Isogeometric Segmentation: Construction of auxiliary curves

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

In the context of segmenting a boundary represented solid into topological hexahedra suitable for isogeometric analysis, it is often necessary to split an existing face by constructing auxiliary curves. We consider solids represented as a collection of trimmed spline surfaces, and design a curve which can split the domain of a trimmed surface into two pieces satisfying the following criteria: the curve must not intersect the boundary of the original domain, it must not intersect itself, the two resulting pieces should have good shape, and the endpoints and the tangents of the curve at the endpoints must be equal to specified values.

1. Introduction

Motivated by the problem of segmenting a solid into pieces suitable for isogeometric analysis (see [2]), we develop a method for segmenting a trimmed surface by constructing a cutting curve. We consider a boundary represented solid to be a collection of trimmed NURBS surfaces, together with incidence information. A trimmed surface consists of (i) a tensor product NURBS map $[0,1]^2 \to \mathbb{R}^3$ defining the *master surface*; (ii) a trimmed domain, a subset of $[0,1]^2$ which itself is represented by a collection of NURBS curves defining its boundary. The surface itself is the image of the restriction of the master spline to the trimmed domain.

In [5, 12] we initiated the development of a method for *isogeometric segmentation*, that is, the segmentation of a contractible boundary represented solid into boundary represented topological hexahedra (sufficiently smooth images of cubes). In order to be suitable for isogeometric analysis, the number of topological hexahedra in the segmentation should be small and it should be possible to convert each of them into a volumetric spline with no singularities and ideally not too much distortion. (Conversion of boundary represented topological hexahedra into volumetric splines can be done, using, for example, the method of [20].) The idea of the method of [5, 12] is as follows:

- Search the *edge graph* of the solid (the graph formed by the edges shared by trimmed surfaces) for a cutting loop: a closed curve in the boundary of the solid which can be used as the boundary of a surface that splits the solid into two simpler solids (see Figure 3). If multiple cutting loops are found, choose the best one according to a combination of combinatorial and geometric criteria.
- By recursively splitting the solid into simpler pieces, we end up with a collection of base solids (topological hexahedra, tetrahedra and prisms) which have predefined segmentations into topological hexahedra.

In [5, 12] the search for a cutting loop is "combinatorial" in the sense that it only results in a description of the list of faces, edges and vertices the closed curve passes through. The cutting loop generally consists of a combination of edges already existing in the solid's edge graph, as well as new *auxiliary* edges that need to be added to split some faces into two. When the solid is split, each face that the curve passes through will be split into two. These two pieces should have "good shape", and the purpose of this paper is to propose a way of measuring "good shape" and a method of constructing curves which can achieve it.

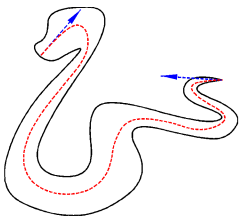

Figure 1: An example of a planar domain (with boundary given by the black curve) and a splitting curve (red dashed curve). The curve is required to meet the domain at specified endpoints with specified tangent vectors, and split the domain into two pieces with reasonable shape.

Our problem is to split a planar domain in two, using a spline curve with the following properties: *(i)* end points and tangents are specified, *(ii)* the curve is reasonably regular, and *(iii)* the curve cuts a simply connected domain into two pieces with good shape.

An example, of somewhat exaggerated difficulty, is given in Figure 1. A domain is shown together with the required tangent vectors at the endpoints. A curve is shown which splits the domain into two pieces and meets the endpoints in the specified directions.

For a complicated domain, it may be easy to split the domain into 2 or more pieces with simple cuts that ignore the endpoint requirements. For example, the snake in Figure 1 might be cut into any number of pieces with

straight or nearly straight lines transversal to the snake's body. Alternatively, the domain might be approximated by a quad mesh. These approaches are more appropriate than ours for segmenting a 2D object without restrictions. However, in order to realize the cutting loop that is central to the 3D segmentation algorithm outlined above, we must construct a curve that meets the specified end points with specified tangents.

The basic idea of our procedure is as follows. We define a cost function on the space of curves, which consists of a measurement of how close the curve comes to intersecting the domain, how close it comes to intersecting itself and a regularity term. An initial curve is found for this optimization problem by first finding a piecewise linear curve inside the domain satisfying the endpoint conditions, refining it and using the vertices as control points for a degree 2 B-spline.

It is important that the resulting curves are non-self-intersecting. In this sense, our problem is related to the problem of self-intersection elimination in curves and surfaces, studied in [15, 17, 3]

Several other approaches to the solution are possible. We might treat the problem as a path planning problem with obstacles (for overviews of this large topic in robotic control, see for example [7] and [8, Part IV]). Path planning using curves of piecewise constant curvature is studied in [4, 1] among other papers. Curves with continuous piecewise linear curvature were proposed in [6] for interpolating between postures, and used in [18] to produce a local path planner (these are not polynomial splines). Path planning with B-splines is studied in e.g., [19, 11, 10, 9]. Our problem differs from path planning problems in several ways. The constructed curve must have endpoints on the boundary of the domain, and elsewhere must not intersect the boundary. The curve cuts the domain into two pieces, and it is the shape of these pieces that is important. Path planning methods are not designed to produce a well shaped splitting of the domain. Many path planning methods seek a curve in a configuration space. Our curves do not represent the trajectory of a rigid body. Also, simplicity of the resulting B-spline curve (in terms of the degree and number of knots) is important.

We considered the following approach to splitting the domain: map the domain to a convex shape (say, a circle) using mean value coordinates (MVC) or harmonic functions. It is easy to produce a B-spline curve connecting points on the boundary, and, by sampling the curve and using a predictor-corrector method, we can construct its preimage in the original domain. This approach is plausible but in our experience, several difficulties arise from this approach: (i) the outcome is a sampled curve, and a fitting step must be included to convert the samples into a B-spline. It is difficult to tell how many samples are needed to ensure that the fitted B-spline stays within the domain; (ii) using the boundary element method to compute MVC or harmonic functions is numerically challenging near the boundary of the domain; (iii) the quality of the result is dependent on how the curve is chosen in the convex domain, and it is not clear how to make a good choice.

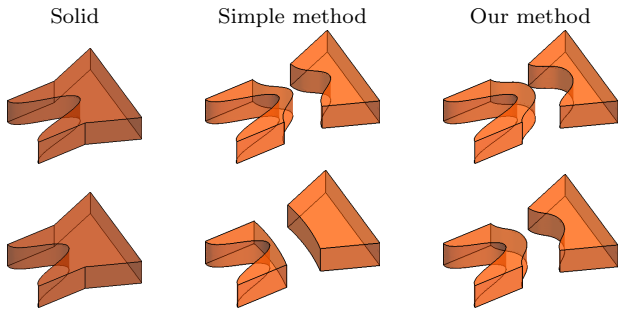

Figure 2: One of the alternative approaches we considered, outlined in Section 1, is based on cutting the domain with line segments and is unaware of the boundary except at the points where the line segments intersect it. As a result it can sometimes create badly shaped pieces. The method of this paper is not susceptible to this problem.

Another approach we considered is to fit a spline curve to a polygonal path which is found using a pathfinding method that stays away from the domain boundary $\partial\Omega$ as follows: draw a line segment beginning at one of the end points in the initial specified direction, and ending at the first intersection point y that the line has with $\partial\Omega$. This divides the domain into two pieces. Discard the piece which does not contain the other endpoint. The first half of the line segment is included in the polygonal path, the midpoint of the line segment is set as the new starting point, and a new initial direction is chosen which is parallel to the tangent direction of $\partial\Omega$ at y. Repeat the procedure until there is a line segment directly to the remaining endpoint. This procedure produces a curve fairly quickly but in certain scenarios can result in unnecessarily thin shapes as shown in Figure 2.

In Section 2 we give precise statements of our assumptions and goals and describe our construction method. Section 3 describes our method for constructing an initial curve. In Section 4 we formulate the optimization problem which we use to find a suitable splitting curve, and outline our strategy for solving the optimization problem. Section 5 provides examples of varying difficulty, and in Section 6 we summarize how our work fits in to the isogeometric segmentation problem and describe the outlook and future work. Appendix A and Appendix B prove regularity properties of the penalty functions, and in Appendix C we provide a method for efficiently computing them.

2. Preliminaries

2.1. The trimmed surface splitting problem

For context, we briefly outline the strategy described in [5, 12] for segmenting a solid. For our purposes, a solid is represented by a collection of trimmed surfaces, together with incidence information. We assume that the solid is

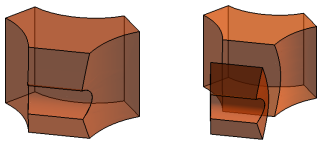

Figure 3: Left: an example of a boundary represented solid. Based on the methods of [5, 12], a segmentation of the solid into topological hexahedra was given in [14] and shown to be suitable for isogeometric analysis. For the examples of that paper, a preliminary version of the present paper's method was used to produce the auxiliary curves. Right: the solid is split into two simpler solids. Two auxiliary edges need to be constructed to make this segmentation.

contractible and that the edges form a simple, 3-vertex-connected edge graph. The edges of the graph are incidences between two faces. An *auxiliary edge* can be constructed between any two vertices that are on the same face but not on the same edge. A cutting loop is a cycle consisting of existing edges and auxiliary edges. The cutting loop can be used to split the edge graph into two, and the cutting loop is *valid* if it can be used as the boundary of a surface which splits the solid into two pieces, each satisfying the original assumptions. A valid cutting loop always exists, with the provision that under some circumstances, additional *auxiliary vertices* may need to be added to the graph. Repeatedly splitting the solid into two simpler pieces using cutting loops, we eventually arrive at a collection of *base solids* which have predefined segmentations into topological hexahedra. Examples of a single step of the solid segmentation method are shown in Figures 2 and 3. In each of these examples, two existing edges and two newly created auxiliary edges are used to form a cutting loop.

