C^1-Smooth Parametrization of Polynomial Shapes over Polygonal Domains

Payam Khanteimouri* Marcel Campen*

Abstract

The generation of meshes and mesh-based parametrizations for given geometric domains is a cornerstone of modeling and analysis. Of particular interest in this context are spaces of increased smoothness, in terms of the continuity order across elements. We describe a method to, by means of higher-order mesh generation, construct a parametrization of 2D shapes with curved boundaries, with a focus on C^1-continuity. Different from previous work, this method guarantees the absence of inversions or degeneracies as well as exact conformance to the given polynomial boundary. It leverages recent results on higher-order triangle mesh generation with guaranteed quality bounds. These bounds enable us to formulate an optimization problem aiming for continuity while maintaining validity.

1 Introduction

In the context of meshing of planar freeform shapes, the use of higher-order elements is of notable interest [40, 1, 24, 35, 10, 20, 21]. The input is a shape described by a piecewise polynomial or rational boundary curve. Mesh generation methods then focus on creating meshes with specific properties, including element regularity (i.e. absence of degeneracies or inversions), element shape quality, and exact or approximate boundary conformance (i.e. interpolation of the boundary).

An additional property of interest in this context is smoothness. Concretely, we are concerned here with C^1-continuity across the mesh's elements. Continuity beyond C^0 plays an important role in isogeometric analysis [12, 37, 11, 39, 3]. C^1-continuous higher-order meshes can yield numerical advantages over methods relying on C^0 inter-element continuity in finite element analysis. These benefits are evident in various aspects, notably through the enhancement of convergence rate per degree-of-freedom (DOF) [5] and improvements in numerical efficiency and accuracy, in areas including structural analysis [4, 6], fluid simulation [2], shape optimization [25, 26, 18], and deformation [17].

A mesh of (higher-order) planar triangles that join C^1-continuously effectively provides a global C^1-smooth parametrization for the underlying shape. In this sense the problem of boundary conforming C^1-mesh generation can also be viewed as the problem of computing C^1-smooth (piecewise polynomial) parametrizations of 2D freeform shapes over polygonal domains, see fig. 1.

1.1 Challenges Generating meshes that are regular, conforming, *and* C^1 proves difficult. Even the simpler problem of generating meshes that are just regular and conforming has found first solutions with success guarantee only recently [20, 21]. This difficulty is exacerbated by the fact that the problem can be infeasible, achieving C^1-continuity globally (in addition to regularity and conformance) can be impossible. At least at some boundary points (*exceptional points*) one may have to accept C^0-continuity, as illustrated in fig. 2. Furthermore, a trade-off in favor of lower parametric distortion may require additional exceptional points beyond those strictly necessary. The necessity of choosing these points, their number and location, adds to the complexity of the problem.

Our method aims to establish C^1-continuity as broadly as possible, leaving only a small (if not minimal) number of exceptional points. To this end an important feature is that the method considers the parametric domain's shape a variable, up for optimization, so as to enable establishing a smoothness-accommodating domain.

Figure 1 underlines the complexity of the problem: Effectively, besides finding a regular conforming higher-order triangle mesh for the given shape (center), a domain polygon (right) needs to be determined that admits a regular conforming linear triangulation that is structurally identical to the curved triangle mesh, and the per-triangle maps from linear triangle to curved triangle need to join C^1-continuously. Together this forms a problem with numerous coupled degrees of freedom and complex constraints.

2 Related Work

In this section, we review previous work on the problems of higher-order mesh generation and higher-order piecewise parametrization, focusing particularly on the properties of regularity and C^1-continuity.

2.1 Regular, not C^1-continuous Many methods are available to generate 2D higher-order meshes of

*Osnabrück University, Germany.

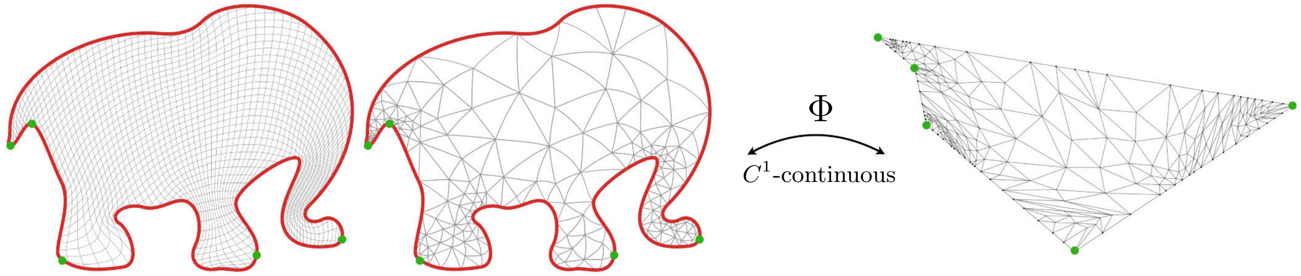

Figure 1: Given a freeform shape, formed by polynomial boundary curves (left), our method is able to compute a parametrization Φ (as indicated by a grid of iso-curves) over a suitably determined polygonal domain (right). The resulting parametrization is piecewise polynomial (based on a suitably generated higher-order mesh, center) and is guaranteed to be bijective and of bounded distortion. Furthermore, it is C^1-smooth almost everywhere; around some exceptional boundary points (highlighted in green) degradation to C^0-continuity is partially inevitable.

polynomial triangles with a particular focus on regularity. One class of methods, indirect methods, start by generating a linear mesh of straight-edge elements, which is then elevated to higher order and progressively deformed towards boundary conformance [10, 7, 29, 34, 8, 28]. By testing for regularity violations during deformation, regularity can be maintained. Achieving conformance, however, is not guaranteed.

In contrast to these, a few approaches offer direct higher-order triangulation with guarantees on regularity and conformance [20, 21], i.e. curved mesh edges exactly interpolate the shape boundary with zero approximation error. Further guarantees concerning bounded distortion are also available [21], and extensions to rational triangles have been described [38, 15].

None of these regularity-focused approaches consider the smoothness of the resulting mesh, only C^0-continuity is established.

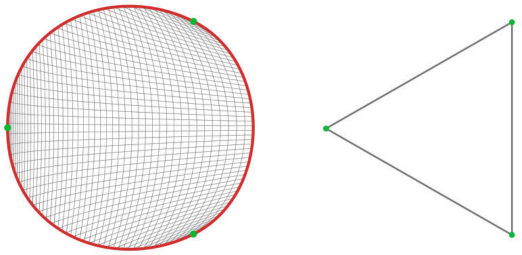

Figure 2: The parametrization of a shape with smooth boundary (left) over a domain with non-smooth boundary (right) cannot be globally C^1-continuous. At least at the spots (green) where a smooth boundary point is in correspondence with a non-smooth corner point, continuity must degrade to C^0 (or the parametrization must degenerate).

2.2 C^1-continuous, not Regular A number of articles address the challenge of generating 2D triangular meshes that offer higher continuity. Qian et al. present methods to generate such meshes for freeform shapes, focusing on ensuring C^r-continuity while minimizing the distortion of the higher-order elements [12, 36]. Their approach utilizes macro-element constructions, such as the Clough-Tocher and Powell-Sabin schemes, subdividing an initial mesh into smaller triangle elements. Similarly, Xia et al. [37] propose a method based on such schemes, with an emphasis on rational elements.

By default, these schemes lead to the desired continuity in the interior, but to only C^0-continuity along the boundary. The above works suggest a modification of the underlying parametric domain mesh, straightening the boundary, so as to raise this to higher continuity in most places. However, no algorithm nor a generally applicable recipe for manual execution of such a modification is described.

Importantly, none of these continuity-focused methods offer any guarantee regarding the regularity of the resulting meshes—the employed macro-element constructions provide no means to control this property. Consequently, the resulting parametrization, while smooth (in the interior), may be non-injective.

2.3 Approximately C^1-continuous As satisfying continuity conditions is a challenge and may require hard trade-offs with other quality aspects, some research efforts aim at relaxing these conditions, trading off strict continuity for better overall geometric quality or lower computational cost [19, 22]. The notion of *approximate continuity*, ε-C^1, can be defined, requiring adjacent elements to meet with the deviation of derivatives bounded by ε. However, such concepts have mainly been used for tangent continuity of 3D surface patches rather than for parametric continuity in planar configurations.

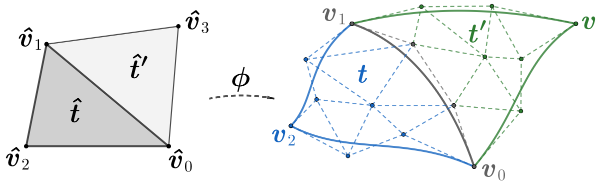

Figure 3: Domains (left) and images (right) of two adjacent Bézier triangles ($n=3$). Control nets are shown dashed, control points p_{ij} as dots.

