf{z}_2 otherwise. As the two new solids satisfy Assumption (A1) and the cutting surface is differentiable, one of the two disjoint sets only contain vectors pointing from \mathbf{v} to the interior of the solid. Note that as the two new solids satisfy Assumption (A2), the two disjoint sets do not contain the unit vectors in the tangent planes of the solid's surfaces at \mathbf{v} except the two vectors \mathbf{z}_1 and \mathbf{z}_2 .

The tangent plane T intersects the unit sphere centered at **v** along two disjoint great arcs ending at \mathbf{z}_1 and \mathbf{z}_2 . As P is the set of all unit vectors pointing from **v** into the interior of the solid, one of the two disjoint great arcs must be completely contained in the interior of P except its two end points, and thereby (C1) is proven.

To facilitate the proof for (C2), we assume that each auxiliary edge of the cutting loop splits a face of the original solid into two faces. The two neighboring spherical edges of \mathbf{z}_i in ∂P correspond to the two tangent planes of the two faces incident to the edge tangent to \mathbf{z}_i . The interior angle at \mathbf{z}_i of P is the angle between the two faces measured at \mathbf{v} . As the edges in the cutting loop become convex in the new subdivided solids, the split angles should be strictly between 0 and π .

Sufficient condition. Assume that for any vertex \mathbf{v} of the cutting loop, there exists a great arc \mathcal{A} ending at \mathbf{z}_1 and \mathbf{z}_2 which lies in the interior of P except its end points. Thus, the plane that contains the great arc will subdivide the spherical polygon P, and thereby a neighborhood of \mathbf{v} in the interior of the solid, into two connected components.

Now consider three consecutive edges $\mathbf{u}^*\mathbf{u}$, \mathbf{uv} , \mathbf{vv}^* of the cutting loop, and suppose the edge \mathbf{uv} is parameterized by the curve $\mathbf{e} : [0, 1] \to \mathbb{R}^3$. We aim to define a continuous unit vector field U(t) satisfying the following properties:

- for $t \in [0, 1]$, the vector U(t) is normal to $\mathbf{e}'(t)$, and forms a convex nonzero angle with each of the faces $\mathbf{f}_1, \mathbf{f}_2$;
- for $t \in (0,1)$, U(t) points into the interior of the solid.
- U(0) is in the plane spanned by $\mathbf{e}'(0)$ and the tangent vector at \mathbf{u} pointing back along the edge $\mathbf{u}^*\mathbf{u}$;
- U(1) is in the plane spanned by $\mathbf{e}'(1)$ and the tangent vector at \mathbf{v} pointing along the edge \mathbf{vv}^* .

Let \mathbf{z}_1 and \mathbf{z}_2 be the two tangent vectors at \mathbf{v} pointing along the edges \mathbf{vu} and \mathbf{vv}^* respectively. Let γ be a great arc satisfying conditions (C1) and (C2). Set U(1) to be the unit tangent vector pointing along γ at \mathbf{z}_1 . Set U(0)analogously at \mathbf{u} . It follows from Condition (C2) that each of U(0) and U(1) makes a convex angle with each of the faces \mathbf{f}_1 and \mathbf{f}_2 .

Define unit vector fields $\mathbf{t}_1(t)$ and $\mathbf{t}_2(t)$ along \mathbf{e} as follows (cf. the vectors \mathbf{t}_1 and \mathbf{t}_2 defined at a single point of the edge in Section 2.1). At each point $\mathbf{e}(t)$, the normal plane to $\mathbf{e}'(t)$ intersects the faces \mathbf{f}_1 and \mathbf{f}_2 in a curve; let $\mathbf{t}_1(t)$ and $\mathbf{t}_2(t)$ be the unit tangent vector at $\mathbf{e}(t)$ of this curve. Now, if **e** is convex at a point $\mathbf{e}(t)$, define $q_1(t) := t_1(t), q_2(t) := t_2(t)$. Otherwise, define $q_1(t) :=$ $-\mathbf{t}_2(t), \mathbf{q}_2(t) := -\mathbf{t}_1(t)$. The construction is shown in Figure 5. The vectors $\mathbf{q}_1(t), \mathbf{q}_2(t)$ divide the normal plane of \mathbf{e} at $\mathbf{e}(t)$ into two connected components. One of these components is the (convex) subset of the plane consisting of those vectors that point into the solid and meet both $\mathbf{f}_1, \mathbf{f}_2$ at convex angles. The vector fields $\mathbf{t}_1(t), \mathbf{t}_2(t)$ vary continuously along **e**, and the choice of $\mathbf{q}_1(t), \mathbf{q}_2(t)$ can only switch at points where $\mathbf{t}_1(t), \mathbf{t}_2(t)$ point in opposite directions. Therefore $\mathbf{q}_1, \mathbf{q}_2$ are continuous. Now, U(0) divides the angle between $\mathbf{q}_1(0)$ and $\mathbf{q}_2(0)$ into some ratio,



