Guaranteed-Quality Higher-Order Triangular Meshing of 2D Domains

MANISH MANDAD and MARCEL CAMPEN, Osnabrück University, Germany

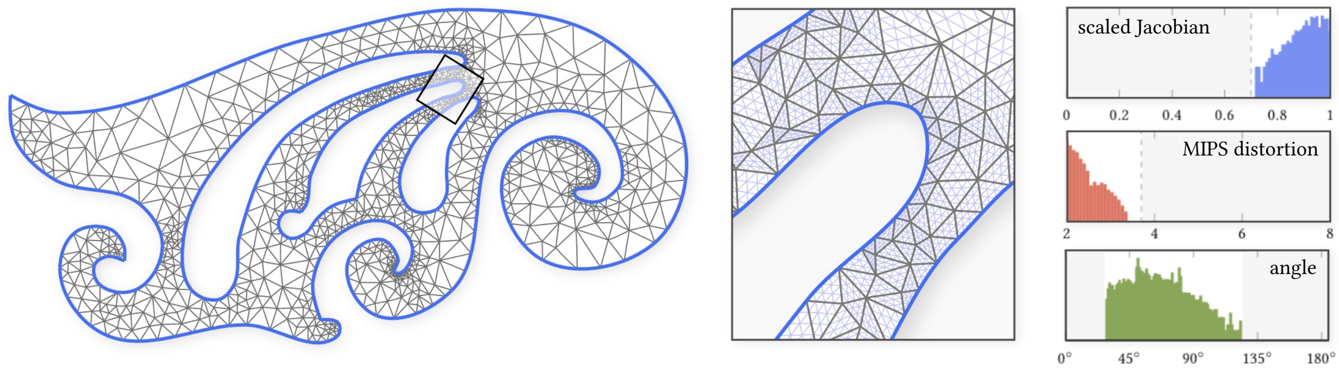


Fig. 1. Higher-order mesh of cubic polynomial triangle elements, precisely conforming to a given domain boundary. By construction our mesh generation algorithm guarantees that the scaled Jacobian quality measure respects a prescribed adjustable bound, here [0.7, 1.0], MIPS distortion of all elements (away from possibly prescribed sharp corners) likewise is bounded, here everywhere below 3.75, and, consequently, inner angles of all curved triangles are bounded as well. The histograms (log scale) show the distribution of these values over the entire mesh.

We present a guaranteed quality mesh generation algorithm for the curvilinear triangulation of planar domains with piecewise polynomial boundary. The resulting mesh consists of higher-order triangular elements which are not only regular (i.e., with injective geometric map) but respect strict bounds on quality measures like scaled Jacobian and MIPS distortion. This also implies that the curved triangles' inner angles are bounded from above and below. These are key quality criteria, for instance, in the field of finite element analysis. The domain boundary is reproduced exactly, without geometric approximation error. The central idea is to transform the curvilinear meshing problem into a linear meshing problem via a carefully constructed transformation of bounded distortion, enabling us to leverage key results on guaranteed-quality straight-edge triangulation. The transformation is based on a simple yet general construction and observations about convergence properties of curves under subdivision. Our algorithm can handle arbitrary polynomial order, arbitrarily sharp corners, feature and interface curves, and can be executed using rational arithmetic for strict reliability.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; **Mesh models**; • **Mathematics of computing** → **Mesh generation**.

Additional Key Words and Phrases: Bézier triangle, curvilinear mesh, higher-order mesh, minimal angle guarantee, scaled Jacobian, bounded distortion

ACM Reference Format:
Manish Mandad and Marcel Campen. 2021. Guaranteed-Quality Higher-Order Triangular Meshing of 2D Domains. *ACM Trans. Graph.* 40, 4, Article 154 (August 2021), 14 pages. https://doi.org/10.1145/3450626.3459673

Author's addresses: Manish Mandad, Marcel Campen, Institute for Computer Science, Osnabrück University, Germany.

1 INTRODUCTION

Triangulation is used for purposes of domain discretization in applications across many different fields. Often, the resulting meshes' *quality* is of relevance for subsequent computational tasks. In the field of finite element methods (FEM), for instance, meshes of low quality easily contribute to ill-conditioning of the system to be solved and to degradation of convergence rates [Babuska and Aziz 1976; Brandts et al. 2011; Fried 1972; Oswald 2015; Shewchuk 2002b; Vavasis 1996]. While poor mesh quality can be masked to some extent, in efforts to decouple simulation accuracy from mesh quality [Schneider et al. 2018], overall the problem of guaranteed-quality mesh generation remains relevant.

While the concrete notion of mesh quality is application dependent, a common key quality indicator—in the context of linear mesh generation, with straight-sided triangle elements—is the range of inner angles. While the conceptual optimum of an all-equilateral mesh cannot be achieved in general [Colin de Verdière and Marin 1990; Shewchuk 2002a], various mesh generation algorithms have been described that guarantee lower and upper bounds on these angles [Chew 1989, 1993; Ruppert 1995; Shewchuk 2002a].

In the field of higher-order mesh generation, however, the notion of quality becomes more intricate. In this case elements are triangular images defined by polynomial maps and exhibit curved edges. In contrast to the linear case, element quality cannot be assessed by the triangles' inner angles alone. The pointwise angular distortion of the map needs to be considered, for instance by means of the MIPS measure, which is relevant in this case [Schneider et al. 2018]. Furthermore, the maps' derivatives (*scaled Jacobian*) as well as the magnitude of higher-order derivatives are of importance particularly in FEM [Ciarlet and Raviart 1972b; Engvall and Evans 2020]. While existing higher-order mesh generation methods produce output of high quality in many cases, this cannot be relied on as they do not provide guarantees on any of these quality measures.

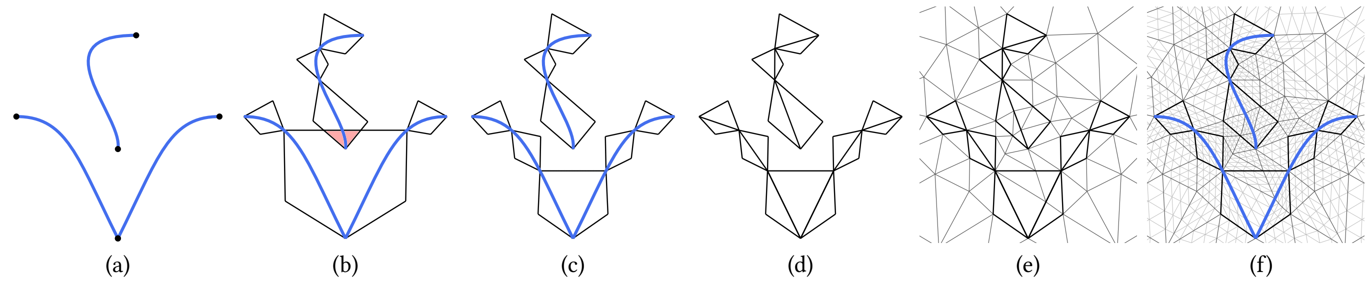


Fig. 2. Approach overview. Prescribed domain curves (a) are covered by envelope elements (b). These are refined so as to satisfy certain geometric properties including spatial disjointness (c). A set of straight line segments derived from these envelopes (d) are used as constraints in a constrained linear meshing problem (e). Each envelope element comes with a polynomial transformation map; it is applied to all triangles lying within the respective envelope element in the linear triangulation. In this way the mesh is transformed into a curve-conforming higher-order mesh (f). Its quality is guaranteed through a combination of the linear mesh's angle guarantees and the transformation maps' distortion properties. (Grey curves indicate isolines of the elements' geometric maps.)

Hence the problem of guaranteed-quality mesh generation largely remains open in the higher-order setting. This setting is of increasing relevance, in particular in the context of FEM and isogeometric techniques, in fluid simulation, animation, and analysis [Wang et al. 2013].

1.1 Contribution

In this paper we introduce a guaranteed-quality higher-order 2D mesh generation method. In particular, the method and its resulting meshes provably posses the following properties:

(1) Elements are injective polynomial triangles.
(2) Arbitrary polynomial order is supported.
(3) Elements conform precisely to curved domain boundaries.
(4) The scaled Jacobian measure is larger than ρ everywhere.
(5) MIPS distortion is smaller than μ away from domain corners.

The involved bounds ρ and μ are parameters that can be chosen within certain ranges. In principle, the scaled Jacobian can be driven arbitrarily close to 1. The desired MIPS distortion bound μ (relative to an ideal, equilateral element) can be set to any value ≥ 3.5. This also implies a minimal angle bound for all curved triangles in the output, away from sharp domain corners. The specific value of 3.5 is in accordance with the currently best minimal angle guarantees available in the field of linear mesh generation (cf. Sec. 5.1).

The output's complexity is sensitive to these parameter choices in the sense that extremely tight settings may lead to very dense meshes, whereas looser settings yield simpler results.

1.2 Approach

Figure 2 illustrates the main steps of our approach: First, we construct *envelopes*, consisting of a series of quadrilateral elements, for each input curve, completely covering it from both sides (Fig. 2b), overall enveloping these curves. As long as these envelope elements either intersect, mutually form corners too sharp, or contain curve pieces too complex, we replace them by smaller substitutes in a recursive bisection process (Fig. 2c). This process, provably, terminates.

Second, we consider a set of straight line segments defined by these envelope elements (Fig. 2d), and generate a high-quality linear triangulation constrained by these segments (Fig. 2e). This can be done using established constrained linear mesh generation methods that guarantee bounded angles [Miller et al. 2003; Ruppert 1995].

Finally, we exploit that, by our manner of construction, each envelope element comes with a polynomial *warp map*. We apply these to the generated triangles lying inside the envelopes, turning them into curved higher-order triangles. This yields a higher-order mesh precisely conforming to the domain curves (Fig. 2f). Through an interplay of distortion bounds (that we ensure for the warp maps by construction) and angle bounds (respected by the linear mesh), the quality of our resulting mesh is guaranteed, with strict bounds on angles and scaled Jacobians.

It is worth remarking that the recent Bézier Guarding method [Mandad and Campen 2020a] follows a similar idea, covering domain curves by explicitly constructed elements in combination with linear mesh generation. As a key difference, though, that method provides no quality guarantees beyond injectivity. In order to achieve quality guarantees, the method we propose differs fundamentally in its envelope construction and the use of warp maps, and requires different and additional convergence and termination arguments.

2 RELATED WORK

A vast amount of work has been spent on both linear and higher order mesh generation problems over the last few decades. We briefly discuss relevant work with a focus on techniques offering injectivity and quality guarantees on the meshes' elements. For a broader overview we refer to surveys [Cheng et al. 2012; Owen 1998; Wang et al. 2013] and literature reviews in recent works such as [Mandad and Campen 2020a; Turner et al. 2018].

2.1 Injectivity Guarantee

Mesh elements can be considered images of an ideal (*reference* or *master*) element under some deforming *geometric map*. This map's injectivity is a vital property in many applications. Generating a 2D conforming triangulation with injective *linear* elements is a long-solved task [de Berg et al. 2000; Fournier and Montuno 1984; Toussaint 1984]; injectivity boils down to ensuring a common orientation for all the triangular elements, i.e., excluding flips.

The problem of *higher-order* 2D triangulation, by contrast, is more involved since injectivity in this case is a per point rather than a per element issue. While methods have been proposed to test for injectivity of such elements [Dey et al. 1999; George and Borouchaki 2012; Gravesen et al. 2014; Hernandez-Mederos et al. 2006], only a few mesh generation methods can actually generate an output with guaranteed injective polynomial elements.

