Generating High-quality High-order Parameterization for Isogeometric Analysis on Triangulations

Songtao Xia

Siemens Corporate Technology
755 College Road East, Princeton, NJ 08540

Xiaoping Qian [*]

Department of Mechanical Engineering
University of Wisconsin-Madison
1513 University Avenue, Madison, WI 53706

Abstract

This paper presents an approach for automatically generating high-quality high-order parameterizations for isogeometric analysis on triangulations. A B-spline represented boundary geometry is parameterized into a collection of high-order Bézier triangles or tetrahedra in 2D and 3D spaces, respectively. Triangular Bézier splines are used to represent both the geometry and physical field over the triangulation. By imposing continuity constraints on the Bézier ordinates of the elements, a set of global C^r smooth basis is constructed and used as the basis for analysis. To ensure high quality of the parameterization, both the parametric and physical mesh are optimized to reduce the shape distortion of the high-order elements relative to well-defined reference elements. The shape distortion is defined based on the Jacobian of the triangular Bézier splines, and its sensitivity with respect to the location of control points is derived analytically and evaluated efficiently. Moreover, a sufficient condition is derived to guarantee the generated mesh is free of local self-intersection, thanks to the convex hull property of triangular Bézier splines. By using a Heaviside projection function, the non-negative Jacobian determinant constraints are formulated efficiently as a single optimization constraint. Several 2D and 3D numerical examples are presented to demonstrate that high-quality high-order elements are generated using our approach.

Keywords Domain parameterization, Isogeometric analysis, Triangular Bézier splines, High-order elements, C^r smoothness

[*] Corresponding author. Email: qian@engr.wisc.edu

1 Introduction

Isogeometric analysis (IGA) is a numerical analysis approach introduced by Hughes et al [1] to integrate Computer-Aided Design (CAD) and Finite Element Analysis (FEA). It uses the same Non-Uniform Rational B-Splines (NURBS) as basis to both represent the geometry and approximate field variables in solving partial differential equations (PDEs). Due to the same basis used in geometric representation and in solution approximation, it eliminates the geometric approximation error commonly occurred in classical FEA procedures. Once the initial mesh is constructed, refinements can also be easily implemented and exact geometry is maintained at all levels without the necessity of interaction with the CAD system [1, 2]. The exact geometry representation has also led to the development of isogeometric shape optimization [3, 4, 5], where the optimized geometry can be directly imported into CAD systems . Another advantage of isogeometric analysis is its computational efficiency on a per-node basis over classical C^0 Lagrange polynomial based FEA. The higher continuity of the NURBS basis has been demonstrated to significantly improve the numerical efficiency and accuracy on a per-node basis in many areas including structural analysis [2, 6], fluid simulation [7] and shape optimization [3, 4, 5, 8].

However, the tensor-product structure of NURBS has restricted its applicability in analysis. First, many CAD models cannot be represented by one single NURBS patch. Therefore, multiple patches have to be used for geometry with complex topology and it is not easy to achieve high-order continuity between such patches. Besides, it is challenging to construct NURBS based volumetric mesh from surface representation of complex geometries [9, 10, 11]. Analysis of trimmed geometries [12] and local mesh refinement are also known to be cumbersome for NURBS based IGA.

As an alternative, triangular Bézier splines (TBS) have recently emerged as a powerful alternative to shape modeling and isogeometric analysis, due to their flexibility in representing domains of complex topology and their high-order of continuity. Local refinement can also be implemented without any difficulty. Normalized basis of Powell-Sabin (PS) splines has been used for numerical solution of PDEs in [13, 14]. More generalized framework of IGA on triangulations are introduced in [15, 16, 17], where C^r smooth rational triangular Bézier splines (rTBS) are used as basis to represent both the geometry and physical field. The rTBS elements can be locally refined and represent any geometric model of complex topology, including trimmed geometries [17]. By using a smooth-refine-smooth scheme, optimal convergence has also been demonstrated with C^r elements for generalized geometries [16, 17]. The rTBS based IGA has also been used in shape optimization where optimized designs with complex topology have been demonstrated and can be directly linked to CAD systems [18]. Recently, an isogeometric approach based on unstructured tetrahedral and mixed-element Bernstein-Bézier elements has also been proposed[19].

Although high-order elements in IGA have exhibited high efficiency in analysis, parameterization quality is a general concern. A good parameterization requires the physical mesh to be valid, that is free of local self-intersection or folding. Although NURBS based IGA has been demonstrated to be robust with severe mesh distortion [20], a good parameterization can be beneficial to both the analysis accuracy and computation efficiency [21, 22, 5, 11, 23]. Both planar and volumetric B-spline based parameterization [5, 22, 11] have been investigated previously, where Jacobian based measurements are used as objectives to improve the parameterization quality through an optimization procedure. T-spline based parameterization has also been studied [24, 25]. Parameterization in IGA based on PS splines is studied in [23], where the planar parameterization is improved by minimizing the Winslow functional. Generating high-quality triangular and tetrahedral Lagrange finite elements has also been studied in [26, 27, 28], where a shape distortion measurement is used as objective to improve the mesh quality. The shape distortion is calculated based on the Jacobian of the parameterization evaluated at sampled points in the elements.

In this work, we focus on generating high-quality high-order Bézier triangular and tetrahedral elements in IGA based on triangular Bézier splines [15, 16, 17]. Given a B-spline represented boundary geometry, we first generate a linear triangular/tetrahedral mesh over a polygonal domain that approximates the original geometry. Then we elevate the degree of the linear mesh to produce the initial parametric mesh. By replacing the boundary control points of the parametric mesh with the Bézier control points extracted from the input B-spline boundary, an initial high-order C^0 physical mesh is created. However, the replacement of boundary control points may result in self-intersection of elements near curved boundaries. Moreover, to construct C^r parameterization, some points in the parametric and physical mesh may need to be relocated to satisfy the high-order continuity constraints, and likely resulting in poor quality or even tangled elements. To untangle the meshes and improve the parameterization quality, we develop an approach to sequentially optimize the parametric and physical meshes with the goal to reduce the shape distortion of the elements relative to well-defined elements. In addition, we derive a sufficient condition to guarantee the generated elements to be free of local self-intersection, taking advantage of the convex hull property of the triangular Bézier splines. Directly incorporating the aforementioned sufficient condition for each element as constraints into the optimization would dramatically slow down the optimization, due to the large number of constraints. Instead we use a Heaviside projection based constraint formulation [29] to cast the large number of constraints into a single constraint, which significantly improves the optimization efficiency.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction of bivariate and trivariate splines on Bézier triangles and tetrahedra respectively. Section 3 describes the process of constructing an initial parameterization in IGA on triangulations. The definition of shape distortion and its calculation is discussed in Section 4. Section 5 describes the details of optimization formulation and analytical sensitivity derivation. Some numerical examples are presented in Section 6 followed by conclusion in Section 7.

2 Triangular Bézier splines

To make the paper self-contained, in this section we briefly introduce bivariate and trivariate splines defined on Bézier triangles and tetrahedra respectively. The constraints for their smooth joins are also presented. For further reading please see [30, 31].

2.1 Bivariate splines on Bézier triangles

B-spline has been widely used to represent curves and surfaces. Each knot span of a B-spline curve corresponds to a Bézier curve which is defined through Bernstein basis functions. A d-th degree Bernstein polynomial is defined as

$$B_{ij,d}(\xi) = \binom{d}{i,j} \xi^i (1-\xi)^{d-i}, \quad \xi \in [0,1], \tag{1}$$

where $\binom{d}{i,j} = \dfrac{d!}{i!j!}$, $i+j=d$. Accordingly a d-th degree bivariate Bernstein polynomial is defined as

$$B_{\mathbf{i},d}(\boldsymbol{\xi}) = \frac{d!}{i!j!k!} \gamma_1^i \gamma_2^j \gamma_3^k, \quad |\mathbf{i}| = i+j+k = d, \tag{2}$$

where $\mathbf{i}$ represents a triple index (i,j,k) and $(\gamma_1, \gamma_2, \gamma_3)$ is the barycentric coordinate of a point $\boldsymbol{\xi} \in \mathbb{R}^2$. Every point $\boldsymbol{\xi} = (\xi_1, \xi_2)$ in a fixed triangle with vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{R}^2$ can be written uniquely in the form

$$\boldsymbol{\xi} = \gamma_1 \mathbf{v}_1 + \gamma_2 \mathbf{v}_2 + \gamma_3 \mathbf{v}_3, \tag{3}$$

with $\gamma_1 + \gamma_2 + \gamma_3 = 1$.

It has been shown that the set $\{B_{\mathbf{i},d}\}_{|\mathbf{i}|=d}$ is a basis for the space of degree d bivariate polynomials $\mathcal{P}_d$ [30]. A triangular Bézier patch is defined as

$$\mathbf{b}(\boldsymbol{\xi}) = \sum_{|\mathbf{i}|=d} \mathbf{p}_{\mathbf{i}} B_{\mathbf{i},d}(\boldsymbol{\xi}), \tag{4}$$

where $\mathbf{p}_{\mathbf{i}}$ represents a triangular array of control points.

Under the isoparametric concept, the same bivariate Bernstein basis defining a triangle $\tau = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ can also be used to define a polynomial function f of degree d over τ as

$$f(\boldsymbol{\xi}) = \sum_{|\mathbf{i}|=d} b_{\mathbf{i}} B_{\mathbf{i},d}(\boldsymbol{\xi}). \tag{5}$$

The $b_{\mathbf{i}}$ (or b_{ijk}) are called the Bézier ordinates of f. Their associated set of domain points is defined as

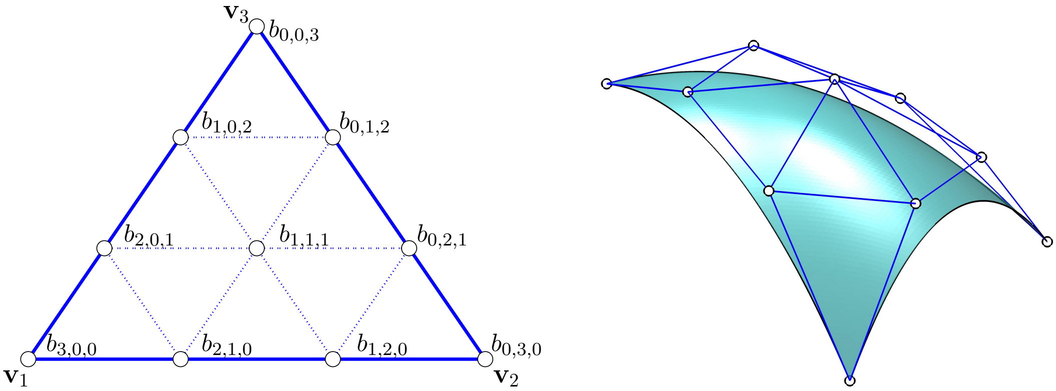
$$\mathcal{D}_{d,\tau} = \left\{ \mathbf{q}_{ijk} = \frac{i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3}{d}, i+j+k=d \right\}. \tag{6}$$

Thus the control polygon of the function f is given by the points $(\mathbf{q}_{ijk}, b_{ijk})$. Figure 1 gives an example of the associated domain points of the Bézier ordinates and triangular Bézier patch.

Two polynomials f and $\tilde{f}$ of degree d join r times differentiably across the common edge of two triangles $\tau = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ and $\tilde{\tau} = \{\mathbf{v}_4, \mathbf{v}_3, \mathbf{v}_2\}$ if and only if [30]

$$\tilde{b}_{\rho,j,k} - \sum_{\mu+\nu+\kappa=\rho} \frac{\rho!}{\mu!\nu!\kappa!} b_{\mu,k+\nu,j+\kappa} \gamma_1^\mu \gamma_2^\nu \gamma_3^\kappa = 0, \quad j+k+\rho=d, \ \rho = 0, \cdots, r, \tag{7}$$

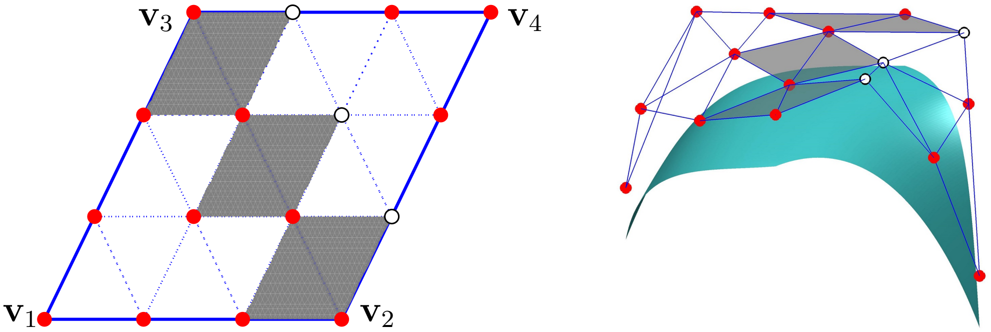
where $\gamma_1, \gamma_2, \gamma_3$ are the barycentric coordinates of vertex $\mathbf{v}_4$ with respect to triangle τ. Figure 2 gives an illustration of two triangular Bézier patches with C^1 continuity constraints. The red solids represent free nodes whose values can be freely chosen, while the three white solids representing dependent nodes are determined by the red free nodes through the continuity constraints. The shaded area indicates the triangles where continuity constraints are imposed. As can be seen in Figure 2b, the control points in each shaded triangle pair are coplanar. For better visualization of the underlying C^1 patch, the control net in Figure 2b is shifted up slightly.



(a) Associated domain points of the Bézier ordinates b_{ijk} in $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$.

(b) Triangular Bézier patch $\mathbf{b}(\boldsymbol{\xi})$.

Figure 1: Domain points and triangular Bézier patch.



(a) Two domain triangles with C^1 constraints on Bézier ordinates.

(b) Two Bézier patches with C^1 continuity.

Figure 2: Triangular Bézier patches with C^1 continuity. The dependent nodes (white solids) are determined by the free nodes (red solids) through the continuity constraints. The shaded areas indicate the triangles with shared edges where the constraints are imposed. As can be seen in Figure 2b, the control points in each shaded triangle pair are coplanar. For better visualization, the control net is shifted up slightly in Figure 2b.

Consider a parametric domain $\widehat{\Omega}$ and its triangulation $\widehat{T}$. We introduce the spline spaces of piecewise polynomials of degree d over $\widehat{T}$ [30]

$$\mathcal{S}_d^r(\widehat{T}) = \{f \in C^r(\widehat{\Omega}) : f|_\tau \in \mathcal{P}_d \ \forall \tau \in \widehat{T}\}, \tag{8}$$

where τ is an arbitrary triangle in $\widehat{T}$ and r is the continuity order of the spline over $\widehat{\Omega}$. In addition, if a bivariate spline has higher smoothness at some vertices, we call it a superspline and denote the associated space as [30]

$$\mathcal{S}_d^{r,\rho}(\widehat{T}) = \{f \in \mathcal{S}_d^r(\widehat{T}) : f \in C^{\rho_{\mathbf{v}}}(\mathbf{v}) \ \forall \mathbf{v} \in \mathcal{V}\}, \tag{9}$$

where $\mathcal{V}$ is the set of all vertices in $\widehat{T}$ and $\rho := \{\rho_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$ with $r \leq \rho_{\mathbf{v}} \leq d$ for each $\mathbf{v} \in \mathcal{V}$.