There are geometric conditions on the validity of a cutting loop which place constraints on the sequence of edges. Once these constraints are known, the search for a good cutting loop becomes a combinatorial problem. However, once the loop is chosen, each auxiliary edge needs to be realized geometrically. In other words, for each auxiliary edge in the cutting loop, we need to solve the following problem.

Problem 1 (Trimmed surface splitting problem)**.** Find a spline curve of the trimmed domain, so that the trimmed surface is split into two new trimmed surfaces with reasonably good shapes.

Due to the geometric conditions for a valid cutting loop, a solution to the trimmed surface splitting problem must be able to handle specified tangent directions at the endpoints.

2.2. The domain splitting problem

We now focus on the *domain splitting problem*, the part of the trimmed surface splitting problem that does not take consideration of the way the surface is embedded.

Consider a simply connected domain $\Omega \subset \mathbb{R}^2$ with connected interior. We restrict our consideration to the case where the boundary $\partial\Omega$ of Ω is formed by spline curves [16]. Suppose two points $\mathbf{A}$ and $\mathbf{B}$ on the boundary of Ω are given, along with two nonzero vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that point from $\mathbf{A}$ and $\mathbf{B}$ to the domain interior respectively. Without losing generality, we assume that the boundary $\partial\Omega$ consists of two spline curves $\mathbf{b}(v)$ and $\widetilde{\mathbf{b}}(v)$, $v \in [0,1]$, so that $\mathbf{b}(0) \equiv \widetilde{\mathbf{b}}(0) \equiv \mathbf{A}$, $\mathbf{b}(1) \equiv \widetilde{\mathbf{b}}(1) \equiv \mathbf{B}$. The notations are illustrated in Figure 4.

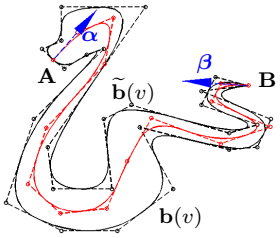

Figure 4: A simply connected snake-shaped domain enclosed by two spline curves $\mathbf{b}(v)$ and $\widetilde{\mathbf{b}}(v)$. A splitting curve of the domain is shown (red).

Additionally, we narrow our consideration to the situation specified by the following assumptions.

(A-1) The two curves $\mathbf{b}(v)$ and $\widetilde{\mathbf{b}}(v)$ have non-vanishing tangents at their end points.

(A-2) Domain Ω has non-zero interior angles at $\mathbf{A}$ and $\mathbf{B}$; the prescribed velocity direction $\boldsymbol{\alpha}$ is not tangent to any of the boundary curves at $\mathbf{A}$; likewise, $\boldsymbol{\beta}$ is not tangent to any of the boundary curves at $\mathbf{B}$.

Throughout the remainder of the paper, we shall often associate discussions with the curve $\mathbf{b}(v)$ while referring to $\widetilde{\mathbf{b}}(v)$ as a similar case. Assume that the curve $\mathbf{b}(v)$ is of the form $\mathbf{b}(v) = \sum_{j=1}^{n} \mathbf{b}_j N_j(v)$, where N_j are B-splines of degree q and associated with an open knot vector $\mathcal{V} = \{0, \ldots, 0 = v_1, v_2, \ldots, v_{n-1}, v_n = 1, \ldots, 1\}$ (the first knot and the last knot are repeated $q+1$ times), and $\mathbf{b}_j \in \mathbb{R}^2$ are associated control points.

Consider a spline curve $\mathbf{c}(u)$ of the form

$$\mathbf{c}(u) = \sum_{i=1}^{m} \mathbf{c}_i M_i(u), \tag{1}$$

where M_i are B-splines of a degree p and associated with an open knot vector

$$\mathcal{U} = \{0, \ldots, 0 = u_1, u_2, \ldots, u_{m-1}, u_m = 1, \ldots, 1\} \tag{2}$$

(again, the first and the last knots are repeated $p+1$ times), and $\mathbf{c}_i \in \mathbb{R}^2$ are control points. The following definition will facilitate our discussions in the remainder of the paper.

Definition 1. A continuous spline curve $\mathbf{c}(u)$, $u \in [0,1]$, is said to be a *splitting curve* of the domain Ω if it satisfies the following conditions.

(C-1) The curve $\mathbf{c}$ starts at $\mathbf{A}$ tangentially to $\boldsymbol{\alpha}$ and ends at $\mathbf{B}$ tangentially to $\boldsymbol{\beta}$. That is,

$$\begin{aligned} \mathbf{c}(0) = \mathbf{A},\ \mathbf{c}(1) = \mathbf{B}, \\ \det[\mathbf{c}'(0), \boldsymbol{\alpha}] = 0,\ \det[\mathbf{c}'(1), \boldsymbol{\beta}] = 0. \end{aligned} \tag{3}$$

(C-2) We divide this condition into the following 3 sub-conditions.

(i) $\mathbf{c}$ is simple over $[0, 1]$, i.e., $\mathbf{c}(u) \neq \mathbf{c}(v)$ for all $u \neq v$.

(ii) $\mathbf{c}'_+(u) \neq 0$ for $0 \leq u < 1$, and $\mathbf{c}'_-(u) \neq 0$ for all $0 < u \leq 1$.

(iii) For all $0 < u < 1$ and $\beta > 0$, $\mathbf{c}'_+(u) \neq -\beta \mathbf{c}'_-(u)$.

(C-3) The curve $\mathbf{c}(u)$ is contained in the interior of the domain Ω except its two end points. That is, $\{\mathbf{c}(u),\ 0 < u < 1\} \subset \text{int}(\Omega)$.

It immediately follows from the definition that a splitting curve of a simply connected domain decomposes the domain into two new simply connected domains.

We can now state our domain splitting problem, which differs from the trimmed surface splitting problem by not considering the embedding in $\mathbb{R}^3$.

Problem 2 (Domain splitting problem)**.** Find a splitting spline curve $\mathbf{c}(u)$, $u \in [0, 1]$, of the domain Ω such that it has few control points and it splits Ω into two new simply connected domains with reasonably good shapes

In Problems 1 and 2, a curve with "good shape" would stay far away from intersecting itself or the domain boundary, and would also be quite regular. Thus the measurement of the quality of shape is based on a combination of penalty functions, which we define in Section 4.

We solve this problem using the algorithm outlined in the next section.

2.3. Outline of the algorithm

Consider the domain Ω and the domain splitting problem stated in Section 2.2. Our algorithm for numerically solving this problem is outlined below.

Curve initialization. First, we construct a splitting piecewise-linear curve $\boldsymbol{\gamma}(u)$ of Ω, i.e., a curve that fulfils Conditions (C1-3). Additionally, it is useful to require the curve to satisfy that (a tighter condition than Condition (C2)): the distance from any one point on the curve to a point on the domain boundary is bounded from below by a given tolerance, except at the two ending segments where the curve meets the domain boundary. The splitting curve is not yet required to satisfy the geometric criterion that it should subdivide Ω into two new domains with good shapes. The purpose of this step is to provide an initial curve, that is a splitting curve, to more sophisticated procedures in the later step of the algorithm. This step is detailed in Section 3.

Curve optimization. Based on the splitting piecewise-linear curve obtained from the last step, we construct a splitting spline curve of higher polynomial degree using a two-stage optimization strategy as follows.

- *Stage 1*: find a splitting piecewise-linear curve that minimizes a *penalty function* which approaches infinity when the curve tends to violate Conditions (C2-3). We use a suitable refinement of the piecewise-linear curve $\boldsymbol{\gamma}(u)$ obtained in the last step as the initial curve for the optimization.

- *Stage 2*: find a splitting spline curve of degree $p \geq 2$ that minimizes the mentioned penalty function. A sufficiently refined spline fitting to the curve obtained from Stage 1 is used as the initial curve for the optimization.

We note that the optimization of Stage 1 is less expensive than that of Stage 2 as the spline curve under optimization is only of degree 1. Stage 1 helps to provide a good feasible initial solution for the optimization of Stage 2. That is, Stage 1 can result in a curve that is closer to a local minimum than a spline fitting to the initial curve from Step 1. The precise description of this step shall be presented in Section 4. The algorithm outline is illustrated in Figure 5.

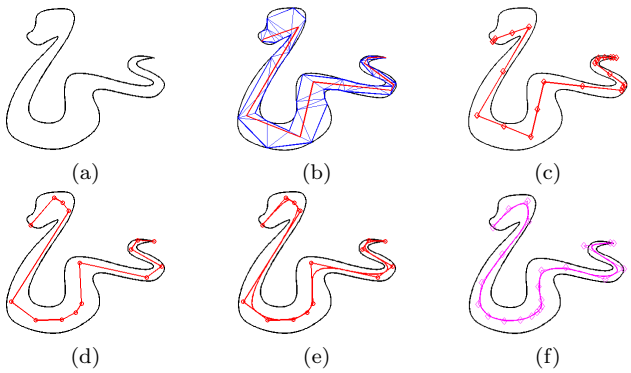

Figure 5: Illustration of the steps of the algorithm outline. (a) The domain. (b) A polygon is constructed contained in the domain, it is triangulated and a pathfinding algorithm is used to construct a piecewise linear spline. (c) The result of Step (b) is refined. (d) The result of Step (c) is used as the initial curve in the first optimization stage. (e) The piecewise linear curve from Step (d) is used as the control polygon for a degree 2 B-spline. (f) The result of Step (e) is used as the initial curve in the second optimization stage.