Methods in this field can generally be classified as indirect or direct [Dey et al. 1999]. Indirect methods generate a linear triangulation (easily with injectivity guarantee) followed by incrementally deforming (i.e., curving) the elements in order to achieve conformance with prescribed domain boundaries [Abgrall et al. 2014; Hu et al. 2019; Moxey et al. 2016; Toulorge et al. 2013]. When constraining the deformation to preserve injectivity, there is no guarantee that conformance will be achieved in all cases. Direct methods create elements with curved edges right away. In this case elements, however, come without geometric maps [Boivin and Ollivier-Gooch 2002], come with non-polynomial geometric maps [Gordon and Hall 1973; Haber et al. 1981], or require additional assumptions (e.g. regarding smoothness) on the input [Ciarlet and Raviart 1972a; Rangarajan and Lew 2014].

A recently proposed approach [Mandad and Campen 2020a] guarantees both, injectivity and conformance to the domain boundary. It, however, provides no further quality guarantees over the curvilinear elements. In the worst case, elements (while injective by construction) can be arbitrarily distorted, their inner angles be arbitrarily small. Higher-order remeshing techniques may often be able to improve the mesh in a post-process [Hu et al. 2019], but these do not provide any quality guarantees (beyond what their input already offers) either.

Our approach, by contrast, in addition to ensuring injectivity, provides guarantees on the quality of the mesh right away, with strict bounds on angles, MIPS distortion, and scaled Jacobians.

2.2 Quality Guarantee

Since the introduction of the first provably good conforming Delaunay refinement algorithm [Chew 1989], many algorithms have been proposed improving upon the guarantees on mesh quality, grading, and size [Bern et al. 1994; Chew 1993; Erten and Üngör 2009; Ruppert 1995; Shewchuk 2002a], relaxing input requirements, and improving the theoretical bounds [Miller et al. 2003; Pav 2003; Rand 2011a,b]. These methods are able to generate guaranteed quality meshes with bounds on angles, but are limited to generating conforming meshes of planar straight line graphs, i.e., meshes with linear elements aligned with piecewise linear domain boundaries.

While a few extensions to accommodate curved boundaries have also been proposed [Boivin and Ollivier-Gooch 2002; Gosselin and Ollivier-Gooch 2007; Pav and Walkington 2005; Rangarajan and Lew 2014], they provide guaranteed bounds only over the curved triangles' inner angles. The construction of valid injective polynomial geometric maps per triangular element (or even just ensuring their existence) with any kind of quality guarantee is not part of the consideration.

By contrast, our approach generates meshes exhibiting an injective higher-order polynomial map with guaranteed quality per element and can handle boundary, feature, and interface curves of arbitrary polynomial order without any smoothness requirements.

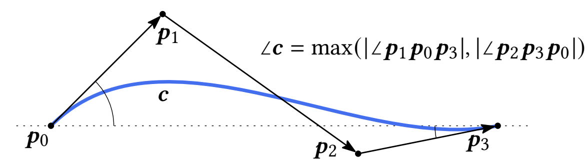


Fig. 3. A Bézier curve c with its 4 control points and its 3 control vectors. This curve's control angle $\angle c$ is defined as the larger of the two absolute angles formed by the curves' end tangents with the curve's *base line* $\overline{p_0 p_3}$.

3 HIGHER-ORDER MESH BASICS

In order to formulate our algorithm, we first introduce a few key definitions and basic constructions that we will leverage throughout our exposition. We represent all polynomials (for curves as well as triangular elements) in the Bernstein basis. Conversion to and from other commonly used bases, e.g., the triangular Lagrange basis, is of course possible.

3.1 Bézier Curves

Let $c : [0, 1] \to \mathbb{R}^2$ be a Bézier curve of order n, i.e., a polynomial curve represented in the Bernstein basis. Let its coefficients be the control points $(p_0, \ldots, p_n)$. The *control vectors* of this curve are the vectors $(p_1 - p_0, \ldots, p_n - p_{n-1})$ (Fig. 3). We assume throughout that curves are always (re)parameterized over $[0, 1]$ such that $c(0) = p_0$ and $c(1) = p_n$.

Definition 3.1 (Control angle). We define the control angle $\angle c$ of a curve c as $\angle c = \max(|\angle p_1 p_0 p_n|, |\angle p_{n-1} p_n p_0|)$, i.e., the larger of the two angles formed by the curves' end tangents with the curve's *base line* $\overline{p_0 p_n}$.

Definition 3.2 (Control width). We define the width $w(c)$ of a curve c as the distance between its first and last control point, i.e. $w(c) = \|p_n - p_0\|$.

Input Assumptions. Input to our method is a set of order n Bézier curves (domain boundaries and possibly feature curves); curves of mixed order are raised to common order n. We assume these are regular (non-vanishing derivative) and no two curves meeting at a joint form a zero angle. These assumptions are necessary requirements for curves to be part of any injective polynomial triangulation [Mandad and Campen 2020a]. In cases where the domain to be meshed is a bounded subset of $\mathbb{R}^2$, the angle criterion can be relaxed in the sense that zero-angles outside of the domain are ignored. We furthermore assume that the input curves intersect only at their end points, which can be ensured by splitting at other intersections.

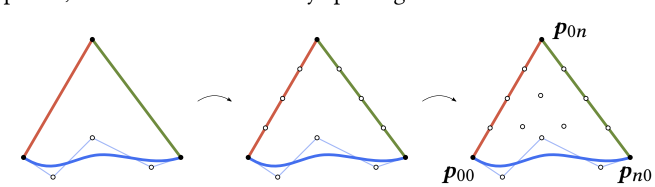
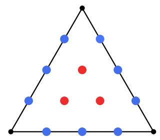


Fig. 4. Left: triangular region bounded by linear (red, green) and curved (blue) edges. Center: linear edges elevated to the same degree (here $n = 4$). Right: the inner control points are computed as affine combinations of outer control points via generalized barycentric coordinates (cf. Construction 3.1).

3.2 Bézier Triangles

Let $\boldsymbol{f}: \triangle \to \mathbb{R}^2$ be a Bézier triangle of order n, i.e., a bivariate polynomial in the simplicial Bernstein basis over some triangular domain $\triangle$. Let its coefficients be the control points $\{\boldsymbol{p}_{ij} \mid i \geq 0, j \geq 0, i+j \leq n\}$. Those control points $\boldsymbol{p}_{ij}$ with $i = 0$, $j = 0$, or $i+j = n$ we call outer control points; the remaining ones inner control points.

Construction 3.1 (Barycentric Extension)**.** Given outer control points for an order n Bézier triangle (Fig. 4 center), we define an extension to the interior by computing each inner control point's position as an affine combination of the outer control points. For an inner control point (red) the affine combination weights are determined as its generalized barycentric coordinates in a linear reference configuration. While various choices are available, empirically cotan coordinates [Pinkall and Polthier 1993] computed relative to the edge control points (blue) of an equilateral triangle elevated to the same degree (degree 4 in the inset example) produce results slightly favourable to alternatives like mean value coordinates for our purpose. It is important though that coordinates offering linear reproduction are chosen; this is exploited to guarantee certain convergence properties (cf. Prop. 4.3).



We will employ this construction for triangles with one or more straight edges (in addition to curved edges of polynomial degree n). In this case we impose a linear parametrization and elevate the degree to n, i.e., we use uniformly distributed control points along these straight edges. Fig. 4 illustrates this construction for a degree 4 example. The question of injectivity and distortion of such a higher-order triangle is considered in Sec. 4.1.

3.3 Quality Measures

We consider two main triangulation quality measures and provide adjustable bounds on these: the scaled Jacobian and MIPS distortion. This choice was made on the grounds that the so-called scaled Jacobian measure has been identified as the key measure relevant in the context of FEM [Engvall and Evans 2020], and the MIPS distortion (with respect to an ideal, equilateral element) is closely related to PDE solution error [Schneider et al. 2018].

3.3.1 Scaled Jacobian. The scaled Jacobian of a Bézier triangle f, following [Engvall and Evans 2020] and [Dey et al. 1999], is defined as $\frac{\min |\det J|}{\max |\det J|}$, where J is the Jacobian of map f, and min and max are computed over the entire triangular domain $\triangle$. When f is non-injective, its scaled Jacobian is 0; for a linear f it is 1. Note that the term scaled Jacobian is also used with different meaning, in particular when referring to a quality measure for quadrilateral and hexahedral elements [Knupp 2000].

3.3.2 MIPS. The MIPS distortion [Hormann and Greiner 2000] of f is defined pointwise as $\frac{\|J\|^2}{\det J}$. MIPS distortion is minimal ($= 2$) iff f is conformal (angle-preserving), and goes to infinity with increased angle distortion. Note that equivalent measures are in use under different names. For instance, the isotropy measure considered in [Johnen et al. 2016] is, up to a constant, simply the inverse of MIPS.

Our approach is quite flexible and further measures that might be relevant for particular applications could be taken into account

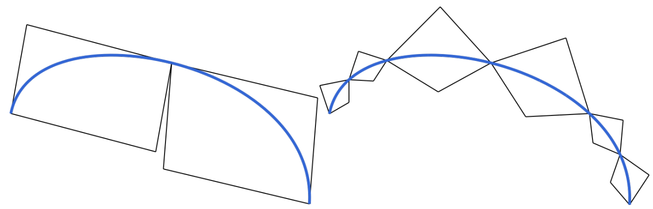


Fig. 5. Left: curved *guarding triangles* of [Mandad and Campen 2020a] along a given curve (blue). The triangles possess injective geometric maps by construction but their distortion is not considered. This results in arbitrarily sharp angles, not only inside but also in between such triangles. Right: our *envelope triangles*. Both inner and outer angles are bounded (here within $[30°, 120°]$) and all geometric maps satisfy prescribed distortion bounds.

under certain convergence conditions. For instance, the magnitudes of higher derivatives can be of relevance [Engvall and Evans 2020]; these could be taken into account as well. In particular, Propositions 4.3 and 4.5 easily extend to include the statement that these higher derivatives converge to 0 in our algorithm's framework, i.e., arbitrary upper bounds on these could be prescribed as well. To keep the exposition simple we focus on the above two main measures.

4 ENVELOPING THE CURVES

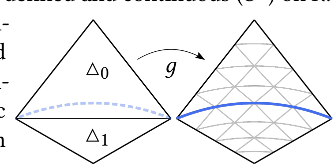
At the heart of our method is the construction of so-called envelopes, defining a spatial transformation in the vicinity of the input curves. This ultimately enables us to leverage linear mesh generation techniques, before transforming the linear mesh into a higher-order curved mesh satisfying all the desiderata.

4.1 Curve Envelopes

In this section, we introduce an algorithm to construct a series of quadrilateral elements enveloping a given curve $\boldsymbol{c}$ of degree n. Each such element has two opposite corners lying on the curve (Fig. 5 right). We associate each element R with a continuous *warp map* $g: R \to R$ that preserves the boundary of R. To this end we view the quadrilateral as union of two triangles $R = \triangle_0 \cup \triangle_1$, such that the splitting diagonal connects the two on-curve corners.

Definition 4.1 (Warp Map). On a triangle $\triangle_i$ of the envelope element R of curve $\boldsymbol{c}$ the warp map g is a higher-order triangular Bézier map of degree n. This map $g|_{\triangle_i}: \triangle_i \to \mathbb{R}^2$ is defined via barycentric extension (Construction 3.1) of the two non-diagonal edges of $\triangle_i$ and the curve c.

Notice that $\triangle_0$ and $\triangle_1$ overlap along the diagonal. Their respective warp maps agree on this diagonal (its common image is the curve $\boldsymbol{c}$), i.e., the combined map is well-defined and continuous (C^0) on R. Intuitively speaking, each quadrilateral envelope element is formed by the union of two curved triangles (with degree n geometric maps), one on each side of c, both conforming with the curve.



