Isogeometric Analysis with Bézier Tetrahedra

Songtao Xia and Xiaoping Qian *

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

Abstract

This paper presents an approach for isogeometric analysis of 3D objects using rational Bézier tetrahedral elements. In this approach, both the geometry and the physical field are represented by trivariate splines in Bernstein Bézier form over the tetrahedrangulation of a 3D geometry. Given a NURBS represented geometry, either untrimmed or trimmed, we first convert it to a watertight geometry represented by rational triangular Bézier splines (rTBS). For trimmed geometries, a compatible subdivision scheme is developed to guarantee the watertightness. The rTBS geometry preserves exactly the original NURBS surfaces except for an interface layer between trimmed surfaces where controlled approximation occurs. From the watertight rTBS geometry, a Bézier tetrahedral partition is generated automatically. By imposing continuity constraints on Bézier ordinates of the elements, we obtain a set of global C^r smooth basis functions and use it as the basis for analysis. Numerical examples demonstrate that our method achieve optimal convergence in C^r spaces and can handle complicated geometries.

Keywords Isogeometric analysis, Bézier tetrahedron, Trimmed NURBS geometry, Watertight geometry, C^r smoothness, Optimal convergence

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 represent both geometry and physical field. The exact CAD geometry is preserved in the analysis mesh, there is no need for the FEA systems to interact with the CAD systems during mesh refinement. The compatibility of IGA with CAD systems also motivates the development of isogeometric based shape optimization [2, 3, 4]. Other benefits of NURBS based IGA include its numerical efficiency and accuracy on a per-node basis, due to the high order of continuity of NURBS basis over classical C^0 Lagrange basis.

However, it is challenging to use tensor-product structure of NURBS to represent complicated geometries and to allow local refinement. A general 3D geometry is usually

^{*}Corresponding author. Email: qian@engr.wisc.edu

represented by multiple NURBS surfaces. It is a known challenge to generate NURBS volumetric parametrization for analysis from such complex geometries [5, 6, 7]. In addition, trimmed geometry is a common occurrence in CAD systems to represent complicated shapes. This poses another challenge for isogeometric analysis because of the model gaps presented in trimmed geometries. Gaps are usually unavoidable when two trimmed NURBS surfaces are stitched together [8]. Fixing the gaps with NURBS often leads to many small NURBS surfaces [9], making it even more difficult to generate volumetric representations.

To overcome such issues, T-splines [10, 8, 11] and Catmull-Clark subdivision surfaces [12] have been used to convert trimmed geometries to watertight representations, which can be used in isogeometric analysis [13, 14, 15, 16]. However, T-splines and Catmull-Clark subdivision surfaces also face the challenge of constructing volumetric parametrization from surface representations. Extraordinary nodes with degenerated continuity are usually unavoidable in the generated solid T-splines [17, 18] or Catmull-Clark subdivision surfaces [16] for geometries with complex topology. Optimal convergence rates in isogeometric analysis of general geometry with extraordinary vertices have yet to be demonstrated.

As an alternative, IGA on triangulations has emerged recently, where Powell-Sabin B-splines [19, 20, 21], triangular B-splines [22] and more generalized rational triangular Bézier splines (rTBS) [23, 24] over 2D triangular mesh are used as basis to solve partial differential equations (PDEs). All these methods possess the merits of efficient local refinement and high order of continuity. The advantages of Powell-Sabin B-splines are its normalized basis [25, 26, 27, 28] with nonnegativity and partition of unity, which are favorable for both geometric modeling and numerical analysis. The advantages of rTBS include their ability to exactly reproduce arbitrary NURBS geometry and to achieve optimal convergence in analysis [24]. Moreover, it can be extended to trivariate splines over tetrahedral partitions, which are able to represent complex solid geometries.

Bézier tetrahedral elements have been used together with trivariate tensor-product B-splines to form a hybrid volumetric representation from untrimmed B-spline surfaces [29]. However, the elements only join with C^0 smoothness and it does not handle trimmed geometries nor demonstrate optimal convergence. Trivariate splines of arbitrary smoothness have been used for data fitting [30] and numerical approximations of the 3D steady state Navier-Stokes equations in a cube [31]. In these works, the smoothness constraints are enforced through Lagrange multipliers. This leads to a singular matrix in the linear system and needs special treatment.