3. Splitting curve initialization

We describe our method for constructing a splitting piecewise linear path of the domain Ω. We construct a polygon P interior to Ω and use a Delaunay triangulation and Dijkstra's algorithm to produce a path.

A polygon is computed such that it is contained in Ω, contains $\mathbf{A}$ and $\mathbf{B}$ and its interior contains $\mathbf{A} + t\boldsymbol{\alpha}$ and $\mathbf{B} + t\boldsymbol{\beta}$ for all sufficiently small t.

(i) Find all the corners and inflection points of $\partial\Omega$, and break it up into segments of three types: where the curvature is positive, zero, or negative. Segments of $\partial\Omega$ where

the curvature is zero can be copied directly into P. Segments where the curvature is positive can be discretized as follows: evaluate the segment of the curve at uniformly spaced parameters, and construct a polygonal path between the resulting points. Segments where the curvature is negative can be discretized as follows: evaluate the curve segment and its derivatives at uniformly spaced parameters, then construct a polygonal path such that the path's endpoints and tangents coincide with the curve segment's endpoints and tangents, and the path is tangent to the curve at the evaluated points.

(ii) Refine P as necessary. Refining in the positively or negatively curved segments simply involves adding more parameter values and recomputing the polygons. The segments with zero curvature do not need to be refined. There are two ways in which refinement can be necessary. The segments containing $\mathbf{A}$ or $\mathbf{B}$ may need to be refined, to ensure that the polygon angles around $\mathbf{A}$ and $\mathbf{B}$ contain the directions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$; refinement may also be required to eliminate self-intersection of P. If two segments of P intersect each other, they are both refined. As the boundary $\partial\Omega$ does not intersect itself, a sufficiently (locally) refined polygon P will not have intersecting non-neighbouring edges.

Once the polygon P is constructed, we compute a triangulation (we use a constrained Delaunay triangulation). A graph $\mathcal{G}$ is formed, whose vertices are *(i)* midpoints of the edges of the Delaunay triangulation that are not edges of P; and *(ii)* the points $\mathbf{A}_1 = \mathbf{A} + \frac{t_1}{2}\boldsymbol{\alpha}$ and $\mathbf{B}_1 = \mathbf{B} + \frac{t_2}{2}\boldsymbol{\beta}$, where t_1 and t_2 are chosen small enough that $\mathbf{A}_1$ and $\mathbf{B}_1$ are inside the domain.

Two vertices of the graph $\mathcal{G}$ have an edge between them if the line segment between them does not intersect the polygon P. Note that there always exists an edge between two nodes which are in the same triangle of the triangulation. Since $\partial\Omega$ has no self-intersections, there must be a sequence of triangles in the triangulation, starting from the triangle containing $\mathbf{A}_1$ and ending with the triangle containing $\mathbf{B}_1$, such that each consecutive pair of triangles in the path shares an edge. Therefore, there must exist a path in $\mathcal{G}$ from $\mathbf{A}_1$ to $\mathbf{B}_1$.

Assign to each edge of $\mathcal{G}$ a cost equal to its Euclidean length. Dijkstra's algorithm (see for example [8, Section 2.2.2]) finds a path of minimal total cost between two vertices in $\mathcal{G}$, where the total cost of a path is the sum of the costs of all edges in that path. Thus, Dijkstra's algorithm produces a path of minimal total length from $\mathbf{A}_1$ to $\mathbf{B}_1$ in $\mathcal{G}$. (Before applying Dijkstra's algorithm, we can remove those edges from $\mathcal{G}$ which are too close to the domain boundary and do not contain $\mathbf{A}_1$ or $\mathbf{B}_1$. This improves the initial state for the next step of the algorithm, but may in rare circumstances prevent the existence of a path.)

The line segments from $\mathbf{A}$ to $\mathbf{A}_1$ and from $\mathbf{B}_1$ to $\mathbf{B}$ are added to form a path from $\mathbf{A}$ to $\mathbf{B}$. If necessary, self-intersections of the resulting path are eliminated by cutting off the loops. The path is treated as a linear spline with evenly spaced knot points. The initial curve constructed in this way satisfies the condition (C1-3).

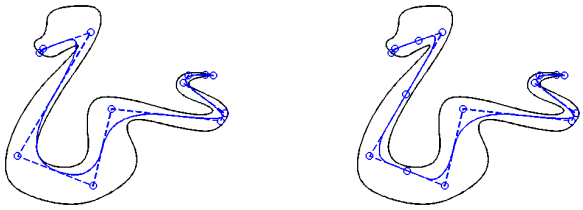

Figure 6: Left: The vertices of the path produced in Step 1 (see Figure 5 (b)) are directly used as the control points of a quadratic B-spline curve. The result intersects the boundary of the domain. Right: The piecewise linear path is refined until the corresponding quadratic B-spline curve does not intersect the boundary of the domain. This is still not a good choice of curve since it can get very close to the boundary.

4. Splitting curve optimization

A splitting spline curve can be obtained as a sufficiently refined spline fitting of a splitting piecewise-linear curve produced by the approach discussed in Section 3. However, the two new domains split from the original domain Ω by such a spline curve may not have good shapes. See, for example, Figure 6. This section will propose an optimization framework which helps to find a splitting spline curve such that the two corresponding split domains have relatively good shapes. The optimization is based on minimization of a penalty function that tends to infinity when the curve tends to violate Conditions (C2-3).

We note that using penalty functions is a fundamental method for numerically solving constrained optimization in general [13], or for particular applications such as for solving discrete HJB equations [21].

In order to represent the constructions of the penalty functions, we shall address each of the treatments of Conditions (C3) and (C2) separately, and formulate the optimization problem afterward.

Throughout the remainder of the paper, if there is a continuous function $f(u, v)$ that has a finite or infinite limit at some point (u_0, v_0) when (u, v) tends to (u_0, v_0), we will write $f(u_0, v_0)$ to mean the limit of the function at the point (u_0, v_0).

4.1. Curve-to-boundary penalty function

In order to treat Condition (C3), we consider the inverse of the squared distance between two points $\mathbf{c}(u)$ and $\mathbf{b}(v)$:

$$\mathcal{I}_{\mathbf{b}}(u, v) = \frac{1}{\|\mathbf{c}(u) - \mathbf{b}(v)\|^2}, \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^2$. The function can be viewed as a penalty function as it penalizes the curve $\mathbf{c}(u)$ from intersecting the boundary curve $\mathbf{b}(v)$ from the point of view of minimization. This is the *curve-to-boundary penalty function*. However, because of Condition (C1), the function $\mathcal{I}_{\mathbf{b}}(u, v)$ is unbounded in any neighborhood of one of the two corner points $(0, 0)$ and $(1, 1)$ of the unit square. As a result, the function is not Riemann

integrable over the unit square. Even in the sense of improper integrals, the integral of the function over the unit square does not converge either, see Lemma 1(ii). By the following theorem, it is possible to construct a multiplier function $r_{\mathbf{b}}(u,v)$ so that the new function $r_{\mathbf{b}}(u,v)\,\mathcal{I}_{\mathbf{b}}(u,v)$ is Riemann integrable over the unit square and it preserves penalizing properties of the original function.

Theorem 1. *Consider the two spline curves $\mathbf{c}(u)$, $0 \le u \le 1$, and $\mathbf{b}(v)$, $0 \le v \le 1$ defined in Section 2.2, with $c(0) = b(0)$ and $c(1) = b(1)$. Assume that*

1. *$p \ge 1$, $q \ge 1$, recalling that p and q are spline degrees of $\mathbf{c}(u)$ and $\mathbf{b}(v)$ respectively;*
2. *$\mathbf{c}'_+(0) \neq 0$ and $\mathbf{c}'_-(1) \neq 0$.*
3. *$\mathbf{c}'_+(0) \neq \beta_0 \mathbf{b}'_+(0)$ and $\mathbf{c}'_-(1) \neq \beta'_1 \mathbf{b}'_-(1)$ for all $\beta_0 > 0$ and $\beta_1 > 0$.*

We can construct a continuous piecewise-rational function $r_{\mathbf{b}}(u,v)$ so that

$$\widehat{\mathcal{I}}_{\mathbf{b}}(u,v) = r_{\mathbf{b}}(u,v)\,\mathcal{I}_{\mathbf{b}}(u,v) = \frac{r_{\mathbf{b}}(u,v)}{\|\mathbf{c}(u) - \mathbf{b}(v)\|^2} \tag{5}$$

satisfies the following properties:

(i) if $\mathbf{c}(u)$ intersects $\mathbf{b}(v)$ only at $u = 0$ and $u = 1$, then $\widehat{\mathcal{I}}_{\mathbf{b}}$ is Riemann integrable over the unit square;

(ii) $\widehat{\mathcal{I}}_{\mathbf{b}}$ equals $C\,\mathcal{I}_{\mathbf{b}}$, for some $C > 0$, everywhere in $[0,1]^2$ except in $[u_1,u_2] \times [v_1,v_2] \cup [u_{n-1},u_n] \times [v_{n-1},v_n]$ (where $u_1,\ldots,u_n$ are the knots of $\mathbf{c}$ and $v_1,\ldots,v_n$ are the knots of $\mathbf{b}$);

(iii) if $\mathbf{c}(u)$ intersects $\mathbf{b}(v)$ at a third point (u^*,v^*), i.e., $0 < u^* < 1$, then $\iint_{[0,1]^2} \widehat{\mathcal{I}}_{\mathbf{b}}(u,v)\,\mathrm{d}u\,\mathrm{d}v = +\infty$.