3 Overview

In this section, after providing essential background definitions, we formalize the problem setting and outline our approach to address this problem.

3.1 Preliminaries Our discussion centers on polynomial elements. We represent these in the Bernstein basis. Hence, we start by defining 2D Bézier curves and 2D Bézier triangles, being fundamental to our method.

Definition 1. (Bézier Curve) *Given control points $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_n \in \mathbb{R}^2$, a 2D (polynomial) Bézier curve $\boldsymbol{c}$ is defined by a univariate geometric map $\varphi : [0,1] \to \mathbb{R}^2$,*

$$\varphi(t) = \sum_{i=0}^{n} \boldsymbol{p}_i B_i^n(t),$$

where $B_i^n(t)$ are the Bernstein polynomials of degree n.

Definition 2. (Bézier Triangle) *A 2D Bézier triangle $\boldsymbol{t}$ represented by a bivariate geometric map $\varphi(u,v) : ◺ \to \mathbb{R}^2$ of order n is defined by its control points (forming its control net) $\boldsymbol{p}_{ij} \in \mathbb{R}^2$, $(i,j \geq 0,\ i+j \leq n)$ as*

$$\varphi(u,v) = \sum_{i+j \leq n} \boldsymbol{p}_{ij} B_{ij}^n(u,v),$$

where $B_{ij}^n(u,v)$ are the triangular Bernstein polynomials and $◺ = \{(u,v) \mid u,v \geq 0, u+v \leq 1\}$ denotes a unit triangular domain.

Note that via pre-composition with an affine map, these geometric maps can also be expressed over any other straight-edge triangle domain instead of the unit triangular domain ◺ (see fig. 9).

These geometric maps, being polynomial, are C^∞. When considering not a single triangle but a mesh of triangles, its triangles' geometric maps join at the edges and vertices (see fig. 3). A join is C^1 if the geometric maps as well as their derivatives agree on the edges and vertices. In fig. 4, we show a comparison of a shape meshed with C^0 (left) and C^1 (right) continuity. C^0-continuity merely requires that adjacent triangles (with adjacent domains) share identical control points along their common edge (as in fig. 3), guaranteeing continuity of the combined map's value across the edges. This implies that iso-curves continue across edges, but commonly with hard kinks. Conversely, in the case of C^1-continuity, in addition adjacent triangles also exhibit agreeing partial derivatives at shared edges and vertices, resulting in smooth transitions without kinks.

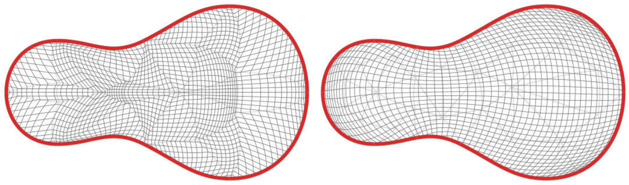

Figure 4: Comparison between a C^0 (left) and a C^1-continuous (right) parametrization of a shape, both defined by the combined geometric maps of conforming curved triangle meshes. Shown is a grid of iso-u and iso-v curves. On the left, notice the prominent kinks in the isocurves.

3.2 Problem Statement The problem that we address is formally defined as follows:

Input: We assume a 2D shape Ω whose boundary is represented by a set of Bézier curves of order n, denoted as $\mathbf{C} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_m\}$. These input curves are assumed to be regular (i.e. with non-vanishing derivative), to intersect only at their endpoints, and to not meet at a zero angle. Besides curves forming a closed shape boundary, the set may contain additional internal curves representing features or interfaces to be respected.

Output: The goal is to generate a higher-order triangular mesh, denoted as $\mathbf{T} = \{\boldsymbol{t}_i\}$, consisting of curved triangles $\boldsymbol{t}_i$. These triangles are defined by a combined geometric map $\Phi : \hat{\Omega} \to \Omega$ over a 2D parametric domain $\hat{\Omega}$. This map $\Phi = \cup_i \phi_i$ consists of per-triangle geometric maps ϕ_i that map straight-edge triangles $\hat{\boldsymbol{t}}_i \in \hat{\Omega}$ onto curved triangles $\boldsymbol{t}_i \in \Omega$. The desired properties of the output mesh include:

- *Conformance:* The boundary of Ω, $\partial\Omega$, is the image of the boundary of $\hat{\Omega}$ under the mapping Φ, i.e. $\Phi\left(\partial\hat{\Omega}\right) = \partial\Omega$.
- *Regularity:* Map Φ is regular within $\hat{\Omega}$, there are no degeneracies or inversions, i.e. $\det\Phi > 0$.
- *Continuity:* Map Φ is close to being globally C^1-continuous on $\hat{\Omega}$.

While $\mathbf{T} = \{\boldsymbol{t}_k\}$ denotes the mesh of curved triangles $\boldsymbol{t}_k$ in *shape space*, with sets of control points $\{\boldsymbol{p}_{ij}\}$, we use $\hat{\mathbf{T}}$, $\hat{\boldsymbol{t}}_{\boldsymbol{k}}$ to refer to the corresponding entities in the *domain space*, cf. fig. 8

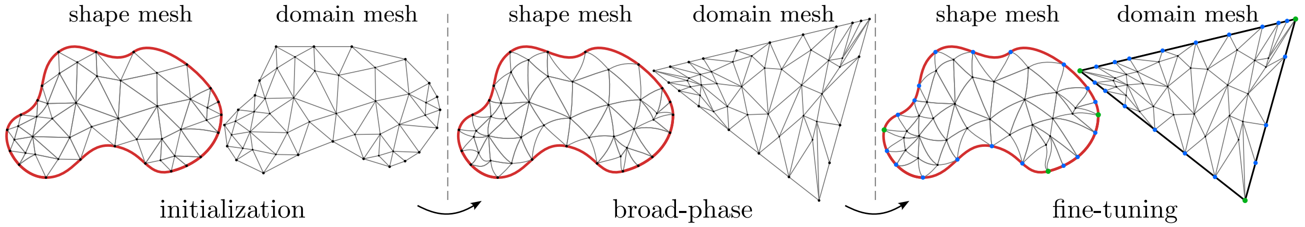

Figure 5: Method overview. Left: Initial triangulation: The input shape (red) is initially triangulated with regular conforming higher-order elements, such that their linearized versions form a valid domain mesh. Center: In a broad phase, both meshes are deformed through a coupled optimization process, resulting in the reduction of exceptional vertices (corners on the domain mesh boundary). Right: Accepting the exceptional vertices (here three green points) and fixing the flatness of the others enables further improvement in a second round of optimization.

3.3 Approach The main steps of our approach are outlined in fig. 5. We start by giving a brief approach overview, before section 4 provides details. First, our method generates an initial higher-order triangulation $\mathbf{T}$ for the given shape using the method proposed in [21]. This triangulation is guaranteed to be regular and conforming—but only C^0. Subsequently, we create a compatible domain mesh $\hat{\mathbf{T}}$, initially obtained as a linearized version of $\mathbf{T}$. We then modify this pair $(\mathbf{T}, \hat{\mathbf{T}})$ towards C^1-continuity, by means of an optimization approach, while maintaining regularity and conformity. The optimization iteratively adjusts the geometry of the shape mesh $\mathbf{T}$ as well as the domain mesh $\hat{\mathbf{T}}$, in two phases. We note that considering not only $\mathbf{T}$ (by means of its inner control point positions $\boldsymbol{p}$) but also $\hat{\mathbf{T}}$ (by means of its vertex positions $\hat{\mathbf{v}}$) a variable in this process is crucial for achieving a low number of exceptional boundary points.

Phase One: Broad-Phase. Both the shape mesh and the domain mesh are deformed by adjusting the control points of the shape mesh and the vertices of the domain mesh. Boundary control points remain fixed, so as to preserve conformance, and the deformation is constrained to preserve regularity.

The main driving force of this deformation process is achieving C^1-continuity across edges, which depends on the shape mesh's control points and the domain mesh's vertices. A particular role is played by the boundary vertices, each of which may initially form a corner or a flat constellation, as an uncontrollable result of the initial mesh generation (fig. 5 left). As shown in fig. 2, a map from a corner to a flat (or smooth) boundary point (or vice versa) cannot be C^1. Hence, aiming for C^1-continuity will cause the domain mesh's boundary vertices to relocate into the necessary corner/flat constellations, as mandated by the shape boundary, to the extent possible (fig. 5 center). This underlines the importance of assuming a variable domain.