In this paper, we extend our earlier work of IGA with Bézier triangles [23, 24] to Bézier tetrahedra, using rational trivariate Bernstein polynomials over tetrahedral mesh as basis for analysis. We first convert a NURBS geometry (either trimmed or untrimmed) into a watertight rTBS geometry. A compatible subdivision scheme is developed to guarantee the watertightness of trimmed geometries. The converted rTBS geometry preserves exactly the original NURBS surfaces except for an interface layer triangles where the approximation controlled by user-specified threshold occurs. High order Bézier tetrahedral elements are then generated automatically from the watertight rTBS geometry, along with a parametric tetrahedral mesh. Finally for analysis we construct a set of C^r basis in the corresponding macro-element space by imposing continuity constraints on the Bézier ordinates of the elements [32, 33]. By using a smooth-refine-smooth scheme [24], optimal convergence rates are achieved with the C^r basis.

To represent trimmed geometries, we extract rTBS patches exactly from the trimmed

NURBS surfaces. This is different from previous approaches in [34, 35], where the trimmed NURBS surfaces are globally approximated using triangular Bézier patches [34] or cubic C^1 Clough-Tocher (CT) splines[35]. In [34], the surfaces are assumed to be trimmed along diametrically opposite patch corners and a system of linear equations is solved to ensure seamless join of two or more intersected surfaces. In [35], the mismatch between different trimmed surfaces are repaired by manually moving the boundary control points. In our approach, to guarantee the watertightness of the geometry, we develop a compatible subdivision scheme. We subdivide the trimming boundary curves on adjoining patches into compatible Bézier curves that can join seamlessly when stitched together. The main properties of the rTBS geometry obtained using our approach include the following.

- 1. The converted rTBS geometry is watertight.
- 2. The rTBS patches preserve the original trimmed NURBS surfaces exactly except for an interface layer of triangles touching the trimming boundary curves.
- 3. The rTBS patches can preserve one of the trimming boundary curves exactly.
- 4. The size of the interface layer can be further reduced by mesh refinement, leading to more rTBS patches that are exact with respect to the original trimmed geometry.
- 5. The approximation accuracy of the rTBS patches in the interface layer can be controlled by prescribed tolerances.

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 converting NURBS surfaces to rational triangular Bézier patches. The volumetric parametrization with Bézier tetrahedra is discussed in Section 4. Section 5 describes the details of isogeometric analysis with Bézier tetrahedra. Some numerical examples are presented in Section 6 followed by conclusion in Section 7.

2 Rational Bézier triangles and tetrahedra

To make the paper self-contained, in this section we briefly introduce bivariate and trivariate splines on Bézier triangles and tetrahedra respectively. The constraints for their smooth joins are also presented. For further reading please see [32, 33].

2.1 Rational Bézier triangles

A bivariate Bernstein polynomial of degree d is defined as

$$B_{ijk,d}(\boldsymbol{\xi}) = \frac{d!}{i!j!k!} \gamma_1^i \gamma_2^j \gamma_3^k, \quad i+j+k=d,$$
(1)

where $(\gamma_1, \gamma_2, \gamma_3)$ is the barycentric coordinate of a point $\boldsymbol{\xi} \in \mathbb{R}^2$ with respect to a triangle $\tau = {\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3}$. Every point $\boldsymbol{\xi} = (\xi_1, \xi_2)$ in the triangle τ can be written uniquely in the form

$$\boldsymbol{\xi} = \gamma_1 \mathbf{v}_1 + \gamma_2 \mathbf{v}_2 + \gamma_3 \mathbf{v}_3, \tag{2}$$

with

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

The set $\{B_{ijk,d}\}_{i+j+k=d}$ forms a basis for the space of degree d bivariate polynomials \mathcal{P}_d [32]. A triangular Bézier patch is defined as

$$\mathbf{b}(\boldsymbol{\xi}) = \sum_{i+j+k=d} \mathbf{p}_{ijk} B_{ijk,d}(\boldsymbol{\xi}), \qquad (3)$$

where \mathbf{p}_{ijk} represents a triangular array of control points. A rational Bézier triangle can be defined similarly as

$$\mathbf{b}(\boldsymbol{\xi}) = \sum_{i+j+k=d} \mathbf{p}_{ijk} \phi_{ijk,d}(\boldsymbol{\xi}), \tag{4}$$

with $\phi_{ijk,d}$ being the rational Bernstein basis

$$\phi_{ijk,d} = \frac{w_{ijk}B_{ijk,d}}{\sum_{i+j+k=d} w_{ijk}B_{ijk,d}} = \frac{w_{ijk}B_{ijk,d}}{w},\tag{5}$$

where w_{ijk} are the weights associated with the control points \mathbf{p}_{ijk} .