There are several approaches to obtain C^r spline spaces on a triangulated domain $\widehat{\Omega}(\widehat{T})$. In this thesis we are interested in the spaces $\mathcal{S}_d^r$ and $\mathcal{S}_d^{r,\rho}$ with full approximation power of d-th degree polynomials. The straightforward way is to apply condition (7) directly on the triangles, which requires the degree of the polynomial much higher than r, such as $d \geq 3r+2$ [32]. The alternative way is splitting each triangle in $\widehat{T}$ into several micro-triangles before imposing the continuity constraints on the micro-triangles. The original triangles are then called macro-triangles. These include the CT split [33] with polynomials of degree $d \geq 3r$ for continuity r-odd and $d \geq 3r+1$ for r-even, and the PS split [34] with polynomials of degree $d \geq \frac{9r-1}{4}$ for r-odd and of degree $d \geq \frac{9r+4}{4}$ for r-even. For example, in this paper we use CT split to obtain $\mathcal{S}_3^1$ spline space with cubic polynomials, and PS split to obtain $\mathcal{S}_2^1$, $\mathcal{S}_5^2$ and $\mathcal{S}_5^{2,3}$ spline spaces with quadratic and quintic polynomials respectively. We also use so-called polynomial macro-element technique to obtain $\mathcal{S}_5^1$ and $\mathcal{S}_5^{1,2}$ spline spaces with quintic polynomials without using any split technique.

Figure 3 shows the CT and PS splits with corresponding free and dependent domain points respectively. In the CT split, each vertex of a triangle in $\widehat{T}$ is connected with its centroid point to form three micro-triangles, as shown in Figure 3a. We denote this resulting triangulation as $\widehat{T}_{ct}$. In the PS split, for each triangle we connect its incenter to each of the three vertices and connect the two incenters of two triangles sharing a common edge. In addition, we connect the middle of each boundary edge to the incenter of the associated triangle, resulting in six micro-triangles. For the ease of implementation in this paper, the centroid point instead of the incenter of each triangle is used as the interior split point and the resulting triangulation is denoted as $\widehat{T}_{ps}$, as shown in Figure 3b.

Uniform refinement can also be performed as needed. For example, each triangle can be subdivided into four sub-triangles by connecting the middle points of the edges. This kind of 1-to-4 split based uniform refinement is used in our subsequent analysis of convergence during mesh refinement.

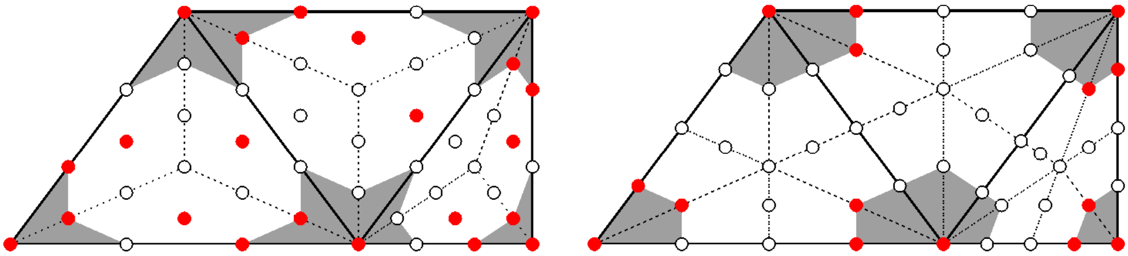
2.2 Trivariate splines on Bézier tetrahedra

Following the bivariate case, the trivariate Bernstein polynomial of degree d with respect to a tetrahedron $\tau = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ is defined as

$$B_{ijk,d}(\boldsymbol{\xi}) = \frac{d!}{i!j!k!l!}\gamma_1^i\gamma_2^j\gamma_3^k\gamma_4^l, \quad i+j+k+l = d, \tag{10}$$

where $(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$ is the barycentric coordinate of a point $\boldsymbol{\xi} \in \mathbb{R}^3$ with respect to τ,

$$\boldsymbol{\xi} = \gamma_1\mathbf{v}_1 + \gamma_2\mathbf{v}_2 + \gamma_3\mathbf{v}_3 + \gamma_4\mathbf{v}_4, \tag{11}$$



(a) Cubic C^1 mesh using CT split.

(b) Quadratic C^1 mesh using PS split.

Figure 3: The CT and PS refinements of macro-triangles with C^1 continuity constraints. The dependent domain points (white solids) are determined by the free domain points (red solids) through the continuity constraints. The shaded areas indicate where the continuity constraints are imposed.

with

$$\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = 1.$$

Similarly $\{B_{ijkl,d}\}_{i+j+k+l=d}$ form a basis for the space of degree d trivariate polynomials $\mathcal{P}_d$ [30]. The associated domain points of a tetrahedron $\tau = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ are

$$\mathcal{D}_{d,\tau} = \left\{ \mathbf{q}_{ijkl} = \frac{i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3 + l\mathbf{v}_4}{d}, i+j+k+l=d \right\}, \tag{12}$$

as shown in Figure 4a.

Suppose $\tau := \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ and $\tilde{\tau} := \{\mathbf{v}_5, \mathbf{v}_2, \mathbf{v}_4, \mathbf{v}_3\}$ are two tetrahedra sharing the face $F := \{\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$. Two polynomials f and $\tilde{f}$ of degree d on τ and $\tilde{\tau}$ join together with C^r continuity across the face F if and only if [30]

$$\tilde{t}_{mijk} - \sum_{\nu+\mu+\kappa+\delta=m} t_{\nu,i+\mu,k+\kappa,j+\delta} B^m_{\nu\mu\kappa\delta}(v_5) = 0, \quad i+j+k = d-m, \ m = 0, \cdots, r, \tag{13}$$

where $B^m_{\nu\mu\kappa\delta}$ are the Bernstein polynomials of degree m associated with the tetrahedron τ. Figure 4b gives an illustration of two cubic Bézier tetrahedra with C^1 continuity constraints. The red solids represent free nodes whose values can be freely chosen, while the blue solids representing dependent nodes are determined by the red free nodes through the continuity constraints. The shaded small tetrahedra indicate the domain points involved in the continuity constraints. Conditions for smooth joins of the two polynomials at the common vertex and across the common edge can also be found in [31].

Consider a parametric domain $\widehat{\Omega}$ and its tetrahedral partition $\widehat{T}$. We introduce the spline spaces of piecewise polynomials of degree d over $\widehat{T}$ [30]

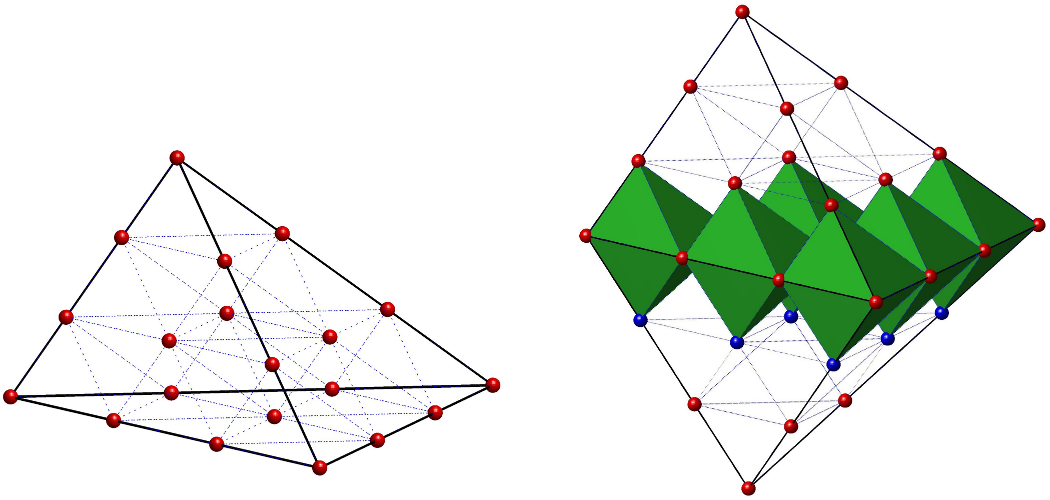
$$\mathcal{S}^r_d(\widehat{T}) = \{f \in C^r(\widehat{\Omega}) : f|_\tau \in \mathcal{P}_d \ \forall \tau \in \widehat{T}\}, \tag{14}$$

where τ is an arbitrary triangle in $\widehat{T}$ and r is the continuity order of the spline over $\widehat{\Omega}$.

Similar to the superspline spaces in 2D, for a trivariate spline given $0 \le r \le \mu \le \rho$, we define

$$\mathcal{S}^{r,\rho,\mu}_d(\widehat{T}) := \{f \in \mathcal{S}^r_d(\widehat{T}) : f \in C^\rho(v), \ \forall v \in \mathcal{V}, f \in C^\mu(e), \ \forall e \in \mathcal{E}\}, \tag{15}$$

where $\mathcal{V}$ and $\mathcal{E}$ are the sets of vertices and edges of $\widehat{T}$ respectively. $f \in C^\rho(v)$ and $f \in C^\mu(e)$ mean the polynomial has C^ρ and C^μ smoothness at the vertex v and across the edge e respectively.



(a) Associated domain points of a Bézier tetrahedron.

(b) Two domain tetrahedra with C^1 constraints on Bézier ordinates. The Bézier ordinates corresponding to the dependent nodes (blue) are determined by the free nodes (red) through the continuity constraints.

Figure 4: Domain points and continuity constraints of Bézier tetrahedra.

Various C^r macro-element spaces have been described in [30]. In this thesis we focus on a quintic C^1 macro-element space defined on the Alfeld split of a tetrahedral partition. In Alfeld split, each tetrahedron is subdivided into four sub-tetrahedra by connecting its barycenter to each of its vertices, as shown in Figure 5. For refinement of a tetrahedral mesh, we consider a quasi-uniform refinement, where a tetrahedron is subdivided into eight subtetrahedra. Four subtetrahedra are obtained by cutting off the four corners by connecting the midpoints of the edges. The other four are obtained by connecting the shortest diagonal of the remaining octahedron.

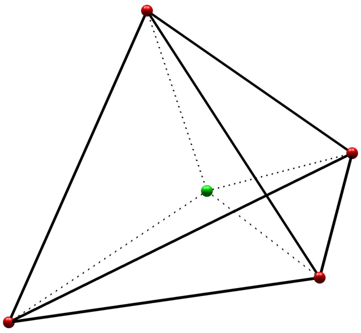


Figure 5: Alfeld split of a tetrahedron. Each tetrahedron is subdivided into four subtetrahedra by connecting the barycenters to the four vertices.

3 Initial parameterization in TBS based IGA

In this section we describe the procedure of constructing an initial parameterization for 2D geometries using triangular Bézier splines, which was introduced in [15, 16]. A similar procedure for constructing initial parameterization for 3D geometries can be found in [17]. The process is illustrated in Figure 6 and can be described as follows.

(1) Given a domain Ω with B-spline boundary curves of degree d (Figure 6a), we subdivide each B-spline curve into a set of Bézier curves via knot insertions (Figure 6b).

(2) The end points of these Bézier curves are connected to form a polygonal parametric domain $\widehat{\Omega}$. The domain $\widehat{\Omega}$ is then triangulated using Delaunay triangulation to obtain $\widehat{T}_0$ and the associated domain points are generated according to Eq. (6) (Figure 6d).

(3) In order to reproduce the exact B-spline boundary in later C^r parametrization, we further adjust those vertices in $\widehat{T}_0$ that correspond to C^k $(k \geq r)$ knot points in the physical boundary. We move each such boundary vertex to the line segment connecting the adjacent vertices of C^q $(0 \leq q < r)$ smoothness, and so that the length-ratio of consecutive edges is the same as the ratio of the corresponding knot intervals in the B-spline boundary curve. For example, in Figure 6d and 6e,

$$|\mathbf{v}_0\mathbf{v}_1| : |\mathbf{v}_1\mathbf{v}_2| : |\mathbf{v}_2\mathbf{v}_3| : |\mathbf{v}_3\mathbf{v}_4| = (t_1 - t_0) : (t_2 - t_1) : (t_3 - t_2) : (t_4 - t_3).$$

If there are too few C^q knots points in the boundary curve to form a suitable domain $\widehat{\Omega}$, some smooth knot points (such as the two points marked as red squares in the inner round boundary in Figure 6a) are also used as corner points to form the polygonal domain (Figure 6c). In this way, we obtain a C^0 parametrization with C^r smoothness along the boundary except at corner vertices where the smoothness is C^0 [15].