Figure 5: Choosing the inward field U(t). In this example the edge **e** is concave at the point $\mathbf{e}(t)$. At each point $\mathbf{e}(t)$ of the edge, the vector U(t) can be chosen anywhere in the arc between \mathbf{q}_1 and \mathbf{q}_2

say a(0): 1 - a(0). Similarly, the vector U(1) divides the angle between $\mathbf{q}_1(1)$ and $\mathbf{q}_2(1)$ into a ratio a(1): 1 - a(1). Define a function $a: [0, 1] \mapsto \mathbb{R}$ by interpolating the values a(0), a(1) linearly, and then choose U(t) to be the unit vector which divides the angle between $\mathbf{q}_1(t)$ and $\mathbf{q}_2(t)$ into the ratio a(t): 1 - a(t). Thus U(t) is defined continuously and satisfies the above conditions.

In this way, a vector field is defined along each edge of the cutting loop. In general, the vector fields cannot be joined up continuously at the vertices of the loop. However, the planes spanned by $\mathbf{e}'(t), U(t)$ define a field of planes along the curves which can be joined up continuously all the way around the loop.

Choose a surface σ contained in the interior of the solid such that (i) σ is a contractible, continuously differentiable surface; (ii) the boundary of σ is the cutting loop; (iii) if edge of the cutting loop is parameterized by a curve **e**, the tangent plane to σ at each point $\mathbf{e}(t)$ is the plane spanned by $\mathbf{e}'(t), U(t)$.

Cutting the solid with the plane σ creates two solids satisfying Assumption (A2), since all the angles between edges are either angles in the original solid, or angles in the new cutting loop; these are all between 0 and 2π since the original solid satisfies Assumption (A2). Additionally, since the vector field U(t) forms a nonzero angle with both faces, no faces in the resulting solids meet tangentially. Finally, Assumption (A3) holds by applying [10, Proposition 5] with our definition of a cutting loop.

We note that Conditions (C1) and (C2) always hold for solids without non-convex edges, thus the concept of validity of a cutting loop in view of Definition 2 is a generalization of that in [10]. However, Proposition 2 also shows that it involves not only combinatorial properties of the solid and the cutting loop (as it does in the context of [10]) but also the geometric properties of the solid and the cutting loop. Moreover, the condition (C1) is the *spherical visibility* problem in computational geometry. If all spherical vertices lie in the interior of one hemisphere then it reduces to the visibility problem in a plane, which has been well studied [7]. The approach discussed in the earlier part of this section is formulated in the solid splitting algorithm SPLITSOLID-NC. It depends on a sub-algorithm CHOOSECUTTINGLOOP-NC for determining a valid cutting loop, which will be described in Section 5.

Algorithm SPLITSOLID-NC is recursive, segmenting solids until they reach base solids, which have predefined segmentations into hexahedra. The base solids are:

- *tetrahedra* which can each be segmented into 4 hexahedra by choosing an internal point and cutting with four surfaces (See [10, Figure 3]);
- *hexahedra*;
- prisms on n-sided polygons $(n \neq 4)$ which can be segmented into 3 hexahedra if n = 3. For n > 4, we segment the base polygon into quads and extend that to a segmentation of the prism into hexahedra. If n is even this gives n/2 - 1 hexahedra. If n is odd we can insert a new vertex on one edge of the base polygon before segmenting it into quads. This gives (n-1)/2 hexahedra.

Assumptions (A1) and (A3) are automatically satisfied for topological hexahedra. Assumption (A2) and the requirement of convex edges can be ensured by choosing the surfaces of the cuts so that they meet only at convex angles.

Algorithm	SPLITSOLID-NC: Splitting the edge graph of
the solid	

1:	1: procedure SPLITSOLID-NC(graph \mathcal{G})				
2:	if \mathcal{G} is a base solid then				
3:	return the predefined decomposition of \mathcal{G}				
4:	else				
5:	$\mathbf{if}\; \mathcal{G}\; \mathrm{contains}\; \mathrm{at}\; \mathrm{least}\; \mathrm{one}\; \mathrm{non-convex}\; \mathrm{edge}\; \mathbf{then}\;$				
6:	Let \mathcal{L} be the set of all valid cutting loops				
	that contain at least one non-convex edge				
7:	else				
8:	Let \mathcal{L} be the set of all valid cutting loops				
9:	end if				
10:	ChooseCuttingLoop-NC(\mathcal{L})				
11:	decompose \mathcal{G} into subgraphs \mathcal{G}_1 and \mathcal{G}_2				
12:	return SPLITSOLID-NC(\mathcal{G}_1) and SPLITSOLID-				
	$\operatorname{NC}(\mathcal{G}_2)$				
13:	end if				

13: end if14: end procedure

Due to the apparent importance of the validity of a cutting loop for the feasibility and the termination of the algorithm SplitSolid-NC, it is crucial to consider the existence of a valid cutting loop. This will be carried out in the following Section 4.