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

$$f(\boldsymbol{\xi}) = \sum_{i+j+k=d} t_{ijk} \phi_{ijk,d}(\boldsymbol{\xi}).$$
(6)

The t_{ijk} are called the Bézier ordinates of f. Their associated set of domain points are defined as

$$\mathcal{D}_{d,\tau} = \left\{ \mathbf{q}_{ijk} = \frac{i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3}{d}, i+j+k=d \right\}.$$
(7)

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



(a) Associated domain points of the Bézier ordinates t_{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.

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 [32]

$$\tilde{t}_{\rho,j,k} - \sum_{\mu+\nu+\kappa=\rho} \frac{\rho!}{\mu!\nu!\kappa!} t_{\mu,k+\nu,j+\kappa} \gamma_1^{\mu} \gamma_2^{\nu} \gamma_3^{\kappa} = 0, \quad j+k+\rho = d, \ \rho = 0, \cdots, r,$$
(8)

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 rTBS 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 2(b), the control points in each shaded triangle pair are coplanar. For better visualization of the underlying C^1 patch, the control net in Figure 2(b) is shifted up slightly.



Figure 2: Two 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 2(b), the control points in each shaded triangle pair are coplanar. For better visualization, the control net is shifted up slightly in Figure 2(b).

2.2 Rational 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

$$B_{ijk,d}(\boldsymbol{\xi}) = \frac{d!}{i!j!k!l!} \gamma_1^i \gamma_2^j \gamma_3^k \gamma_4^l, \quad i+j+k+l = d,$$
(9)

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 ,

$$\boldsymbol{\xi} = \gamma_1 \mathbf{v}_1 + \gamma_2 \mathbf{v}_2 + \gamma_3 \mathbf{v}_3 + \gamma_4 \mathbf{v}_4, \tag{10}$$

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 [32]. A rational Bézier tetrahedron and its basis are defined similarly as in Eq. (4) and (5), with the triple index (i, j, k) replaced by the quadruple index (i, j, k, l). The associated domain points of a tetrahedron $\tau = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ are

$$\mathcal{D}_{d,\tau} = \left\{ \mathbf{q}_{ijkl} = \frac{i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3 + l\mathbf{v}_4}{d}, i+j+k+l = d \right\},\tag{11}$$

as shown in Figure 3(a).

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 [32]

$$\tilde{t}_{mijk} - \sum_{\nu+\mu+\kappa+\delta=m} t_{\nu,i+\mu,k+\kappa,j+\delta} B^m_{\nu\mu\kappa\delta}(v_5) = 0, \quad i+j+k = d-m, \ m = 0, \cdots, r, \quad (12)$$

where $B^m_{\nu\mu\kappa\delta}$ are the Bernstein polynomials of degree *m* associated with the tetrahedron τ . Figure 3(b) 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 [33].



(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.



2.3 Spline spaces on tetrahedral partitions

Before introducing the spline spaces defined over tetrahedral partitions, we present two ways of refining a tetrahedron that will be used in this paper. One is called the Alfeld split, in which a tetrahedron is subdivided into four subtetrahedra by connecting its barycenter to each of its vertices. The other is 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. Let \widehat{T} be a tetrahedral partition of a bounded domain $\widehat{\Omega} \in \mathbb{R}^3$. The space of C^0 polynomials \mathcal{P}_d of degree d over \widehat{T} is defined as

$$\mathcal{S}_d^0(\widehat{T}) := \{ f \in C^0(\widehat{\Omega}) : f|_\tau \in \mathcal{P}_d \; \forall \tau \in \widehat{T} \}.$$
(13)

Given $0 \leq r < d$, the space of C^r polynomials of degree d and smoothness r over \widehat{T} is defined as

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

We will also use certain superspline subspaces of $\mathcal{S}_d^r(\widehat{T})$. Given $0 \leq r \leq \mu \leq \rho$, we define

$$\mathcal{S}_{d}^{r,\rho,\mu}(\widehat{T}) := \{ f \in \mathcal{S}_{d}^{r}(\widehat{T}) : f \in C^{\rho}(v), \ \forall v \in \mathcal{V}, f \in C^{\mu}(e), \ \forall e \in \mathcal{E} \},$$
(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.