The idea to define an envelope element for a given curve is to choose the position of the *apex* (the off-curve corner) of each of the two triangles such that

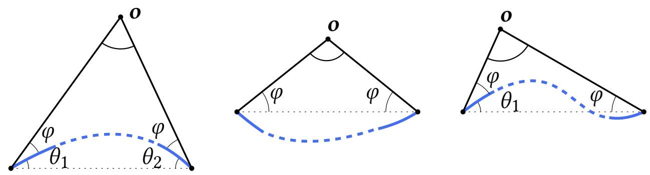


Fig. 6. Given angle parameter φ, the apex point $\boldsymbol{o}$ for a curve's side is chosen based on three different scenarios, depending on the curve's end tangents relative to the base line. From left to right it subtends an angle of $180^\circ - 2\varphi - \theta_1 - \theta_2$, $180^\circ - 2\varphi$, and $180^\circ - 2\varphi - \theta_1$, respectively. This angle is at least φ (and therefore the apex valid) as long as $\theta_1 + \theta_2 \leq 180^\circ - 3\varphi$ (left), $0 \leq 180^\circ - 3\varphi$ (center) or $\theta_1 \leq 180^\circ - 3\varphi$ (right).

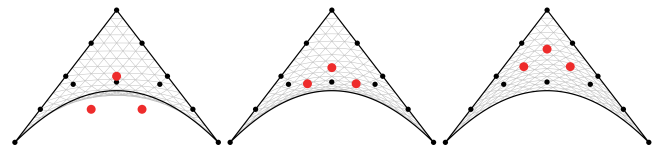


Fig. 7. Left: computing the inner control points (red) as weighted corner averages based on the rule $\boldsymbol{p}_{ij} = \frac{k}{n}\boldsymbol{p}_{00} + \frac{i}{n}\boldsymbol{p}_{n0} + \frac{j}{n}\boldsymbol{p}_{0n}$ here results in a non-injective map. Middle: computing them via barycentric extension leads to lower distortion, here MIPS distortion 4.5. Right: By (non-linearly) optimizing the inner control points one could reduce it even further (here down to 3.8) albeit at a significantly higher cost.

- warp map g is injective (and thus a homeomorphism $R \to R$) and satisfies bounds on its distortion,
- corner angles formed within and between envelope elements respect a lower bound,
- envelope elements are mutually disjoint.

The latter two properties are required to enable a high quality linear triangulation aligned to these envelopes in a subsequent step of our overall algorithm (cf. Sec. 5.2), as illustrated in Fig. 2e.

It is easy to see that a curve cannot generally be covered by a single envelope element satisfying these requirements. Instead, we use a series of such elements. To this end the curve is bisected recursively in an adaptive manner until each sub-curve permits a single proper envelope element. Therefore, the key is to design the envelope element construction such that under bisection the above requirements will eventually be satisfied in any case.

Construction 4.1 (Apex point)**.** Given a curve $\boldsymbol{c}$ with control points $(\boldsymbol{p}_0, \ldots, \boldsymbol{p}_n)$ and angle parameter φ, a valid apex point $\boldsymbol{o}$ for an envelope triangle is one such that all three inner angles of the curved triangular region formed by segments $\overline{\boldsymbol{p}_n\boldsymbol{o}}$, $\overline{\boldsymbol{o}\boldsymbol{p}_0}$, and curve $\boldsymbol{c}$ are in the range $\Phi = [\varphi, 180^\circ - 2\varphi]$, i.e.,

$$\angle \boldsymbol{p}_n\boldsymbol{o}\boldsymbol{p}_0 \in \Phi, \quad \angle \boldsymbol{o}\boldsymbol{p}_0\boldsymbol{p}_1 \in \Phi, \quad \angle \boldsymbol{p}_{n-1}\boldsymbol{p}_n\boldsymbol{o} \in \Phi.$$

Among all valid apexes $\boldsymbol{o}$ we choose one that forms a small envelope triangle. Concretely, at the curve's end points, $\boldsymbol{p}_0$, $\boldsymbol{p}_n$, we rotate by angle φ and $-\varphi$, respectively, either the curve's end tangent or the curve's base line, depending on their relative position (see Fig. 6). The intersection point of these two rotated lines is chosen as apex.

We discuss the choice of the parameter φ (dependent on the desired output mesh quality bounds) in Sec. 5.1. Note that not for all $\boldsymbol{c}$ and φ a valid apex exists; the above intersection point may not exist or may lie on the curve's wrong side. However:

Proposition 4.2 (Apex Existence). *For any $\varphi < 60^\circ$ repeated bisection of a curve eventually yields sub-curves that all have valid apexes on both sides.*

Proof. This follows from convergence of control polygons to a flat state under repeated bisection [Li et al. 2012; Morin and Goldman 2001]. Concretely, in Fig. 6 the angles θ_i converge to 0, enabling an isosceles triangle $\boldsymbol{p}_0\boldsymbol{p}_n\boldsymbol{o}$ with base angles φ in the limit. □

We note that a similar adaptive enveloping idea was used in [Mandad and Campen 2020a], based on *guard triangles* (see Fig. 5). In that case the focus, however, is purely on injectivity; angles and distortion can be arbitrarily extreme. In our construction, by using a different construction for the apex as well as for the inner control points, we guarantee that all the inner and outer angles of envelopes as well as the warp maps' distortion are bounded.

With this definition at hand we can now show the following.

Proposition 4.3 (Envelope Distortion). *Under repeated bisection of a curve the envelopes' warp maps' distortion behaves as follows: the scaled Jacobian converges to 1 from below and the MIPS distortion converges to 2 from above.*

Proof. Under repeated bisection each sub-curve's control polygon does not only converge to a flat state [Li et al. 2012; Morin and Goldman 2001] but also to same-length control vectors, i.e., control points are equidistant in the limit (Appendix A). The limits of the Bézier triangles forming the envelopes therefore are linear isosceles triangles with base angles φ, isometric to their respective domain triangle $\triangle_i$. Due to being linear, the Jacobian is constant, the scaled Jacobian measure is 1. Due to being isometric, the MIPS distortion is 2. □

Note that a positive scaled Jacobian implies $\det J \neq 0$, therefore a locally injective warp map. As a curve intersecting its own envelope's edges implies $\det J = 0$ somewhere, this extends to global injectivity due to the envelope triangle's non-intersecting boundary.

We note that one may choose the inner control points defining the warp maps (Def. 4.1) based on other rules than the barycentric extension from Construction 3.1, for instance, simply as weighted averages of the corner points. The only requirement is convergence to a linear map as the edges become linear. We opted for the barycentric extension as it empirically generates maps with significantly lower distortion, at a low cost, as illustrated in Fig. 7.

Proposition 4.4 (Outer Angles). *Under repeated bisection of a curve the envelope's outer angles (between adjacent envelope elements) converge to $180^\circ - 2\varphi$.*

Proof. The outer angle β is related to φ and the sub-curves' control angles as follows: $\beta \geq 180^\circ - 2\varphi \pm \angle c_1 \pm \angle c_2$ (as illustrated in Fig. 8). The signs depend on the curve being convex or concave at the respective point. Under repeated bisection, both $\angle c_1$ and $\angle c_2$ converge to 0 (cf. Prop. 4.2), resulting in the stated limit. □

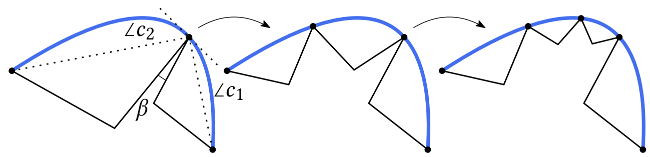


Fig. 8. The outer angle between our envelopes along a curve are related (in the worst case) via $2\varphi + \angle c_1 + \angle c_2 + \beta = 180°$, where $\angle c_1$ and $\angle c_2$ denote the two sub-curves' control angles (Def. 3.1). Repeated bisection eventually ensures that all outer angles are in the range $[\varphi, 180° - 2\varphi]$.

4.2 Corner Envelopes

At joints, i.e., points where two or more curves meet and intersect at their end points, a different treatment is required whenever a corner of angle $\phi < 3\varphi$ is formed. This is because the outer angle bound of Prop. 4.4 only holds between envelope elements of one curve (or of C^1-continuous curves). At corners of angle $\phi < 2\varphi$ the two curves' envelopes would overlap regardless of bisection level. For $2\varphi \leq \phi < 3\varphi$ they might not, but they would form an outer angle smaller than φ, adversely affecting the quality of the linear triangulation (Sec. 5.2). Even for corners somewhat larger than 3φ, a lot of curve bisection may be necessary to reach satisfactory outer angles. We therefore conservatively treat corners of angle $\phi < 4\varphi$ differently, avoiding excessive refinement.

At such corners we construct special *corner envelopes*. Intuitively, the two curve envelope triangles in a corner are replaced by one shared triangle—equipped with a Bézier map that conforms to *two* curved edges and one straight segment (Fig. 9).

Construction 4.2 (Corner Envelope)**.** We intersect the two curves $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ forming a corner with a circle of radius r around the common end point $\boldsymbol{p}_0$ (Fig. 9 center). Such a circle intersection was used previously to handle small input angles, in linear and curved meshing [Boivin and Ollivier-Gooch 2002; Ruppert 1995]. The two intersection points $\boldsymbol{q}_1$, $\boldsymbol{q}_2$ and the joint point $\boldsymbol{p}_0$ form a triangle $\triangle$ over which we define, in analogy to Definition 4.1, a warp map g. It is built via barycentric extension of the two curved edges formed by $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ between $\boldsymbol{p}_0$ and $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, and the straight segment $\overline{\boldsymbol{q}_1\boldsymbol{q}_2}$.

Proposition 4.5 (Corner Envelope Distortion). *As $r \longrightarrow 0$ the corner envelope's warp map's distortion behaves as follows: the scaled Jacobian converges to 1 from below, the MIPS distortion converges to 2 from above. Further, the outer angles (α in Fig. 9) converge to $90° + \phi/2$.*

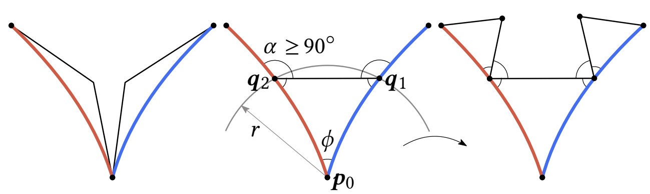


Fig. 9. Left: when each curve is enveloped individually (as in [Mandad and Campen 2020a]) in corners this implies either overlaps or small angles. Center: we treat such corners specially, with leverage on both interior and exterior angles (Construction 4.3). In the limit of $r \longrightarrow 0$ outer angles α are $\geq 90°$. Right: as a result, the (possibly sharp) corner angle does not degrade further and the remainder of the curves can be enveloped as before (Sec. 4.1).

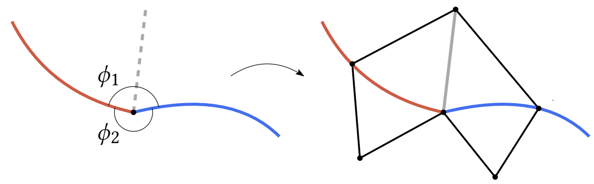


Fig. 10. If corner angle ϕ_1 between two curves is larger than $180° - 2\varphi$, we bisect it by introducing a virtual curve (dashed). The joint can then be enveloped as in Sec. 4.2. In case the angle (ϕ_2) is larger than 4φ, no special treatment is required; we can envelope the curves' sides following Sec. 4.1.

Proof. The proof of distortion limits is analogous to that of Proposition 4.3: the limit is a linear isosceles triangle with, in this case, base angles $90° - \phi/2$ (and third angle ϕ). As a consequence, the outer angles converge to $90° + \phi/2$. □

The remainder of the curves (beyond $\boldsymbol{q}_i$) is then enveloped as described in Sec. 4.1 (Fig. 9 right).