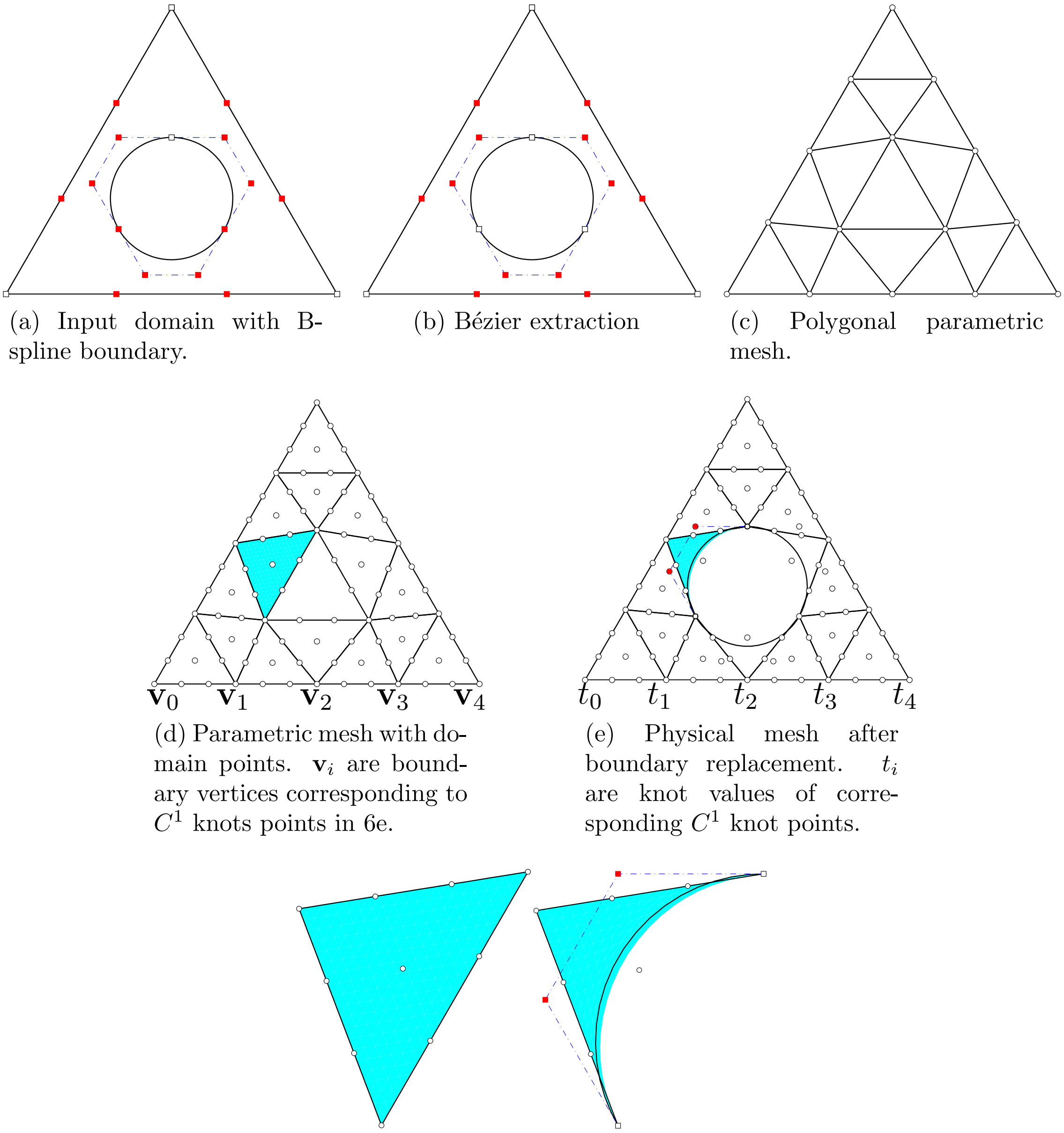
(4) Replace the boundary control points of $\widehat{\Omega}_{\widehat{T}}$ with corresponding control points of the Bézier curves in the physical domain to obtain a triangulation T_0 on the physical domain (Figure 6e).

The above procedure gives us an initial C^0 parameterization that can be used for isogeometric analysis. However, a couple of issues need to be noted in this procedure. First, although the Delaunay triangulation generated in Step (2) is usually of high quality, the adjustment of vertices in Step (3) to ensure C^r smoothness of the boundary will change the shape of elements along the boundary and thus reduces their quality. Second, the boundary replacement in Step (4) may result in self-intersected elements, as shown in Figure 6f. Because of the over recessed control points after replacement, the control polygon of the curved boundary intersects the other two boundaries of the element.

In addition, when constructing C^r parameterization, some of the interior control points may need to be relocated to satisfy the high-order smoothness constraints [16, 17], and likely resulting in poor quality or even tangled elements. Since the parameterization quality is crucial for analysis accuracy and computational efficiency, these issues need to be addressed to generate high-quality parameterization.

4 Shape Distortion of Bézier triangles and tetrahedra

In this section, we formulate an optimization problem to improve the quality of TBS based parameterization by optimizing the positions of the control points in the mesh.



(a) Input domain with B-spline boundary.

(b) Bézier extraction

(c) Polygonal parametric mesh.

(d) Parametric mesh with domain points. $\mathbf{v}_i$ are boundary vertices corresponding to C^1 knots points in 6e.

(e) Physical mesh after boundary replacement. t_i are knot values of corresponding C^1 knot points.

(f) Detailed view of Figure (e). Because of the over recessed control points, the control polygon of the curved boundary intersects with the other two boundaries of the element, causing self-folding of the mesh.

Figure 6: Construction of initial parameterization for a given input domain bounded by B-spline curves (Figure 6a). White and red squares are the end and interior control points respectively of the boundaries. White circles are the domain and control points of the mesh.

The optimization formulation can simultaneously untangle the mesh and improve its quality even if the initial mesh has invalid elements. Particularly, we develop a sufficient condition to guarantee all Bézier triangular and tetrahedral elements in the final mesh to be valid.

The shape distortion measure is defined in terms of the Jacobian of the geometric mapping, and has been widely used to measure the quality of triangular and tetrahedral elements [27, 35, 36]. Thanks to the spline basis of Bézier triangles and tetrahedral, the Jacobian determinant of TBS based parameterization can also be expressed in the TBS form. This allows efficient evaluation of Jacobian based functions and derivation of a sufficient condition that guarantees the validity of the parameterization.

4.1 Definition of shape distortion

Given a reference element $\hat{\tau}_e$, a master element τ_e and the mapping G_0 between them: $G_0 : \hat{\tau}_e \mapsto \tau_e$, as shown in Figure 7, the Jacobian of the mapping $\mathbf{J} = \mathbf{D}G_0$ is usually used to check whether the element τ_e is valid as following:

$$|\mathbf{J}| \begin{cases} > 0 & \text{valid,} \\ = 0 & \text{degenerated,} \\ < 0 & \text{invalid.} \end{cases} \tag{16}$$

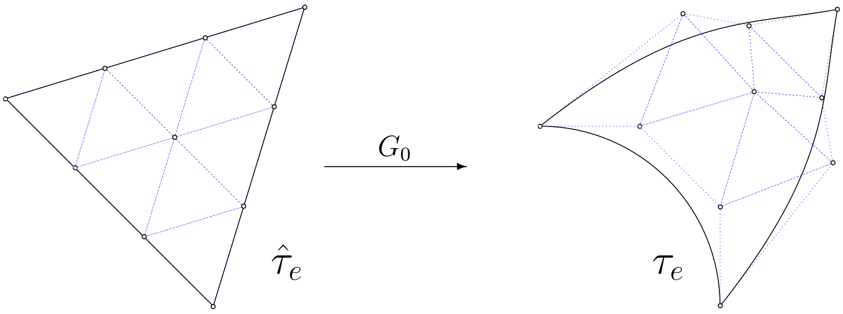


Figure 7: Geometric mapping between a reference and master element.

To further measure the quality of τ_e, a shape distortion of τ_e with respect to $\hat{\tau}_e$ is defined as [37]

$$\eta = \frac{\|\mathbf{J}\|_F^2}{\alpha |\mathbf{J}|^{2/\alpha}}, \tag{17}$$

where $\|\mathbf{J}\|_F$ is the Frobenius norm, $|\mathbf{J}|$ is the determinant of $\mathbf{J}$, and α is the spatial dimension, i.e. $\alpha = 2$ for Bézier triangular element and $\alpha = 3$ for Bézier tetrahedral element. As can be seen in Eq. (17), the value of η equals 1 when τ_e has no distortion with respect to $\hat{\tau}_e$, i.e. the two elements only differ by a scale factor. The value of η tends to ∞ when τ_e degenerates, i.e. $|\mathbf{J}| = 0$. When the element τ_e is inverted, i.e. $|\mathbf{J}| < 0$, η becomes negative too. Therefore, η is not continuous when $|\mathbf{J}|$ varies from negative to positive and not differentiable at $|\mathbf{J}| = 0$.

To deal with invalid elements, a regularized distortion measure is proposed in [27] as

$$\eta = \frac{\|\mathbf{J}\|_F^2}{\alpha h^{2/\alpha}}, \quad h = \frac{1}{2}\left(|\mathbf{J}| + \sqrt{|\mathbf{J}|^2 + 4\delta^2}\right), \tag{18}$$

where δ is an element-wise regularization parameter. For valid elements, i.e. $|\mathbf{J}| > 0$, δ is 0, while for other elements, δ is a small positive number. A detailed description of determining the value of δ can be found in [38]. In this work, we choose $\delta = 0.001$ for all the examples. An illustration of the function h, the Jacobian determinant $|\mathbf{J}|$ and the regularization parameter δ is shown in Figure 8. With the regularized distortion measure, the value of h is monotonically increasing when the element transits from invalid to valid, allowing the optimization to recover from invalid meshes.

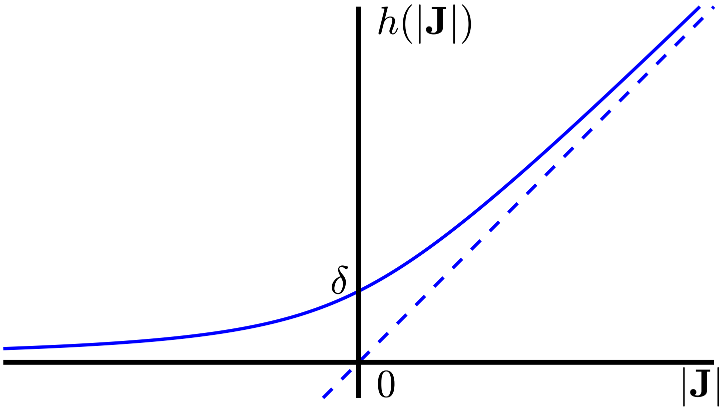


Figure 8: The regularization function h is monotonically increasing when the element transits from invalid to valid.

Note that the reference element can be any valid element. In this work, we choose the ideal element (equilateral triangle or tetrahedra) as the reference when measuring the parametric element in parametric mesh optimization, and choose the corresponding parametric element as the reference when measuring the physical element in physical mesh optimization.

4.2 Jacobian of TBS based parameterization

We first show the 2D case, where the Jacobian determinant of the bivariate spline based parameterization can be written in the form of bivariate spline as well [11, 23]. Then the 3D case can be derived similarly.

Let $\mathbf{G}(\xi, \eta) : \widehat{\Omega} \mapsto \Omega$ be the geometric map, i.e.

$$\mathbf{G}(\xi, \eta) = (\mathbf{G}^x, \mathbf{G}^y), \tag{19}$$

where (x, y) and (ξ, η) represent points in the physical and parametric domain respectively. The Jacobian of the map is

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{G}}{\partial \xi} & \frac{\partial \mathbf{G}}{\partial \eta} \end{bmatrix}, \tag{20}$$

where $\frac{\partial \mathbf{G}}{\partial \xi}, \frac{\partial \mathbf{G}}{\partial \eta}$ are the first order directional derivatives of $\mathbf{G}$ in the directions $(1, 0)$ and $(0, 1)$ respectively. The geometric map including C^0 and C^r for 2D parameterization is in the form

$$\mathbf{G} = \sum_{i+j+k=d} \mathbf{p}_{ijk} B^d_{ijk},$$

where $\mathbf{p}_{ijk}$ are the physical control points and B^d_{ijk} are the bivariate Bernstein basis of degree d. Then the directional derivative of $\mathbf{G}$ in direction $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$ can be written as

$$\frac{\partial \mathbf{G}}{\partial \xi} = d \sum_{i+j+k=d-1} \mathbf{p}^{(1)}_{ijk}(\boldsymbol{\xi}) B^{d-1}_{ijk} \tag{21}$$

with

$$\mathbf{p}^{(1)}_{ijk}(\boldsymbol{\xi}) = \xi_1 \mathbf{p}_{i+1,j,k} + \xi_2 \mathbf{p}_{i,j+1,k} + \xi_3 \mathbf{p}_{i,j,k+1}, \tag{22}$$

where (ξ_1, ξ_2, ξ_3) are the barycentric directional coordinates in direction $\boldsymbol{\xi}$ with respect to the given element. Eq. (21) shows that the directional derivative of a degree d bivariate spline is another bivariate spline with degree $d-1$.

To calculate the Jacobian determinant of an element, we need to find the barycentric directional coordinates $\boldsymbol{\gamma}^1 = (\gamma^1_1, \gamma^1_2, \gamma^1_3)$ and $\boldsymbol{\gamma}^2 = (\gamma^2_1, \gamma^2_2, \gamma^2_3)$ in directions $(1,0)$ and $(0,1)$ respectively. Let (ξ_1, η_1), (ξ_2, η_2), (ξ_3, η_3) be the coordinates of the vertices of an element, then $\boldsymbol{\gamma}^1$ and $\boldsymbol{\gamma}^2$ can be found as

$$\boldsymbol{\gamma}^1 = \frac{1}{M}(\eta_2 - \eta_3, \eta_3 - \eta_1, \eta_1 - \eta_2), \boldsymbol{\gamma}^2 = \frac{1}{M}(\xi_3 - \xi_2, \xi_1 - \xi_3, \xi_2 - \xi_1), \tag{23}$$

where

$$M = \begin{vmatrix} \xi_1 & \xi_2 & \xi_3 \\ \eta_1 & \eta_2 & \eta_3 \\ 1 & 1 & 1 \end{vmatrix}. \tag{24}$$

Substituting Eq. (23) and (21) into Eq. (20), we obtain

$$\det \mathbf{J} = |\mathbf{J}| = d^2 \sum_{|\mathbf{s}|=2d-2} B^{2d-2}_{\mathbf{s}} J_{\mathbf{s}} \tag{25}$$

with

$$J_{\mathbf{s}} = \sum_{\substack{\mathbf{i}_1+\mathbf{i}_2=\mathbf{s} \\ |\mathbf{i}_1|=d-1 \\ |\mathbf{i}_2|=d-1}} \frac{\binom{d-1}{\mathbf{i}_1}\binom{d-1}{\mathbf{i}_2}}{\binom{2d-2}{\mathbf{i}_1+\mathbf{i}_2}} \det \left[\mathbf{p}^{(1)}_{\mathbf{i}_1}(\boldsymbol{\gamma}^1) \quad \mathbf{p}^{(1)}_{\mathbf{i}_2}(\boldsymbol{\gamma}^2) \right], \tag{26}$$

where $\mathbf{s}, \mathbf{i}_1, \mathbf{i}_2$ are index tuples. As can be seen, the Jacobian determinant is another bivariate spline. Moreover, due to the convex hull property of bivariate spline, if all coefficients $J_{\mathbf{s}} > 0$, then $|\mathbf{J}| > 0$, i.e. the element is guaranteed to be valid when all $J_{\mathbf{s}} > 0$. This sufficient condition is used as a constraint in our optimization formulation to ensure the final optimized mesh is valid.