3 Conversion of NURBS geometries to watertight rTBS patches

Since geometries are usually represented by NURBS in CAD systems, we first need to convert them to rTBS patches. In this section we describe the process of converting both untrimmed and trimmed NURBS surfaces to watertight rTBS patches.

3.1 Conversion of untrimmed NURBS surfaces

A NURBS surface of degree p, q in directions of u, v respectively is defined as

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} N_{i,p}(u) N_{j,q}(v) w_{ij} \mathbf{P}_{ij}}{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} N_{i,p}(u) N_{j,q}(v) w_{ij}}, \quad 0 \le u, v \le 1,$$
(16)

where \mathbf{P}_{ij} are the control points forming a $(n_1+1) \times (n_2+1)$ bidirectional control net, w_{ij} are the weights, and $N_{i,p}(u), N_{j,q}(v)$ are the nonrational B-spline basis functions defined with the knot vectors

$$U = \{u_0, u_1, \cdots, u_{n_1+p+1}\}, \quad V = \{v_0, v_1, \cdots, v_{n_2+q+1}\},\$$

in directions of u, v respectively. Usually it is convenient to represent a NURBS surface using homogeneous coordinates as

$$\mathbf{S}^{w}(u,v) = \sum_{i=0}^{n_{1}} \sum_{j=0}^{n_{2}} N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{ij}^{w},$$
(17)

where $\mathbf{P}_{ij}^w = (w_{ij}\mathbf{P}_{ij}, w_{ij})$. In this paper all calculations are performed using homogeneous coordinates, for convenience we omit the superscript w in $\mathbf{S}^w(u, v)$ and \mathbf{P}_{ij}^w when referring to the homogeneous form.

By increasing the multiplicity of the distinct knots in U and V to p and q respectively, for each knot interval $[u_{i_u}, u_{i_u+1}] \times [v_{i_v}, v_{i_v+1}]$ in the parametric domain, the corresponding surface can be extracted as a tensor-product Bézier patch

$$\widetilde{\mathbf{S}}(u,v) = \sum_{i=0}^{p} \sum_{j=0}^{q} B_{i,p}(u) B_{j,q}(v) \widetilde{\mathbf{P}}_{ij}, \quad 0 \le u, v \le 1,$$
(18)

where $B_{i,p}(u)$ and $B_{j,q}(v)$ are 1D Bernstein polynomials defined as $B_{i,d}(u) = \binom{d}{i}u^i(1-u)^{d-i}$. $\tilde{\mathbf{P}}_{ij}$ are the control points of the Bézier patch, which can be obtained from the NURBS control points \mathbf{P}_{ij} through Bézier extraction. For convenience, later in this paper we will omit the ~ in $\tilde{\mathbf{S}}(u, v)$ and $\tilde{\mathbf{P}}_{ij}$ when referring to Bézier patches and their control points. There should be no ambiguity aroused by this change considering the context.

After the Bézier extraction, each NURBS surface is represented by a collection of tensor-product Bézier patches. Each tensor-product Bézier patch can be exactly converted to two Bézier triangles [36].



(a) A bicubic tensor-product Bézier patch.

(b) The parametric domain of the Bézier patch is decomposed to two triangles by connecting one of the diagonals.

(c) The bicubic tensor-product Bézier patch is converted to two Bézier triangles of degree 6.

Figure 4: Exact conversion of a tensor-product Bézier patch to two Bézier triangles. The red points are control points and the blue dashed lines are control nets.

Given the control points \mathbf{P}_{mn} of a tensor-product Bézier patch of degree p, q, the control points \mathbf{p}_{ijk} of the Bézier triangles that represent the same patch can be explicitly computed as

$$\mathbf{p}_{ijk} = \frac{1}{\binom{p+q}{q}} \sum_{m=0}^{i} \sum_{n=\max\{0,j-p+m\}}^{\min\{j,q-i+m\}} \mathbf{P}_{mn} \binom{i}{m} \binom{j}{n} \times \binom{p+q-i-j}{p+n-m-j}$$
(19)

where i + j + k = p + q. While Eq.(19) gives the control points of one Bézier triangle from the rectangular array of control points $\{\mathbf{P}_{mn} : 0 \le m \le p; 0 \le n \le q\}$, the control points of the other Bézier triangle can be obtained using the same equation from the reversed array of control points $\{\mathbf{P}_{p-m,q-n} : 0 \le m \le p; 0 \le n \le q\}$. Note that the control points \mathbf{P}_{mn} and \mathbf{p}_{ijk} are in the form of homogeneous coordinates, so weights of all control points can be calculated at the same time. An illustration of converting a bicubic Bézier patch to two Bézier triangles of degree 6 is given in Figure 4.