Phase Two: Fine-Tuning Not all boundary vertices may reach the desired corner/flat state, as this may be infeasible (see fig. 2) or only achievable under extreme parametric distortion. Those boundary vertices not in the desired state after phase one are identified and classified as exceptional vertices (green points in fig. 5 right). The other boundary vertices (blue) are ensured to be perfectly flat. With these states fixed accordingly, we proceed with another round of optimization. This time C^0-continuity at the exceptional points is accepted and the optimization can focus on instead fine-tuning distortion and C^1-continuity where it is actually achievable, i.e. everywhere else.

Remark Let us remark that it is crucial to maintain bounded mapping distortion during the deformation process. Otherwise it can be energetically cheaper to bring the mesh into near-degenerate states than to properly modify the domain, as illustrated in fig. 6.

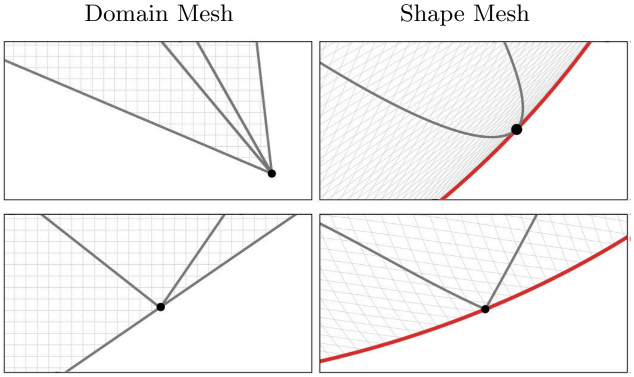

Figure 6: Top: When not enforcing a bound on mapping distortion, a near-degeneracy may evolve where a corner is mapped to a smooth boundary point, effectively hiding the C^1 violation. Bottom: When maintaining a bound, instead the corner is driven towards a flat state when aiming for C^1-continuity.

4 Method

To provide a comprehensive description, we start by reviewing the initial higher-order triangulation approach. Next, we delve into our optimization algorithm, detailing its key components, including the objective function and constraints. Following this, we present the complete optimization algorithm, illustrating how it exploits these components to achieve the desired result.

4.1 Initial Triangulation As starting point we generate a higher-order shape mesh that is regular and conforming to the curves given as input – without regard for continuity. A few mesh generation methods offer these properties in a guaranteed manner [20, 21, 38]. We pick the method by Mandad and Campen [21], because it allows prescribing an upper bound on mesh element distortion, in particular in terms of the MIPS distortion measure [9, 32]. This is beneficial towards being able to maintain low distortion during our optimization process, so as to prevent the issue illustrated in fig. 6.

This mesh generation method first constructs a layer of curved triangular *warp elements* that cover the input curves (see fig. 7 left). These elements are constructed such that they conform to the curves while their inner edges are linear. By means of a constrained Delaunay triangulation, constrained by these edges, a mesh of linear triangles is then constructed. Those of its triangles that lie inside a warp triangle are then deformed according to the warp triangle's geometric map, turning the linear mesh into a higher-order mesh that conforms to the input shape (see fig. 7 right) [21].

Importantly, the thus generated initial higher-order shape mesh $\mathbf{T}$ possesses a known valid corresponding linear domain mesh $\hat{\mathbf{T}}$. While it is tempting to simply use the linearized version of $\mathbf{T}$ (with triangles spanned by the curved triangle's corner points) as $\hat{\mathbf{T}}$, this mesh may exhibit degeneracies or overlaps. The above Delaunay triangulation *before* applying the warp deformation, however, is valid by construction and can thus be used as $\hat{\mathbf{T}}$ (generally or as fallback). Defining the curved shape mesh triangles over these domain triangles, their geometric maps ϕ_i therefore together form a C^0-parametrization Φ of the shape over the domain.

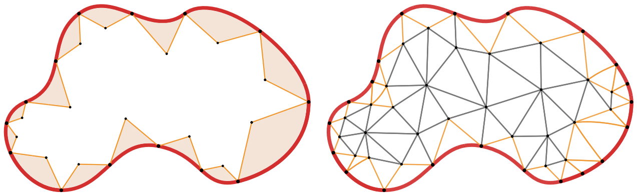

Figure 7: Initial triangulation process: The input curves (red) are covered with curved warp elements (left). Then a linear mesh is constructed, constrained by these elements' straight edges. Triangles lying inside a warp element (orange) are then warped into a curved boundary conforming state (right).

Note that this map Φ restricted to the non-warp region is actually the identity. This is because shape mesh triangles in this region are identical with their domain mesh counterparts by construction. Hence, in this region the map is already C^1-continuous (even C^∞). Concretely, we already have the desired C^1-continuity where these linear elements join (black in fig. 7 right). Across edges (and vertices) that are incident to at least one warp-affected triangle, however, the map is only guaranteed to be C^0 (orange in fig. 7 right).

We accordingly categorize the edges:

- *Soft edges* $\mathbf{E}^{\mathbf{T}}_{\mathbf{S}}$*:* Interior edges of $\mathbf{T}$ that are incident to at least one warped triangle, as indicated in orange in fig. 7 right.
- *Hard edges* $\mathbf{E}^{\mathbf{T}}_{\mathbf{H}}$*:* Interior edges of $\mathbf{T}$ that are incident to only non-warped triangles, as indicated in black in fig. 7 right.

4.2 Optimization Components Starting from the above initial configuration $(\mathbf{T}, \hat{\mathbf{T}})$, the central goal of our method is to adjust it so as to achieve C^1-continuity also at the soft edges, while maintaining it at the hard edges. Therefore, in our optimization strategy, we impose C^1-maintaining constraints at hard edges and C^1-promoting objective terms at soft edges. Additional constraints ensure that conformance and bounded distortion are maintained. The importance of the latter is illustrated in fig. 6. Note that these distortion constraints imply preservation of regularity. Finally, as all these constraints still leave a potentially large space of minimizers, we further add a regularizing term, favoring low distortion relative to the initial state.

Below, these ingredients are described in detail and then combined in an overall optimization strategy.

C^1-Energy for Soft Edges For an inner directed soft edge $e = (\boldsymbol{v}_0, \boldsymbol{v}_1)$ of initial triangulation $\mathbf{T}$, consider the two adjacent triangles $\boldsymbol{t} = \{\boldsymbol{p}_{ij}\}$ and $\boldsymbol{t}' = \{\boldsymbol{p}'_{ij}\}$ in $\mathbf{T}$ sharing this edge. Correspondingly, $\hat{\boldsymbol{t}} = \triangle(\hat{\boldsymbol{v}}_0\hat{\boldsymbol{v}}_1\hat{\boldsymbol{v}}_2)$ and $\hat{\boldsymbol{t}}' = \triangle(\hat{\boldsymbol{v}}_1\hat{\boldsymbol{v}}_0\hat{\boldsymbol{v}}_3)$ represent the domain mesh triangles in the parameter space, as illustrated in fig. 8. Let $\boldsymbol{\gamma} = \{\gamma_0, \gamma_1, \gamma_2\}$ be the barycentric coordinates of $\hat{\boldsymbol{v}}_3$ with respect to triangle $\hat{\boldsymbol{t}}$. The join of $\boldsymbol{t}$ and $\boldsymbol{t}'$ across the common edge e is C^1 if [12]:

1. $\boldsymbol{p}'_{0(n-i)} = \boldsymbol{p}_{0i}$ $(0 \leq i \leq n)$, i.e. the positions of corresponding control points along the edge e match, ensuring C^0-continuity.

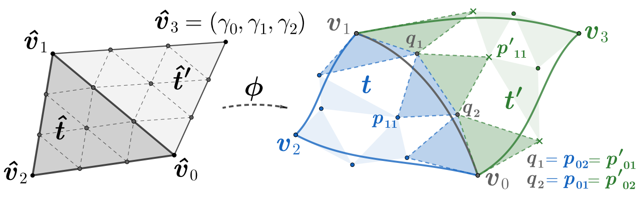

Figure 8: Two Bézier triangles sharing an edge. It is assumed that their control points are indexed such that the shared edge is the first edge in each triangle. C^1-continuity across the shared edge is given iff the second row control points in the green triangle (indicated by cross points) lie at specific positions implied by the control points of the first and second row in the blue triangle.

2. $\boldsymbol{p}'_{1(n-i-1)} = \gamma_0 \boldsymbol{p}_{0i} + \gamma_1 \boldsymbol{p}_{0(i+1)} + \gamma_2 \boldsymbol{p}_{1i}$ $(0 \le i \le n-1)$, i.e. the control points $\boldsymbol{p}'_{1(n-i-1)}$ of $\boldsymbol{t}'$ match a specific affine combination of neighboring control points of $\boldsymbol{t}$, with coefficients $\{\gamma_0, \gamma_1, \gamma_2\}$.