4. Existence of a valid cutting loop

We show that it is always possible to find a cutting loop through a given non-convex edge. **Theorem 1.** For any solid with an edge graph which satisfies the assumptions (A1-3), there always exists a valid cutting loop that passes through a given non-convex edge.

Theorem 1 would not be possible without the use of auxiliary vertices (See Figure 3). On the other hand, the criterion for validity at a vertex (Definition 3) holds automatically for two auxiliary edges meeting at an auxiliary vertex. So, given a non-convex edge **e** between vertices \mathbf{u}, \mathbf{v} , the goal of the proof is to build up a cutting loop which, apart from $\mathbf{u}, \mathbf{v}, \mathbf{e}$, uses only auxiliary vertices and edges. The idea of the proof can be seen in Figure 6 (left).

Proof. Consider a non-convex edge \mathbf{uv} that is incident to two vertices \mathbf{u} and \mathbf{v} . We will show that a valid cutting loop can be constructed following the two steps described below.

Step 1: First find two auxiliary vertices \mathbf{u}^* and \mathbf{v}^* such that the cutting loop will be valid at \mathbf{u} and \mathbf{v} .

Let $\mathbf{u}_0, \ldots, \mathbf{u}_m$, where $\mathbf{u}_0 = \mathbf{u}$, be all vertices of the faces that contain \mathbf{u} but do not contain \mathbf{v} . We order the \mathbf{u}_i so that each \mathbf{u}_i is on the same edge as $\mathbf{u}_{i+1}, i = 0, \ldots, m - 1$, and the edge loop with consecutive vertices $\mathbf{u}_0, \ldots, \mathbf{u}_m$ form a counterclockwise path along the boundary of the union of the faces containing \mathbf{u} or \mathbf{v} when viewed from outside of the contractible solid, see Fig. 6 (left).



Figure 6: Left: Construction of a valid cutting loop (comprised of edges in green) that passes through a given non-convex edge \mathbf{uv} . Right: The spherical polygon at a vertex \mathbf{u} of a solid whose spherical vertices are the unit tangent vectors of the incident edges to \mathbf{u} .

We consider the subsequence $\mathbf{u}_{i_1}, \ldots, \mathbf{u}_{i_u}$ of those vertices which are adjacent to \mathbf{u} . We order them so that $i_1 = 1 < \ldots < i_u = m$.

In order to analyze the validity of a cutting loop passing through \mathbf{u} , we again consider the filled spherical polygon $P_{\mathbf{u}}$ at \mathbf{u} which is the closure of the collection of the unit vectors pointing from \mathbf{u} into the interior of the solid. The spherical vertices of $P_{\mathbf{u}}$ are the unit tangent vectors $\hat{\mathbf{v}}$, $\hat{\mathbf{u}}_{i_k}$ of the edges \mathbf{uv} , \mathbf{uu}_{i_k} at \mathbf{u} (which point away from \mathbf{u}) respectively, see Fig. 6 (right).

Consider the great circle \mathcal{A} which bisects the interior angle α at $\hat{\mathbf{v}}$ of $P_{\mathbf{u}}$. Note that $\alpha < 2\pi$ due to Assumption (A2). Rotate \mathcal{A} around the line between $\hat{\mathbf{v}}$ and its antipole an angle ε to obtain a great circle $\mathcal{A}_{\varepsilon}$ so that $\mathcal{A}_{\varepsilon}$ splits the interior angle between \mathcal{A} and $\hat{\mathbf{u}}_1\hat{\mathbf{u}}_2$. If $\varepsilon \in [0, \pi - \frac{\alpha}{2}]$ then $\mathcal{A}_{\varepsilon}$ splits the interior angle at $\hat{\mathbf{v}}$ into two angles in $(0, \pi)$. There must exist $\varepsilon \in [0, \pi - \frac{\alpha}{2})$ and a spherical edge $\widehat{\mathbf{u}}_{i_k} \widehat{\mathbf{u}}_{i_{k+1}}$ such that the intersection of $\mathcal{A}_{\varepsilon}$ and the interior of $P_{\mathbf{u}}$ contains a great arc incident to $\widehat{\mathbf{v}}$ and an interior point $\widehat{\mathbf{w}}$ of the spherical arc $\widehat{\mathbf{u}}_{i_k} \widehat{\mathbf{u}}_{i_{k+1}}$, see Fig. 6 (right).