An important property of the resulting rTBS representation is that, it preserves the original NURBS surfaces and their parametric continuity exactly. For example, for a



Figure 5: Conversion of NURBS surfaces to rTBS patches. The red points are control points, the blue dashed and black solid lines represent control nets and element edges respectively. (a) A CAD geometry represented by six bicubic NURBS surfaces $\{\mathbf{S}_i\}, i = 1, \dots, 6$. (b) Extracted tensor-product Bézier patches. (c) rTBS representation converted from the tensor-product Bézier patches.

bicubic NURBS surface without repeated knots in the interior, it has global C^2 continuity everywhere. After converting it to rational triangular Bézier patches using the process described above, the resulting rTBS of degree 6 represents the exact geometry as the original NURBS and is C^2 continuous globally.

Figure 5 gives an example of converting a NURBS geometry to rTBS geometry. Given a CAD geometry represented by a collection of NURBS surfaces $\{\mathbf{S}_i\}$, as shown in Figure 5(a), we first extract tensor-product Bézier patches through Bézier extraction (Figure 5(b)). Using the Eq.(19), each tensor-product Bézier patch is converted to two Bézier triangles. Thus all NURBS surfaces are converted exactly to rTBS representation, as shown in Figure 5(c), and the parametric continuity is preserved as well.

3.2 Conversion of trimmed NURBS surfaces

For complex geometries involving surfaces intersection and trimming, it is challenging to construct a watertight representation because of the trimmed boundaries leading to gaps between intersected surfaces. When two NURBS surfaces are intersected, the actual intersection curve is generally not rational, instead it is approximated using a NURBS curve in CAD system. This approximated intersection curve is projected back to the two parametric domains of the NURBS surfaces, which are again approximated using B-spline curves and referred as parametric trimming curves. The compositions of the NURBS surfaces and the parametric trimming curves give two physical trimming curves on the respective NURBS surface. Since the two physical trimming curves are mapped from two different parametric curves under different geometric maps, they usually do not match with each other. As a consequence, for a two-surface intersection, where there should be only one intersection curve in theory, there are actually three different representations of the intersection curve in the CAD system: the approximated intersection curve and two physical trimming curves. The differences among these curves, particularly the difference between the two physical trimming curves leads to gaps when stitching the two trimmed surfaces together.

Figure 6 gives an example where two bicubic NURBS surfaces \mathbf{S}_1 and \mathbf{S}_2 are intersected. The blue NURBS curve \mathbf{C}_0 (quintic in this example) in Figure 6(a) is the approximated NURBS intersection curve. \mathbf{c}_1 , \mathbf{c}_2 (both are quadratic in this example)

are the approximated parametric trimming curves and $\mathbf{C}_1 = \mathbf{S}_1(\mathbf{c}_1)$, $\mathbf{C}_2 = \mathbf{S}_2(\mathbf{c}_2)$ are the composited physical trimming curves. Each parametric trimming curve has a direction and the domain on its right side is trimmed off, which is indicated by the dashed lines in Figure 6(b). The NURBS surfaces \mathbf{S}_1 and \mathbf{S}_2 are then trimmed along \mathbf{C}_1 and \mathbf{C}_2 respectively (Figure 6(d)). However, since \mathbf{C}_0 , \mathbf{C}_1 and \mathbf{C}_2 are all different from each other, there will be gaps when stitching the two trimmed surfaces together, as can be seen in Figure 6(e) and 6(f). The gaps need to be fixed before the geometry is discretized for analysis.

The challenge of fixing the gaps is how to represent the trimmed patches accurately and efficiently, while allowing patches joining at the trimmed boundaries seamlessly. To solve this issue, we propose an approach to represent trimmed NURBS geometries using watertight rTBS patches. Specifically, we directly extract rTBS patches from the trimmed NURBS surfaces and develop a compatible subdivision scheme to guarantee the watertightness. The advantage of our approach is that we represent the original trimmed NURBS surfaces exactly except for an interface layer of triangles touching the trimming boundary curves. The size of this interface layer can be further reduced by mesh refinement, leading to more rTBS patches that are exact with respect to the original trimmed geometry. In addition, the approximation accuracy of the rTBS patches in the interface layer can be controlled by prescribed tolerances.