Complex Joints. When more than two curves meet at one joint, forming multiple corners, a common radius r is used (initialized to the minimum of incident curves' control widths) in the corner envelope constructions so as to yield conforming elements.

Obtuse Corners. At corners of angle $\phi < \varphi$ (where φ is the parameter used in Construction 4.1), the envelopes exhibit inner angles smaller than those of curve envelopes in the limit. This is inevitable given the prescribed sharp corner. At corners of angle $\phi \geq 180° - 2\varphi$ the use of a single corner envelope triangle would likewise imply smaller angles. This, by contrast, can be avoided. To this end, we split corners with $\phi \geq 180° - 2\varphi$ but $\phi < 4\varphi$ in advance by inserting short virtual bisecting curves (Fig 10).

4.3 Numerics

Many of the operations required to implement the above curve and corner envelope construction (such as curve bisection, barycentric extension) are rational calculations. This opens up possibilities to implement the method using exact rational arithmetic if desired—so as to avoid any potential numerical issues arising due to the limited precision of floating point arithmetic. Two operations, the apex construction and the circle intersection, however, are not generally rational and therefore require special attention to enable this.

The apex (Construction 4.1) is constructed by intersecting two lines (the base line or the end tangents, depending on the three different scenarios illustrated in Fig. 6) that are rotated by φ. By instead of φ using $\tan\varphi$ as parameter, and setting it to a rational value, the lines' rotation and intersection can be computed without numerical inconsistencies in the rational numbers.

For corner envelopes (Construction 4.2) we need intersection points between circles and polynomial curves. These are irrational in general. We can, however, use curve points with rational coordinates as substitutes instead, as long as they are sufficiently close to the true intersection points. Concretely, we use a curve point $\boldsymbol{c}(t)$, for rational parameter t, and let t converge to the true circle intersection parameter in the rationals, until angle conditions are met that are relevant for the corner envelope construction in Sec. 4.2:

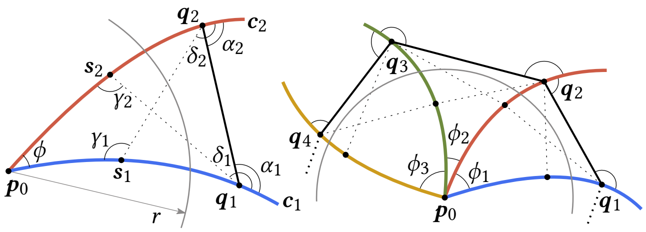


Fig. 11. Left: Two curves $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ meeting at their end point $\boldsymbol{p}_0$ at an angle ϕ. $\boldsymbol{s}_i$ and $\boldsymbol{q}_i$ refer to points on the curve $\boldsymbol{c}_i$, chosen such that $\|\boldsymbol{p}_0 - \boldsymbol{s}_i\| < r \leq \|\boldsymbol{p}_0 - \boldsymbol{q}_i\|$. All marked angles ($\phi$, α, δ, γ) are with respect to the curves' tangents at the corresponding points. Algorithm 1 tightens the points $\boldsymbol{s}_i$, $\boldsymbol{q}_i$ (and reduces r if necessary) until the outer angles α_i satisfy $90° \leq \alpha_i \leq 180° - \varphi$, as desired. Right: The same algorithm can also be used to envelope a joint where multiple curves meet by sharing $\boldsymbol{s}_i$, $\boldsymbol{q}_i$ and r.

Algorithm 1: Rational Corner Envelope

Function *CornerEnvelope*($\boldsymbol{c}_1, \boldsymbol{c}_2, \varphi$):
  while *true* **do**
1   **if** $\gamma_1 \leq \varphi \vee \gamma_2 \leq \varphi \vee \delta_1 \leq 90° \vee \delta_2 \leq 90°$ **then**
2      $r^2 \leftarrow r^2/2$
3      $\boldsymbol{c}_1.t_s \leftarrow 0$
4      $\boldsymbol{c}_2.t_s \leftarrow 0$
5   **else if** $\alpha_1, \alpha_2 \notin [90°, 180° - \varphi]$ **then**
6      $\boldsymbol{c}_1$.TIGHTEN_T(r^2)
7      $\boldsymbol{c}_2$.TIGHTEN_T(r^2)
    else
      return ($\boldsymbol{q}_1 = \boldsymbol{c}_1(\boldsymbol{c}_1.t_q)$, $\boldsymbol{q}_2 = \boldsymbol{c}_2(\boldsymbol{c}_2.t_q)$)
Function *TIGHTEN_T*(r^2):
  $t \leftarrow \frac{t_s + t_q}{2}$
  if $\|\boldsymbol{p}_0 - \boldsymbol{c}(t)\|^2 \geq r^2$ **then**
    $t_q \leftarrow t$
  else
    $t_s \leftarrow t$

Construction 4.3 (Rational Corner Envelope)**.** Given two curves $\boldsymbol{c}_1 = (\boldsymbol{p}_0, \boldsymbol{p}_1^1, \ldots, \boldsymbol{p}_n^1)$ and $\boldsymbol{c}_2 = (\boldsymbol{p}_0, \boldsymbol{p}_1^2, \ldots, \boldsymbol{p}_n^2)$ intersecting only at their common end point $\boldsymbol{p}_0$, we define a pair of parameters ($t_s^i = 0, t_q^i = 1$) for each curve $\boldsymbol{c}_i$, ($i = 1, 2$). Let $\boldsymbol{s}_i = \boldsymbol{c}_i(t_s^i)$ and $\boldsymbol{q}_i = \boldsymbol{c}_i(t_q^i)$ be the respective curve points (Fig. 11 left). We initialize radius $r^2 = \min\{w(\boldsymbol{c}_1), w(\boldsymbol{c}_2)\}^2$ (cf. Def. 3.2). Using Algorithm 1, the two ranges $[t_s^i, t_q^i]$ are then iteratively tightened around the true intersection parameter associated with the circle of radius r. Upon termination, the points $\boldsymbol{q}_1, \boldsymbol{q}_2$ are suitable rational substitutes:

Proposition 4.6. *Assuming corner angle $\phi < 180° - 2\varphi$ (which we ensure, Sec. 4.2), Algorithm 1 terminates. Upon termination, the outer angles α_i are obtuse, the opposite inner angles are $\geq \varphi$.*

Proof. As $r \longrightarrow 0$, the sub-curves contained in the circle converge to straight lines. Consequently, the first condition (line 1) must become false below some $r > 0$, regardless of t_s, t_q. Therefore the first conditional block (r^2-halving, lines 2-4) will only be executed a finite number of times. The second conditional block (lines 6-7) tightens $\boldsymbol{s}_i$ and $\boldsymbol{q}_i$ around the true circle intersection point. Notice that under this tightening δ_i converges to α_i from above, and $180° - \gamma_i$ converges to α_i from below. This implies that repeated execution of this tightening will eventually lead to either $\alpha_i \in [90°, 180° - \varphi]$ (in which case it terminates, line 5) or to violation of the first condition (line 1)—which can only occur a finite number of times. □

The angle conditions in lines 1 and 5 can be checked exactly based on the rational $\tan\varphi$ parameter and simple dot products.

The algorithm readily extends to handle multiple corners (at joints where more than two curves meet) simultaneously (Fig. 11 right).

5 GUARANTEED-QUALITY CURVED TRIANGULATION

Equipped with the definitions and algorithmic components introduced above, we can now formulate our overall algorithm.

5.1 Input

The input to our algorithm is a set of 2D polynomial curves, satisfying conditions stated in Sec. 3.1, and parameters ρ and μ corresponding to the desired bounds the output triangulation shall respect in terms of scaled Jacobian and MIPS distortion, respectively.

In accordance with the best lower angle bound that is currently offered by linear constrained mesh generation techniques [Rand 2011b], we fix the parameter $\varphi = 28.6°$ in our envelope construction. This ensures that the envelopes, away from sharp curve corners, do not form any smaller (neither inner nor outer) angles. In this way the quality of the linear envelope-constrained triangulation we make use of is not forced to deteriorate. In case of future improvements in this field, φ could readily be adjusted (up to 45°) accordingly.

We denote the MIPS distortion (relative to an equilateral element) associated with the worst case triangle that could occur in a linear triangulation with minimal angle bound φ as μ_φ. For the above angle bound we have $\mu_\varphi = 3.4915... \approx 3.5$ (Appendix B). The desired bound μ on the curved output mesh can be set to any value larger than μ_φ. One of the key responsibilities of our algorithm then is to ensure that the distortion of the envelopes' warp maps g is below some threshold μ_g. In Sec 5.4 we discuss how this threshold needs to be set, dependent on μ_φ, so as to guarantee bounded distortion μ in the output—in addition to respecting the other quality bounds.

5.2 Complete Algorithm

Given a set of input curves and quality bounds we perform the following steps (cf. Fig. 2):

(1) Construct corner envelopes (Construction 4.2 or 4.3) and split the curves at the chosen corner envelope vertices.
(2) While there is a (sub-)curve such that any of the following conditions hold, bisect it. In the case of a corner curve, instead recompute its corner envelopes starting from a halved radius.
  (a) valid apex points do not exist (Construction 4.1).
  (b) an outer angle between its envelope and an envelope of a neighbor curve (with smaller control angle) is $< \varphi$ (Fig. 12).
  (c) the MIPS distortion of its envelope's warp map is $> \mu_g$.
  (d) the scaled Jacobian of the warp map is not bounded by ρ.
  (e) its envelope element intersects another envelope element of equal or smaller size (area).
(3) Compute angle-bounded linear mesh constrained by the segments (including diagonals) of all envelope elements.
(4) Obtain the output higher-order mesh by applying the warp map to all the linear triangles lying inside the envelopes.

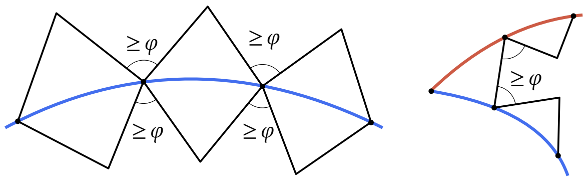


Fig. 12. Step (2b) of our algorithm ensures that (besides the envelopes' inner angles) the angles between two consecutive curve envelopes along a curve (left), and the angle between a corner envelope and a curve envelope (right) are at least φ. This ensures that the lower angle bound of φ can be respected (away from sharp corners) in the linear mesh.

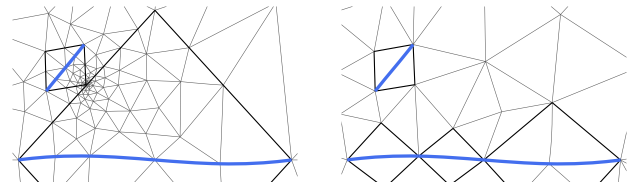


Fig. 13. Left: non-intersecting but close-by envelopes may result in very small local feature size resulting in very dense meshes. Right: through curve bisection the local feature size is increased between such envelopes, overall resulting in a simpler output mesh.

In the final step (4), for a mesh triangle that corresponds to an entire envelope triangle, applying the warp map simply means replacing it by this envelope triangle's warp map (which is a Bézier triangle, see Def. 4.1). For a mesh triangle that is just a part of an envelope triangle an explicit Bézier representation of its warped counterpart can be obtained as a part of this warp map Bézier triangle through subdivision (Appendix C). Note that the resulting mesh has straight edges outside and, in general, curved edges inside the envelopes (see, e.g., Fig. 26 top right).

The result is a mesh conforming to each curve from both sides. If the input curves form a closed domain, exterior triangles can be discarded in the end, or—more efficiently—already the envelope construction be restricted to the interior, as done in Figs. 1, 20, 25.

5.2.1 Strict Bounded Distortion Tests. In steps (2c) and (2d), we need to test whether a given higher-order map (in the form of a Bézier triangle) satisfies certain distortion bounds. In contrast to the linear setting, where the Jacobian is constant, in a higher-order triangle distortion varies pointwise. To ensure that the input bounds are respected everywhere, one can exploit the convex hull property of the Bernstein basis to test this conservatively. For the scaled Jacobian this has been spelled out before [Engvall and Evans 2020].