Let \mathbf{u}^* be an auxiliary vertex of the edge $\mathbf{u}_{i_k}\mathbf{u}_{i_k+1}$ at which the two tangent planes of the two faces incident to the edge are distinct. On the face containing \mathbf{u} and $\mathbf{u}_{i_k}\mathbf{u}_{i_k+1}$, connect \mathbf{u} and \mathbf{u}^* by an auxiliary edge $\mathbf{u}\mathbf{u}^*$, which is tangent to $\hat{\mathbf{w}}$ at \mathbf{u} and is not tangent to $\mathbf{u}_{i_k}\mathbf{u}_{i_k+1}$ at \mathbf{u}^* . As Conditions (C1) and (C2) of Proposition 2 are satisfied, any cutting loop which passes through $\mathbf{u}\mathbf{v}$ and $\mathbf{u}\mathbf{u}^*$ is valid at \mathbf{u} .

For vertex \mathbf{v} , we define vertices $\mathbf{v}_0, \ldots, \mathbf{v}_n$ in a completely similar way to the vertices $\mathbf{u}_0, \ldots, \mathbf{u}_m$. As a result, there exists an auxiliary vertex \mathbf{v}^* of an edge $\mathbf{v}_{\ell}\mathbf{v}_{\ell+1}$ and an auxiliary edge $\mathbf{v}\mathbf{v}^*$ so that any cutting loop passing though $\mathbf{u}\mathbf{v}$ and $\mathbf{v}\mathbf{v}^*$ is valid at \mathbf{v} .

Step 2: Find auxiliary vertices that connect \mathbf{u}^* and \mathbf{v}^* to complete a valid cutting loop.

Let \mathbf{p}_i , $i = 0, \ldots, n_{\mathbf{p}} + 1$ be the vertices of the face containing $\mathbf{u}\mathbf{v}$ and \mathbf{u}_1 so that \mathbf{p}_i is adjacent to \mathbf{p}_{i+1} , and $\mathbf{p}_0 = \mathbf{v}_n$, and $\mathbf{p}_{n_{\mathbf{p}}+1} = \mathbf{u}_1$; see Fig. 6 (left). Similarly, \mathbf{q}_j , $j = 0, \ldots, n_{\mathbf{q}} + 1$ be the vertices of the face containing $\mathbf{u}\mathbf{v}$ and \mathbf{u}_m so that \mathbf{q}_j is adjacent to \mathbf{q}_{j+1} , and $\mathbf{q}_0 = \mathbf{u}_m$, and $\mathbf{q}_{n_{\mathbf{q}}+1} = \mathbf{v}_1$.

Let \mathcal{G}_1 denote the sub-graph which is obtained from the given graph by deleting **u** and **v**. Because of Assumption (A3), \mathcal{G}_1 is connected. Compared to the original edge graph, \mathcal{G}_1 has a new face \mathbf{f}^* with vertices $\{\mathbf{u}_1, \ldots, \mathbf{u}_m, \mathbf{q}_1, \ldots, \mathbf{q}_{n_q}, \mathbf{v}_1, \ldots, \mathbf{v}_n, \mathbf{p}_1, \ldots, \mathbf{p}_{n_p}\}$.

If \mathbf{u}^* and \mathbf{v}^* belong to a face which is different from \mathbf{f}^* , then we simply connect them by an auxiliary edge on the face to form a valid cutting loop. Otherwise, when walking along the path: $\mathbf{u}_{i_k}, \ldots, \mathbf{u}_1, \mathbf{p}_{n_p}, \ldots, \mathbf{p}_1, \mathbf{v}_n, \mathbf{v}_{\ell+1}$, each time we meet an edge which connects one of the vertices of the path to another vertex not in the path, we collect an auxiliary vertex of that edge. The auxiliary vertex is chosen so that the tangent planes of the two faces incident to the edge are not identical. In the order of the collection, let the auxiliary vertices be denoted by $\mathbf{w}_1, \ldots, \mathbf{w}_N$.

If there is a pair of non-adjacent auxiliary vertices \mathbf{w}_i and \mathbf{w}_j which are on a face different from \mathbf{f}^* , we skip the vertices lying between \mathbf{w}_i and \mathbf{w}_j and connect the two vertices by an auxiliary edge. Therefore, we can assume that this is not the case. Argued in a similar way, we can assume that any \mathbf{w}_i is on a different face with \mathbf{u}^* or with \mathbf{v}^* , not counting \mathbf{f}^* .

As all auxiliary vertices \mathbf{w}_i are in \mathcal{G}_1 , none of them share a face with \mathbf{u} or \mathbf{v} in the original edge graph. Hence, the edge loop with vertices: $\mathbf{v}, \mathbf{u}, \mathbf{u}^*, \mathbf{w}_1, \ldots, \mathbf{w}_N, \mathbf{v}^*$ is a cutting loop. By choosing auxiliary edges between auxiliary vertices so that they are not tangent to the edges of the auxiliary vertices, the cutting loop is valid. \Box

Corollary 1. Consider a solid that satisfies Assumptions (A1-3). Algorithm SPLITSOLID-NC can segment the solid

into a collection of topological hexahedra satisfying Assumptions (A1-3), but without non-convex edges.