We present the proof for the theorem in Appendix A. The multiplier function has the following form

$$r_{\mathbf{b}} = w_0\,\alpha_0\,f_0 + w_\tau\,\alpha_\tau\,1_{[0,1]^2} + w_1\,\alpha_1\,f_1, \tag{6}$$

where α_0, α_τ, and α_1 are some positive constants; w_0, w_τ, and w_1 are piecewise rational functions given by (A.9) that only depend on the knot vectors $\mathcal{U}$ and $\mathcal{V}$; f_0 and f_1 are polynomials given by (A.5) that are quadratic functions of the coordinates of the control points of $\mathbf{c}(u)$. The function $1_{[0,1]^2}$ is the constant 1 function on the domain. The construction depends on the curves having at least one internal knot.

4.2. Curve-to-itself penalty function

Similar to the treatment of Condition (C3) discussed above, we consider the following penalty function which is related to the inverse of the squared distance between two points $\mathbf{c}(u)$ and $\mathbf{c}(v)$ of the splitting curve:

$$\mathcal{J}(u,v) = \frac{(u-v)^2}{\|\mathbf{c}(u) - \mathbf{c}(v)\|^2}. \tag{7}$$

We refer to this function as a *curve-to-itself penalty function*. It turns out that it is much easier to handle this function compared to the curve-to-boundary penalty function. This will be shown by the following theorem.

Theorem 2. *Let $\mathbf{c}(u)$, $0 \le u \le 1$, be a spline curve of degree $p \ge 1$ and associated with a knot vector $\mathcal{U}$ given by (2). Assume that $\mathbf{c}$ fulfils Condition (C-2). Then we have the following conclusions.*

(i) $\mathcal{J}(u,v)$ is continuous over $[0,1]^2$ if and only if $\mathbf{c}$ is differentiable over $[0,1]$.

(ii) $\mathcal{J}(u,v)$ is Riemann integrable over $[0,1]^2$ (even if $\mathbf{c}$ is not differentiable).

If $\mathbf{c}$ violates Condition (C-2)/(i) or (C-2)/(ii), and all zeros of $\mathbf{c}'$ are isolated, then

(iii) $$\iint_{[0,1]^2} \mathcal{J}(u,v)\,\mathrm{d}u\mathrm{d}v = +\infty.$$

On the other hand, if $\mathbf{c}'$ has non-isolated zeros, the integral is undefined.

The proof is provided in Appendix B.

4.3. Optimization formulation

The uses of the penalty functions can be summarized in the following optimization formulation.

$$\begin{aligned}
\underset{\mathbf{c}_1,\ldots,\mathbf{c}_m}{\text{minimize}} \quad & \frac{\omega_e}{\omega_{e0}} \int_0^1 \|\mathbf{r}''(u)\|^2\,\mathrm{d}u \\
& + \iint_{[0,1]^2} \Big(\omega_b\,\widehat{\mathcal{I}}_{\mathbf{b}}(u,v) + \omega_b\,\widehat{\mathcal{I}}_{\widetilde{\mathbf{b}}}(u,v) + \omega_c\,\mathcal{J}(u,v)\Big)\,\mathrm{d}u\,\mathrm{d}v && \text{(8a)} \\
\text{such that} \quad & \mathbf{c}(0) = \mathbf{A},\ \mathbf{c}(1) = \mathbf{B} && \text{(8b)} \\
& \det[\mathbf{c}'(0), \boldsymbol{\alpha}] = 0,\ \det[\mathbf{c}'(1), \boldsymbol{\beta}] = 0 && \text{(8c)} \\
& \{\mathbf{c}(u),\ 0 < u < 1\} \subset \mathrm{int}(\Omega). && \text{(8d)}
\end{aligned}$$

The first term in (8a) is related to one of the standard regularity conditions for a curve. The curve $\mathbf{r}(u)$ is chosen as follows: if $p > 1$, $\mathbf{r}(u) \equiv \mathbf{c}(u)$; if $p = 1$, $\mathbf{r}(u)$ is a quadratic spline curve with a uniform and open knot vector where its control points are inherited from those of $\mathbf{c}(u)$. The indirect imposition of regularization on $\mathbf{c}$ when $p = 1$ is related to the two-stage optimization introduced in Section 2.3. This is to make Stage 1 capable of providing a better initial solution to Stage 2.

In Equations (8), ω_e is the weighting parameter associated with the regularity term. We use $\omega_{e0} > 0$ as a reference value for ω_e. In this work, we define ω_{e0} as $\max\left(\varepsilon, \int_0^1 \|\mathbf{r}''_0(u)\|^2\right)$ where $\varepsilon = 10^{-6}$, and $\mathbf{r}_0$ is the initial value of $\mathbf{r}$ in the optimization. The coefficient ω_b is the weighting for the penalty function of the curve to the boundary, and ω_c is the weighting for the penalty function of the curve to itself. The linear constraints (8b) and (8c) are to make sure the resulting curve satisfies Condition (C1).

We note that if the domain boundary consists of more than two spline curves, the extension is straightforward

(as long as the domain is still simply connected), as several spline curves can be converted into a single spline curve. However, in order to lessen the effect of the discrepancy between the speed of different parametrized boundary curves, we use a weighted average where we weight the integral with respect to each boundary curve according to its length. More sophisticated approaches can be employed, such as reparameterizing the combined single curve into a unit-speed curve, however we observe that the averaging according to curve lengths can produce sufficiently satisfactory results in practice.

Numerical approaches for solving the optimization problem 8 shall be addressed in Section 5.1. Our approach to computing the penalty functions depends on methods presented in Appendix C.

4.4. Two-stage optimization strategy

We apply standard nonlinear optimization methods to compute a B-spline curve of degree 2 or more solving Eq. (8). It is necessary to find a feasible solution to use as an initial point. Our approach is to choose as an initial control polygon a solution of Eq (8) among a space of linear B-splines.

Our approach to solving Eq (8) can be summarized as follows:

- Use a pathfinding algorithm to choose a knot vector and initial curve for the first optimization stage (Section 3).
- First optimization stage: solve Eq (8) in the space of linear splines with the given knot vector.
- Second optimization stage: solve Eq (8) in the space of splines with desired degree and the given knot vector. To choose the initial spline, we treat the solution of the previous optimization stage as a control polygon. To ensure that the initial spline satisfies the conditions, we detect polynomial pieces of the spline where the conditions fail, and refine those pieces by adding control points along the corresponding segments of the control polygon.

Using the result of the first optimization stage as the control polygon has two advantages (compared to, for example, fitting a higher order spline to the polygon): firstly, it saves on calculation while resulting in a curve that fulfils the tangential boundary conditions. Secondly, it makes it easy to include in the penalty function for the first stage a regularity term corresponding to a higher order spline with the same control net.

The requirement that the initial curve for the first stage is a splitting curve ensures the stability of the optimization. Since this initial curve has uniformly spaced knots, the result for $p = 1$ should be a control polygon with edges of approximately equal length. This enables the spline approximation with $p = 2$ in the second stage to have speed that is not far from uniform.

5. Examples

5.1. Implementation remarks

In order to solve the optimization problem (8), we have used the following numerical optimization approaches.

- *Gradient descent.* We use a backtracking inexact line-search (see [13, Chapter 3]) where, in addition to the Arjimo condition, we check if a point is visible. This approach works stably but is rather slow.
- *Broyden-Fletcher-Goldfarb-Shanno (BFGS)-based quasi-Newton method.* (See [13, Chapter 8].) This method often has faster convergence. Again, apart from checking if a point satisfies the Wolfe condition, we have to check the feasibility of the point.

We note that both approaches convert the constrained optimization problem (8) into an unconstrained one by taking the constraint (8d) as a feasibility check for the associated line-search. This is done in a similar fashion to, e.g., the methods using barrier functions to handle generic constrained optimizations [13, Chapter 17]. We observe that it suffices to perform this feasibility check in only a few iterations when the optimization stepsize is relatively large. In our examples, we neglect the check if the stepsize is less than 10^{-2}. Further analyses of these numerical approaches are beyond the scope the present paper.

For the examples below, we use Gauss quadrature to compute the penalty functions. In each dimension we use 10 Gaussian points distributed along each knot interval. In this way we obtain high accuracy so that the quadrature error does not influence the resulting curves. We have not studied whether it suffices to use less Gaussian points. The values of the B-spline basis functions at the Gaussian points are precomputed to reduce the computation time.

5.2. Reducing shape dependency for optimization weighting parameters

In order to make the the optimization problem (8) invariant to scaling, the following coefficients have to be defined: *(i)* the coefficient for the curve-to-boundary penalty function, *(ii)* the coefficient for the curve-to-itself penalty function, and *(iii)* the reference energy for the regularity term. Thanks to the way the penalty functions are defined, varying the coefficients has a predictable effect on every shape. This is demonstrated in Figures 7 and 8 where we show that varying the coefficient for the curve-to-boundary penalty affects the distance of the curve to the boundary in a similar way in 4 different examples. We also find out that the weighting parameters $(30, .1, 1)$ can produce good results.

In Figure 9, we show other examples of the maze-shaped domains. This, in turn, shows the robustness of our approach.

In order to demonstrate the current work's connection to the segmentation problem, we show in Fig. 10 a decomposition of a mechanical part into topological hexahedra.

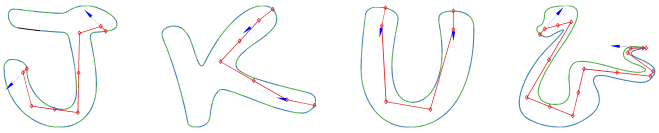

Figure 7: Example domains and initial curves used for the examples in Figure 8.