We exploit this property to additionally devise a conservative test for the upper bound on MIPS distortion, i.e., $\max(\|J\|^2/\det J) \leq \mu_g$ over the domain. We make use of the relation

$$\max \frac{\|J\|^2}{\det J} \leq \frac{\max \|J\|^2}{\min \det J} \leq \mu_g$$

and test for the second inequality. Each entry of the Jacobian matrix is a polynomial of degree $n-1$. Hence, both $\|J\|^2$ and $\det J$ are polynomials of degree $2(n-1)$. By expressing them in the Bernstein basis, owing to the convex hull property the numerator's maximum (the denominator's minimum) over the triangular domain is bounded from above (below) by the maximum (minimum) of its coefficients. The coefficient computation for the denominator is spelled out in [Mandad and Campen 2020b], for the numerator in Appendix D.

By virtually subdividing the domain triangle, these two bounds can be tightened [Leroy 2008], reducing unnecessary curve bisections. We choose to adaptively use up to 10 levels of subdivision before declaring a map possibly violating the distortion bound.

5.2.2 Envelope Intersection Test. In step (2e), to avoid a naive pairwise test for envelope element intersection, an obvious approach is to employ a spatial search data structure, like an interval tree of the envelopes' bounding boxes. This is also done in the implementation provided by the authors of [Mandad and Campen 2020a], where mutual intersections of envelope-like elements (called *guards*) need to be determined.

We propose an even more efficient acceleration tailored to our setting, using constrained Delaunay triangulation (CDT). We initially build a CDT of all input curves' end points. Each curve keeps pointers to its two corresponding CDT vertices. During the algorithm, we then add, one by one, the envelope elements' segments (using the vertex pointers of its curve for constant-time location) as constraints to the triangulation [Boissonnat et al. 2000]. If a conflict is discovered, i.e., as soon as two constraint segments intersect, curve bisection (step (2)) can be performed, the outdated constraint segments be removed from the CDT, and the algorithm can proceed.

As a demonstration of the significant benefit, we modified the implementation of [Mandad and Campen 2020a] to use this CDT approach instead of a bounding box tree. The run time for this slowest step of their algorithm improved by a large factor, see Table 1.

As an additional benefit, the final CDT can be re-used as initialization for step (3)—which then reduces to improving the angles of this CDT triangulation via incremental Delaunay refinement.

Table 1. Comparison of timing (in milliseconds) for Step 2 (ensuring disjoint triangles) from [Mandad and Campen 2020a, §5.5] between the original approach and our CDT approach (Sec 5.2.2), on three example inputs of increasing complexity.

Number of Curves =	10	100	1000
Original (using AABB Tree)	20.8	411.3	6010.6
Ours (using CDT)	4.0	61.7	400.7

5.2.3 Envelope Margins. The complexity of the angle-bounded linear mesh created in step (3) depends on the constraint segments' local feature size. There can be cases where two envelopes are very close, implying a small local feature size, leading to high mesh density. We can avoid this by performing the intersection test, step (2e), between non-adjacent envelope elements *with a margin*. The effect is illustrated in Fig. 13. This margin must be relative to the elements' size so as to maintain termination properties. Concretely, we check whether the distance between two envelope elements is smaller than a factor λ of their largest segment length—but ignore the case where this minimum distance is attained between two on-curve vertices of these envelopes, as these are invariant to further bisection. A choice of $\lambda = 0.2$ is used in all of our experiments.

5.3 Termination

Termination of step (1) (initial corner envelope construction) follows directly from Prop. 4.6.

Step (2) terminates because all conditions that trigger further curve bisection (or corner radius reduction) will no longer be met after a certain level of bisection. For condition (a) this follows directly from Prop. 4.2, for condition (b) from Prop. 4.4 and Prop. 4.5/4.6. For the distortion conditions (c) and (d) Prop. 4.3 and Prop. 4.5 provide the respective guarantees. Intersections considered by condition (e) will vanish because two adjoining envelopes (along a curve) do not intersect (except at their shared vertex) due to Prop. 4.4 and because bisecting curves or decreasing the corner envelope radius reduces the envelopes' size; eventually they will be smaller than the local feature size between any two disjoint curves and therefore non-intersecting. By, in case of conflicting envelopes, choosing for bisection the curve with larger control angle (2b) or larger envelope area (2e), it is ensured that all curves involved in violations will eventually be bisected.

Step (3) involves using existing linear constrained meshing algorithms, such as [Miller et al. 2003], which come with convergence guarantees. As a result, the overall algorithm will terminate and the output will satisfy the expected properties by construction.

5.4 Quality Guarantees

While it is obvious that all the desired quality bounds are satisfied by the linear triangulation with angle bound φ generated in step (3) (which has scaled Jacobian 1 and MIPS distortion no larger than μ_φ away from sharp corners), we still need to show that they are respected after applying the warp maps to triangles inside envelopes in step (4).

The Bézier map defining a warped triangle of the final mesh (whose distortion properties we are interested in) can be viewed as the composition $g \circ \ell$ of a linear map ℓ from an ideal equilateral triangle to the linear mesh triangle (an envelope triangle or part thereof) with the warp map g, as illustrated in Fig. 14.

As the linear map ℓ has a constant Jacobian, the composite map adopts the scaled Jacobian of the warp map g—which is bounded due to condition (d) in algorithm step (2). Regarding MIPS distortion: that of ℓ is bounded by μ_φ (except at sharp corners) due to the linear triangulation minimal angle bound (Appendix B), that of g by μ_g due to condition (c) in algorithm step (2). We now derive how the threshold μ_g needs to be chosen such that the MIPS distortion of the composite map is bounded by μ as prescribed.

5.4.1 Choice of Warp Limit μ_g. Let $\sigma_1 \geq \sigma_2$ be the singular values of the Jacobian of overall map $g \circ \ell$. Bounded MIPS distortion in the output mesh then means that this is supposed to respect $\sigma_1/\sigma_2 + \sigma_2/\sigma_1 \leq \mu$. This is equivalent to

$$\frac{\sigma_1}{\sigma_2} \leq \frac{\mu + \sqrt{\mu^2 - 4}}{2}.$$

Similarly, let $\sigma_1^\ell \geq \sigma_2^\ell$ and $\sigma_1^g \geq \sigma_2^g$ be the singular values corresponding to the linear and the warp map, respectively. Clearly, $\sigma_1 \leq \sigma_1^\ell \sigma_1^g$ and $\sigma_2 \geq \sigma_2^\ell \sigma_2^g$, and therefore $\sigma_1^g/\sigma_2^g \geq \sigma_1\sigma_2^\ell/\sigma_2\sigma_1^\ell$. Together, this yields that $g \circ \ell$ has MIPS distortion $< \mu$ if (but not only if) the

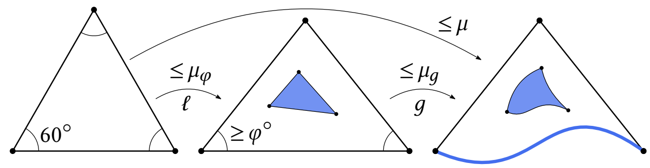


Fig. 14. The overall geometric map of a curved triangle is a composite of linear map ℓ from an equilateral triangle to a linear mesh triangle (center, white or blue) with warp map g. Due to angle bounds in the linear triangulation (center), MIPS distortion of ℓ is at most μ_φ. By construction of the envelopes, MIPS distortion of g is bounded by μ_g. We choose μ_g such that the distortion of the overall map is guaranteed to be bounded by μ.

MIPS distortion of g is bounded by:

$$\mu_g = \frac{\mu + \sqrt{\mu^2 - 4}}{\mu_\varphi + \sqrt{\mu_\varphi^2 - 4}} + \frac{\mu_\varphi + \sqrt{\mu_\varphi^2 - 4}}{\mu + \sqrt{\mu^2 - 4}},$$

i.e., based on μ, this is the parameter we use in condition (c) of step (2).

At sharp curve corners of angle $\phi < \varphi$ formed by input curves, the constrained linear triangulation of course inevitably contains inner angles smaller than φ. Using the μ_ϕ values corresponding to the concrete corner angles instead of μ_φ in the above formula, one can derive stricter individual bounds μ_g for each corner envelope's warp map, essentially for compensation—within certain limits of course: at corner angles ϕ where μ_ϕ already is larger than μ this would ask for $\mu_g < 2$, which cannot be achieved.

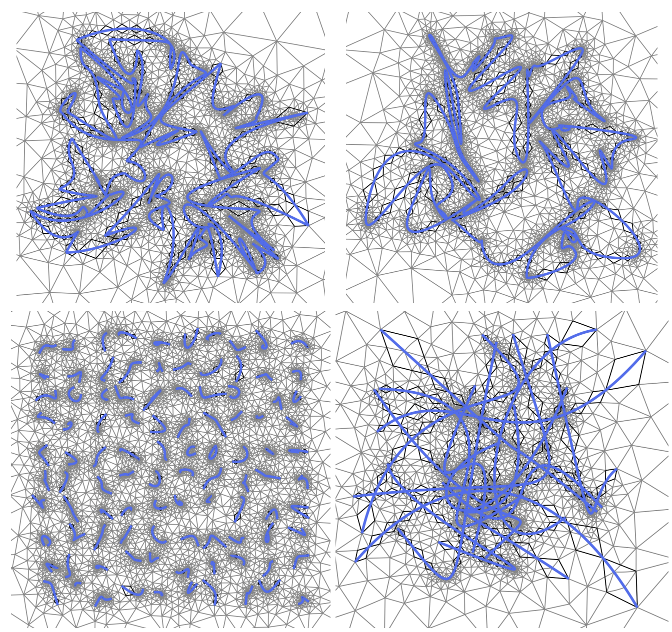


Fig. 15. Our method's output for $\mu = 10.0$, $\rho = 0.5$ on examples from the ABCD datasets of challenging random curve configurations [Mandad and Campen 2020a]. Left to right, top to bottom: A: C^0-domains, B: C^1-domains, C: isolated curves, D: curve networks with numerous corners.

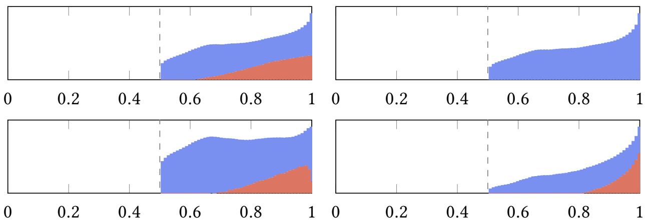


Fig. 16. Histograms (log scale) of scaled Jacobian over result meshes' curved triangles for datasets A, B, C, D (left to right, top to bottom) for bound $\rho = 0.5$. We distinguish values of triangles in curve envelopes (blue) and in corner envelopes (red).

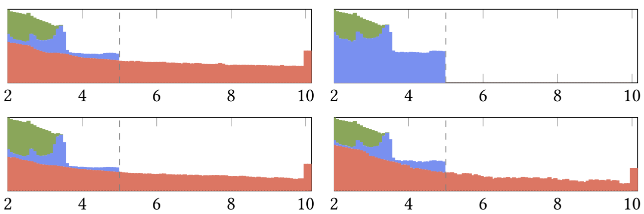


Fig. 17. Histograms (log scale) of MIPS distortion over the result meshes for datasets A, B, C, D (left to right, top to bottom) for bound $\mu = 5.0$. The red, blue, and green colors correspond to values on triangles belonging to corner envelopes, curve envelopes, and the rest of the domain, respectively. Notice that only values on corner triangles (red) lie beyond the bound.