Proof. Let S be the solid that is not topologically equivalent to a tetrahedron. Theorem 1 implies that after no more steps than the number of non-convex edges of S, using Algorithm SPLITSOLID-NC, S can be segmented into new solids so that none of them contain any non-convex edges. For these solids, Algorithm SPLITSOLID-NC performs the same as Algorithm SPLITSOLID of [10], but with a modified cost function. Therefore by [10, Corollary 7], each of them will be decomposed into a collection of topological hexahedra without non-convex edges. Since each step of Algorithm SPLITSOLID-NC preserves Assumptions (A1-3) we conclude that the resulting topological hexahedra without non-convex edges.

The construction in the proof of Theorem 1 provides a valid cutting loop, but it requires many auxiliary vertices. This means that cutting along it will result in new solids with many vertices. The construction also ignores the difficulty of constructing a cutting surface for the given cutting loop. One can do much better in practice, as described in the following section.

5. Selection of cutting loops

In order to perform the solid splitting algorithm SPLITSOLID-NC, we shall present a strategy to choose a valid cutting loop from the set \mathcal{L}_N of all valid cutting loops containing at most N edges. Similar to [10], we assign to each cutting loop $\lambda \in \mathcal{L}_N$ a positive value $\mathcal{C}(\lambda)$, which from now on we will refer to as *a cost*, and choose the one with the smallest cost (note that in [10], a cutting loop with the highest cost is chosen). We extend the strategy proposed in [10] by considering more complex combinatorial and geometric criteria. For a valid cutting loop λ , we regard $\mathcal{C}(\lambda)$ as *a total cost* with the following *component costs*.

• $C_E(\lambda)$: related to the number of edges of λ . For a given vector $\boldsymbol{\alpha} = (\alpha_3, \ldots, \alpha_N)$, we define

$$\mathcal{C}_E(\lambda) = \alpha_{|\lambda|}.\tag{2}$$

where $|\lambda|$ denotes the number of edges in λ .

• $C_A(\lambda)$: related to the splitting properties of an auxiliary edge of λ . For a sequence $\boldsymbol{\beta} = (\beta_4^4, \beta_0^4, \beta_0^0, \beta_1^0, \beta_1^1)$, first we define a cost for each auxiliary edge \mathbf{e} of λ as follows: Let n_1 and n_2 be the number of edges of the two new faces split by \mathbf{e} :

$$c_{\boldsymbol{\beta}}(\mathbf{e}) = \begin{cases} \beta_4^4 & \text{if } n_1 = 4 \text{ and } n_2 = 4, \\ \beta_0^4 & \text{if } n_1 = 4 \text{ and } n_2 \text{ is even}, \\ \beta_{n_2 \mod 2}^{n_1 \mod 2} & \text{otherwise} \end{cases}$$
(3)

The considered cost is then defined as

$$\mathcal{C}_A(\lambda) = \sum_{\text{auxiliary edge } \mathbf{e} \text{ of } \lambda} c_{\boldsymbol{\beta}}(\mathbf{e}). \tag{4}$$

• $C_T(\lambda)$: related to the numbers $n_n(\lambda)$, $n_c(\lambda)$, and $n_a(\lambda)$ of non-convex, convex, and auxiliary edges in λ respectively. For a vector $\boldsymbol{\gamma} = (\gamma_n, \gamma_c, \gamma_a)$, we define the component cost as

$$C_T(\lambda) = n_n(\lambda)\gamma_n + n_c(\lambda)\gamma_c + n_a(\lambda)\gamma_a.$$
 (5)

• $C_P(\lambda)$, $\eta = (\eta_b, \eta_{GB})$: related to the planarity of an associated cutting surface of λ . Defining a cost that measures the planarity of a potential cutting surface is more complicated than the previous costs. In order to do this, we define two terms described below; in the following section an example shows the necessity of both terms. We first consider the planarity of the cutting loop $\lambda(t), t \in [0, l_{\lambda}]$ where l_{λ} is the length of the curve; herein we assume that λ is a simple, closed, and unit speed curve. In order to quantify this property, we let \mathcal{P}_{λ} be the plane of regression of the curve λ in the sense of least squares. We define

$$p_b(\lambda) = \frac{1}{l_\lambda} \int_0^{l_\lambda} d_\mathcal{P}(\lambda(t)) \,\mathrm{d}t.$$
 (6)

where $d_{\mathcal{P}}(\lambda(t))$ denotes the distance from the point $\lambda(t)$ to the plane \mathcal{P} . Note that this is not scaling invariant (it could be made scaling invariant by using $\frac{1}{l_{\lambda}^2}$ as the scaling factor) because we want to emphasize the importance of planarity for large cutting surfaces.