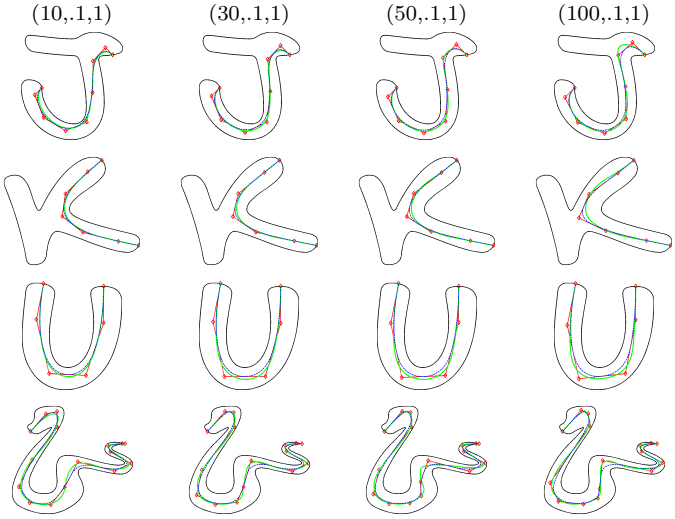

Figure 8: Effects of the weighting parameters $(\omega_b, \omega_c, \omega_e)$.

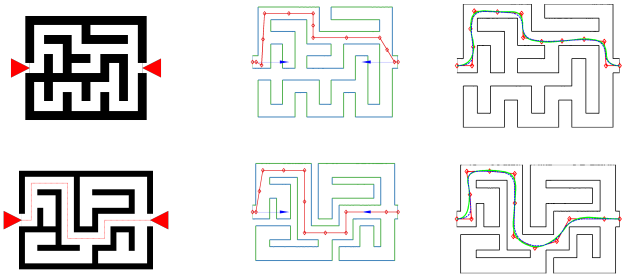

Figure 9: Applying our method to complex maze-like domains still produces high quality curves.

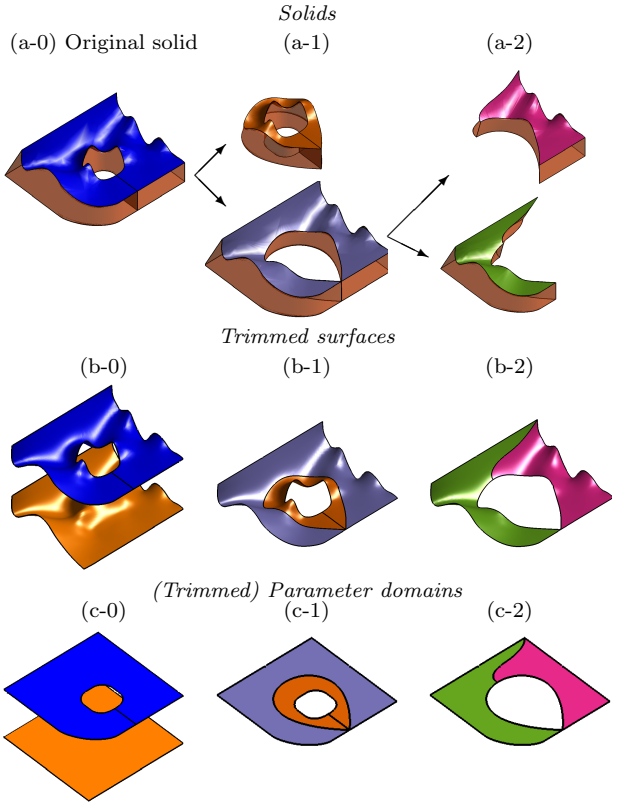

Figure 10: An isogeometric segmentation of a mechanical part requiring auxiliary curve constructions. The solid in (a-0) is first cut into the two pieces shown in (a-1). The top piece is a topological hexahedron. The bottom piece is then segmented into the two topological hexahedra shown in (a-2). Cutting the solid into just three topological hexahedra requires highly curved cuts. The marked face in (a-0), reproduced in blue in (b-0), is a trimmed surface with base NURBS surface shown in orange. In (c-0) we show the trimmed domain (blue) and the domain of the base NURBS surface (orange). The marked trimmed surface in (a-0) is split into 2 as shown by (a-1), (b-1) and (c-1). The marked face of the lower solid in (a-1) is also split into 2 as shown in (a-2), (b-2) and (c-2).

The mechanical part and the cutting loops used for the segmentations are intentionally made complicated so that the example can best show the connection and the robustness of our approach.

6. Conclusion and outlook

The results of this paper form an important step in the segmentation of a boundary represented solid into topological hexahedra. The segmentation procedure described in [5, 12] requires the construction of auxiliary edges which are used as part of a cutting loop to segment a solid into two pieces. Geometric conditions on the valid cutting loops can dictate boundary conditions for the new auxiliary edges. This paper provides a construction of edges which can split a face into two pieces while satisfying these boundary conditions. The method has been demonstrated to work even on quite complicated domains. In Figure 2 we show how our method can help to avoid badly shaped topological hexahedra in a segmentation by comparing the result of our method against an older approach we used.

Our method focuses on splitting the (2D) domain of a (3D) trimmed surface. Since a trimmed surface is defined by its domain and an embedding, a splitting of the 2D domain implies a splitting of the 3D surface. As a result, no more constraints are required to ensure that the generated curve lies within the surface. However, for highly distorted surfaces, the distortion could be taken into account in the penalty function to improve the resulting curves. We have not considered this yet.

Once a cutting loop has been chosen and its auxiliary edges have been constructed, the next step is the construction of a *cutting surface* having the given loop as its boundary. This problem can be seen as a higher dimen-

sional version of the problem studied in this paper, and, besides being more computationally expensive, comes with additional difficulties including the choice of an appropriate trimmed domain for the surface. The cutting surface problem is the topic of ongoing work.

Acknowledgment: This research was supported by the EC through the FP7 projects INSIST (GA 289361) and EXAMPLE (GA no. 324240), and by the FWF, NFN S117.

Appendix A. Regularization of the curve-to-boundary penalty function

Lemma 1 aids the proof of Theorems 1 and 2.

Lemma 1. *Let* $\mathbf{c}(u), \mathbf{b}(v) : [0, \epsilon] \to \mathbb{R}^n$ *be spline curves of degree 1 or higher. Assume* $\mathbf{c}(0) = \mathbf{b}(0)$ *and* $\mathbf{c}'_+(0) \neq 0 \neq \mathbf{b}'_+(0)$. *Then:*

(i) if $\mathbf{c}'_+(0) \neq \beta \mathbf{b}'_+(0)$ *for any* $\beta > 0$ *then*

$$\frac{u+v}{\|\mathbf{c}'_+(0)u - \mathbf{b}'_+(0)v\|} \tag{A.1}$$

is bounded for $u, v \geq 0$ *with* $(u, v) \neq (0, 0)$, *and*

$$\lim_{(u,v)\to(0^+,0^+)} \frac{\|\mathbf{c}(u) - \mathbf{b}(v)\|}{\|\mathbf{c}'_+(0)u - \mathbf{b}'_+(0)v\|} = 1; \tag{A.2}$$

(ii)

$$\iint_{[0,\epsilon]^2} \frac{1}{\|\mathbf{c}(u) - \mathbf{b}(v)\|^2} \mathrm{d}u\,\mathrm{d}v \tag{A.3}$$

is divergent in the sense of improper integrals.

Proof. Since $\mathbf{c}(u), \mathbf{b}(v)$ are splines of degree at least 1,

$$\begin{aligned} \mathbf{c}(u) &= \mathbf{c}(0) + \mathbf{c}'_+(0)u + o(u) \text{ as } u \to 0^+ \\ \mathbf{b}(v) &= \mathbf{b}(0) + \mathbf{b}'_+(0)v + o(v) \text{ as } v \to 0^+. \end{aligned} \tag{A.4}$$

(i) For $u, v \geq 0$ and $(u, v) \neq (0, 0)$, we have $\|\mathbf{c}'_+(0)u - \mathbf{b}'_+(0)v\| \geq G(u+v)$, where $G = \min_{u\geq 0, v\geq 0, u+v=1} \|\mathbf{c}'_+(0)u - \mathbf{b}'_+(0)v\|$, which exists by compactness of the set $\{(u, v), u \geq 0, v \geq 0, u + v = 1\}$ and is positive by the assumption of part (i) of the lemma. Therefore (A.1) is bounded. So by Equation (A.4) and the triangle inequality, as $(u, v) \to (0^+, 0^+)$,

$$\frac{\|\mathbf{c}(u) - \mathbf{b}(v)\|}{\|\mathbf{c}'_+(0)u - \mathbf{b}'_+(0)v\|} = 1 + \frac{o(u) + o(v)}{G(u+v)} = 1 + o(1)$$

and Equation (A.2) follows.

(ii) By Equation (A.4) there are $M > 0$ and ϵ_1 with $0 < \epsilon_1 < \epsilon$ such that $\|\mathbf{c}(u) - \mathbf{b}(v)\|^2 < M(u^2 + v^2)$ for all $0 < u, v < \epsilon_1$. Thus $\iint_{[0,\epsilon]^2} \frac{1}{\|\mathbf{c}(u)-\mathbf{b}(v)\|^2} \mathrm{d}u\mathrm{d}v$ diverges. □

By transforming the domain, Lemma 1 can be applied to two splines $\mathbf{c}, \mathbf{b}$ such that $\mathbf{c}(\bar{u}) = \mathbf{b}(\bar{v})$ for arbitrary $\bar{u}, \bar{v}$ in their domains and even to any of the four quadrants around $(\bar{u}, \bar{v})$, to see the behaviour of the function in any of the four limits $(\bar{u}^\pm, \bar{v}^\pm)$. In two cases, the sign of β in the condition of Lemma 1 (i) must be reversed.