Curved Angle Bounds. Note that since all angles in the linear triangulation are bounded by φ (except at sharp curve corners) and the angle distortion (MIPS) of warp maps is bounded by μ_g, the final curved triangles have bounded angles as well (Appendix E).

6 RESULTS

We test our method on the four datasets (ABCD) of 1000 configurations of input curves each, with varying characteristics, from [Mandad and Campen 2020a]. Fig. 15 shows example results on one case from each dataset.

Validation. In Fig. 16 we show histograms of the scaled Jacobian accumulated over output meshes obtained when applying our method to the ABCD datasets. It can be observed that the generated higher-order meshes indeed do not contain any elements with a scaled Jacobian value below the set bound of $\rho = 0.5$.

In Fig. 17 analogously histograms of the MIPS distortion are shown. As expected, in those datasets that contain sharp corners of angle $< \varphi$ (datasets A, C, and D) there are MIPS values beyond the set bound of $\mu = 5.0$, but those are strictly confined to triangles in corners (red) where this is inevitable.

Timing. In Table 2 a breakdown of run times relative to the choice of μ is given. It can be observed that for very high quality requirements (μ close to the limit of 3.5), the adaptive envelope refinement until warp maps are of sufficiently low distortion accounts for the largest share. For loser bounds run time decreases quickly, and the envelope intersection test (2e) remains as dominating item.

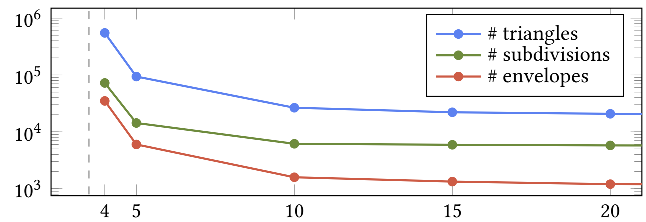


Fig. 18. MIPS distortion bound μ vs mesh complexity, average of dataset A.

The plot in Fig. 18 illustrates how the number of envelope elements, the number of sub-triangles inside envelopes after constrained linear meshing, and the total number of triangles in the output mesh depend on the choice of μ.

Comparison. In Fig. 19 differences to the method from [Pav and Walkington 2005] are illustrated. That method handles curved input constraints as well and yields coarser meshes than our method. However, it does not output a polynomial representation of the curved triangles, nor does it guarantee the existence of injective polynomial maps. In the blow-ups it can furthermore be observed that the method may split sharp input corners further, producing very small as well as large angles. Our dedicated corner enveloping prevents further quality deterioration at curve corners.

Table 2. Average timing (in seconds) over all 1000 instances from dataset A for various MIPS quality bounds μ.

μ =	4.0	4.5	5.0	10.0	20.0	50.0
Step (1)	0.1	0.1	0.1	0.1	0.1	0.1
Step (2a-d)	91.4	16.9	8.7	0.7	0.3	0.1
Step (2e)	24.6	29.4	5.7	2.4	2.2	2.4
Step (3)	5.4	1.3	0.8	0.2	0.1	0.1
Total Time	121.5	47.7	15.3	3.4	2.7	2.7

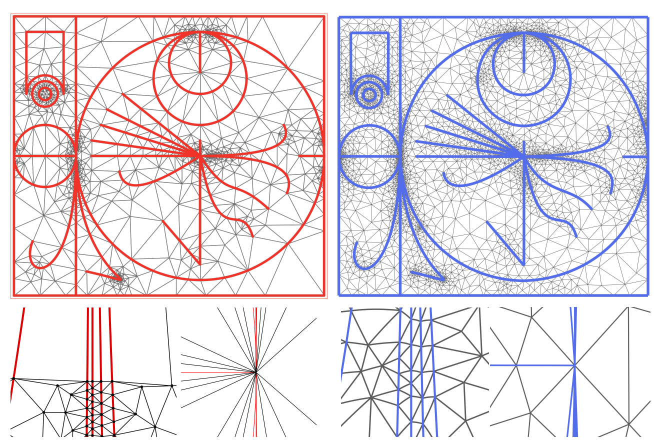


Fig. 19. Left: result of [Pav and Walkington 2005] on an example input. Right: result of our method. Notice in particular the quality difference in the lower blow-ups. Note: the original input contained zero angles which we remedied by slight perturbation.

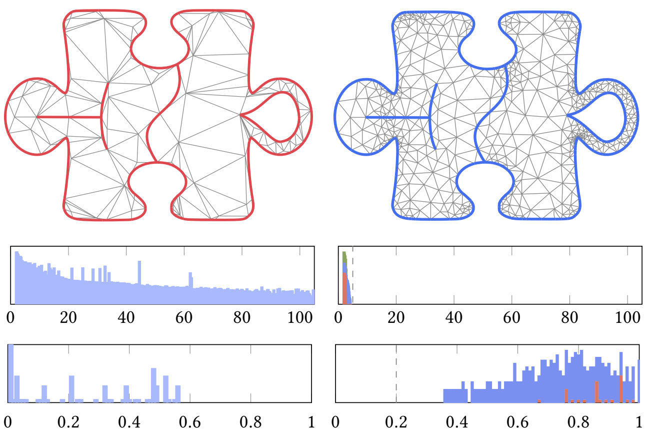


Fig. 20. Left: Result of Bézier Guarding [Mandad and Campen 2020a] on a given curve configuration. Right: Our result (for $\mu = 5.0$ and $\rho = 0.2$). MIPS and scaled Jacobian histograms (log scale) are shown below. Linear triangles are excluded from the scaled Jacobian histograms for clarity.

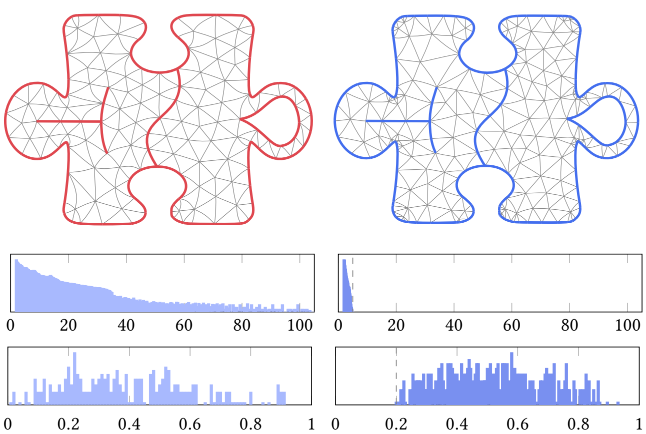


Fig. 21. Higher-order mesh optimization towards a target edge length, as described in [Mandad and Campen 2020a], following [Hu et al. 2019], applied to the results from Fig. 20. While some quality improvement (see the histograms) is achieved on the output of [Mandad and Campen 2020a] (left), guarantees about the amount of improvement are unavailable. In case of our output (right), we are able to preserve the quality guarantees.

Fig. 20 shows an example comparison of our proposed method to the recent method from [Mandad and Campen 2020a]; for an aggregate comparison over the entire ABCD dataset see Appendix F. That method guarantees polynomial triangles with injective geometric maps. It does not, however, consider distortion in any way. We observe many triangles with scaled Jacobian close to 0, down to $7 \cdot 10^{-5}$, and MIPS values up to $8 \cdot 10^{4}$. This is in stark contrast to the strictly bounded distortion guaranteed by our method. On the downside, a denser triangulation is generated to achieve these quality benefits.

Let us point out that [Mandad and Campen 2020a] propose a subsequent remeshing and optimization procedure. While this may commonly reduce distortion (see Fig. 21 left), it does not provide guarantees and may even be hampered by the poor numerical condition of the initial triangulation with arbitrarily high distortion. In contrast, when the mesh respects quality bounds due to our method, by omitting mesh modification operations that would break the bounds in the remeshing process we can yield a simplified mesh that still respects the desired bounds (Fig. 21 right).

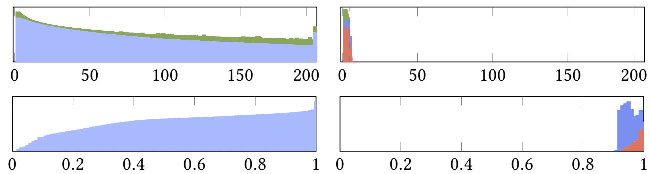


Fig. 22. Histograms (log scale) of aggregated MIPS distortion and scaled Jacobian for output meshes generated by TriWild [Hu et al. 2019] (left) and our method with $\mu = 5$ (right).

Fig. 22 shows a distortion comparison to the TriWild method [Hu et al. 2019] aggregated over inputs from that paper's OpenClipart dataset. It becomes apparent that, like the above, this method offers no quality bounds. Furthermore, while our method conforms to all input curves by construction, this method may trade a small amount of curve conformance to be able to guarantee injectivity, as discussed in [Hu et al. 2019, §4.2] and illustrated in [Mandad and Campen 2020a, §5.4] and Fig. 23. As an advantage, processing time on this dataset is lower by an average factor of 2.1 with this method.

Parameter Effects. In Figs. 24 and 25 the effects of quality bound parameters μ and ρ are demonstrated separately. Note that due to the way envelope elements are formed and inner control points are distributed, there is some correlation between these measures, i.e., decreasing μ has an effect related, but not identical, to increasing ρ.

As explained in Sec. 5.1 we fix the parameter φ (which controls minimal inner and outer envelope angles as well as minimal linear mesh angles) to 28.6° by default. A value that high allows driving MIPS distortion down to 3.5. In cases where loser (higher) MIPS bounds are sufficient, one may opt to use a lower φ value in the method; e.g., for $\mu = 20$ we may decrease φ down to $\sim 5°$ (Appendix B spells out the relation). Furthermore, one may use a different φ value for the envelope construction than for the linear mesh generator. This results in output meshes of different characteristics in terms of density and uniformity, as demonstrated in Fig. 26.

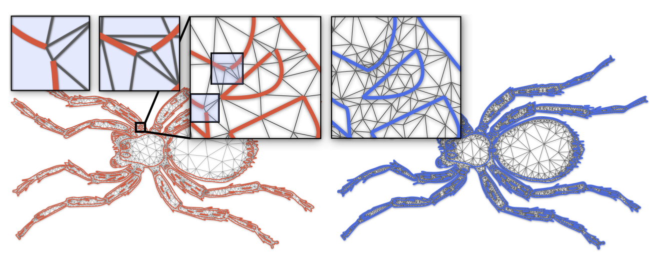


Fig. 23. Even if the input is entirely regular and free of intersections, TriWild [Hu et al. 2019] alters the input curves by clipping away parts to meet certain separation criteria (left). As a result, the output mesh does not fully conform to the input curves. Our method (right) ensures full conformance.

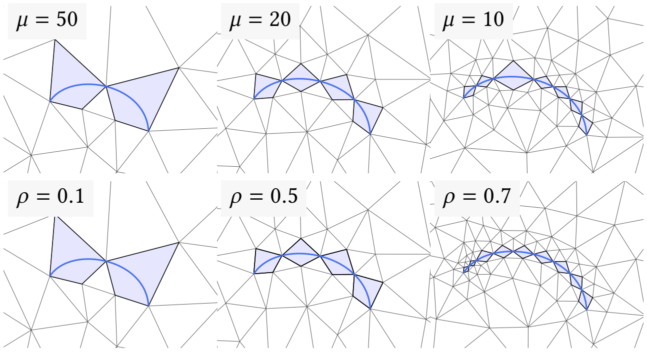


Fig. 24. Top row: effect of decreasing μ from 50 over 20 to 10 (while $\rho = 0$). Bottom row: effect of increasing ρ from 0.1 over 0.5 to 0.7 (while $\mu = \infty$). Envelopes are highlighted (light blue) for orientation.