Based on this, we introduce C^1-continuity deviation $D_e \ge 0$ per $e \in \mathbf{E}^{\mathbf{T}}_{\mathbf{S}}$, such that $D_e = 0$ implies a C^1-smooth join of the triangles incident to e. It is defined as follows:

- $\boldsymbol{p}^{\times}_{1(n-i-1)} = \gamma_0 \boldsymbol{p}_{0i} + \gamma_1 \boldsymbol{p}_{0(i+1)} + \gamma_2 \boldsymbol{p}_{1i}$ $(0 \le i \le n-1)$, is the expected position of the control point $\boldsymbol{p}'_{1(n-i-1)}$.
- $d_{e,i} = \left\| \boldsymbol{p}'_{1(n-i-1)} - \boldsymbol{p}^{\times}_{1(n-i-1)} \right\|_2$ is the Euclidean distance between the current position of a control point and its expected position.
- D_e aggregates the deviations $d_{e,i}$ along e, providing an indication of how well the current configuration of edge e aligns with the expected one. Concretely,

$$D_e = \sum_{0 \le i \le n-1} d_{e,i}. \tag{4.1}$$

With this, the total C^1-continuity objective term, which serves as an indicator of the smoothness along the soft edges $\mathbf{E}^{\mathbf{T}}_{\mathbf{S}}$ is expressed as:

$$E_s = \sum_{e \in \mathbf{E}^{\mathbf{T}}_{\mathbf{S}}} D_e. \tag{4.2}$$

Note that this term depends on both the shape mesh $\mathbf{T}$ (via $\boldsymbol{p}$) and the domain mesh $\hat{\mathbf{T}}$ (via $\boldsymbol{\gamma}$).

Regularizer To facilitate the formulation of the regularizer over the mesh $\mathbf{T} = \{\boldsymbol{t}_i\}$, we define the following terms, as illustrated in fig. 9:

- $M_{ij} = [\boldsymbol{a}_{ij}, \boldsymbol{b}_{ij}]$ is the matrix formed by *control vectors* $\boldsymbol{a}_{ij} = \boldsymbol{p}_{(i+1)j} - \boldsymbol{p}_{ij}$ and $\boldsymbol{b}_{ij} = \boldsymbol{p}_{i(j+1)} - \boldsymbol{p}_{ij}$ at control point $\boldsymbol{p}_{ij}$,

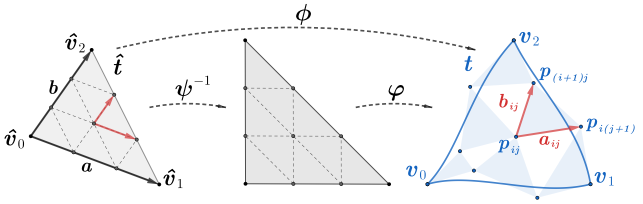

Figure 9: The transformation function ϕ, which maps a triangle $\hat{\boldsymbol{t}}$ in domain space to a curved triangle $\boldsymbol{t}$ in shape space, can be considered as composition of two mappings: $\phi = \varphi \circ \psi^{-1}$. The first mapping, ψ^{-1}, transforms the domain triangle to the unit triangle, while the second, φ, is a Bézier triangle map. The indicated vectors $\frac{1}{n}\boldsymbol{a}$, $\frac{1}{n}\boldsymbol{b}$ and $\boldsymbol{a}_{ij}$, $\boldsymbol{b}_{ij}$ are employed in formulating our regularizer.

- $\hat{M} = \frac{1}{n}[\boldsymbol{a}, \boldsymbol{b}]$ is the matrix formed by the corresponding control vectors in the linear domain triangle, where $\boldsymbol{a} = \hat{\boldsymbol{v}}_{\mathbf{1}} - \hat{\boldsymbol{v}}_{\mathbf{0}}$ and $\boldsymbol{b} = \hat{\boldsymbol{v}}_{\mathbf{2}} - \hat{\boldsymbol{v}}_{\mathbf{0}}$ represent two edge vectors of $\hat{\boldsymbol{t}}$ in their pre-optimization state,
- $R_{ij} = \|J_{ij}\|_F^2 + \left\|J_{ij}{}^{-1}\right\|_F^2$, with $J_{ij} = M_{ij}\hat{M}^{-1}$, then effectively measures the local deviation of the shape triangle control net from the domain triangle control net, in the style of the symmetric Dirichlet energy [27, 33].

With this and the area $A^{\boldsymbol{t}} = \frac{1}{2}\|\boldsymbol{a} \times \boldsymbol{b}\|_2$ of $\hat{\boldsymbol{t}}$, a per-triangle notion of deviation can be defined as $E^{\boldsymbol{t}}_{\text{reg}} = A^{\boldsymbol{t}} \sum_{i+j<n} R_{ij}$. Aggregating this over all triangles $\boldsymbol{t}_i$ in the mesh $\mathbf{T}$ provides an overall deviation measure

$$E_{\text{reg}} = \sum_{\boldsymbol{t}_i \in \mathbf{T}} E^{\boldsymbol{t}_i}_{\text{reg}}, \tag{4.3}$$

which we employ as regularizing objective term.

C^1-Constraint for Hard Edges In section 4.1 we highlighted that the initial triangulation method yields C^1-continuity across the set of hard edges, shared by linear triangles. To preserve C^1-continuity along these edges throughout the deformation process, we introduce constraints based on (4.1):

$$d_{e,i} = 0 \quad \forall e \in \mathbf{E}^{\mathbf{T}}_{\mathbf{H}}, 0 \le i \le n-1. \tag{4.4}$$

Note that these are satisfied initially by construction.

Distortion Constraint In our approach it is important that triangles remain within a specified range of distortion during the deformation process. To establish this, we utilize the (pointwise) definition of MIPS distortion [9], defined as $\|J_\phi\|^2 \det(J_\phi)^{-1}$, where ϕ is the map of a triangle (see fig. 9). Its maximum over triangle $\boldsymbol{t}$ is bounded from above by

$$\max_{(u,v)\in\hat{\boldsymbol{t}}} \|J_\phi\|^2 / \min_{(u,v)\in\hat{\boldsymbol{t}}} \det(J_\phi),$$

which in turn is bounded from above by $g(\boldsymbol{t}) = \max_i n_i / \min_i d_i$, where n_i and d_i are the Bernstein coefficients of $\|J_\phi\|^2$ and $\det(J_\phi)$, respectively, for which closed form expressions are available [21]. Based on this, we define distortion bounding constraints as

$$g(\boldsymbol{t}) \le \alpha\mu \quad \forall \boldsymbol{t} \in \mathbf{T}. \tag{4.5}$$

Here, μ denotes an upper bound of the *initial* MIPS distortion, and the factor $\alpha > 1$ determines the amount of "wiggle room" provided for deformation. Note that for any choice $\alpha \ge 1$ these constraints are satisfied initially. These constraints ensure that the mesh elements' distortion does not exceed $\alpha\mu$ throughout the optimization process.

4.3 Optimization Strategy Based on the above ingredients, objective terms (4.2) and (4.3) and constraints (4.4) and (4.5), we formulate the following overall optimization problem:

$$\begin{aligned}
&\text{(4.6a)} \qquad && \min \quad && E_s + \lambda_{\text{reg}} E_{\text{reg}} \\
&\text{(4.6b)} && \text{s.t.} && g(\boldsymbol{t}) \le \alpha\mu,\ \forall \boldsymbol{t} \in \mathbf{T} \\
&\text{(4.6c)} && && d_{e,i} = 0,\ \forall e \in E_H^T, i \\
&\text{(4.6d)} && && \partial\Omega = \text{fixed},
\end{aligned}$$

where $\lambda_{\text{reg}} > 0$ is a small prefactor for the regularizing term. The latter constraint implies that during the deformation process, the boundary of the shape, $\partial\Omega$, maintains its initial configuration satisfying conformance. To this end the control points along the boundary simply remain fixed.

As pointed out above, achieving $E_s^{\mathbf{T}} = 0$, i.e. a globally C^1 state, can be infeasible, as exceptional points can be inevitable. We therefore approach the problem (4.6) in two phases. The overarching goal of the first phase is to determine a suitable set of exceptional points. In the second phase these are accepted as given, and continuity is optimized for only away from them.