As a consequence of Lemma 1, the following two functions can be used to eliminate the two singular points $(0, 0)$ and $(1, 1)$ for the curve-to-boundary function:

$$\begin{aligned} f_0(u, v) &= \|\mathbf{c}'_+(0)u - \mathbf{b}'_+(0)v\|^2 \\ f_1(u, v) &= \|\mathbf{c}'_-(1)(u-1) - \mathbf{b}'_-(1)(v-1)\|^2. \end{aligned} \tag{A.5}$$

In order to obtain a multiplier function $r_{\mathbf{b}}(u, v)$ as mentioned in Section 4.1, we will define weighting functions w_0, w_τ, and w_1 associated with the three functions: f_0, the indicator function $1_{[0,1]^2}$, and f_1 so that

(W1) $w_0 f_0$ and $w_1 f_1$ are supported over a neighborhood of the corner points $(0, 0)$ and $(1, 1)$ respectively;

(W2) $w_{0,1}$ isolates the two corner points $(0, 0)$ and $(1, 1)$ from the support of $1_{[0,1]^2}$;

(W3) the three weighting functions form a set of coefficients of a convex interpolation, i.e., they are non-negative and form a partition of unity.

The grid lines corresponding to the knots u_i of the knot vector $\mathcal{U}_p$ and the the knots v_j of the knot vector $\mathcal{V}_q$ partition the unit square into rectangular boxes $[u_i, u_{i+1}] \times [v_j, v_{j+1}]$, $i = 1, \ldots, m-1$, $j = 1, \ldots, n-1$. Let $\mathcal{D}_0 = [u_1, u_2] \times [v_1, v_2]$ and $\mathcal{D}_1 = [u_{m-1}, u_m] \times [v_{n-1}, v_n]$ be the boxes that contain the two corner points $(0, 0)$ and $(1, 1)$ respectively as illustrated in Figure A.11(a).

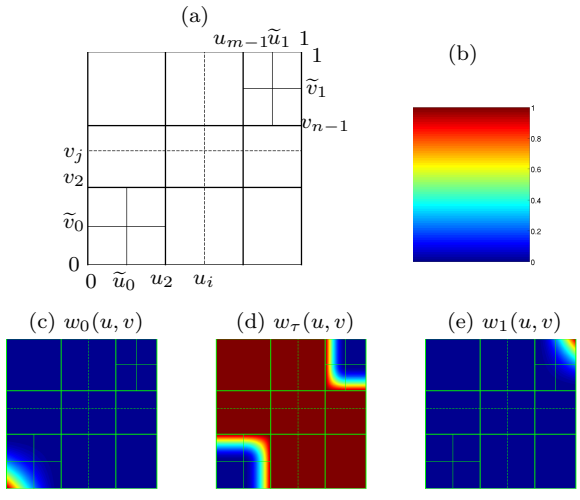

Figure A.11: Construction of weighting functions for regularizing the curve-to-boundary penalty function. (a): Partition of the unit square induced by grid lines defined by horizontal knots $0 = u_1, \ldots, u_m = 1$ and vertical knots $0 = v_1, \ldots, v_n = 1$; domains $\mathcal{D}_0$ (in red) and $\mathcal{D}_1$ (in blue) are subdivided into 4 boxes. (b): Color scale used to depict the weighting functions. (c), (d), (e): normalizations of these functions (by its summation) give the weighting functions.

First, we define the following auxiliary functions:

$$\widehat{w}_0(u,v) = 1_{\mathcal{D}_0}\Big(\frac{u_2-u}{u_2-u_1}\Big)^{p_0}\Big(\frac{v_2-v}{v_2-v_1}\Big)^{q_0} \tag{A.6}$$

$$\widehat{w}_1(u,v) = 1_{\mathcal{D}_1}\Big(\frac{u-u_{m-1}}{u_m-u_{m-1}}\Big)^{p_1}\Big(\frac{v-v_{n-1}}{v_n-v_{n-1}}\Big)^{q_1} \tag{A.7}$$

where p_0, q_0, p_1, and q_1 are positive integers. Note that $w_{0,0}$ is the bivariate Bernstein polynomial of degrees (p_0, q_0) over $\mathcal{D}_0$ which does not vanish at the lower left corner of $\mathcal{D}_0$. Similarly, w_1 is the bivariate Bernstein polynomial of degrees (p_1, q_1) over $\mathcal{D}_1$ which does not vanish at the upper right corner of $\mathcal{D}_1$. See Figure A.11(c)(e).

In order to construct the weighting function w_τ, subdivide $\mathcal{D}_0$ into four new boxes by inserting line segments $\{(\widetilde{u}_0, v)\,|\,v_1 \le v \le v_2\}$ and $\{(u, \widetilde{v}_0)\,|\,u_1 \le u \le u_2\}$; and subdivide $\mathcal{D}_1$ by inserting line segments $\{(\widetilde{u}_1, v)\,|\,v_{n-1} \le v \le v_m\}$ and $\{(u, \widetilde{v}_1)\,|\,u_{m-1} \le u \le u_m\}$; see Figure A.11(a). Letting $h_p(x,a,b) = 1 - \max(\frac{x-a}{b-a}, 0)^p$, define another auxiliary function $\widehat{w}_\tau$:

$$\begin{aligned}\widehat{w}_\tau(u,v) &= 1_{[0,1]^2} - 1_{\mathcal{D}_0} h_{p_\tau}(u, \widetilde{u}_0, u_2) h_{q_\tau}(v, \widetilde{v}_0, v_2)\\ &\quad - 1_{\mathcal{D}_1} h_{p_\tau}(u, \widetilde{u}_1, u_{m-1}) h_{q_\tau}(v, \widetilde{v}_1, v_{n-1}).\end{aligned} \tag{A.8}$$

Figure A.11(d) gives an illustration for the function. The weighting functions can now be defined:

$$w_\bullet(u,v) = \frac{\widehat{w}_\bullet}{\widehat{w}_0(u,v) + \widehat{w}_\tau(u,v) + \widehat{w}_1(u,v)}, \tag{A.9}$$

where $\bullet$ stands for one of the symbols 0, 1, and τ. It is straightforward that the weighting functions defined in (A.9) satisfy the conditions (W1-3).

We can now define a multiplier function as follows.

$$r_{\mathbf{b}} = w_0\,\alpha_0\,f_0 + w_\tau\,\alpha_\tau\,1_{[0,1]^2} + w_1\,\alpha_1\,f_1, \tag{A.10}$$

where α_0, α_τ, and α_1 are some positive constants. We use α_0 and α_1 to control the scaling of the two functions f_0 and f_1; we used $\alpha_0 = \alpha_1 = 10^{-3}$. Meanwhile, α_τ helps to make the optimization problem invariant under similarities. For this purpose, we define α_τ as the inverse of the integral of the corresponding curve-to-boundary function associated with the initial curve for the optimization problem (8). We are now ready to prove Theorem 1.

Proof of Theorem 1. First, we note that as the weighting functions and f_0, f_1 are piecewise rational functions continuous over $[0,1]^2$, $r_{\mathbf{b}}(u,v)$ must be a piecewise rational function continuous over $[0,1]^2$.

(i) By Assumptions (A-1), (A-2), Hypotheses 1-3 of this theorem, Lemma 1(i) and Equation (5),

$$\frac{1}{\alpha_0}\lim_{(u,v)\to(0^+,0^+)}\widehat{\mathcal{I}}_{\mathbf{b}}(u,v) = \frac{1}{\alpha_1}\lim_{(u,v)\to(1^-,1^-)}\widehat{\mathcal{I}}_{\mathbf{b}}(u,v) = 1.$$

As we define $\widehat{\mathcal{I}}_{\mathbf{b}}(0,0) = \alpha_0$ and $\widehat{\mathcal{I}}_{\mathbf{b}}(1,1) = \alpha_1$, $\widehat{\mathcal{I}}_{\mathbf{b}}$ is bounded and continuous, and therefore Riemann integrable over $[0,1]^2$.

(ii) Equations (A.6), (A.7), (A.8) and (A.9) directly imply that $w_\tau \equiv 1$ and $w_0 \equiv w_1 \equiv 0$ over $[0,1]^2 \setminus (\mathcal{D}_0 \cup \mathcal{D}_1)$. Therefore, $\widehat{\mathcal{I}}_{\mathbf{b}} \equiv \alpha_\tau \mathcal{I}_{\mathbf{b}}$ over $[0,1]^2 \setminus (\mathcal{D}_0 \cup \mathcal{D}_1)$.