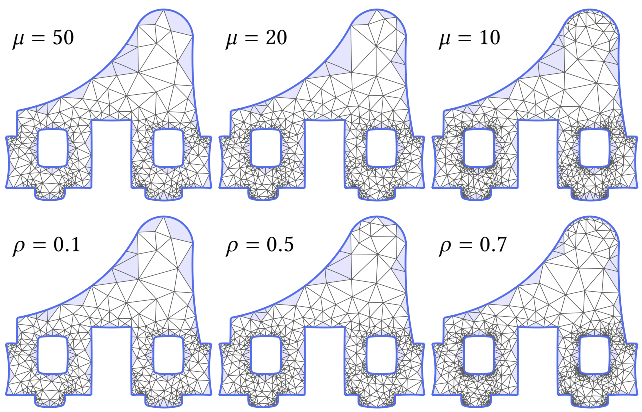


Fig. 25. Top row: effect of decreasing μ from 50 over 20 to 10 (while $\rho = 0$). Bottom row: effect of increasing ρ from 0.1 over 0.5 to 0.7 (while $\mu = \infty$).

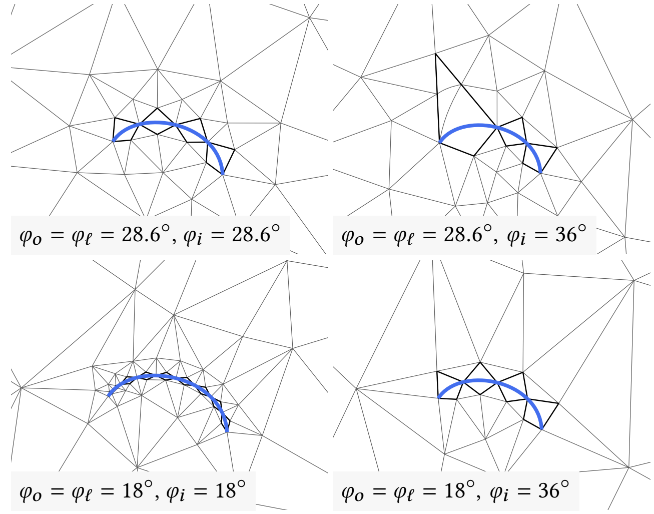


Fig. 26. For simplicity we assumed a common lower bound φ for inner envelope angles (Constructions 4.1 and 4.3), outer envelope angles (step (2b)), and linear mesh angles (step (3)). By using separate bounds φ_i, φ_o, and φ_ℓ, respectively, output mesh characteristics can be influenced. A lower minimal angle bound φ_ℓ (bottom row) coarsens the linear mesh [Ruppert 1995], but at the same time implies a tighter μ_g (cf. Sec. 5.4.1), leading to more curve bisections to guarantee meeting the desired bound μ. A larger value for φ_i (right column) reduces the amount of bisections needed to meet μ_g, but generates taller envelopes which may in turn intersect more often.

7 LIMITATIONS AND FUTURE WORK

Mesh Density. One of the application-dependent benefits of higher-order meshes is the fact that they can be coarser than linear meshes while still approximating a given boundary well, or even conforming to it exactly. From that perspective the meshes resulting from our method may be suboptimal and potentially too dense (cf. Fig. 30) for some applications. Nevertheless, no method to generate coarser higher-order meshes that offers quality bounds is available. We therefore view our method as an important step and envision various routes for follow-up work to address the aspect of parsimony.

One path is to use the guaranteed-quality results as starting point for mesh decimation techniques that are tailored to preserve the input mesh's quality. As our method's results commonly respect the bounds with quite some margin in large parts, there is room for simplification while preserving quality guarantees. As indicated in Fig. 21 right, this general direction, which has not received much attention yet, has potential and should be explored further.

As another avenue, adjustments to the various pieces of the algorithm could be made, so as to directly yield simpler meshes. One may strive to minimize the number of curve bisections (e.g., by partitioning curves based on curvature distribution); or the envelope apex could be chosen (from the space of valid apexes) in a more sophisticated manner (e.g., attempting to share the apex with a nearby (neighbor or opposite) curve, see Fig. 27; or outside of the envelopes, where no warp map is applied, looser angle bounds could be used in the linear triangulation, just to name a few options.

Finally, a more precise understanding of which mesh quality vs mesh simplicity balance is favorable for which use case and which quality measures are most relevant would be of value. This would better inform further developments in this direction.

Extension to 3D. It will be interesting to explore generalization to the 3D setting, where higher-order tetrahedral meshes are of interest [Feng et al. 2018], so as to reduce the problem to that of constrained linear tetrahedral meshing with guaranteed-quality [Cheng et al. 2012]. One challenge will be to deal with trimmed surface patches in a practical manner, which are commonly used for the precise representation of piecewise smooth 3D domain boundaries.

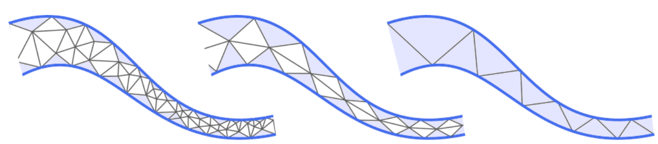


Fig. 27. Our method's result between two curves, with disjoint envelopes (left). Different choices of apex positions, e.g., shared with or on nearby conflicting curves, as sketched here (center and right), could reduce the number of curve bisections and as a result generate coarser meshes.

REFERENCES

Remi Abgrall, Cécile Dobrzynski, and Algiane Froehly. 2014. A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems. *International Journal for Numerical Methods in Fluids* 76, 4 (2014), 246–266.

I. Babuska and A. K. Aziz. 1976. On the Angle Condition in the Finite Element Method. *SIAM J. Numer. Anal.* 13, 2 (1976), 214–226.

Marshall Bern, David Eppstein, and John Gilbert. 1994. Provably good mesh generation. *J. Comput. System Sci.* 48, 3 (1994), 384–409.

Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, and Mariette Yvinec. 2000. Triangulations in CGAL. In *Proc. 16th Symp. on Computational Geometry.* 11–18.

Charles Boivin and Carl Ollivier-Gooch. 2002. Guaranteed-quality triangular mesh generation for domains with curved boundaries. *Int. J. Numer. Methods Eng.* 55, 10 (2002), 1185–1213.

Jan Brandts, Antti Hannukainen, Sergey Korotov, and Michal Křížek. 2011. On angle conditions in the finite element nethod. *SeMA Journal* 56 (09 2011), 81–95.

Siu-Wing Cheng, Tamal K Dey, and Jonathan Shewchuk. 2012. *Delaunay mesh generation.* CRC Press.

L. Paul Chew. 1989. *Guaranteed-Quality Triangular Meshes.* Tech. Rep. Department of Computer Science, Cornell University.

L. Paul Chew. 1993. Guaranteed-Quality Mesh Generation for Curved Surfaces. In *Proc. 9th Symp. on Computational Geometry.* 274–280.

P.G. Ciarlet and P.-A. Raviart. 1972a. The Combined Effect of Curved Boundaries and Numerical Integration in Isoparametric Finite Element Methods. In *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, A.K. Aziz (Ed.). Academic Press, 409 – 474.

P.G. Ciarlet and P.-A. Raviart. 1972b. Interpolation theory over curved elements, with applications to finite element methods. *Comput. Methods Appl. Mech. Eng.* 1, 2 (1972).

Y. Colin de Verdière and A. Marin. 1990. Triangulations presque équilatérales des surfaces. *J. Differential Geom.* 32, 1 (1990), 199–207.

Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Cheong Schwarzkopf. 2000. *Polygon Triangulation.* Springer Berlin Heidelberg, Berlin, Heidelberg, 45–61.

Saikat Dey, Robert M. O'Bara, and Mark S. Shephard. 1999. Curvilinear Mesh Generation In 3D. In *Proc. International Meshing Roundtable.* John Wiley & Sons, 407–417.

Luke Engvall and John A. Evans. 2020. Mesh quality metrics for isogeometric Bernstein–Bézier discretizations. *Comput. Methods Appl. Mech. Eng.* 371 (2020).

Hale Erten and Alper Üngör. 2009. Quality Triangulations with Locally Optimal Steiner Points. *SIAM Journal on Scientific Computing* 31, 3 (2009), 2103–2130.

Gerald Farin. 1986. Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design* 3, 2 (1986), 83–127.

Leman Feng, Pierre Alliez, Laurent Busé, Hervé Delingette, and Mathieu Desbrun. 2018. Curved Optimal Delaunay Triangulation. *ACM Trans. Graph.* 37, 4 (2018).

A. Fournier and D. Y. Montuno. 1984. Triangulating Simple Polygons and Equivalent Problems. *ACM Trans. Graph.* 3, 2 (April 1984), 153–174.

Isaac Fried. 1972. Condition of finite element matrices generated from nonuniform meshes. *Aiaa Journal* 10, 2 (1972), 219–221.

P.L. George and H. Borouchaki. 2012. Construction of tetrahedral meshes of degree two. *Int. J. Numer. Methods Eng.* 90, 9 (2012), 1156–1182.

William J. Gordon and Charles A. Hall. 1973. Transfinite element methods: Blending-function interpolation over arbitrary curved element domains. *Numer. Math.* 21, 2 (1973), 109–129.

Serge Gosselin and Carl Ollivier-Gooch. 2007. *Revisiting Delaunay Refinement Triangular Mesh Generation on Curve-bounded Domains.* Technical Report. U British Columbia.

Jens Gravesen, Anton Evgrafov, Dang-Manh Nguyen, and Peter Nørtoft. 2014. Planar Parametrization in Isogeometric Analysis. In *Mathematical Methods for Curves and Surfaces.* Springer Berlin Heidelberg, 189–212.

Robert Haber, Mark S. Shephard, John F. Abel, Richard H. Gallagher, and Donald P. Greenberg. 1981. A general two-dimensional, graphical finite element preprocessor utilizing discrete transfinite mappings. *Int. J. Numer. Methods Eng.* 17, 7 (1981).

Victoria Hernandez-Mederos, Jorge Estrada-Sarlabous, and Dionne León Madrigal. 2006. On local injectivity of 2D triangular cubic Bezier functions. *Investigación Operacional* 27, 3 (2006), 261–275.

Kai Hormann and Günther Greiner. 2000. MIPS: An efficient global parametrization method. *Curve and Surface Design '99* (2000), 153–162.

Yixin Hu, Teseo Schneider, Xifeng Gao, Qingnan Zhou, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2019. TriWild: Robust Triangulation with Curve Constraints. *ACM Trans. Graph.* 38, 4 (2019).

A. Johnen, C. Geuzaine, T. Toulorge, and J.-F. Remacle. 2016. Efficient Computation of the Minimum of Shape Quality Measures on Curvilinear Finite Elements. *Procedia Engineering* 163 (2016), 328 – 339.

Patrick M. Knupp. 2000. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I—a framework for surface mesh optimization. *Internat. J. Numer. Methods Engrg.* 48, 3 (2000), 401–420.

Richard Leroy. 2008. *Certificates of positivity and polynomial minimization in the multivariate Bernstein basis.* Ph.D. Dissertation. University of Rennes 1.

J. Li, T. J. Peters, and J. A. Roulier. 2012. Angular Convergence during Bézier Curve Approximation. arXiv:1210.2686 [math.GT]

Manish Mandad and Marcel Campen. 2020a. Bézier Guarding: Precise Higher-Order Meshing of Curved 2D Domains. *ACM Trans. Graph.* 39, 4, Article 103 (2020).

Manish Mandad and Marcel Campen. 2020b. Efficient piecewise higher-order parametrization of discrete surfaces with local and global injectivity. *Computer-Aided Design* 127 (2020).

Gary L. Miller, Steven E. Pav, and Noel J. Walkington. 2003. When and Why Ruppert's Algorithm Works. In *Proc. 12th International Meshing Roundtable.* 91–102.