We further consider the planarity of an associated cutting surface to λ which is orthogonal to a prescribed continuous unit normal vector field $\mathbf{N}(t)$ along the curve $\lambda(t)$. The vectors $\mathbf{N}(t)$, λ' can be used to define the geodesic curvature $\kappa_q(t)$ (see for example [12, section 49]). Let

$$p_{GB}(\lambda) = \left| 2\pi - \int_0^{t_\lambda} \kappa_g(t) \, \mathrm{d}t - \sum_{\text{turning angles } \theta} \theta \right|.$$
(7)

Now we define

$$C_P(\lambda) = \eta_b \, p_b(\lambda) + \eta_{GB} \, p_{GB}(\lambda). \tag{8}$$

The Gauss-Bonnet theorem implies that for a planar cutting surface, the costs p_b and p_{GB} vanish. The converse may not always be true. However, we will show in the next section that looking for loops with small values of p_b and p_{GB} makes it easier to construct cutting surfaces.

For the present purpose, in computing $p_b(\lambda)$ and $p_{GB}(\lambda)$ we treat edges as straight lines. More complicated schemes will be considered in a follow-up paper.

Once the parameter vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$, and $\boldsymbol{\eta}$ are given by the user, we sort all valid cutting loops λ in \mathcal{L}_N according to their associated total cost

$$\mathcal{C}(\lambda) = \mathcal{C}_E(\lambda) + \mathcal{C}_A(\lambda) + \mathcal{C}_T(\lambda) + \mathcal{C}_P(\lambda).$$

We choose the cutting loop with the smallest total cost. alo Similar to [10], if more than one have the smallest total cost, we arbitrarily choose one of them. This selection strategy is summarized by the algorithm CHOOSECUTTINGLOOP-NC.

Algorithm CHOOSECUTTINGLOOP-NC: Selection of the cutting loop

- 1: **procedure** CHOOSECUTTINGLOOP-NC(set \mathcal{L}_N of valid cutting loops that have at most N edges)
- 2: compute for each cutting loop λ the value C with given parameter vectors α , β , γ , and η .
- 3: choose a cutting loop λ_{max} that realizes the smallest value C
- 4: return λ_{max}
- 5: end procedure

The algorithm CHOOSECUTTINGLOOP-NC depends on the construction of the set \mathcal{L}_N of valid cutting loops having at most N edges. The loopless paths between two specified vertices can be efficiently listed in order of increasing length by Yen's algorithm [21]. A straightforward way to generate \mathcal{L}_N is by iterating over possible choices of a starting edge and applying Yen's algorithm to generate loops containing that edge. Invalid loops can then be filtered out.

6. Examples

In this section, after covering some of the experimental settings, we show several examples of cutting loops to demonstrate the costs p_b and p_{GB} . We then apply our algorithm to the example solids in Figure 7, investigating the effect of parameter choices on the resulting number of topological hexahedra.



Figure 7: The example solids we use for our comparison of parameter choices. *Solid 1:* A slotted cube. *Solid 2:* A vase. *Solid 3:* The building "JKU Science Park 2". *Solid 4:* A chair, divided into two along its plane of symmetry. *Solid 5:* The half-chair of Solid 4, with the bottom moved up.

6.1. Experimental settings

The examples presented in this section are produced in accordance with the following settings.

- *Choice of auxiliary vertices.* In general, auxiliary vertices could be placed anywhere along an edge. For the purpose of our experiments, all auxiliary vertices are placed at midpoints.
- Base solids. Once a solid is segmented into solids with only convex edges, those pieces are further segmented into three types of base solids: topological hexahedra, tetrahedra and prisms. Predefined segmentations can be applied to tetrahedra and prisms to reduce them to hexahedra. In [10], topological pyramids are also considered base solids, but we exclude them here as they would be automatically segmented into tetrahedra by the present method.
- We only consider cutting loops with at most 8 edges.
- For a solid with only convex edges, we do not permit auxiliary vertices. We also only consider cutting loops which contain at least one edge that is already in the edge graph of the solid (that is, a nonauxiliary edge). This helps to speed up the search for valid cutting loops, and in our experience, such a cutting loop was always found. (As stated in Algorithm SPLITSOLID-NC, for a solid with at least one non-convex edge, we only consider cutting loops that contain at least one non-convex edge.)

6.2. Effect of the planarity cost

We will show that in several examples, the planarity cost helps to find a nearly optimal cutting loop.

In Figure 8 we show a "vase" shape formed by cutting a small cube out of the top of a larger one. We compare several potential cutting loops, measuring the value of p_b and p_{GB} .

Loop (a) is ideal because it can be interpolated with a plane. The planarity costs p_b and p_{GB} are both equal to zero. Loop (b) is contained entirely in a plane, and $p_b = 0$. but it is not possible to cut it with a plane because the surface needs to be inside the solid. Therefore any surface cutting through this loop must be highly curved. The large value of p_{GB} for Loop (b) reflects this. For Loop (c), $p_{GB} = 0$ but the loop is not contained in a plane, showing that p_{GB} is not sufficient to describe the deviation from planarity. Loops (d) and (e) are more twisted than the first 3 and have nonzero values for both p_b and p_{GB} .