1. *Phase One:* In this phase, problem formulation (4.6) is used as is. Variables up for optimization in this phase are:
    - $V_{\hat{\mathbf{T}}}$, the set of vertex positions of the domain mesh.
    - $\{\boldsymbol{p}_{ij}\} \setminus \{\boldsymbol{c}_{ij}\}$, the set of all control points of all triangles, without its subset of boundary control points $\{\boldsymbol{c}_{ij}\}$. The latter is effectively invariable due to constraint (4.6d).
2. *Phase Two:* Based on the result of phase one, exceptional points are identified as described below. Let $\mathbf{E}_\mathbf{S}'^\mathbf{T} \subset \mathbf{E}_\mathbf{S}^\mathbf{T}$ be the subset of soft edges that are *not* incident to exceptional points. In phase two we employ this subset in place of the full set, so as to accept that around the exceptional points only C^0-continuity can be asked for. Concretely, we limit E_s in (4.6) to edges in $\mathbf{E}_\mathbf{S}'^\mathbf{T}$. Furthermore, the domain mesh is now considered fixed, i.e. the set of variables in this phase is only $\{\boldsymbol{p}_{ij}\} \setminus \{\boldsymbol{c}_{ij}\}$.

Exceptional Point Determination Between phases one and two, we need to identify the exceptional points. Conceptually, after phase one, the domain mesh typically exhibits a few corners and otherwise flat boundary vertices (fig. 5 center). This is because boundary vertices in correspondence with a smooth boundary point of the shape are implicitly pushed towards a flat state by the C^1-objective (4.2). Vertices in correspondence with a corner of the shape do not experience this effect and remain corners—plus possibly a few more because a certain minimum number is required to even form a valid polygonal domain, and because the distortion bound (4.6b) limits deformation.

Due to the numerical nature of the deformation process, the decision is not always clear-cut, vertices may be near-flat. We therefore move these near-flat vertices into a flat state, as illustrated in fig. 10. While it is often possible to do this by simply projecting onto straight lines between corners, in general this may unintentionally introduce degenerate or flipped triangles in the domain mesh, violating the required properties. To safely achieve a state with perfectly flat vertices between corners in any case, we proceed as follows, as illustrated in fig. 11. We begin by marking only the sharpest vertices as corners. Then, for each sequence of boundary vertices between two marked corners, we

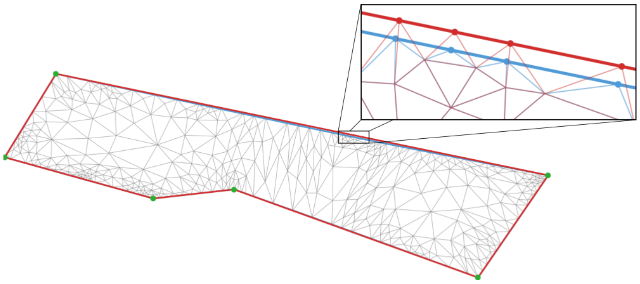

Figure 10: The domain mesh resulting from phase one typically has a piecewise near-straight boundary (red), here for one of the example input shapes from section 5.3. The straightened version is shown in blue, with a blow-up of the region of largest deviation.

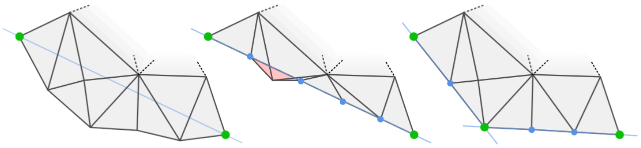

Figure 11: Illustration of exceptional point determination, on an example exaggerated for clarity. Left: Initial state between two corners (green). Center: Straightening of the entire sequence causes irregularity (red). Right: Accepting an additional corner enables straightening of the subsequences.

tentatively move them orthogonally onto the straight line between the two corners, to make them flat. If this invalidates the mesh, we restore their positions and mark the sharpest vertex among them as additional corner. This is repeated recursively. Upon termination, each boundary vertex is either marked as a corner, or is truly flat. Those corners in correspondence with a smooth boundary point in the shape mesh are the exceptional points that are then exploited in phase two.

Local vs Global Injectivity The domain mesh deformation may result in global overlaps. In this case, while the map Φ is still a bijection between the domain mesh and the shape mesh, it is no longer well-defined over a subset of the plane, due to multiple $\hat{\boldsymbol{t}}$ coinciding with a point (u, v). The optional use of a surrounding scaffold mesh [13] can prevent the occurrence of such overlaps.

5 Experiments

In this section, we evaluate the effectiveness of an implementation of the method described. We present results obtained from executing it on diverse sets of input shapes, including randomly generated shapes for stress testing purposes. The polynomial order is $n = 3$ in these experiments.

5.1 Implementation The implementation we use for the experiments employs a second-order Newton-style optimization method with line search to solve the two variants of optimization problem (4.6) in the two phases. To facilitate the easy computation of the required first and second partial derivatives, we employ TinyAD [31], a library for automatic differentiation tailored to mesh-based processing.

The distortion bounding inequality constraint (4.6b) is taken into account by means of a primal log-barrier [23], adding a barrier function per triangle $\boldsymbol{t}_i \in \mathbf{T}$:

$$E_q = \sum_{\boldsymbol{t}_i \in \mathbf{T}} -\rho \log\left(\alpha\mu - g(\boldsymbol{t}_i)\right),$$

where ρ is a positive constant. The nonlinear equality constraint (4.6c) is taken into account by means of the penalty method [23], with penalty factor λ_h, adding

$$E_h = \lambda_h \sum_{e \in \mathbf{E}_{\mathbf{H}}^{\mathbf{T}}} D_e.$$

The constraint (4.6d), being linear, is incorporated simply by means of variable elimination.

Parameters For our experiments we make use of the following default setting of the method's parameters, unless stated otherwise.

- λ_{reg} is set to 10^{-6} in phase one, and subsequently lowered further to 10^{-8}, so as to reduce regularization effects to a minimum. λ_h is set to 10^2 by default. Reduction to 1 in phase two, after the exceptional points have been fixed, proved to offer small numerical benefits. These values have been selected via a grid search parameter sweep over the example input data reported on below.

- ρ: A value of 0.1 proved to offer a practical balance between convergence speed and deformation flexibility.

- α allows a trade-off between the quality of the resulting mesh and parametrization, in terms of distortion, and the number of exceptional points. A higher value of α enlarges the search space during deformation, allowing for greater flexibility in achieving flat boundary vertices wherever beneficial. At the same time, this flexibility may lead to more severe deformations and thereby distortion. Figure 12 illustrates this effect. A comprehensive comparison is part of the experiment reported on in section 5.3. In phase two, the distortion bound is less crucial; lifting α (e.g. by a factor of 3) proved to enable slightly improved fine tuning in this phase.

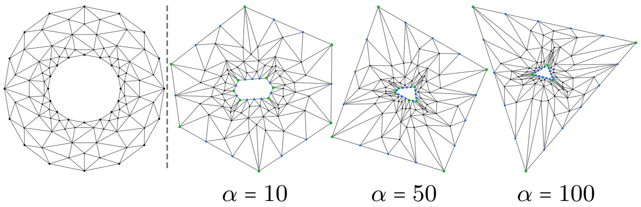

Figure 12: The image on the left shows the initial domain mesh of an annulus shape (MIPS bound $\mu = 10$), the others are resulting domain meshes. Note how a higher value leads to less corners, thus less exceptional points, but more distorted elements.

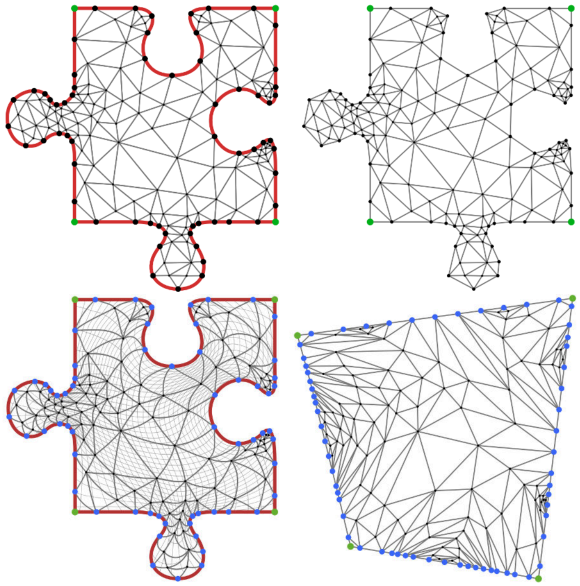

Figure 13: Top: Initial shape mesh and initial domain mesh of a puzzle piece shape that has four non-smooth corners (green). Bottom: The shape mesh and domain mesh resulting from our method, exhibiting global C^1-continuity without any exceptional points.