Géraldine Morin and Ron Goldman. 2001. On the smooth convergence of subdivision and degree elevation for Bézier curves. *Comput. Aided Geom. Des.* 18, 7 (2001).

D. Moxey, D. Ekelschot, Ü. Keskin, S.J. Sherwin, and J. Peiró. 2016. High-order Curvilinear Meshing Using a Thermo-elastic Analogy. *Comput. Aided Des.* 72, C (2016).

Peter Oswald. 2015. Divergence of FEM: Babuška-Aziz triangulations revisited. *Applications of Mathematics* 60, 5 (2015), 473–484. http://eudml.org/doc/271633

Steven J Owen. 1998. A survey of unstructured mesh generation technology. *IMR* 239 (1998), 267.

Steven Elliot Pav. 2003. *Delaunay Refinement Algorithms.* Ph.D. Dissertation. Carnegie Mellon University.

Steven E. Pav and Noel J. Walkington. 2005. Delaunay Refinement by Corner Lopping. In *Proceedings of the 14th International Meshing Roundtable*, Byron W. Hanks (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 165–181.

Ulrich Pinkall and Konrad Polthier. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* 2, 1 (1993), 15–36.

Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny. 2013. *Bézier and B-spline techniques.* Springer Science & Business Media.

Alexander Rand. 2011a. Improved Examples of Non-Termination for Ruppert's Algorithm. arXiv:1103.3903 [cs.CG]

Alexander Rand. 2011b. Where and How Chew's Second Delaunay Refinement Algorithm Works. In *Proc. 23rd Canadian Conference on Computational Geometry.*

Ramsharan Rangarajan and Adrián J. Lew. 2014. Universal meshes: A method for triangulating planar curved domains immersed in nonconforming meshes. *Int. J. Numer. Methods Eng.* 98, 4 (2014), 236–264.

J. Ruppert. 1995. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *Journal of Algorithms* 18, 3 (1995), 548 – 585.

Teseo Schneider, Yixin Hu, Jérémie Dumas, Xifeng Gao, Daniele Panozzo, and Denis Zorin. 2018. Decoupling Simulation Accuracy from Mesh Quality. *ACM Trans. Graph.* 37, 6 (2018).

Jonathan R. Shewchuk. 2002a. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry* 22, 1 (2002), 21 – 74. 16th ACM Symp. Comput. Geom.

Jonathan Richard Shewchuk. 2002b. What is a good linear element? Interpolation, Conditioning, and Quality Measures. In *Proc. 11th Int. Meshing Roundtable.* 115–126.

Thomas Toulorge, Christophe Geuzaine, Jean-François Remacle, and Jonathan Lambrechts. 2013. Robust untangling of curvilinear meshes. *J. Comput. Phys.* 254 (2013).

Godfried T Toussaint. 1984. A new linear algorithm for triangulating monotone polygons. *Pattern Recognition Letters* 2, 3 (1984), 155 – 158.

Michael Turner, Joaquim Peiró, and David Moxey. 2018. Curvilinear mesh generation using a variational framework. *Computer-Aided Design* 103 (2018), 73 – 91.

S.A. Vavasis. 1996. Stable finite elements for problems with wild coefficients. *SIAM J. Numer. Anal.* 33 (1996), 890–916.

Z.J. Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, H.T. Huynh, Norbert Kroll, Georg May, Per-Olof Persson, Bram van Leer, and Miguel Visbal. 2013. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids* 72, 8 (2013), 811–845.

A EQUIDISTANT CONTROL POINTS

W.l.o.g. consider the first sub-curve of a curve after ℓ repeated subdivisions at parameter t ($0 < t < 1$). Its k-th control point is $\boldsymbol{p}'_k = \sum_{i=0}^{k} \boldsymbol{p}_i B_i^k(t^\ell)$. The distance between two consecutive control points of this sub-curve is given by

$$|\boldsymbol{p}'_{k+1} - \boldsymbol{p}'_k| = |\boldsymbol{p}_{k+1} B_{k+1}^{k+1}(t^\ell) + \sum_{i=0}^{k} \boldsymbol{p}_i (B_i^{k+1}(t^\ell) - B_i^k(t^\ell))|.$$

As $\ell \to \infty$, the ratio of distances between consecutive control points $|\boldsymbol{p}'_1 - \boldsymbol{p}'_0| / |\boldsymbol{p}'_{k+1} - \boldsymbol{p}'_k| \to 1$. This, together with the convergence of the control polygon to a flat state, shows convergence to a state of uniformly distributed control points.

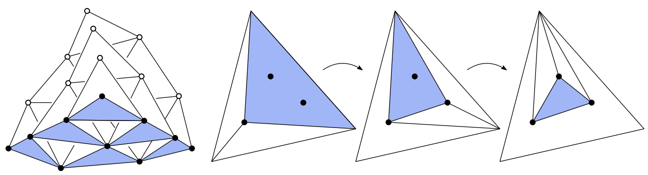


Fig. 28. Left: de Casteljau's algorithm can be used to compute the control points of the Bézier sub-triangles under 1-3 subdivision. Right: threefold application yields the control points corresponding to any sub-triangle domain [Prautzsch et al. 2013, §11.3].

B MIPS DISTORTION OF STRAIGHT-EDGE TRIANGLES

Consider the linear map that maps an equilateral triangle to a triangle with angles θ_1, θ_2, and $\theta_3 = 180^\circ - \theta_1 - \theta_2$. Up to a constant factor, its Jacobian is

$$\begin{bmatrix} \sin\theta_3 & \cos\theta_1 \sin\theta_2 \\ 0 & \sin\theta_1 \sin\theta_2 \end{bmatrix} \begin{bmatrix} \sqrt{3} & -1 \\ 0 & 2 \end{bmatrix}.$$

Its MIPS distortion therefore is given by

$$\frac{2}{\sqrt{3}}(\sin\theta_2 \csc\theta_1 \csc\theta_3 + \cot\theta_2).$$

This is a convex function over the convex domain $\theta_1, \theta_2, \theta_3 \geq \varphi$ (of all triangles with inner angles at least φ) so the maximum is attained at this domain's corners, where two of the triangle's angles are φ. For a lower angle bound of $\varphi = 28.6^\circ$ this yields a worst case MIPS value of 3.49159, slightly below 3.5.

C BÉZIER TRIANGLE SUBDIVISION

Application of de Casteljau's algorithm to evaluate a Bézier triangle at an interior point yields (as byproduct) the control points of three Bézier sub-triangles corresponding to a 1-3 subdivision, as illustrated in Fig. 28 left. Threefold application allows to compute the control points corresponding to any sub-triangle domain (Fig. 28 right).

D BERNSTEIN COEFFICIENTS OF JACOBIAN NORM

Assuming $\boldsymbol{p}_{ijk} = (x_{ijk}, y_{ijk})$, the Jacobian of the warp map g is given by

$$J = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix} \begin{bmatrix} J_{00} & J_{10} \\ J_{01} & J_{11} \end{bmatrix} = \begin{bmatrix} J_a & J_b \\ J_c & J_d \end{bmatrix},$$

where the second matrix represents the Jacobian of the mapping from the domain triangle to a unit leg right triangle. The partial derivatives [Farin 1986] are given by

$$\frac{\partial x}{\partial u} = n \sum_{i+j+k=n-1} (x_{(i+1)jk} - x_{ij(k+1)}) B^{n-1}_{ijk},$$

$$\frac{\partial x}{\partial v} = n \sum_{i+j+k=n-1} (x_{i(j+1)k} - x_{ij(k+1)}) B^{n-1}_{ijk}.$$

The ijk-th coefficient of the squared Jacobian's norm $\|J\|^2 = J_a^2 + J_b^2 + J_c^2 + J_d^2$ in the triangular Bernstein basis is then constructed by summing the squared corresponding individual entries

$$\left(J_a^2\right)_{ijk} = \sum_{\substack{|r|=|s| \\ r+s=(i,j,k)}} \frac{i!j!k!}{n!r!s!} (J_{00} x_r^u + J_{01} x_r^v)(J_{00} x_s^u + J_{01} x_s^v),$$

where $r = (r_1, r_2, r_3)$, $|r| = r_1 + r_2 + r_3$, $r! = r_1!r_2!r_3!$, x_r^u denotes the r-th control point of $\partial x/\partial u$, analogously for the other entries of J.

E MINIMUM CURVED ANGLE

Consider one corner of angle τ of some triangle, w.l.o.g. formed by vectors $(1, 0)$ and $(\cos\tau, \sin\tau)$, and its image under a continuous map g. Via singular value decomposition, at the corner point, g is a rotation by some angle θ followed by a non-uniform scaling by σ_1 and σ_2 followed by a second rotation (that we can ignore in the following as it does not affect angles). Application of this map to the two corner vectors yields

$$\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & \cos\tau \\ 0 & \sin\tau \end{bmatrix} = \begin{bmatrix} \sigma_1 \cos\theta & \sigma_1 \cos(\theta+\tau) \\ \sigma_2 \sin\theta & \sigma_2 \sin(\theta+\tau) \end{bmatrix}$$

The angle τ' between the two mapped vectors is

$$\cot\tau' = \frac{{\sigma_1}^2 \cos\theta \cos(\theta+\tau) + {\sigma_2}^2 \sin\theta \sin(\theta+\tau)}{\sigma_1 \sigma_2 \sin\tau}.$$

Differentiating wrt. θ and equating to zero yields $({\sigma_2}^2 - {\sigma_1}^2)\sin(2\theta + \tau) = 0$. Hence, the mapped angle is minimal (assuming $\sigma_1 \geq \sigma_2$, and $\sigma = \sigma_1/\sigma_2$), regardless of the scaling, for $\theta = -\tau/2$ (and maximal for $\theta = 90^\circ - \tau/2$). The warped angle τ' therefore is, regardless of orientation, not smaller than

$$\tau' = \cot^{-1}\left(\frac{\sigma \cos^2 \frac{\tau}{2} - \sigma^{-1} \sin^2 \frac{\tau}{2}}{\sin\tau}\right).$$

F ADDITIONAL COMPARISONS

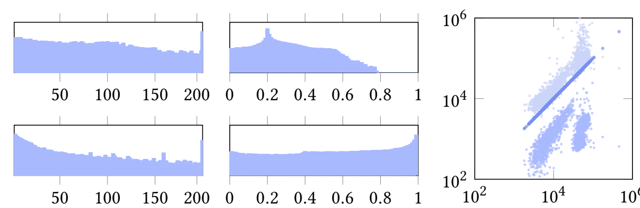


Fig. 29. Histograms (log scale) of MIPS distortion and scaled Jacobian of Bézier Guarding output aggregated over the ABCD dataset. Top: immediate output. Bottom: after subsequent remeshing and mesh optimization. In contrast to our method's results for the same input (see Figs. 16 and 17), no bounds (other than $\det J > 0$) are respected. Right: Scatter plot showing processing time (ms) of Bézier Guarding (native • or with remeshing •) relative to our method • (with $\mu = 5$) over the ABCD dataset cases.

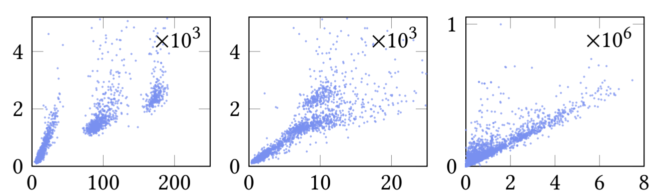


Fig. 30. Mesh complexity (number of elements) comparison scatter plots. Left: Ours (x-axis) with $\mu = 5$ vs Bézier Guarding (y-axis). Each dot represents one input case from the ABCD dataset. Center: Same, with looser bound (MIPS < 300). Right: Ours (x-axis) vs TriWild (y-axis). Each dot represents one input case from the OpenClipart dataset of [Hu et al. 2019]. On average, our output has 7.8× as many elements.