(iii) Lemma 1(ii) directly implies this conclusion for $(u^*, v^*) \in [0,1]^2 \setminus (\mathcal{D}_0 \cup \mathcal{D}_1)$. Consider the case where $(u^*, v^*) \in \mathcal{D}_0$, the case where $(u^*, v^*) \in \mathcal{D}_1$ is similar. Hypothesis 3 then implies that either $p > 1$ or $q > 1$. On the other hand, Assumptions (A-1), (A-2), and Hypotheses 1–3 imply that f_0 only vanishes at $(0,0)$. Thus, $f_0(u^*, v^*) \neq 0$. Consequently, in the limiting process when $(u,v) \to (u^*, v^*)$, $\widehat{\mathcal{I}}_{\mathbf{b}}(u,v)$ is equivalent to $\mathcal{I}_{\mathbf{b}}(u,v)$[1]. Therefore, Lemma 1(ii) again leads to the conclusion. □

Appendix B. Regularity properties of the curve-to-itself penalty function

Proof of Theorem 2. Partition the unit square into the sets $\mathcal{B}_{i,j} = [u_i, u_{i+1}] \times [u_j, u_{j+1}]$, $1 \le i \le m-1$, $1 \le j \le n-1$. From Equation (7) and Lemma 1(i),

$$\lim_{(u,v)\to(u_i^-,u_i^+)}\frac{\mathcal{J}(u,v)}{g(u,v)} = 1 = \lim_{(u,v)\to(u_i^+,u_i^-)}\frac{\mathcal{J}(u,v)}{g(u,v)}, \tag{B.1}$$

$$\text{where } g(u,v) := \frac{(u-v)^2}{\|\mathbf{c}'_-(u_i)(u-u_i) - \mathbf{c}'_+(u_i)(v-u_i)\|^2}.$$

(i) By Condition (C-2), $\mathcal{J}$ is continuous at any (u,v) where $u \neq v$. For $(u,v) \in \mathcal{B}_{i,i}$ which includes the case $u = v$,

$$\mathbf{c}(u) - \mathbf{c}(v) = (u-v)\big(\mathbf{c}'(v) + (u-v)\mathbf{P}_i(u,v)\big), \tag{B.2}$$

for a vector-valued polynomial $\mathbf{P}_i(u,v)$. Condition (C-2) implies $\|\mathbf{c}'(u^*)\| > 0$ for all $u^* \in [u_i, u_{i+1}]$, $1 < i < m$, and with Equations (7), (B.2) this implies that $\mathcal{J}$ is continuous at every (u,u) where $u \neq u_i$, $1 < i < m$. Therefore we only need to consider discontinuities of $\mathcal{J}$ at (u_i, u_i), $1 < i < m$. We divide the proof of (i) into two parts.

Sufficient condition. Assume $\mathbf{c}$ is differentiable at each (u_i, u_i). Then $c'_-(u_i) = c'_+(u_i)$, and Equations (B.1), (B.2) can be used to show that the four limits of $\mathcal{J}$ at $(u_i^\pm, u_i^\pm)$ are equal. Thus $\mathcal{J}$ is continuous at each (u_i, u_i) and, we conclude, over the whole unit square.

Necessary condition. If $\mathbf{c}$ is not differentiable at some u_i, $1 < i < m$, so that $\mathbf{c}'_+(u_i) \neq \mathbf{c}'_-(u_i)$, then for $\alpha > 0$, $\beta > 0$ such that $\alpha + \beta = 1$, we see from (B.1) that $\lim_{n\to\infty}\mathcal{J}(u_i - \frac{\alpha}{n}, u_i + \frac{\beta}{n})$ is not independent of α and β. Therefore $\mathcal{J}$ is not continuous at (u_i, u_i).

(ii) By Lemma 1 (i) and condition C-2/(ii), $g(u,v)$ is bounded on the sets $\{(u,v) : u < u_i, v > u_i\}$ and $\{(u,v) : u > u_i, v < u_i\}$. By (B.1), $\mathcal{J}(u,v)$ is bounded on the interiors of $\mathcal{B}_{i-1,i}$ and $\mathcal{B}_{i,i-1}$. By Equation (B.2), $\mathcal{J}(u,v)$ is bounded on the interiors of each $\mathcal{B}_{i,i}$. Thus $\mathcal{J}$ is

[1] this is not true, if Hypothesis (iii) is disregarded and $p = q = 1$.

almost everywhere continuous and bounded, and therefore Riemann integrable.

(iii) We treat the violations of the conditions separately:

(C-1)/(i): Suppose u, v are such that $u \neq v$ and $\mathbf{c}(u) = \mathbf{c}(v)$. Apply Lemma 1/(ii) to the two curves formed by restricting $\mathbf{c}$ to neighbourhoods of u and v.

(C-1)/(ii): Since $\mathbf{c}$ is a spline, either all zeros of $\mathbf{c}'$ are isolated or $\mathbf{c}'$ is zero on a knot interval. If $\mathbf{c}'$ is zero on an interval (u_i, u_{i+1}) then $\mathbf{c}$ is constant on the interval and $\mathcal{J}$ is undefined on $\mathcal{B}_{i,i}$. Thus the integral is undefined.

If there is isolated u^* such that $\mathbf{c}'_+(u^*) = 0$, then, applying Equations (7), (B.2), $\mathcal{J}(u,v) = \|O(u-u^*) + O(v-u^*)\|^{-2}$ as $(u,v) \to (u^{*+}, u^{*+})$. Therefore there exist $\epsilon, C > 0$ such that $\mathcal{J}(u,v) > (C((u-u^*)+(v-u^*)))^{-2}$ for all $u, v \in (u^*, u^* + \epsilon)$, so the integral diverges. □

Appendix C. Computation of the regularized penalty functions

Before discussing our approach for computing the regularized penalty functions, we will introduce several kinds of bivariate spline functions which help to make the computation more efficient.

Representation of the difference between two points of one spline curve In order to compute the denominator of the curve-to-itself penalty function $\mathcal{J}(u,v)$ given by (7) in a diagonal box $\mathcal{B}_{i,i} = [u_i, u_{i+1}] \times [u_i, u_{i+1}]$, we let $\mathbf{Q} : \mathcal{B}_{i,i} \to \mathbb{R}^2$ be the vector-valued polynomial satisfying

$$\mathbf{c}(u) - \mathbf{c}(v) = (u-v)\mathbf{Q}(u,v). \tag{C.1}$$

It is obvious that $\mathbf{Q}(u,v)$ is a bivariate spline of degree $p-1$ in both u and v. Pekerman et al. [15] derived explicitly the control points of its spline representation for the case where $\mathbf{c}(u)$ is a Bézier curve. For this case, $\mathbf{Q}(u,v)$ is a Bézier surface described by $(m-1)^2$ control points and $(m-1)^2$ basis functions. Extending to the case of a general spline curve requires either implicit representation of the control points of the spline surface $\mathbf{Q}(u,v)$ or breaking the curve into Bézier segments. In the remainder of the section, we will represent the spline surface $\mathbf{Q}(u,v)$ in terms of just m control points and m functions, and the control points are the same as those of the curve.

Let $D = \bigcup_j \big([u_j, u_{j+1}) \times [u_j, u_{j+1})\big)$, the union of the diagonal semi-open boxes associated with the knot vector $\mathcal{U}$ given by (2). Consider B-splines $N_{i,p,\mathcal{U}}$ of degree p associated with $\mathcal{U}$. We assume that $p \geq 1$. Define

$$Q_{i,p}(u,v) = \frac{N_{i,p,\mathcal{U}}(u) - N_{i,p,\mathcal{U}}(v)}{u-v}. \tag{C.2}$$

$Q_{i,p}$ is well defined because of the fundamental polynomial remainder theorem. Further, as $\sum_{i=1}^m Q_{i,p} \equiv 0$, the functions $Q_{i,p}$ are not linearly independent[2]. The following proposition shows that these functions form a spanning set of the space of functions $\mathbf{Q}(u,v)$ given by (C.1).

Proposition 1. *For $p \geq 1$, the functions $Q_{i,p}$ defined by* (C.2) *have the following properties.*

(i) For any spline curve $\mathbf{c}(u)$ of the form (1)*, the following identity holds for all $(u,v) \in D$:*

$$\mathbf{c}(u) - \mathbf{c}(v) = (u-v) \sum_{i=1}^m \mathbf{c}_i \, Q_{i,p}(u,v). \tag{C.3}$$

(ii) They can be computed recursively as below:

$$\begin{aligned} Q_{i,p}(u,v) = {} & (a_i v + b_i)\, Q_{i,p-1}(u,v) \\ & + (c_i v + d_i)\, Q_{i+1,p-1}(u,v) \\ & + a_i \, N_{i,p-1,\mathcal{U}}(u) + c_i \, N_{i+1,p-1,\mathcal{U}}(u), \end{aligned} \tag{C.4}$$

for all $(u,v) \in D$, where

$$(a_i, b_i) = \frac{(1, -u_i)}{u_{i+p} - u_i}, \; (c_i, d_i) = \frac{(-1, u_{i+p+1})}{u_{i+p+1} - u_{i+1}}. \tag{C.5}$$

In (C.4)*, we assume that $Q_{i,0} \equiv 0$ and with the standard convention that a fraction equals 0 if its denominator equals 0.*

(iii) $Q_{i,p}(u,u)$ is the derivative of $N_{i,p,\mathcal{U}}(u)$.

Proof. (i) can be derived directly from (C.1) and (C.3). *(ii)* Equation (C.4) can be derived from Equation (C.2) and the recursive evaluation of B-splines $N_{i,p,\mathcal{U}}$,

$$N_{i,p,\mathcal{U}}(t) = (a_i t + b_i) N_{i,p-1,\mathcal{U}}(t) + (c_i t + d_i) N_{i+1,p-1,\mathcal{U}}(t).$$

(iii) The following Taylor expansion of $N_{i,p,\mathcal{U}}$ proves *(iii)*:

$$N_{i,p,\mathcal{U}}(u) = N_{i,p,\mathcal{U}}(v) + N'_{i,p,\mathcal{U}}(v)(u-v) + o(u-v)^2. \quad \square$$

Representation of Taylor-like expansion coefficients of a spline curve A one-sided Taylor expansion of $\mathbf{c}$ around $\bar{u}$, $u_i \leq \bar{u} < u_{i+1}$, can be written as below for all $u \in [u_i, u_{i+1}]$

$$\mathbf{c}(u) = \mathbf{c}(\bar{u}) + \mathbf{c}'_+(\bar{u})(u-\bar{u}) + \widehat{\mathbf{c}}_+(u,\bar{u})(u-\bar{u})^2, \tag{C.6}$$

$$\text{where } \widehat{\mathbf{c}}_+(u,\bar{u}) = \sum_{k=2}^{p} \frac{1}{k!} \mathbf{c}_+^{(k)}(\bar{u})(u-\bar{u})^{k-2}, \tag{C.7}$$

and $\mathbf{c}_+^{(k)}$ denotes a right-sided derivative of order k of $\mathbf{c}$. We shall see in (C.13) that the representation (C.6) allows for stable computations for the penalty functions.