At the same time, the squared Frobenius norm can be computed as

$$\begin{aligned} \|\mathbf{J}\|^2_F &= \left(\frac{\partial \mathbf{G}}{\partial \xi}\right)^2 + \left(\frac{\partial \mathbf{G}}{\partial \eta}\right)^2 \\ &= d^2 \sum_{|\mathbf{t}|=2d-2} B^{2d-2}_{\mathbf{t}} J_{\mathbf{t}} \end{aligned} \tag{27}$$

with

$$J_{\mathbf{t}} = \sum_{\substack{\mathbf{i}_1+\mathbf{i}_2=\mathbf{t} \\ |\mathbf{i}_1|=d-1 \\ |\mathbf{i}_2|=d-1}} \frac{\binom{d-1}{\mathbf{i}_1}\binom{d-1}{\mathbf{i}_2}}{\binom{2d-2}{\mathbf{i}_1+\mathbf{i}_2}} \left(\mathbf{p}^{(1)}_{\mathbf{i}_1}(\boldsymbol{\gamma}^1) \cdot \mathbf{p}^{(1)}_{\mathbf{i}_2}(\boldsymbol{\gamma}^1) + \mathbf{p}^{(1)}_{\mathbf{i}_1}(\boldsymbol{\gamma}^2) \cdot \mathbf{p}^{(1)}_{\mathbf{i}_2}(\boldsymbol{\gamma}^2) \right). \tag{28}$$

Similar to the 2D case, the Jacobian of trivariate spline based parameterization can also be derived. The geometric map based on trivariate spline parametrization is

$$\mathbf{G} = \sum_{i+j+k+l=d} \mathbf{p}_{ijkl} B^d_{ijkl},$$

where $\mathbf{p}_{ijkl}$ are the physical control points and B^d_{ijkl} are the bivariate Bernstein basis of degree d. The Jacobian of the map is

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{G}}{\partial \xi} & \frac{\partial \mathbf{G}}{\partial \eta} & \frac{\partial \mathbf{G}}{\partial \zeta} \end{bmatrix}, \tag{29}$$

where $\frac{\partial \mathbf{G}}{\partial \xi}, \frac{\partial \mathbf{G}}{\partial \eta}, \frac{\partial \mathbf{G}}{\partial \zeta}$ are the first order directional derivatives of $\mathbf{G}$ in the directions $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ respectively. Then the directional derivative of $\mathbf{G}$ in direction $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3, \xi_4)$ can be written as

$$\frac{\partial \mathbf{G}}{\partial \xi} = d \sum_{i+j+k+l=d-1} \mathbf{p}^{(1)}_{ijkl}(\boldsymbol{\xi}) B^{d-1}_{ijkl} \tag{30}$$

with

$$\mathbf{p}^{(1)}_{ijkl}(\boldsymbol{\xi}) = \xi_1 \mathbf{p}_{i+1,j,k,l} + \xi_2 \mathbf{p}_{i,j+1,k,l} + \xi_3 \mathbf{p}_{i,j,k+1,l} + \xi_4 \mathbf{p}_{i,j,k,l+1}, \tag{31}$$

where $(\xi_1, \xi_2, \xi_3, \xi_4)$ are the barycentric directional coordinates in direction $\boldsymbol{\xi}$ with respect to the given element. Eq. (30) shows that the directional derivative of a degree d trivariate spline is another trivariate spline with degree $d-1$.

Let $\boldsymbol{\gamma}^1$, $\boldsymbol{\gamma}^2$ and $\boldsymbol{\gamma}^3$ be the barycentric directional coordinates in directions $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ respectively. The Jacobian determinant of a trivariate spline based parameterization can be calculated as

$$\det \mathbf{J} = |\mathbf{J}| = d^3 \sum_{|\mathbf{s}|=3d-3} B^{3d-3}_{\mathbf{s}} J_{\mathbf{s}} \tag{32}$$

with

$$J_{\mathbf{s}} = \sum_{\substack{\mathbf{i}_1+\mathbf{i}_2+\mathbf{i}_3=\mathbf{s} \\ |\mathbf{i}_1|=d-1 \\ |\mathbf{i}_2|=d-1 \\ |\mathbf{i}_3|=d-1}} \frac{\binom{d-1}{\mathbf{i}_1}\binom{d-1}{\mathbf{i}_2}\binom{d-1}{\mathbf{i}_3}}{\binom{3d-3}{\mathbf{i}_1+\mathbf{i}_2+\mathbf{i}_3}} \det \begin{bmatrix} \mathbf{p}^{(1)}_{\mathbf{i}_1}(\boldsymbol{\gamma}^1) & \mathbf{p}^{(1)}_{\mathbf{i}_2}(\boldsymbol{\gamma}^2) & \mathbf{p}^{(1)}_{\mathbf{i}_3}(\boldsymbol{\gamma}^3) \end{bmatrix}. \tag{33}$$

As can be seen, the Jacobian determinant is another trivariate spline, and the sufficient condition on element validity also applies here. That is, the Bézier tetrahedral element is guaranteed to be valid when all $J_{\mathbf{s}} > 0$.

At the same time, the squared Frobenius norm can be computed as

$$\begin{aligned} \|\mathbf{J}\|_F^2 &= \left(\frac{\partial \mathbf{G}}{\partial \xi}\right)^2 + \left(\frac{\partial \mathbf{G}}{\partial \eta}\right)^2 + \left(\frac{\partial \mathbf{G}}{\partial \zeta}\right)^2 \\ &= d^2 \sum_{|\mathbf{t}|=2d-2} B^{2d-2}_{\mathbf{t}} J_{\mathbf{t}} \end{aligned} \tag{34}$$

with

$$J_{\mathbf{t}} = \sum_{\substack{\mathbf{i}_1+\mathbf{i}_2=\mathbf{t} \\ |\mathbf{i}_1|=d-1 \\ |\mathbf{i}_2|=d-1}} \frac{\binom{d-1}{\mathbf{i}_1}\binom{d-1}{\mathbf{i}_2}}{\binom{2d-2}{\mathbf{i}_1+\mathbf{i}_2}} \left(\mathbf{p}^{(1)}_{\mathbf{i}_1}(\boldsymbol{\gamma}^1) \cdot \mathbf{p}^{(1)}_{\mathbf{i}_2}(\boldsymbol{\gamma}^1) + \mathbf{p}^{(1)}_{\mathbf{i}_1}(\boldsymbol{\gamma}^2) \cdot \mathbf{p}^{(1)}_{\mathbf{i}_2}(\boldsymbol{\gamma}^2) + \mathbf{p}^{(1)}_{\mathbf{i}_1}(\boldsymbol{\gamma}^3) \cdot \mathbf{p}^{(1)}_{\mathbf{i}_2}(\boldsymbol{\gamma}^3) \right). \tag{35}$$

5 Optimizing TBS based parameterization

To improve the quality of TBS based parameterization, we formulate an optimization problem to minimize the shape distortion of Bézier triangle and tetrahedral elements. The position of the free control points that do not affect the boundary are chosen as the design variables.

5.1 Objective function and design variables

The shape distortion measure η is a point-wise value evaluated at a single point. The element-wise measure is defined as the integration of η over the element, i.e.

$$\|\eta\|_{\tau_e} = \int_{\tau_e} \eta d\Omega. \tag{36}$$

To evaluate and improve the quality of a mesh composed by a set of elements, we define the objective function as

$$\mathcal{F} = \frac{1}{2}\sum_{e=1}^{n_e} \int_{\tau_e} \left(\eta(|\mathbf{J}|) - 1\right)^2 d\Omega, \tag{37}$$

where n_e is the number of Bézier triangle or tetrahedral elements τ_e in the mesh.

In this work, the parametric and physical mesh are optimized in sequential steps. First the linear parametric mesh is optimized with respect to the ideal elements, then the optimized linear mesh is degree-elevated to the same degree as the physical mesh and used as the reference to optimize the physical mesh.

When optimizing the linear parametric mesh, all boundary nodes are fixed, only positions of the interior nodes $\hat{\mathbf{p}}_\imath$ are used as optimization variables. Although it has been shown that minimizing the objective function (37) alone without constraints can lead to good mesh quality in practice [27, 36, 28], there is no theoretical guarantee that the optimized mesh is free of local self-intersection. The reason is that in numerical implementation, the Jacobian determinant in Eq. (37) is only evaluated at a limited number of points for each element, which are usually the numerical integration points. For a highly curved geometry of complex topology, it is possible that some local negative Jacobian values may not be detected. Therefore the optimization may easily be trapped into a local minimum and take many iterations to converge. Thanks to the convex hull property of the spline representation for the Jacobian of TBS based parameterization in Eq. (26) and (33), the Jacobian determinant is guaranteed to be positive if all of its coefficients J_s are positive [39]. Therefore we can incorporate this sufficient condition into the optimization formulation to ensure the validity of the optimized mesh. The optimization formulation can be given as

$$\begin{cases} \min\limits_{\hat{\mathbf{p}}_\imath} & \mathcal{F} = \dfrac{1}{2}\displaystyle\sum_{e=1}^{\hat{n}_e} \int_{\hat{\tau}_e} \left(\eta(|\hat{\mathbf{J}}|) - 1\right)^2 d\widehat{\Omega}, \\ \text{s.t.} & \hat{J}_s > 0, \quad s = 1, 2, \cdots, \hat{n}_s, \end{cases} \tag{38}$$

where $\hat{n}_e$ and $\hat{n}_s$ are the total number of parametric elements and the Jacobian coefficients for all elements in the mesh respectively. As been confirmed in our later numerical results, the optimization formulation with explicit constraints on the Jacobian determinant converges in much fewer iterations than the formulation without any constraints.

For physical mesh optimization, besides the control points that affect the boundary, the positions of the dependent points for C^r elements are also excluded from optimization variables, since their positions depend on the free points. Thus only the internal free points are allowed to change positions. We denote such points as $\mathbf{p}_\jmath$, $\jmath = 1, 2, \cdots, n_0$ and use them as the optimization variables. The optimization problem now can be formulated as

$$\begin{cases} \min\limits_{\mathbf{p}_\jmath} & \mathcal{F} = \dfrac{1}{2}\sum\limits_{e=1}^{n_e}\displaystyle\int_{\tau_e} \left(\eta(|\mathbf{J}|) - 1\right)^2 d\Omega, \\ \text{s.t.} & J_s > 0, \quad s = 1, 2, \cdots, n_s, \end{cases} \tag{39}$$

with n_e and n_s being the total number of physical elements and the Jacobian coefficients for all elements in the mesh respectively.

Note that, the degree of the Jacobian determinant of a degree d Bézier tetrahedral element is $3d - 3$ and it has $(3d-2)(3d-1)3d/6$ coefficients. This means there would be tens of thousands of constraints if we use the formulation Eq. (39), which is extremely inefficient for optimization. To increase the efficiency, we use a Heaviside projection based constraint formulation introduced in [29]. The Heaviside projection function is defined as

$$H(J_s) = \frac{1}{1 + e^{2\beta J_s}} = \begin{cases} 1 & J_s < 0, \\ \frac{1}{2} & J_s = 0, \\ 0 & J_s > 0, \end{cases} \tag{40}$$

where β is a positive number to control the projection behavior. A plot of the Heaviside function with different β parameters is shown in Figure 9. The Heaviside function projects the Jacobian value on to a $0 - 1$ scale. The larger the β value, closer the projection curve to a step function, more non-linear the function becomes. In this work , we choose the initial value of β to be 2, and gradually increase its value till it reaches 16. The projected constraint is

$$\sum_{s=1}^{n_s} H(J_s) \leq \varepsilon, \tag{41}$$

where ε is small positive value close to 0. As ε tends to zero during the optimization, the projection forces all J_s to be positive. In this work, the value of ε is gradually decreased from 1 to 10^{-8} during optimization.

The new optimization formulation becomes

$$\begin{cases} \min\limits_{\hat{\mathbf{p}}_\imath} & \mathcal{F} = \dfrac{1}{2}\sum\limits_{e=1}^{\hat{n}_e}\displaystyle\int_{\hat{\tau}_e} \left(\eta(|\hat{\mathbf{J}}|) - 1\right)^2 d\widehat{\Omega}, \\ \text{s.t.} & \sum\limits_{s=1}^{\hat{n}_s} \dfrac{1}{1 + e^{2\beta \hat{J}_s}} \leq \varepsilon, \end{cases} \tag{42}$$

for parametric mesh optimization and

$$\begin{cases} \min\limits_{\mathbf{p}_\jmath} & \mathcal{F} = \dfrac{1}{2}\sum\limits_{e=1}^{n_e}\displaystyle\int_{\tau_e} \left(\eta(|\mathbf{J}|) - 1\right)^2 d\Omega, \\ \text{s.t.} & \sum\limits_{s=1}^{n_s} \dfrac{1}{1 + e^{2\beta J_s}} \leq \varepsilon, \end{cases} \tag{43}$$

for physical mesh optimization. By using the Heaviside projection based formulation, the number of constraints reduces to 1 and the optimization efficiency is significantly improved.

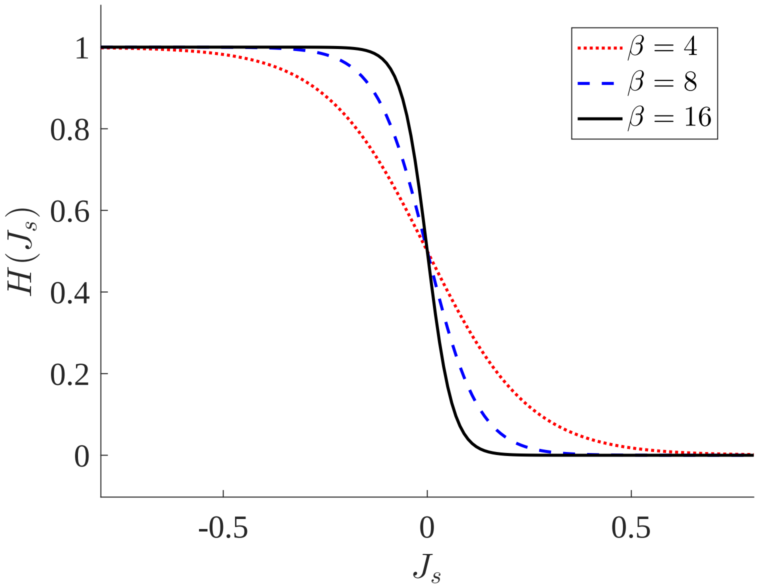


Figure 9: Heaviside projection function with different parameters.

5.2 Sensitivity analysis

We use a gradient based optimization algorithm to solve the optimization problem in Eq. (42) and (43), which requires sensitivity information. In this section, we describe how to derive the gradient of the objective and constraint functions with respect to the design variables. From Section 4.2, it can be seen that the Jacobian coefficients J_s, determinant $|\mathbf{J}|$ and the squared Frobenius norm $\|\mathbf{J}\|_F^2$ are just linear functions of the control points, so it is straightforward to obtain their sensitivities with respect to the control points. In the following, we focus on the sensitivity of $\mathcal{F}$ with respect to the $|\mathbf{J}|$ and $\|\mathbf{J}\|_F^2$.