6.3. Effects on the number of resulting topological hexahedra

In order to examine the effects of the combinatorial and geometric criteria discussed in Section 5 on the number of resulting hexahedra, we perform several segmentations on five example solids in which we vary the parameters associated with the combinatorial and geometric criteria, see



Figure 8: Some cutting loops together with the values of p_b , which measures the deviation of the curve from a plane, and p_{GB} , which measures the total geodesic curvature. In (b), p_b is zero and in (c), p_{GB} is zero, showing that neither of these values tells the whole story by itself.

Fig. 7. All examples in this section use the same parameter vectors: $\boldsymbol{\alpha} = (100, 0, 10, 10, 20, 20)$, and $\boldsymbol{\gamma} = (-1, 9, 20)$. The remaining parameter vectors are listed in Table 1. We note that as the two component costs C_E and C_T are examined in [10], we keep $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ as constant parameter vectors in the current study.

	$\boldsymbol{\beta} = (\beta_4^4, \beta_0^4, \beta_0^0, \beta_1^0, \beta_1^1)$	$\boldsymbol{\eta} = (\eta_b, \eta_{GB})$
Seg. 1	(0,0,0,0,0)	(0,0)
Seg. 2	(0, 0.5, 5, 20, 25)	(0,0)
Seg. 3	(0, 0, 0, 0, 0)	(100, 1000)
Seg. 4	(0, 0.5, 5, 20, 25)	(100, 1000)

Table 1: Four different choices of the parameters of the strategy for selecting a cutting loop presented in Section 5.

	Solid 1	Solid 2	Solid 3	Solid 4	Solid 5
Seg. 1	9	57	22	42	42
Seg. 2	15	51	30	28	28
Seg. 3	14	36	22	41	41
Seg. 4	7	16	19	31	42

Table 2: The numbers of the resulting topological hexahedra for the five example solids considered in this section obtained from the corresponding segmentations associated with the parameter vectors in Table 1.

In Table 2, we report the numbers of resulting topological hexahedra obtained by performing segmentations of the example solids with each set of parameters listed in Table 1. In the first three cases (for Solids 1–3) the segmentation with the smallest number of hexahedra is obtained in the presence of both of the component cost C_A , which is related to the splitting properties of the auxiliary edges of a cutting loop, and the planarity cost C_P . This may be because of the following facts:

- the cost C_A encourages the creation of new four-sided faces when a face is segmented into two by an auxiliary edge,
- the planarity cost can help to find a cutting loop which fits better with the global structure of a solid.

As a demonstration for the latter point, it is shown in Fig. 8 that the cutting loop in figure (a), which admits the vanishing planarity costs p_b and p_{GB} , splits the solid into two new solids both of which have simpler structures; each of the cutting loops (b)–(e), with non-vanishing planarity costs, splits the solid into two solids at least one of which has an even more complex structure.

Each of Figures 9, 10 and 11 shows the resulting base solids obtained from the segmentation using the parameter vectors "Segmentation 4" in Table 1.

We expect that solids which do not have (approximate) symmetries might not be segmented into a smaller number of topological hexahedra using the planarity cost C_P than using the purely combinatorial cost C_A . To examine this situation, we consider the segmentation of a "chair", see Figure 12. As the full chair is symmetric, only a half of it is considered. Such a half chair does not possesses approximate symmetries as desired. We observe that the resulting number of hexahedra in Table 2 is slightly higher for Segmentation 4 than for Segmentation 2, which does not consider C_P . Moving the vertices of the bottom of the chair slightly (See Figure 7, Solid 5) results in a significantly worse outcome for Segmentation 4. In this case, a trade-off must be made between having a small number of hexahedra, and more planar loops which are easier to cut.

Not all of the examples have been segmented in a geometrically intuitive way, and in the case of Solid 4, both the number and the degree of distortion of the base solids are not optimal. As in [10], our segmentations are only locally optimal, since the best cutting loop is chosen one step at a time. Optimizing over the set of complete segmentations would improve the outcome, but the computation time would increase enormously. An alternative way to improve the segmentations is to allow more general types of cutting loops; some possibilities are discussed in Section 7. Nevertheless, adaptive splines such as THB-splines [8, 11] can provide high quality parameterizations of the distorted volumes.

7. Conclusions

We have developed a method of segmenting the edge graph of a contractible solid given by a boundary representation. In comparison to previously known methods, our



Figure 9: *Solid 1*: Isogeometric segmentation of a "slotted cube", obtained using the parameters "Seg. 4" listed in Table 1. A loop of edges in green indicates a cutting loop which helps to subdivide the solid. The segmentation is terminated in 3 steps, and results in 4 base solids.

approach focuses on segmentation into a small number of topological hexahedra. We have extended the method of [10] by treating both convex and non-convex edges.