Equation (C.7) provides a direct approach for computing the term. However, this approach is not preferable. Similar to the evaluation of the quantity $\mathbf{c}(u) - \mathbf{c}(v)$ in Proposition 1, we introduce the following spline functions that also allow for a computation of the term as a linear combination of only m functions.

We assume $p \geq 2$. Define the following spline functions:

$$H_{i,p} = \frac{Q_{i,p}(u,v) - Q_{i,p}(v,v)}{u-v}. \tag{C.8}$$

[2] In fact, if all inner knots have multiplicity less than $p+1$, the dimension of the linear space spanned by $Q_{i,p}$, $1 \leq i \leq m$, are $m-1$. We leave the proof as an exercise for readers.

Similar to the definition of $Q_{i,p}$ given by (C.2), this definition is also well-defined. Also, the set of $H_{i,p}$ is not linearly independent as the sum of all these functions vanishes. However, different from $Q_{i,p}$, $H_{i,p}$ is not symmetric in its variables. That is, $H_{i,p}(u,v)$ is in general different from $H_{i,p}(v,u)$. The following proposition provides a recursive formula for the evaluation of these spline functions.

Proposition 2. *For $p \geq 2$, the functions $H_{i,p}$ defined by* (C.8) *have the following properties.*

(i) For any spline curve $\mathbf{c}(u)$ given by (1)*, we have the following identity:*

$$\begin{aligned}\mathbf{c}(u) = \ & \mathbf{c}(\bar{u}) + \mathbf{c}'_+(\bar{u})(u-\bar{u}) \\ & + \Big(\sum_{i=1}^{m} \mathbf{c}_i\, H_{i,p}(u,\bar{u})\Big)(u-\bar{u})^2,\ \forall u \in [u_i, u_{i+1}) \quad \text{(C.9)}\end{aligned}$$

(ii) $H_{i,p}$ can be computed recursively as follows

$$\begin{aligned}H_{i,p}(u,v) = \ & (a_i v + b_i)\, H_{i,p-1}(u,v) \\ & + (c_i v + d_i)\, H_{i+1,p-1}(u,v) \\ & + a_i\, Q_{i,p-1}(u) + c_i\, Q_{i+1,p-1}(u),\end{aligned} \quad \text{(C.10)}$$

for all $(u,v) \in D$, where a_i, b_i, c_i, d_i are given by (C.5) *and where we assume that $H_{i,1} \equiv 0$.*

Proof. The proof is similar to that of Proposition 1. □

Evaluation of the regularized penalty functions With the introduction of the spline functions $Q_{i,p}$, and $H_{i,p}$, together with Equations (C.3), and (C.9), the computation of the penalty functions $\widehat{\mathcal{I}}$ and $\mathcal{J}$ are ready, except that we have to compute the two functions in the neighborhoods of a point at which one of the two functions has the form $0/0$. At such a point, we will convert each function into a form in polar coordinate that is no longer a $0/0$ form. As both of the penalty functions can be derived from the following asymptotically $0/0$ function

$$\mathcal{R}(u,v) = \frac{\|\mathbf{c}(u) - \mathbf{b}(v)\|}{\|\mathbf{c}'_+(0)u - \mathbf{b}'_+(0)v\|} = \frac{\|\mathbf{c}(u) - \mathbf{b}(v)\|}{f_0} \quad \text{(C.11)}$$

that appears in (A.2) in Lemma 1, it suffices to consider this function alone. Similar to (C.6), Taylor-like expansions for $\mathbf{c}(u)$ near $u = 0$ and $\mathbf{b}(v)$ near $v = 0$ read

$$\begin{aligned}\mathbf{c}(u) &= \mathbf{c}(0) + \mathbf{c}'_+(0)u + \widehat{\mathbf{c}}_+(0)u^2,\ \forall u \in [0,\epsilon) \\ \mathbf{b}(v) &= \mathbf{b}(0) + \mathbf{b}'_+(0)v + \widehat{\mathbf{b}}_+(0)v^2,\ \forall v \in [0,\epsilon)\end{aligned} \quad \text{(C.12)}$$

where $\widehat{\mathbf{c}}, \widehat{\mathbf{b}}$, defined analogously to (C.7), can be calculated using the approach stated in Proposition 2. We perform the change of variables $u = r\cos\phi$, $v = r\sin\phi$. As we consider $0 < u, v < \epsilon$, we assume that $0 < \phi < \pi/2$. In the form of new variables, dividing both numerator and denominator of $\mathcal{R}^2(r\cos\phi, r\sin\phi)$ by the common term r we have

$$\mathcal{R}^2(r\cos\phi, r\sin\phi) = 1 + 2r\,\frac{\langle \tilde{\mathbf{f}}_0, \widetilde{\mathbf{R}}\rangle}{\|\tilde{\mathbf{f}}_0\|^2} + r^2\,\frac{\langle \widetilde{\mathbf{R}}, \widetilde{\mathbf{R}}\rangle}{\|\tilde{\mathbf{f}}_0\|^2} \quad \text{(C.13)}$$

where $\tilde{\mathbf{f}}_0(r,\phi) = \mathbf{c}'_+(0)\cos\phi - \mathbf{b}'_+(0)\sin\phi$ and $\tilde{\mathbf{R}}(r,\phi) = \widehat{\mathbf{c}}_+(0)\cos^2\phi - \widehat{\mathbf{b}}_+(0)\sin^2\phi$. Because of Lemma 1(i), $\|\tilde{\mathbf{f}}_0(r,\phi)\|^2$ is bounded away from zero. Thus, the penalty functions can be evaluated using (C.13) and the spline functions $H_{i,p}$ analyzed in Proposition 2.

References

[1] P.K. Agarwal, P. Raghavan, and H. Tamaki. Motion planning for a steering-constrained robot through moderate obstacles. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 343–352, New York, NY, USA, 1995. ACM.

[2] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, 2009.

[3] G.A Elber, T.B Grandine, and M.-S. Kim. Surface self-intersection computation via algebraic decomposition. *CAD Computer Aided Design*, 41(12):1060–1066, 2009.

[4] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 2–7 vol.1, May 1989.

[5] B. Jüttler, M. Kapl, D.-M. Nguyen, Q. Pan, and M. Pauley. Isogeometric segmentation: The case of contractible solids without non-convex edges. *Computer-Aided Design*, 57(0):74 – 90, 2014.

[6] Y. Kanayama and B.I. Hartman. Smooth local path planning for autonomous vehicles. In *1989 IEEE International Conference on Robotics and Automation, Proceedings*, pages 1265–1270, May 1989.

[7] J.-P. Laumond, S. Sekhavat, and F. Lamiraux. Guidelines in nonholonomic motion planning for mobile robots. In *Robot Motion Planning and Control*, pages 1–53. Springer-Verlag, 1998.

[8] S.M. LaValle. *Planning algorithms*. Cambridge University Press, Cambridge, 2006.

[9] Z. Li, D. S. Meek, and D. J. Walton. A smooth, obstacle-avoiding curve. *Computers & Graphics*, 30(4):581–587, 2006.

[10] T. Maekawa, T. Noda, S. Tamura, T. Ozaki, and K.-I. Machida. Curvature continuous path generation for autonomous vehicle using B-spline curves. *Computer-Aided Design*, 42(4):350–359, 2010.

[11] A. Malhotra, J. H. Oliver, and W. Tu. Synthesis of spatially and intrinsically constrained curves using simulated annealing. *Journal of Mechanical Design*, 118(1):53–61, Mar 1996.

[12] D.-M. Nguyen, M. Pauley, and B. Jüttler. Isogeometric segmentation. Part II: On the segmentability of contractible solids with non-convex edges. *Graphical Models*, 76(5):426 – 439, 2014.

[13] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer, New York, NY, 2. ed. edition, 2006.

[14] M. Pauley, D.-M. Nguyen, D. Mayer, J. Špeh, O. Weeger, and B. Jüttler. The isogeometric segmentation pipeline. Submitted.

[15] D. Pekerman, G. Elber, and M.-S. Kim. Self-intersection detection and elimination in freeform curves and surfaces. *Computer-Aided Design*, 40(2):150 – 159, 2008.

[16] L. Piegl and W. Tiller. *The NURBS Book (2nd Ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[17] T. Samoilov and G. Elber. Self-intersection elimination in metamorphosis of two-dimensional curves. *Visual Computer*, 14(8-9):415–428, 1998.

[18] A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. In *IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 997–1003, Sep 1997.

[19] C.-C. Tsai, H.-C. Huang, and C.-K. Chan. Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. *IEEE Transactions on Industrial Electronics*, 58(10):4813–4821, Oct 2011.

[20] X. Wang and X. Qian. An optimization approach for constructing trivariate B-spline solids. *Comput.-Aided Des.*, 46:179–191, 2014.

[21] J. Witte and C. Reisinger. Penalty methods for the solution of discrete HJB equations – continuous control and obstacle problems. *SIAM Journal on Numerical Analysis*, 50(2):595–625, 2012.