Taking the derivative of the objective function Eq. (37) with respect to an arbitrary control point $\mathbf{p}_\imath$, we obtain

$$\frac{\partial \mathcal{F}}{\partial \mathbf{p}_\imath} = \sum_{e=1}^{n_e} \int_{\tau_e} (\eta - 1) \frac{\partial \eta}{\partial \mathbf{p}_\imath} d\Omega. \tag{44}$$

We first proceed with the 2D case and the 3D case can be derived similarly. Using the chain rule, we can compute the term $\dfrac{\partial \eta}{\partial \mathbf{p}_\imath}$ based on Eq. (18) as

$$\begin{aligned} \frac{\partial \eta}{\partial \mathbf{p}_\imath} &= \frac{\partial \eta}{\partial \left(\|\mathbf{J}\|_F^2\right)} \frac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath} + \frac{\partial \eta}{\partial h} \frac{\partial h}{\partial |\mathbf{J}|} \frac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath} \\ &= \frac{1}{2h} \frac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath} - \frac{\|\mathbf{J}\|_F^2}{2} \frac{2}{h^2} \frac{1}{2} \left(1 + \frac{2|\mathbf{J}|}{2\sqrt{|\mathbf{J}|^2 + 4\delta^2}}\right) \frac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath} \\ &= \frac{1}{2h} \frac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath} - \frac{|\mathbf{J}|_F^2}{4h^2} \frac{2h}{\sqrt{|\mathbf{J}|^2 + 4\delta^2}} \frac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath} \\ &= \frac{1}{2h} \frac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath} - \frac{\eta}{\sqrt{|\mathbf{J}|^2 + 4\delta^2}} \frac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath} \end{aligned} . \tag{45}$$

The gradients $\dfrac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath}$ and $\dfrac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath}$ can be easily derived from Eq. (26) and (27). Combining all equations together, we can obtain the sensitivity of the functions with respect to all control points. However, the boundary control points are fixed during optimization. Especially for C^r parameterizations, only the free points inside the domain are the design variables. To map the sensitivity from all control points to the design variables, we recall the continuity constraints

$$\mathbf{p} = \mathbf{C}^T \mathbf{p}^f. \tag{46}$$

From Eq. (46) we can get the sensitivity of a control point $\mathbf{p}_\imath$ with respect to a design control point $\mathbf{p}_\jmath$ as

$$\frac{\partial \mathbf{p}_\imath}{\partial \mathbf{p}_\jmath} = \mathbf{C}_{\jmath\imath}, \tag{47}$$

where $bC_{\jmath\imath}$ is the element in the continuity matrix $\mathbf{C}$ with row and column indices $\jmath, \imath$ respectively. Substituting Eq. (47) and (45) into Eq. (44) will give the sensitivity of the objective function with respect to the design variables.

The derivation for 3D parameterization is the same as 2D, the only difference is the sensitivity of the shape distortion with respect to all control points, which we give below as

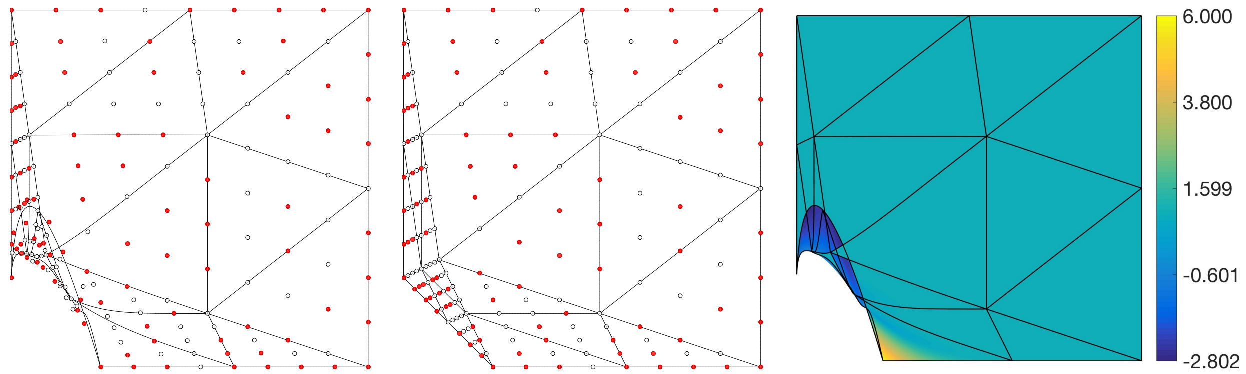
$$\begin{aligned}
\frac{\partial \eta}{\partial \mathbf{p}_\imath} &= \frac{\partial \eta}{\partial \left(\|\mathbf{J}\|_F^2\right)} \frac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath} + \frac{\partial \eta}{\partial h} \frac{\partial h}{\partial |\mathbf{J}|} \frac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath} \\
&= \frac{1}{3h^{2/3}} \frac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath} - \frac{\|\mathbf{J}\|_F^2}{3} \frac{2}{3h^{5/3}} \frac{1}{2} \left(1 + \frac{2|\mathbf{J}|}{2\sqrt{|\mathbf{J}|^2 + 4\delta^2}}\right) \frac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath} \\
&= \frac{1}{3h^{2/3}} \frac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath} - \frac{|\mathbf{J}|_F^2}{9h^{5/3}} \frac{2h}{\sqrt{|\mathbf{J}|^2 + 4\delta^2}} \frac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath} \\
&= \frac{1}{3h^{2/3}} \frac{\partial \left(\|\mathbf{J}\|_F^2\right)}{\partial \mathbf{p}_\imath} - \frac{2\eta}{3\sqrt{|\mathbf{J}|^2 + 4\delta^2}} \frac{\partial |\mathbf{J}|}{\partial \mathbf{p}_\imath}
\end{aligned}. \tag{48}$$

6 Numerical examples

In this section, we present some numerical results on optimizing TBS based parameterization. We use a gradient based algorithm named the method of moving asymptotes (MMA) [40] to solve the optimization problems. The initial parameterization used in the optimization are constructed using the method described in Section 3 for 2D geometries and in [17] for 3D geometries. Although both the parametric and physical mesh are included in our formulation, only the parametric mesh of the second example is optimized. The initial parametric mesh constructed for all other examples are already close to optimal and thus are not included in the optimization implementation.

6.1 Optimizing Bézier triangles

We first present some results of optimizing Bézier triangles for 2D parameterization. In the first example, we demonstrate the need for explicit constraints on the Jacobian determinant in the optimization formulation. Figure 10 shows an initial quartic C^1 physical and parametric mesh along with the Jacobian determinant. As can be seen, there is local self-intersection in the initial physical mesh. The minimum value of the Jacobian determinant is about -2.8. First, we optimize the physical mesh using the objective function shown in Eq. (37) without any constraints. Since there is no explicit constraint on the Jacobian determinant, the optimization is trapped into a local minimum and it takes about 450 iterations to converge, as shown in Figure 11. $\min(J_0)$ and $\min(J_s)$ are the minimum values of the Jacobian determinant in Eq. (25) evaluated at quadrature points and Jacobian coefficients in Eq. (26) respectively. Note in Figure 11c that, after the optimization converges, $\min(J_s)$ is still negative, meaning there are still negative values for the Jacobian coefficient, resulting the minimum value $\min(J_0)$ of only 0.038.

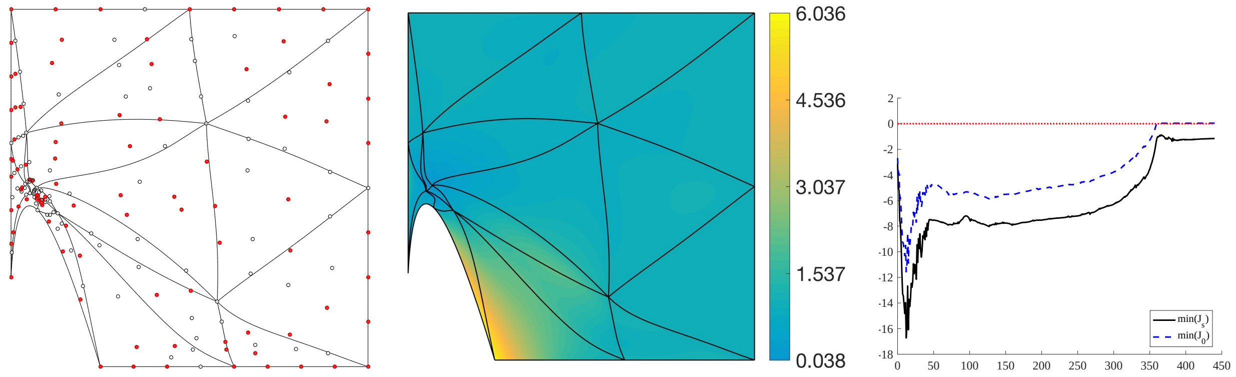


(a) Initial tangled C^1 physical mesh.

(b) Parametric mesh.

(c) Jacobian determinant of the initial C^1 physical mesh.

Figure 10: Initial quartic C^1 physical and parametric mesh. The red and white points represent free and dependent control points respectively. The initial physical mesh is tangled with a minimum Jacobian determinant value around -2.8.



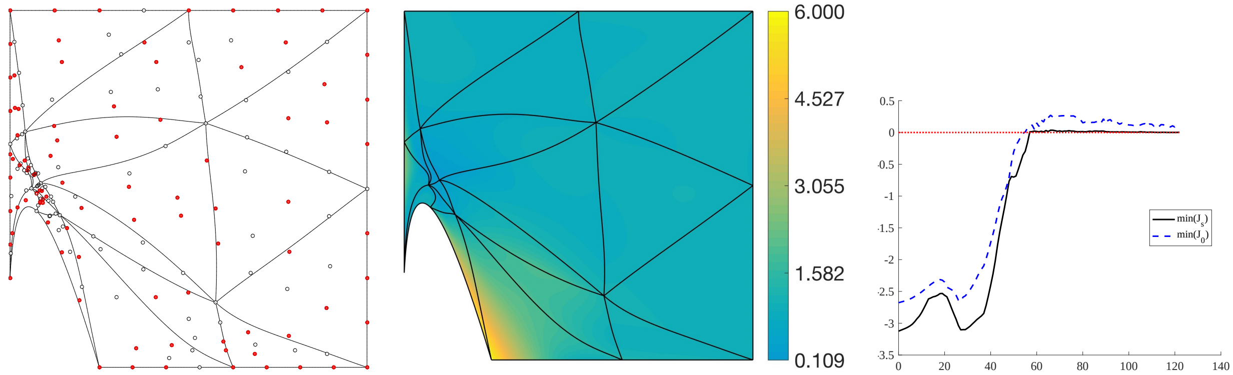
(a) Optimized C^1 mesh after 450 iterations.

(b) Jacobian determinant of the optimized mesh.

(c) Iteration history.

Figure 11: Optimization of the C^1 mesh in Figure 10 without explicit constraints on the Jacobian determinant. $\min(J_0)$ and $\min(J_s)$ are the minimum values of the Jacobian determinant in Eq. (25) evaluated at quadrature points and Jacobian coefficients in Eq. (26) respectively.

Next we optimize the mesh using the formulation in Eq. (39) with explicit constraints on the Jacobian determinant. As shown in Figure 12, this time it takes only 122 iterations for the optimization to converge, with all Jacobian coefficients positive. Particularly, it only takes about 50 iterations for the minimum Jacobian determinant to be above zero, while it takes about 380 iterations if no constraints are imposed. The minimum value of the evaluated Jacobian determinant is 0.109, about three folds of the value obtained without the constraints.



(a) Optimized C^1 mesh after 122 iterations.

(b) Jacobian determinant of the optimized mesh.

(c) Iteration history.

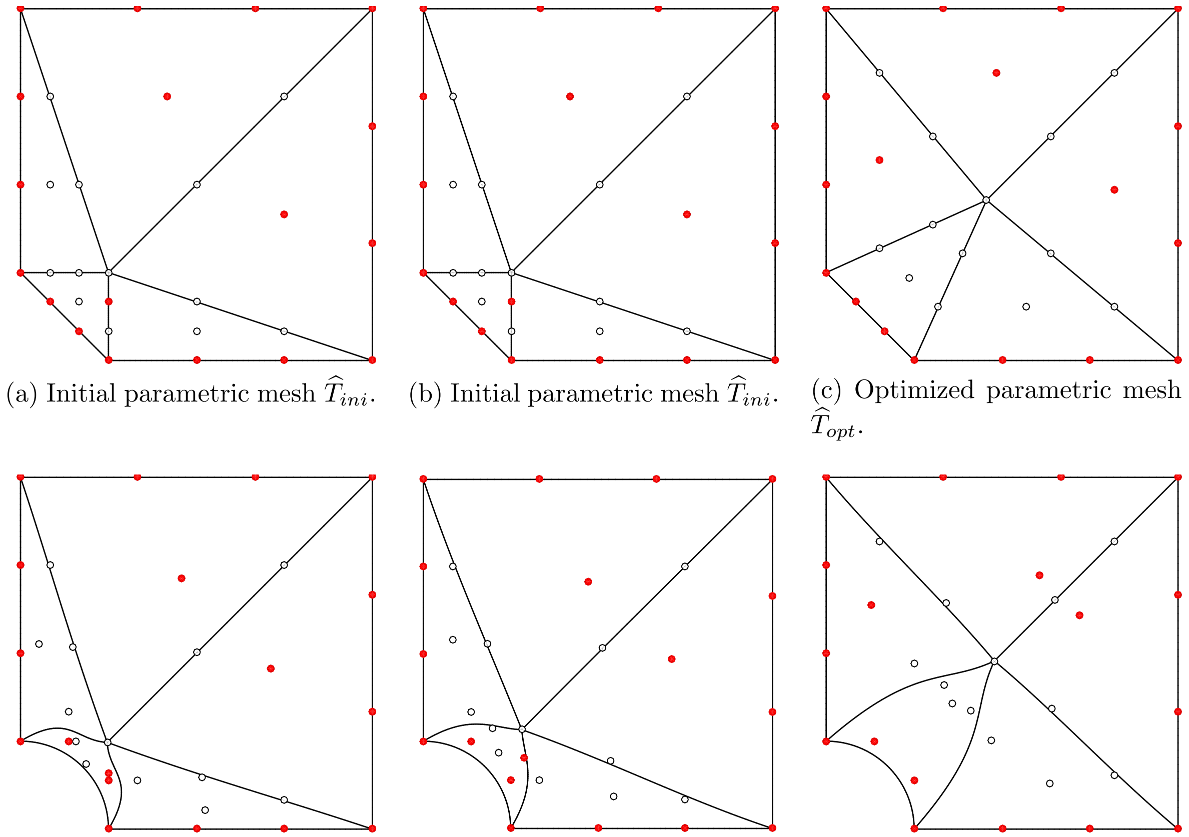
Figure 12: Optimization of the C^1 mesh in Figure 10 with explicit constraints on the Jacobian determinant.

It should be noted that, generally the more complex the geometry is, the more important the constraints on J_s play a role in improving the optimization efficiency. The reason is that the regularized distortion measure in Eq. (18) itself has the ability to untangle invalid meshes. For complex geometries, however, the explicit constraints on J_s enable the optimization to recover much quicker from invalid meshes. In addition, without the explicit constraints, some Jacobian coefficients J_s may still be negative after optimization even if all Jacobian determinant evaluated are positive. While with the constraints, all J_s are enforced to be positive after optimization, leading to a larger value of the minimum Jacobian determinant $\min(J_0)$ than without the constraints. In all of the following 2D and 3D examples, explicit constraints on Js are always imposed.