We only considered segmentation of the edge graph, although our algorithm makes use of geometric and combinatorial data. In a follow-up paper we will discuss strategies for constructing the new surfaces that are needed to pro-



(four congruent hexes)

(four congruent prisms: \times 3 hexes)

Figure 10: *Solid 2*: Isogeometric segmentation of a "vase", obtained using the parameters "Seg. 4" listed in Table 1. The original solid is segmented into 8 base solids. In practice, since the vase has symmetries, we would expect it to be segmented into quarters first. However, running the algorithm with the full vase helps to demonstrate the effects of the choice of parameters.

duce boundary representations of the resulting solids. This includes constructing curves that realize auxiliary edges.

There are several possible directions for extending the present research. A limitation of our algorithm is that it always makes cuts using new surfaces, when it may be possible to extend existing surfaces into the solid. The reader may be able to find a segmentation of Solid 4 into 4 nicely shaped topological hexahedra, by extending existing surfaces. A second limitation is the restriction that non-convex edges must always be eliminated first. A segmentation of Solid 1 into 5 topological hexahedra can be constructed by abandoning this restriction.

Finally, we have not considered solids which do not have a 3-vertex-connected edge graph (e.g. cylinders) or which are not contractible. It seems possible to treat these cases with a pre-processing step that creates auxiliary edges and surfaces.

Acknowledgment

The authors acknowledge the financial support from the European Union through the 7th Framework program, project "Towards Enhanced Integration of Design and Production in the Factory of the Future through Isogeometric Technologies" (TERRIFIC, grant agreement no. 284981).

References

- M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, and A.-V. Vuong. Swept volume parameterization for isogeometric analysis. In E. R. Hancock, R. R. Martin, and M. A. Sabin, editors, *Mathematics of Surfaces XIII*, volume 5654 of *Lecture Notes in Computer Science*, pages 19–44. Springer Berlin Heidelberg, 2009.
- [2] C.L. Bajaj and T.K. Dey. Convex decomposition of polyhedra and robustness. SIAM J. Comput., 21(2):339–364, 1992.
- [3] B. Chazelle. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. SIAM J. Comput., 13(3):488– 507, 1984.
- [4] B. Chazelle and L. Palios. Triangulating a nonconvex polytope. Discrete Comput. Geom., 5(5):505–526, 1990. ACM Symposium on Computational Geometry (Saarbrücken, 1989).
- [5] E. Cohen, R. F. Riesenfeld, and G. Elber. Geometric Modeling with Splines: An Introduction. A. K. Peters, Ltd., 2001.
- [6] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. Isogeometric Analysis: Toward Integration of CAD and FEA. John Wiley & Sons, 2009.
- [7] S.K. Ghosh. Visibility algorithms in the plane. Cambridge University Press, Cambridge, 2007.
- [8] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29:485–498, 2012.
- [9] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [10] B. Jüttler, M. Kapl, D.-M. Nguyen, Q. Pan, and M. Pauley. Isogeometric segmentation. Part I: Decomposing contractible solids without non-convex edges. Submitted, 2013.
- [11] G. Kiss, C. Giannelli, U. Zore, B. Jüttler, D. Großmann, and G. Barner. Adaptive cad model (re-)construction with thbsplines. *Graphical Models*, 2014, to appear.
- [12] E. Kreyszig. Differential geometry. Dover Publications Inc., New York, 1991. Reprint of the 1963 edition.
- [13] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin. Harmonic volumetric mapping for solid modeling applications. In *In Proc. ACM symp. on Solid and physical modeling*, pages 109–120, 2007.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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, $45(2){:}351-360,\ 2013.$

- [18] W. Wang, Y. Zhang, G. Xu, and T. J. R. Hughes. Converting an unstructured quadrilateral/hexahedral mesh to a rational Tspline. *Comput. Mech.*, 50(1):65–84, 2012.
- [19] X. Wang and X. Qian. An optimization approach for constructing trivariate B-spline solids. Comput.-Aided Des., 46:179–191, 2014.
- [20] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Analysissuitable volume parameterization of multi-block computational domain in isogeometric applications. *Comput.-Aided Des.*, 45(2):395–404, 2013.
- [21] J.Y. Yen. Finding the K shortest loopless paths in a network. Management Sci., 17:712–716, 1970/71.
- [22] W. Yu, K. Zhang, S. Wan, and X. Li. Optimizing polycube domain construction for hexahedral remeshing. *Comput.-Aided Des.*, 46:58–68, 2014.
- [23] 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.
- [24] 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.

Original solid



Figure 11: Solid 3: Isogeometric segmentation of a building named "JKU Science Park 2", obtained using the parameters "Seg. 4" listed in Table 1. The segmentation results in 19 topological hexahedra. Note that the solid is not symmetric, the front block is slightly bent to the right in the current view.



Figure 12: Solid 4: Isogeometric segmentation of a symmetric half of a "chair", obtained using the set of parameters "Seg. 4" listed in Table 1.