5.2 Qualitative Analysis Figure 13 shows an example shape with piecewise smooth boundary between four corners. As can be seen, the domain mesh resulting from our method is quadrilateral-shaped, with corner vertices corresponding to the shape's four corner points. All the other boundary vertices, corresponding to smooth boundary points, were successfully brought into a flat state, enabling a globally C^1-continuous result, i.e. a globally smooth parametrization of the freeform shape over a polygonal domain.

Figure 15 illustrates an example of smooth spiral-shaped input. The algorithm effectively straightens this intricate structure, producing a bar-shaped domain mesh with only four exceptional points. This example demonstrates that major large-scale domain deformations, beyond mere local straightening, can be necessary to avoid surplus exceptional points, and that the algorithm is able to perform these.

Figure 14 shows another, more complex example, with a mix of curved and straight segments, smooth and sharp boundary points, and nontrivial topology. A state without exceptional points is achieved.

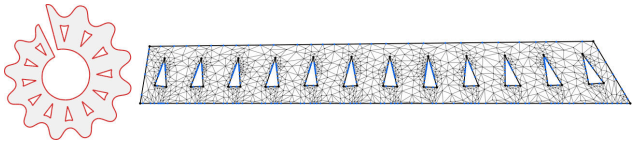

Figure 14: Example shape and resulting domain mesh.

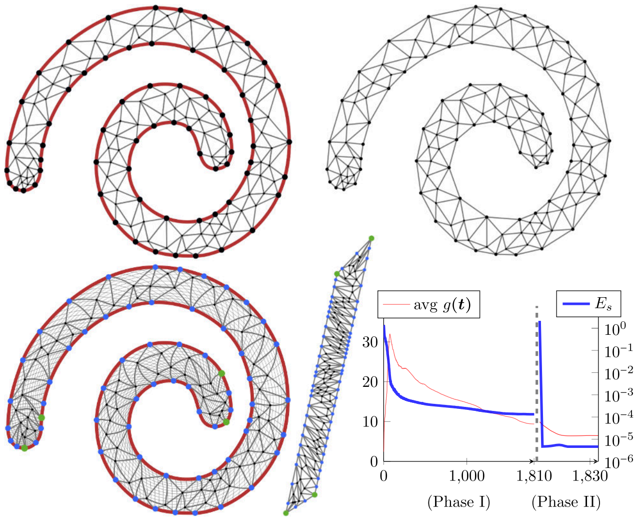

Figure 15: Top: Initial shape mesh and domain mesh of a smooth spiral shape. Bottom: The shape mesh and domain mesh resulting from our method, exhibiting four exceptional points (green). The graphs illustrate the progression of the main objective, total C^1-deviation E_s, and the distortion $g(\boldsymbol{t})$ over the optimization iterations. Notice the additional reduction in phase two, having accepted the exceptional points.

Figure 16 illustrates a shape with internal curves, representing features or interfaces. These can be handled by treating them like boundary curves, not requiring C^1-continuity across them, and marking curve branch points (pink) exceptional from the start.

5.3 Quantitative Analysis For a quantitative analysis, we make use of a larger-scale evaluation based on datasets of randomly generated shapes. We created

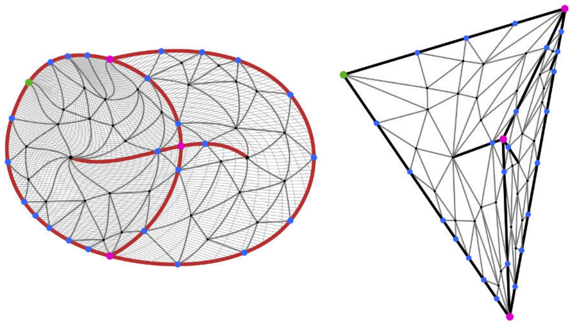

Figure 16: Final shape and domain mesh of an input composed of two sub-shapes with internal features.

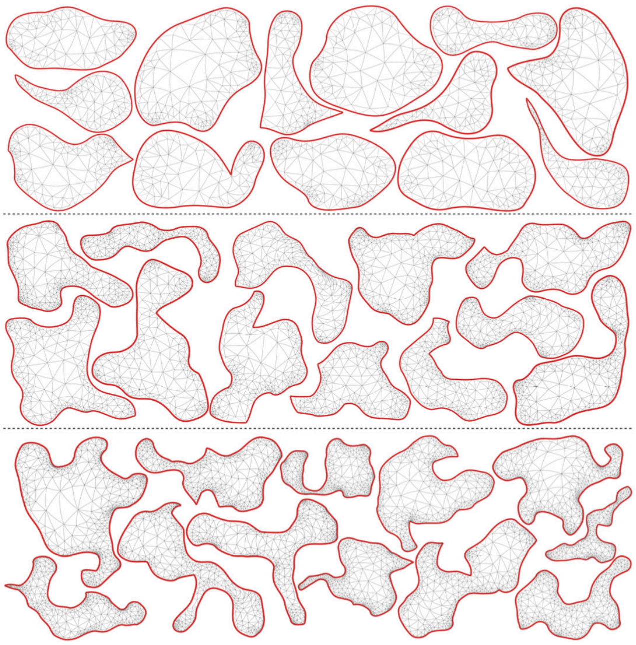

Figure 17: Some of the instances of random piecewise smooth shapes from (top to bottom) set A, set B, set C. Their resulting higher-order mesh is shown in black.

three sets of 100 shapes each, sets A, B, C (fig. 17). Each contained shape's boundary is formed by 10, 20, and 30 curves, respectively. These curves were generated such that the fraction of their joints that are non-smooth is uniformly distributed in the range $[0\%, 20\%]$.

Metrics To assess the quality of our method's results, we make use of the following indicators:

- *Flatness ratio*: The proportion of boundary vertices that correspond to smooth shape boundary points and that the method successfully makes flat, thereby preventing exceptional points. The higher and closer this value is to 1, the better. However, note that the best theoretically achievable value can be lower than 1. For instance, a completely smooth simple shape requires at least three exceptional points (fig. 5), smooth non-simple shapes can require even more (fig. 12).
- *Continuity deviation*: Due to the numerical nature of the approach (and possibly due to nonconvexity), there is a chance for small deviations from the perfect C^1-continuity aimed for. This deviation can be assessed directly in terms of the values $d_{e,i}$, introduced in eq. (4.1). More intuitively, we also examine the mismatch of gradients of adjacent triangles along their common edge, in terms of their angle and of their magnitude.

Results Figure 18 presents evaluation statistics for the method's results on the random shape sets. We observe:

- The continuity deviation $d_{e,i}$ (blue) is concentrated around 10^{-6} (relative to the shape's bounding box extent). The per shape maximum of these values (red) is concentrated around 10^{-5}. The tendency is towards even smaller values as the shapes get more complex (set A → set B → set C).
- The per-edge maximum of the angular deviation of the gradient along edges (evaluated at ten equidistant sample points per edge), is always below 0.5°, with the vast majority way below 0.01°. The maximum relative magnitude deviation of the derivatives along edges is always below 0.004, with the vast majority way below 0.0001.
- The number of exceptional points per shape (red) shows a dependence on parameter α. For $\alpha = 10$, it is concentrated around 5 or 6, while for $\alpha = 100$, providing more room for deformation, it is at 3 or 4. As for the type of shapes used here the theoretical minimum is known, $\max(3 - n_c, 0)$, where n_c is the number of input boundary corners, also the number of *surplus* exceptional points (blue) can be analyzed. For the $\alpha = 100$ setting for set A it most often is 0, for set B 1, and for the most complex set C most commonly 2 or 3, overall indicating good performance of the method in this regard.
- The flatness ratio, which is 1 if no exceptional points remain, is mostly well above 0.9.
- The run time of our current prototype implementation of the method ranges from around 1 minute for the simplest shapes up to around 50 minutes for the most complex outlier instance.

5.4 Post-Processing

While deviations from C^1-continuity turn out to be small in the results, it may be tempting to modify the mesh into a perfectly C^1-state. This can be achieved by projecting the variables into the solution space of the C^1-constraints.