In the second example, we demonstrate the advantages of optimized parameterization in producing stiffness matrices with smaller condition number and achieving more accurate analysis results. A linear elasticity problem is solved using cubic C^1 elements with CT split.

As explained in [16], a pre-smooth C^1 geometric map is needed in order to obtain optimal convergence rate. To demonstrate the advantage of using optimized parameterization, three different pre-smooth C^1 geometric map are constructed. The first C^1 map is constructed using the initial parametric and physical mesh $\widehat{T}_{ini}$ and T_{ini}, as shown in the first column in Figure 13. Then we fix the parametric mesh $\widehat{T}_{ini}$ and optimize the physical mesh T_{ini} by minimizing its distortion with respect to $\widehat{T}_{ini}$. The optimized physical mesh T_{opt} and initial parametric mesh $\widehat{T}_{ini}$ is used in the second C^1 map, as shown in the second column in Figure 13. To construct the third C^1 map, we first optimize the parametric mesh $\widehat{T}_{ini}$ by minimizing the element distortion with respect to the equilateral triangle. Since the boundary points are fixed, only the location of the control point inside of the domain is changed. The optimized parametric mesh is denoted as $\widehat{T}_{opt}$, as shown in Figure 13c. Then the physical mesh corresponding to $\widehat{T}_{opt}$ is also optimized. The final optimized mesh T_{opt2} together with $\widehat{T}_{opt}$ are used to construct the third C^1 geometric map, as shown in the third column in Figure 13.

The Jacobian determinant of the three physical meshes are also shown in Figure 14. As can be seen, the unoptimized physical mesh T_{ini} has the least overall Jacobian determinant, and the optimized physical mesh T_{opt2} corresponding to the optimized parametric mesh has the largest overall Jacobian determinant.



(a) Initial parametric mesh $\widehat{T}_{ini}$.

(b) Initial parametric mesh $\widehat{T}_{ini}$.

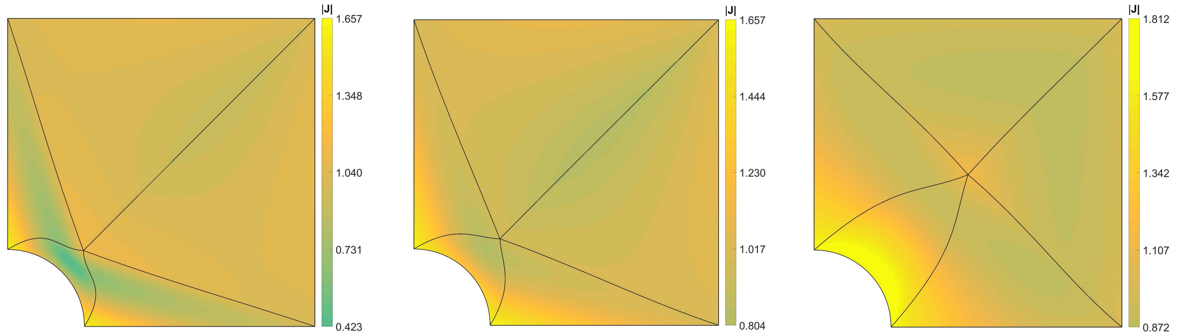
(c) Optimized parametric mesh $\widehat{T}_{opt}$.

(d) Initial physical mesh T_{ini} for $\widehat{T}_{ini}$.

(e) Optimized physical mesh T_{opt} for $\widehat{T}_{ini}$.

(f) Optimized physical mesh T_{opt2} for $\widehat{T}_{opt}$.

Figure 13: Construction of three different pre-smooth C^1 geometric maps. The first column shows the first C^1 map using the initial parametric and physical meshes. The second column shows the second C^1 map using the initial parametric mesh and the optimized physical mesh. The third column shows the third C^1 map using the optimized parametric mesh and physical mesh.



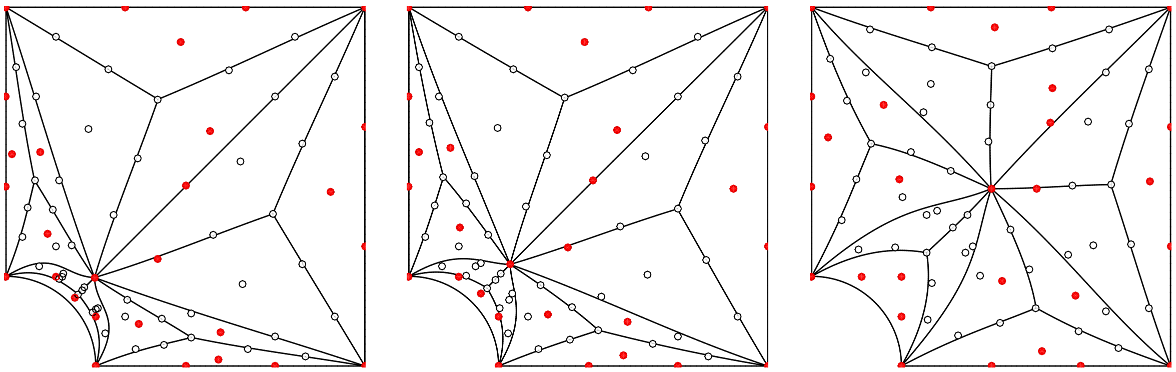
(a) Jacobian determinant of mesh T_{ini} in Figure 13d.

(b) Jacobian determinant of mesh T_{opt} in Figure 13e.

(c) Jacobian determinant of mesh T_{opt2} in Figure 13f.

Figure 14: Jacobian determinant of the C^1 meshes in Figure 13.

After constructing the C^1 pre-smooth geometric maps, we construct cubic C^1 basis with CT split for the three maps. Figure 15 shows the physical meshes for the three different parameterizations. The condition number of stiffness matrices and the L^2 error are shown in Figure 16. While the three parameterizations exhibit the same optimal convergence rates as expected, the first parameterization with unoptimized parametric and physical meshes gives the largest condition number of stiffness matrices and L^2 error, the third parameterization with both optimized parametric and physical mesh gives the smallest condition number and L^2 error.

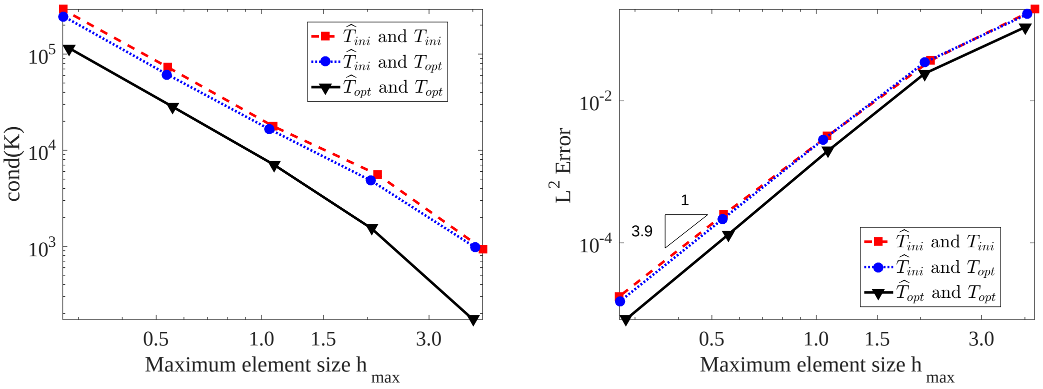


(a) Physical mesh in the first parameterization.

(b) Physical mesh in the second parameterization.

(c) Physical mesh in the third parameterization.

Figure 15: Cubic C^1 physical meshes with CT split for the three different parameterizations in Figure 13.

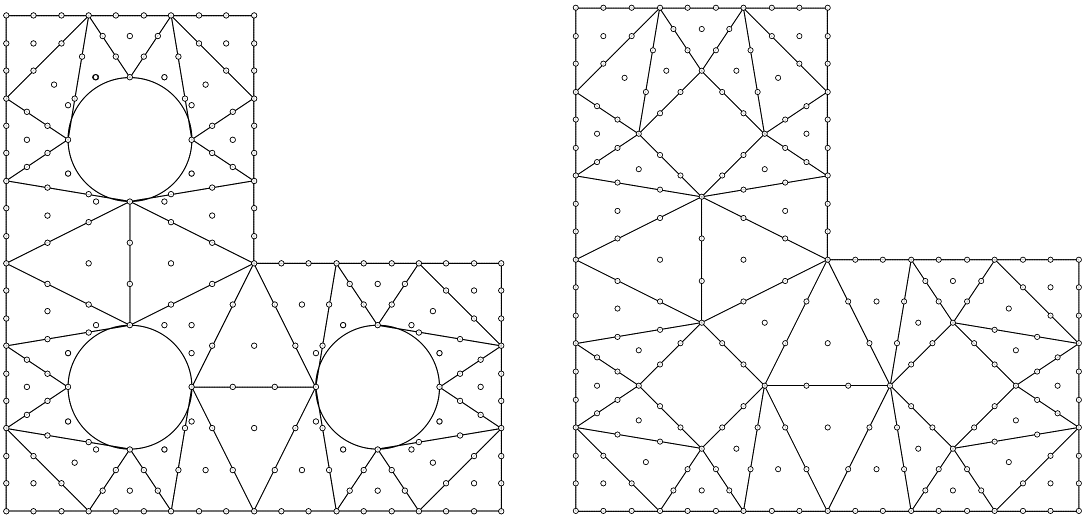


(a) Condition number of stiffness matrices.

(b) L^2 error.

Figure 16: Comparison of condition number and L^2 error for the three different parameterizations. While the three parameterizations exhibit the same optimal convergence rates as expected, the first parameterization with unoptimized parametric and physical meshes gives the largest condition number of stiffness matrices and L^2 error, the third parameterization with both optimized parametric and physical mesh gives the smallest condition number and L^2 error.

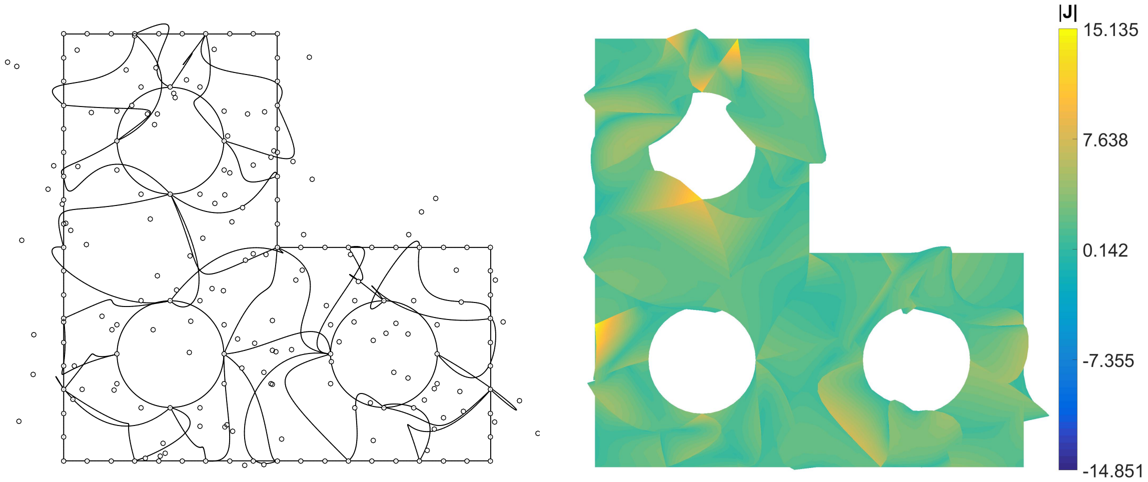
In the third example, we demonstrate that our method not only untangles mesh with local self-intersections but also improves its quality for both C^0 and C^r meshes. Figure 17 shows the initial physical and parametric meshes. In the physical mesh, there are some local self-intersections around the round boundaries represented by B-splines. In order to demonstrate our method is robust to severe distortions, we perturbed the physical mesh by randomly moving the interior control points, as shown in Figure 18a. The corresponding Jacobian determinant is also shown in Figure 18b.



(a) Physical mesh.

(b) Parametric mesh

Figure 17: Initial cubic physical and parametric meshes.



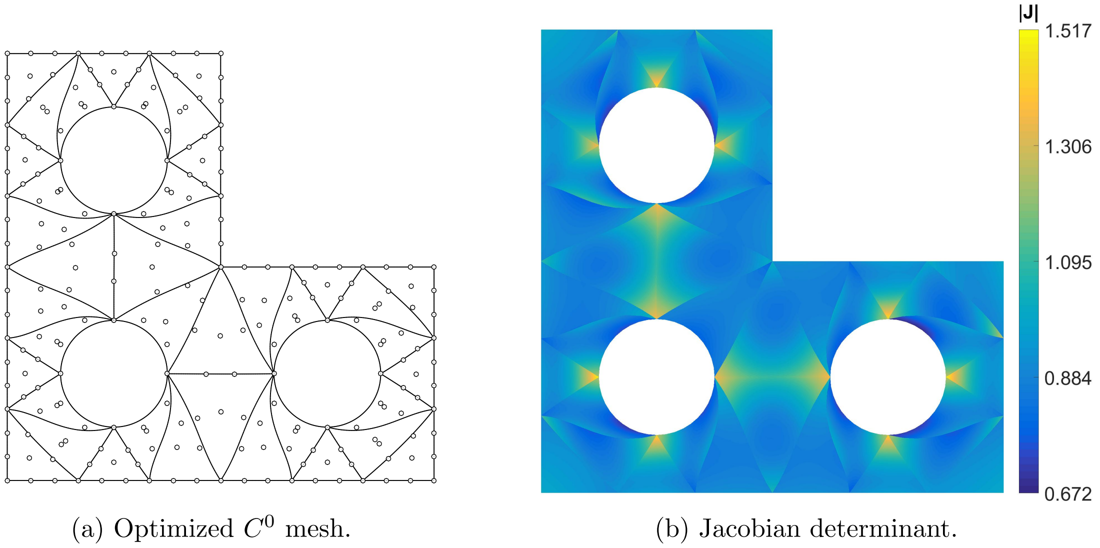
(a) Perturbed mesh.

(b) Jacobian of the perturbed mesh.

Figure 18: Perturbed C^0 mesh and its Jacobian determinant.

Figure 19 shows the optimized mesh with its Jacobian determinant. The optimization history is also shown in Figure 20. $\min(J_s)$ and $\min(J_0)$ are the minimum values of the Jacobian coefficients in Eq. (26) and evaluated Jacobian determinant at Gaussian points respectively. As can be seen, $\min(J_s)$ is positive in the optimized mesh. Due to the convex hull property, $J_s > 0$ is a conservative condition for $|\mathbf{J}| > 0$, this implies all elements are guaranteed to be valid.

To demonstrate our method can handle C^r parameterizations, we perturbed the cubic C^1 mesh with CT split, as shown in Figure 21. After optimization, the Jacobian



(a) Optimized C^0 mesh.

(b) Jacobian determinant.

Figure 19: Optimized C^0 mesh and its Jacobian determinant.

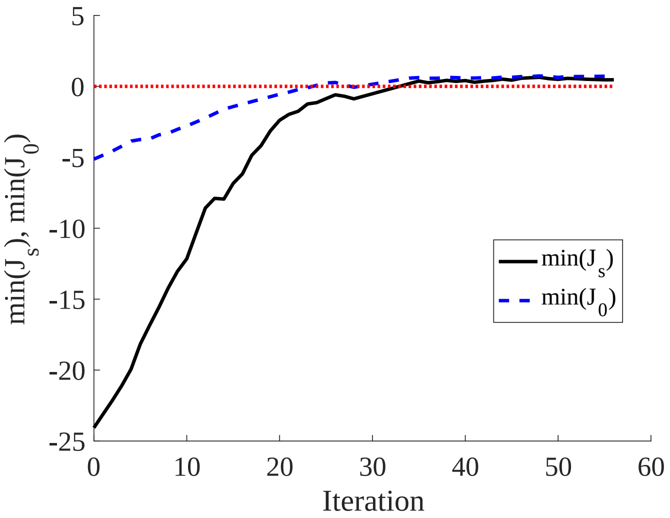
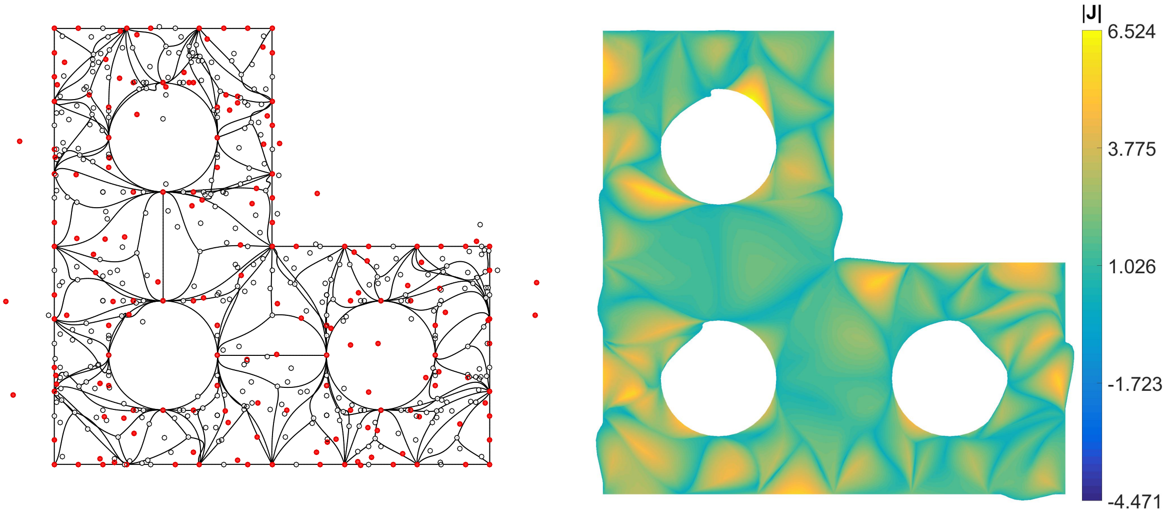


Figure 20: Optimization history for C^0 parameterization. $\min(J_s)$ becomes positive in the optimized mesh, meaning all elements are guaranteed to be valid.

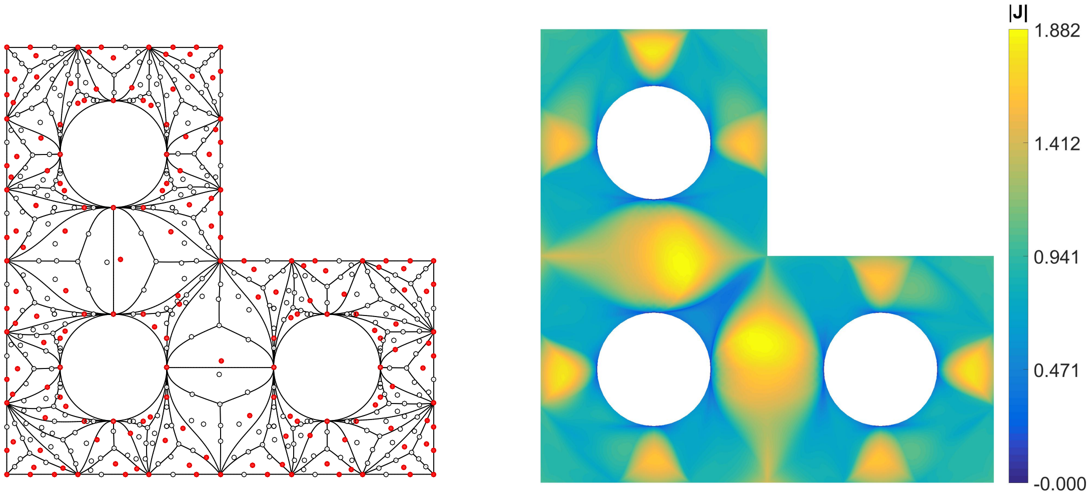
determinant all become positive except at some singular points on the round boundary, as shown in Figure 22. This is expected as explained in [16], degeneracy occurs when points are mapped from C^0 corner points in the parametric domain to C^1 points on the physical boundary. Nonetheless the optimization can still untangle the mesh and improve its quality. Note that the Jacobian determinant is smooth over the domain since the parameterization is C^1 continuous. The optimization history is also shown in Figure 23.



(a) Perturbed C^1 mesh.

(b) Jacobian determinant.

Figure 21: Perturbed C^1 mesh and its Jacobian determinant.



(a) Optimized C^1 mesh.

(b) Jacobian determinant.

Figure 22: Optimized C^1 mesh and its Jacobian determinant. The zero value of Jacobian only occurs at the singular points on the round boundary. The Jacobian determinant is smooth over the domain since the parameterization is C^1 continuous.

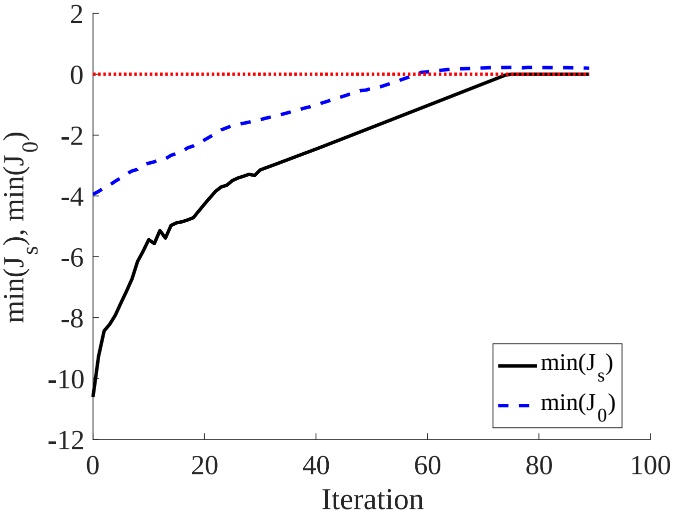
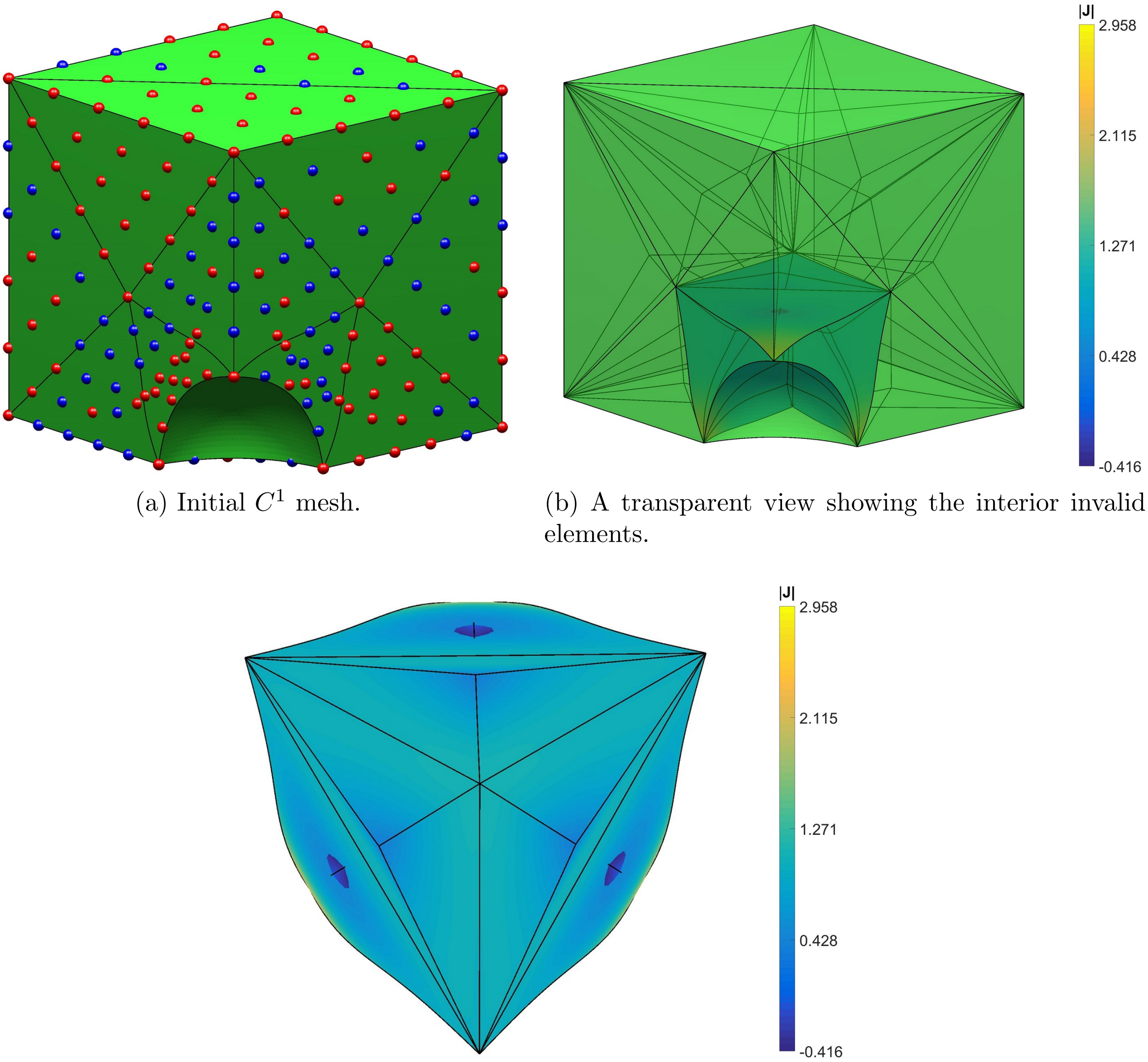


Figure 23: Optimization history for C^1 parameterization.

6.2 Optimizing Bézier tetrahedra

In this section, we present the numerical results of optimizing the parameterization of two 3D objects. The first example demonstrates the optimization of C^1 parameterization with Alfeld split. The geometry is shown in Figure 24. The initial C^1 mesh has several invalid elements around the cut-out area, with smallest Jacobian value of -0.416. Figure 24c shows an enlarged view of the invalid elements viewing from the back in Figure 24b. Now we show a valid parameterization can be obtained automatically through optimization.



(a) Initial C^1 mesh.

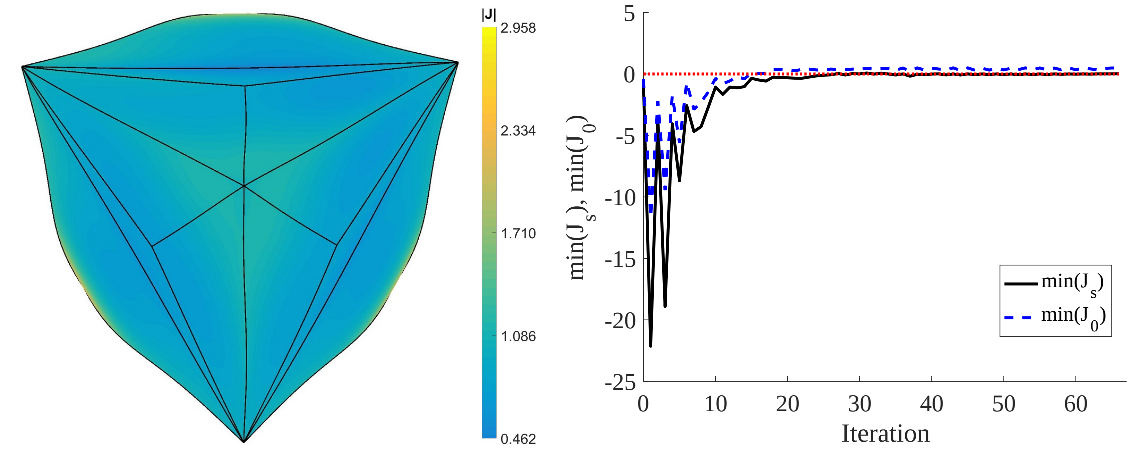
(b) A transparent view showing the interior invalid elements.

(c) Invalid elements viewed from the back in Figure 24b.

Figure 24: Initial C^1 mesh with invalid interior elements. The dark blue area are where negative Jacobian occurs.

Using the formulation presented in Section 5, a valid C^1 parameterization is obtained with smallest Jacobian value of 0.462, as shown in Figure 25. A plot of the optimization history is also shown in Figure 25b.