C^1-Projection Following the derivation of equation (4.2), a mesh is C^1 across an edge e if

$$\gamma_0 \boldsymbol{p}_{0i} + \gamma_1 \boldsymbol{p}_{0(i+1)} + \gamma_2 \boldsymbol{p}_{1i} - \boldsymbol{p}'_{1(n-i-1)} = 0 \quad (0 \leq i \leq n-1).$$

Compiling these equations for all edges from $\mathbf{E}_{\mathbf{H}}^{\mathbf{T}} \cup \mathbf{E}_{\mathbf{S}}'^{\mathbf{T}}$ yields a linear equation system $A\boldsymbol{p} = 0$, where A is a matrix whose rows represent the coefficients of the above constraints, and $\boldsymbol{p}$ is the vector consisting of the shape

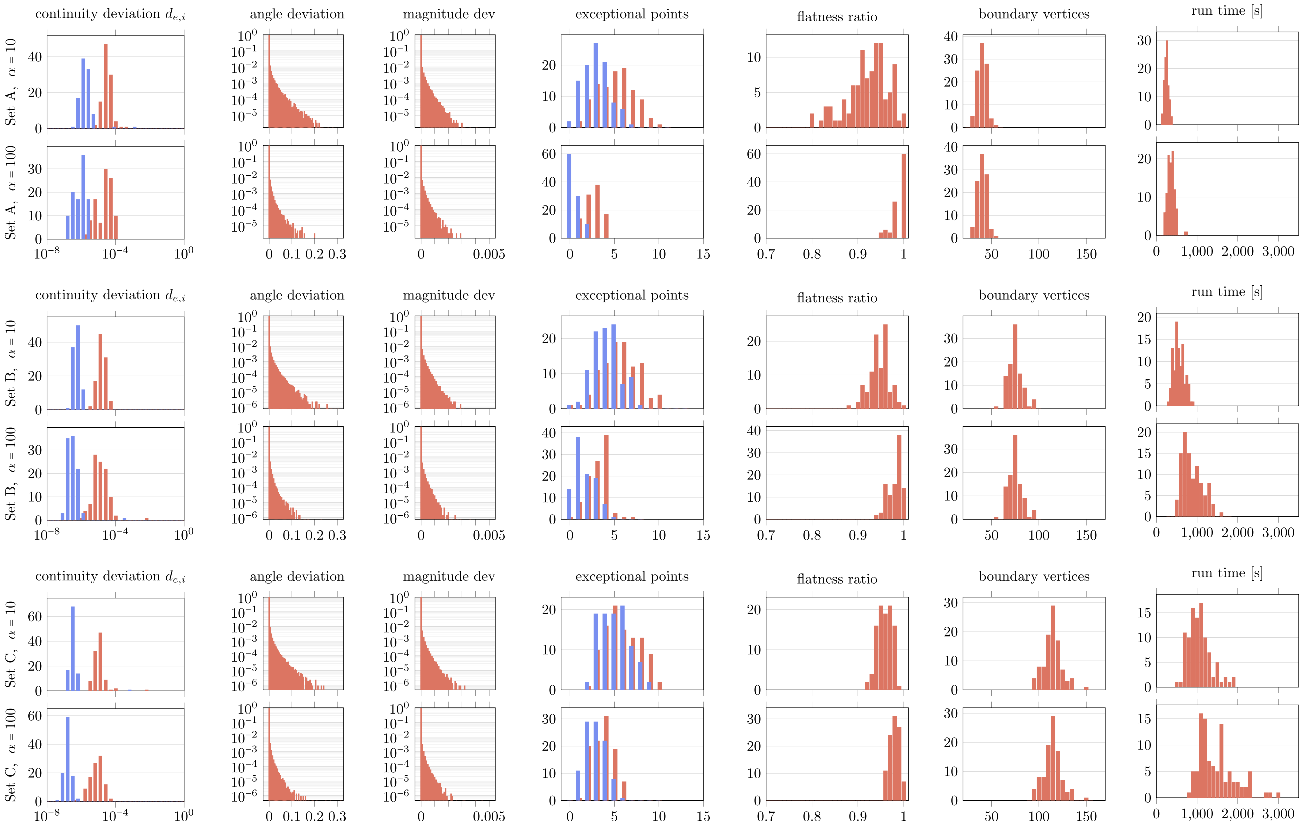

Figure 18: Histograms of various evaluation metrics over results for the random shape data sets A, B, and C, for two different settings of $\alpha = 10$ and $\alpha = 100$.

mesh variables, i.e. all control points. The matrix

$$P = I_n - A^{\dagger}A, \tag{5.7}$$

where $A^{\dagger}$ is the pseudoinverse of A, performs an orthogonal projection into the nullspace of A, thus into the solution space of $A\boldsymbol{p} = 0$. The matrix $A^{\dagger}A$ can be computed [30] via singular value decomposition as $V_r V_r^T$, where V_r consists of the first r right-singular vectors of A and r is the rank of A.

Applying this projection to the control points $\boldsymbol{p}$ of a mesh, $\bar{\boldsymbol{p}} = P\boldsymbol{p}$, yields a perfectly C^1-continuous mesh defined by control points $\bar{\boldsymbol{p}}$. The cost of this, however, is the potential loss of regularity, though the orthogonality minimizes the deviation from the regular state $\boldsymbol{p}$, reducing the risk. Furthermore, as also the boundary control points are projected, conformance to the original shape may be affected.

Analyzing the effects of this C^1-projection, applied to the above results reported on in section 5.3 and fig. 18, we make the following observations:

- *Continuity deviation:* Figure 19 confirms that, after projection, the mesh indeed is C^1, up to the numerical accuracy limit. Namely, values $d_{e,i}$ are around 10^{-14}, their maxima typically around 10^{-13}.
- *Regularity:* No triangle in any mesh became irregular due to the projection, underlining that the former C^1-deviations could be considered minuscule.
- *Conformance:* Boundary control points are moved by the projection commonly in the range 10^{-6}–10^{-4}, relative to the shape's bounding box extent. In this sense conformance is slightly approximate after the projection. Depending on the use case this deviation may be preferable in exchange for exact C^1-continuity.

5.5 Improvement through Repetition As a final note, further experiments indicate that iterating the optimization process (section 4.3) twice (i.e. phases one-two-one-two) yields significant further accuracy improvements on a numerical level. Namely, the continuity deviations reported in fig. 18 are reduced by around two to three orders of magnitude. This improvement translates into a similar reduction in conformance deviation in case the C^1-projection is applied. See fig. 20.

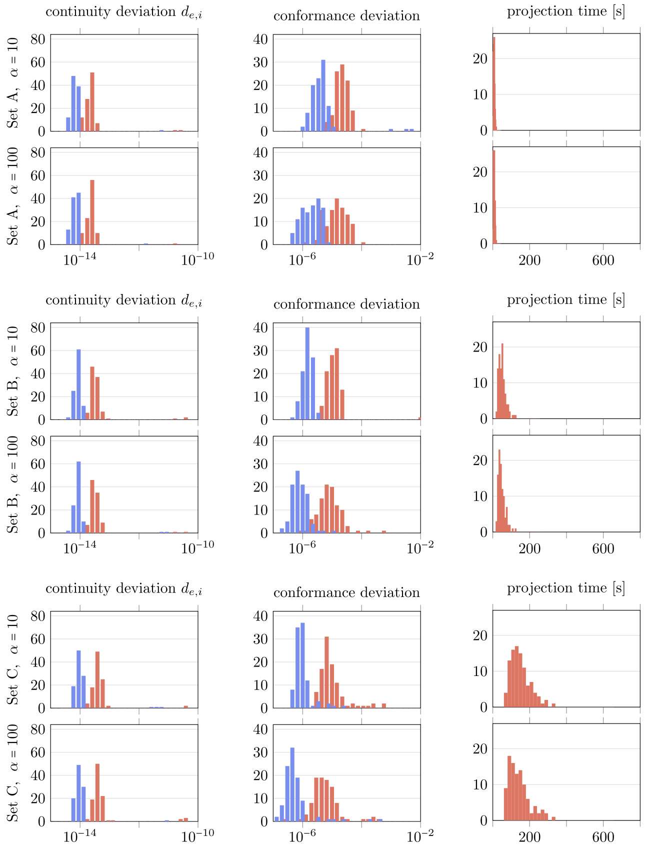

Figure 19: Histograms of various evaluation metrics over results for the random shape data sets A, B, C, *after applying C^1-projection*.

6 Limitations and Future Work

We introduced a method for C^1-focused higher-order mesh generation, that can also be viewed as constructing parametrizations of 2D freeform shapes over polygonal domains. The parametrization is expressed in a piecewise polynomial manner by means of a mesh of higher-order triangles. Our approach utilizes an optimization strategy, constrained to ensure that mesh elements are free from inversions while guaranteeing that the resulting mesh matches the given input without geometric error. The following limitations and areas for future work can be highlighted:

- *Efficiency Improvement:* There is clear potential for improving the optimization process. This concerns in particular the overall run time, where precomputed derivative expressions, parallelization, and preconditioning could aid. It also concerns the convergence behavior, affecting the number of surplus exceptional points (due to time limits, getting trapped in local minima, or stuck due to numerical issues).

- *Bounded C^1-Deviation:* While our method was shown to be capable of commonly producing results with minuscule C^1-deviations, there is no guarantee in this regard. Future research may aim for variants that allow providing a guarantee of meeting a prescribed deviation bound, providing stronger assurances about the smoothness of the generated meshes. This will likely require an adaptive combinatorial mesh modification strategy.