In the second 3D example, we present the results of optimizing a complex geometry. As shown in Figure 26, the initial mesh consists of 1965 degree 6 Bézier tetrahedral elements and 82682 control points. There are 457 invalid elements, as shown in blue

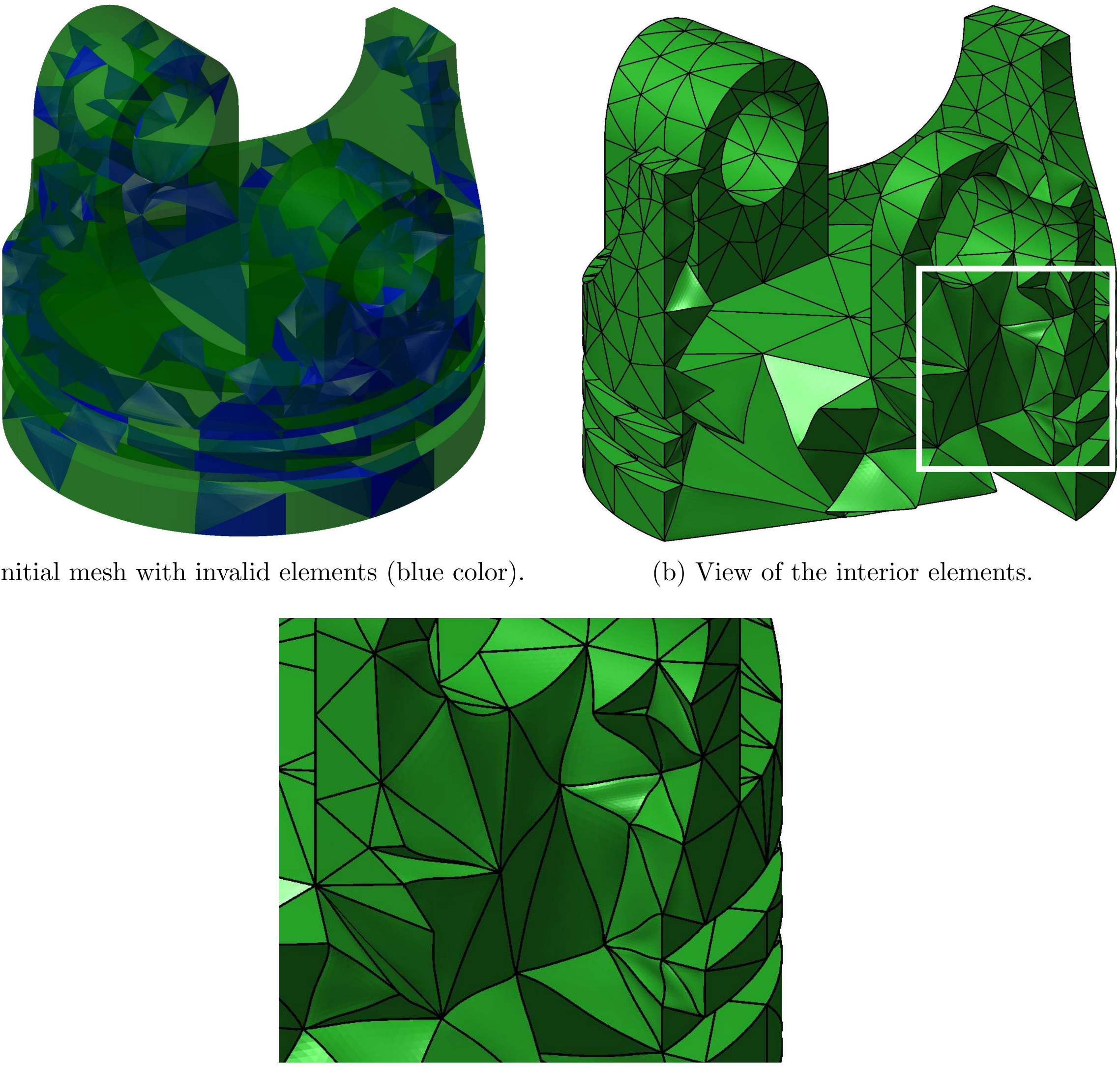


(a) Jacobian determinant after optimization.

(b) Optimization history.

Figure 25: Jacobian determinant after optimization and iteration history.

color in Figure 26a. The optimized mesh is shown in Figure 27. Comparing the zoom-in view of Figure 26c and 27b, we can see that the element edges after optimization are straighter than before, indicating the distortion has been reduced. The optimization history is also shown in Figure 28.

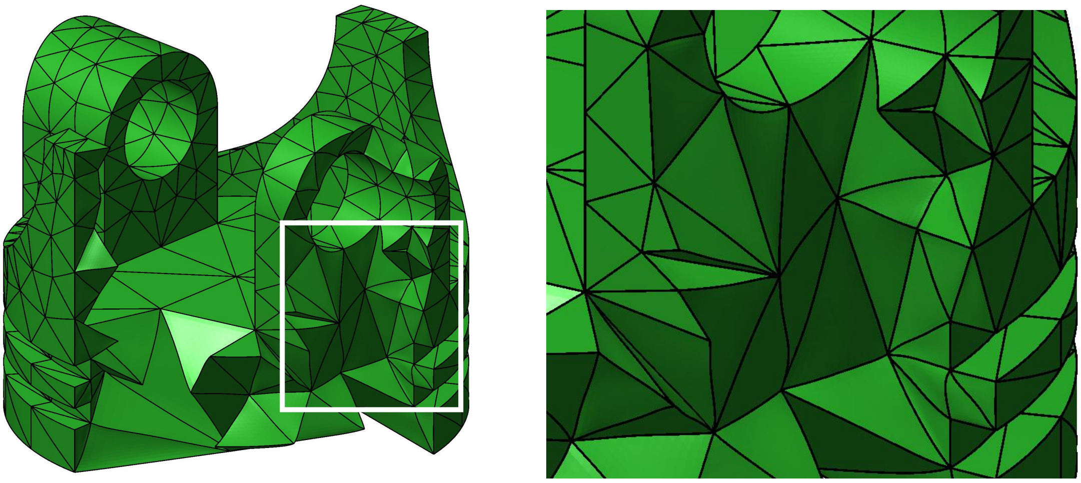


(a) Initial mesh with invalid elements (blue color).

(b) View of the interior elements.

(c) A zoom-in view of the interior elements in the white box in (b) showing the curved element edges.

Figure 26: Initial mesh of the piston head model. The mesh consists of 1965 degree 6 Bézier tetrahedral elements and 82682 control points, out of which 457 elements are invalid.



(a) Optimized mesh.

(b) A zoom-in view of the interior elements in the white box in (a).

Figure 27: Optimized mesh with view of interior elements. Figure 27b shows the element edges are much straighter than shown in Figure 26c, indicating the distortion has been reduced.

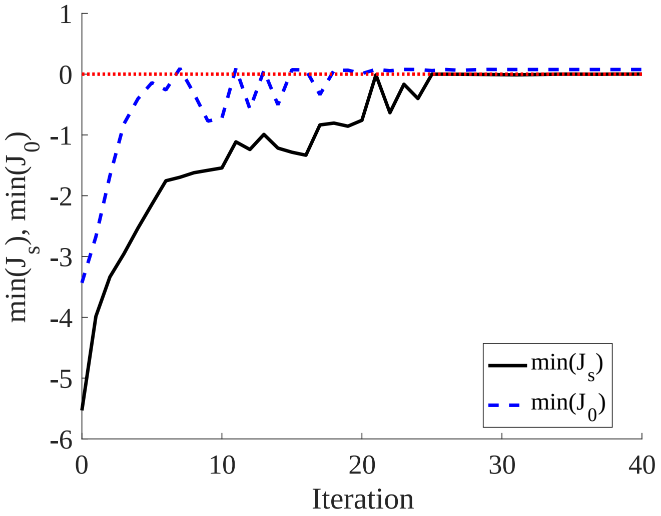


Figure 28: Optimization history.

7 Conclusion

In this paper, we present an optimization based approach to automatically generate high-quality high-order parameterization for isogeometric analysis on triangulations. The parameterization is based on triangular Bézier splines, where a B-spline represented boundary geometry is parameterized into a collection of either Bézier triangles for a 2D geometry or Bézier tetrahedra for a 3D geometry. Continuity constraints can be imposed on the Bézier ordinates of the elements to achieve C^r smoothness parameterizations. To obtain a high-quality parameterization, we developed an approach to optimize both the parametric and physical meshes to minimize the shape distortion of the elements with respect to well-defined elements. The shape distortion measured is defined based on the Jacobian of the parameterization, and its sensitivity is derived analytically and evaluated efficiently. Moreover, based on the convex hull property of triangular Bézier splines, we derive a sufficient condition to guarantee the optimized parameterization is free of local self-intersection. We show that improved parameterization led to better analysis results, in terms of smaller condition number of stiffness matrix and smaller error per same degree-of-freedoms. Various 2D and 3D numerical examples are presented. They demonstrate that our approach is able to simultaneously untangle and smooth the mesh and lead to high-quality high-order parameterization, even if the initial mesh has folded elements.

We note that in this work, the parametric and physical meshes are optimized sequentially, we first optimize the parametric mesh and then optimize the physical mesh. A better approach would be to optimize both meshes simultaneously. Another possible direction for future research is to optimize the boundary points of the parametric domain as well, in addition to the interior points. This will allow the optimization algorithm to search in a larger optimization space and possibly lead to better parameterization and mesh quality. One further possibility for future work would be to isolate the problematic elements (e.g. those with negative Jacobian) and their affected neighboring elements so that optimization is only conducted for these elements to improve overall efficiency.

8 Acknowledgment

This work is supported by NSF grant 1435072 and ARO grant W911NF-17-1-0020.

References

[1] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005.

[2] JA Cottrell, TJR Hughes, and A Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(41):4160–4183, 2007.

[3] Wolfgang A Wall, Moritz A Frenzel, and Christian Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33):2976–2988, 2008.

[4] Xiaoping Qian. Full analytical sensitivities in NURBS based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199(29):2059–2071, 2010.

[5] Xiaoping Qian and Ole Sigmund. Isogeometric shape optimization of photonic crystals via Coons patches. *Computer Methods in Applied Mechanics and Engineering*, 200(25):2237–2255, 2011.

[6] J Austin Cottrell, Alessandro Reali, Yuri Bazilevs, and Thomas JR Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5257–5296, 2006.

[7] 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(9):3402–3414, 2010.

[8] Kang Li and Xiaoping Qian. Isogeometric analysis and shape optimization via boundary integral. *Computer-Aided Design*, 43(11):1427–1437, 2011.

[9] M Aigner, C Heinrich, B Jüttler, E Pilgerstorfer, B Simeon, and A-V Vuong. Swept volume parameterization for isogeometric analysis. In *Proceedings of the 13th IMA International Conference on Mathematics of Surfaces XIII*, pages 19–44. Springer-Verlag, 2009.

[10] Gang Xu, Bernard Mourrain, Régis Duvigneau, and André Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design*, 45(2):395–404, 2013.

[11] Xilu Wang and Xiaoping Qian. An optimization approach for constructing trivariate B-spline solids. *Computer-Aided Design*, 46:179–191, 2014.

[12] Robert Schmidt, Roland Wüchner, and Kai-Uwe Bletzinger. Isogeometric analysis of trimmed NURBS geometries. *Computer Methods in Applied Mechanics and Engineering*, 241:93–111, 2012.

[13] Hendrik Speleers. A normalized basis for quintic Powell–Sabin splines. *Computer Aided Geometric Design*, 27(6):438–457, 2010.

[14] Hendrik Speleers, Paul Dierckx, and Stefan Vandewalle. Numerical solution of partial differential equations with Powell–Sabin splines. *Journal of Computational and Applied Mathematics*, 189(1):643–659, 2006.

[15] Noah Jaxon and Xiaoping Qian. Isogeometric analysis on triangulations. *Computer-Aided Design*, 46:45–57, 2014.

[16] Songtao Xia, Xilu Wang, and Xiaoping Qian. Continuity and convergence in rational triangular Bézier spline based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 297:292–324, 2015.

[17] Songtao Xia and Xiaoping Qian. Isogeometric analysis with bézier tetrahedra. *Computer Methods in Applied Mechanics and Engineering*, 316:782–816, 2017.

[18] Cunfu Wang, Songtao Xia, Xilu Wang, and Xiaoping Qian. Isogeometric shape optimization on triangulations. *Computer Methods in Applied Mechanics and Engineering*, 331:585–622, 2018.

[19] Luke Engvall and John A Evans. Isogeometric unstructured tetrahedral and mixed-element bernstein–bézier discretizations. *Computer Methods in Applied Mechanics and Engineering*, 319:83–123, 2017.

[20] Scott Lipton, John A Evans, Yuri Bazilevs, Thomas Elguedj, and Thomas JR Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199(5):357–373, 2010.

[21] Yuri Bazilevs, L Beirao da Veiga, J Austin Cottrell, Thomas JR Hughes, and Giancarlo Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(07):1031–1090, 2006.

[22] Gang Xu, Bernard Mourrain, Régis Duvigneau, and André Galligo. Parameterization of computational domain in isogeometric analysis: methods and comparison. *Computer Methods in Applied Mechanics and Engineering*, 200(23):2021–2031, 2011.

[23] Hendrik Speleers and Carla Manni. Optimizing domain parameterization in isogeometric analysis based on Powell–Sabin splines. *Journal of Computational and Applied Mathematics*, 289:68–86, 2015.

[24] Yongjie Zhang, Wenyan Wang, and Thomas JR Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering*, 249:185–197, 2012.

[25] Wenyan Wang, Yongjie Zhang, Lei Liu, and Thomas JR Hughes. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design*, 45(2):351–360, 2013.

[26] Xevi Roca, Abel Gargallo-Peiró, and Josep Sarrate. Defining quality measures for high-order planar triangles and curved mesh generation. In *Proceedings of the 20th International Meshing Roundtable*, pages 365–383. Springer, 2011.

[27] JM Escobar, E Rodrıguez, R Montenegro, G Montero, and JM González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2775–2787, 2003.

[28] Abel Gargallo-Peiró, X Roca, J Peraire, and J Sarrate. Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 103(5):342–363, 2015.

[29] Xiaoping Qian. Undercut and overhang angle control in topology optimization: A density gradient based integral approach. *International Journal for Numerical Methods in Engineering*, 111(3):247–272, 2017.

[30] Ming-Jun Lai and Larry L Schumaker. *Spline Functions on Triangulations*, volume 110. Cambridge University Press, 2007.

[31] Michael Andreas Matt. *Trivariate local lagrange interpolation and macro elements of arbitrary smoothness.* Springer Science & Business Media, 2012.

[32] Hong Dong. Spaces of bivariate spline functions over triangulation. *Approx. Theory Appl*, 7(1):56–75, 1991.

[33] Ray W Clough and James L Tocher. Finite element stiffness matrices for analysis of plates in bending. In *Proceedings of Conference on Matrix Methods in Structural Analysis*, pages 515–545, 1965.

[34] Michael JD Powell and Malcolm A Sabin. Piecewise quadratic approximations on triangles. *ACM Transactions on Mathematical Software (TOMS)*, 3(4):316–325, 1977.

[35] Abel Gargallo-Peiró, Xevi Roca, Jaime Peraire, and Josep Sarrate. Defining quality measures for mesh optimization on parameterized CAD surfaces. In *Proceedings of the 21st International Meshing Roundtable*, pages 85–102. Springer, 2013.

[36] Abel Gargallo-Peiró, Xevi Roca, and Josep Sarrate. A surface mesh smoothing and untangling method independent of the CAD parameterization. *Computational Mechanics*, 53(4):587–609, 2014.

[37] Patrick M Knupp. Algebraic mesh quality metrics. *SIAM journal on scientific computing*, 23(1):193–218, 2001.

[38] Abel Gargallo-Peiró, Xevi Roca, Jaime Peraire, and Josep Sarrate. Distortion and quality measures for validating and generating high-order tetrahedral meshes. *Engineering with Computers*, 31(3):423–437, 2015.

[39] Les Piegl and Wayne Tiller. *The NURBS book.* Springer Science & Business Media, 2012.

[40] K. Svanberg. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 2005.