- *Adaptive Distortion:* Our method allows setting a distortion bound, and the number of exceptional points follows. The inverse may also be interesting, being able to choose the number of exceptional points, while the distortion is internally adapted as necessary to achieve this.

- *3D Extension:* Our method targets 2D curved shapes. Extending this approach to 3D volumetric shapes, based on curved tetrahedral meshes, represents a worthwhile direction for future research, expanding the utility of the approach for practical scenarios. The initial mesh could be obtained using a recent higher-order tetrahedralization method [14], the C^1-energy and C^1-constraint expressions extend to 3D naturally [16]. More work will be necessary regarding the exceptional point determination; in 3D a network of exceptional edges (between planar regions) needs to be determined, e.g. by some form of clustering of near-coplanar boundary facets in the domain mesh.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) - 451286978. The authors wish to thank Teseo Schneider for background discussions.

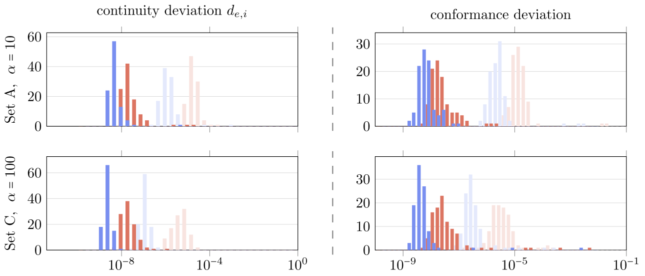

Figure 20: Repeating the two optimization phases *twice* fine-tunes the results, yielding further accuracy improvements. Left: Continuity deviation as in fig. 18. Right: conformance deviation as in fig. 19. The former results are shown light-shaded for comparison.

References

[1] I. Babuška and B. Guo, *Approximation properties of the hp version of the finite element method*, Comput. Methods Appl. Mech. Eng., 133 (1996), pp. 319–346.

[2] Y. Bazilevs and I. Akkerman, *Large eddy simulation of turbulent taylor–couette flow using isogeometric analysis and the residual-based variational multiscale method*, Journal of Computational Physics, 229 (2010), pp. 3402–3414.

[3] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. Hughes, *Isogeometric finite element data structures based on Bézier extraction of NURBS*, International Journal for Numerical Methods in Engineering, 87 (2011), pp. 15–47.

[4] J. Cottrell, T. Hughes, and A. Reali, *Studies of refinement and continuity in isogeometric structural analysis*, Computer methods in applied mechanics and engineering, 196 (2007), pp. 4160–4183.

[5] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs, *Isogeometric analysis: toward integration of CAD and FEA*, John Wiley & Sons, 2009.

[6] J. A. Cottrell, A. Reali, Y. Bazilevs, and T. J. Hughes, *Isogeometric analysis of structural vibrations*, Computer methods in applied mechanics and engineering, 195 (2006), pp. 5257–5296.

[7] C. Dobrzynski and G. El Jannoun, *High order mesh untangling for complex curved geometries*, PhD thesis, INRIA Bordeaux, équipe CARDAMOM, 2017.

[8] C. Geuzaine, A. Johnen, J. Lambrechts, J. F. Remacle, and T. Toulorge, *The generation of valid curvilinear meshes*, IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach, (2015), pp. 15–39.

[9] K. Hormann and G. Greiner, *MIPS: An efficient global parametrization method*, Curve and Surface Design '99, (2000), pp. 153–162.

[10] Y. Hu, T. Schneider, X. Gao, Q. Zhou, A. Jacobson, D. Zorin, and D. Panozzo, *Triwild: robust triangulation with curve constraints*, ACM Transactions on Graphics (TOG), 38 (2019), pp. 1–15.

[11] T. J. 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 (2005), pp. 4135–4195.

[12] N. Jaxon and X. Qian, *Isogeometric analysis on triangulations*, Computer-Aided Design, 46 (2014), pp. 45–57.

[13] Z. Jiang, S. Schaefer, and D. Panozzo, *Simplicial complex augmentation framework for bijective maps*, ACM Trans. Graph., 36 (2017).

[14] P. Khanteimouri and M. Campen, *3D Bézier Guarding: Boundary-conforming curved tetrahedral meshing*, ACM Trans. Graph., 42 (2023).

[15] P. Khanteimouri, M. Mandad, and M. Campen, *Rational Bézier Guarding*, Computer Graphics Forum, 41 (2022), pp. 89–99.

[16] M.-J. Lai and L. L. Schumaker, *Spline functions on triangulations*, no. 110, Cambridge University Press, 2007.

[17] M.-J. Lai and P. Wenston, *Bivariate splines for fluid flows*, Computers & fluids, 33 (2004), pp. 1047–1073.

[18] K. Li and X. Qian, *Isogeometric analysis and shape optimization via boundary integral*, Computer-Aided Design, 43 (2011), pp. 1427–1437.

[19] Y. Liu, *Triangular Bézier surfaces with approximate continuity*, PhD thesis, University of Waterloo, 2008.

[20] M. Mandad and M. Campen, *Bézier Guarding: Precise higher-order meshing of curved 2D domains*, ACM Transactions on Graphics (TOG), 39 (2020).

[21] ———, *Guaranteed-quality higher-order triangular meshing of 2D domains*, ACM Trans. Graph., 40 (2021), pp. 1–14.

[22] S. Mann, *Surface approximation using geometric Hermite patches*, University of Washington, 1992.

[23] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2006.

[24] J. T. Oden, *Optimal hp finite element methods*, Comput. Methods Appl. Mech. Eng., 112 (1994), pp. 309–331.

[25] X. Qian, *Full analytical sensitivities in nurbs based isogeometric shape optimization*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 2059–2071.

[26] X. Qian and O. Sigmund, *Isogeometric shape optimization of photonic crystals via coons patches*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 2237–2255.

[27] M. Rabinovich, R. Poranne, D. Panozzo, and O. Sorkine-Hornung, *Scalable locally injective mappings*, ACM Trans. Graph. (TOG), 36 (2017), p. 16.

[28] X. Roca, A. Gargallo-Peiró, and J. Sarrate, *Defining quality measures for high-order planar triangles and curved mesh generation*, in Proceedings of the 20th international meshing roundtable, Springer, 2012, pp. 365–383.

[29] E. Ruiz-Gironés, J. Sarrate, and X. Roca, *Generation of curved high-order meshes with optimal quality and geometric accuracy*, Procedia engineering, 163 (2016), pp. 315–327.

[30] C. Sanderson and R. Curtin, *Armadillo: a template-based C++ library for linear algebra*, Journal of Open Source Software, 1 (2016), p. 26.

[31] P. Schmidt, J. Born, D. Bommes, M. Campen, and L. Kobbelt, *Tinyad: Automatic differentiation in geometry processing made simple*, Computer graphics forum, 41 (2022), pp. 113–124.

[32] T. Schneider, Y. Hu, J. Dumas, X. Gao, D. Panozzo, and D. Zorin, *Decoupling simulation accuracy from mesh quality*, ACM Transactions on Graphics, 37 (2018).

[33] J. Smith and S. Schaefer, *Bijective parameterization with free boundaries*, ACM Trans. Graph., 34 (2015), pp. 70:1–70:9.

[34] T. Toulorge, J. Lambrechts, and J.-F. Remacle, *Optimizing the geometrical accuracy of curvilinear meshes*, Journal of Computational Physics, 310 (2016), pp. 361–380.

[35] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal, *High-order CFD methods: current status and perspective*, Int. J. Numerical Methods in Fluids, 72 (2013), pp. 811–845.

[36] S. Xia and X. Qian, *Generating high-quality high-order parameterization for isogeometric analysis on triangulations*, Computer Methods in Applied Mechanics and Engineering, 338 (2018), pp. 1–26.

[37] S. Xia, X. Wang, and X. Qian, *Continuity and convergence in rational triangular Bézier spline based isogeometric analysis*, Computer Methods in Applied Mechanics and Engineering, 297 (2015), pp. 292–324.

[38] J. Yang, S. Liu, S. Chai, L. Liu, and X.-M. Fu, *Precise high-order meshing of 2D domains with rational Bézier curves*, Computer Graphics Forum, 41 (2022), pp. 79–88.

[39] M. Zareh and X. Qian, *C^1 triangular isogeometric analysis of the von Karman equations*, in Geometric Challenges in Isogeometric Analysis, Springer, 2022, pp. 365–384.

[40] M. Zlamal, *The finite element method in domains with curved boundaries*, Int. J. Numer. Methods Eng., 5 (1973), pp. 367–373.