Our approach can be described as following. We first check the parametric domains of the trimmed NURBS surfaces to identify the intact triangles and an interface layer near the parametric trimming curves. Then for each intact triangle a corresponding rTBS patch can be extracted directly in the physical space. For the interface layers of two adjoining trimmed surfaces, we want the extracted rTBS patches to be compatible and join seamlessly at the trimming boundary curve. Thus we develop a compatible subdivision scheme that subdivides the two physical trimming curves of adjoining NURBS surfaces and a pre-selected physical trimming boundary curve into compatible Bézier curves, i.e. Bézier curves with the same set of corresponding end points. Then we extract rTBS patches from the interface layer as the compositions of the parametric triangles and the tensor-product Bézier patches. Finally, the compatible physical trimming curves are replaced with the same pre-selected trimming boundary curves, leading to a watertight representation with high accuracy.

The extraction of triangular Bézier subpatches from tensor-product Bézier patches is based on the results in [37]. The composition of a degree n Bézier triangle in the parametric domain and a tensor-product Bézier patch of degree (p,q) can be exactly represented as a Bézier triangle of degree n(p+q). The control points of the resulted triangular Bézier subpatch can also be calculated explicitly [37]. In this paper we only work with n = 1, that is, we use linear triangle in the parametric domain to extract rTBS patches from tensor-product Bézier patches. Let $\boldsymbol{\tau}(\boldsymbol{\xi}) = (u(\boldsymbol{\xi}), v(\boldsymbol{\xi}))$ be a linear Bézier triangle with control points $\boldsymbol{\tau}_{i'j'k'} = (u_{i'j'k'}, v_{i'j'k'})$, and $\mathbf{S}(u, v) = (x(u, v), y(u, v), z(u, v))$ be a tensor-product Bézier patch of degree (p, q) in the direction (u, v) respectively with control points \mathbf{P}_{mn} . The composition $\mathbf{b}(\boldsymbol{\xi}) = \mathbf{S}(\boldsymbol{\tau}(\boldsymbol{\xi})) = \mathbf{S}(u(\boldsymbol{\xi}), v(\boldsymbol{\xi}))$ is a rTBS patch of degree p + q:

$$\mathbf{b}(\boldsymbol{\xi}) = \sum_{|\mathbf{I}|=p+q} \mathbf{p}_{\mathbf{I}} B_{\mathbf{I},p+q}(\boldsymbol{\xi})$$
(20)

with control points

$$\mathbf{p}_{\mathbf{I}} = \sum_{\mathbf{I}^u + \mathbf{I}^v = \mathbf{I}} \frac{1}{\binom{p+q}{\mathbf{I}}} \mathbf{P}_{0,0}^{p,q}(u_{\mathbf{I}^u}^p, v_{\mathbf{I}^v}^q),$$
(21)



Figure 6: Two trimmed bicubic NURBS surfaces. (a) The two intersecting NURBS surfaces \mathbf{S}_1 and \mathbf{S}_2 and their intersection curve \mathbf{C}_0 . (b) Parametric domains and trimming curves \mathbf{c}_1 and \mathbf{c}_2 of \mathbf{S}_1 and \mathbf{S}_2 . (c) The original NURBS surfaces with control nets. (\mathbf{S}_2 has been moved away from \mathbf{S}_1 for better visualization.) The white points are the control points of \mathbf{S}_1 and \mathbf{S}_2 . The red and green curves $\mathbf{C}_1, \mathbf{C}_2$ are the composited physical trimming curves. (d) Trimmed NURBS surfaces where unwanted parts of \mathbf{S}_1 and \mathbf{S}_2 are trimmed off along \mathbf{C}_1 and \mathbf{C}_2 respectively. (e) The two trimmed surfaces stitched together. (f) Zoom-in view showing the gap between the trimmed surfaces inside the black rectangle in (d).

where $\mathbf{I}, \mathbf{I}^{u}, \mathbf{I}^{v}$ represent index triples of the form $(i, j, k), \mathbf{I}^{u} = \mathbf{I}_{1}^{u} + \dots + \mathbf{I}_{p}^{u}, \mathbf{I}^{v} = \mathbf{I}_{1}^{v} + \dots + \mathbf{I}_{q}^{v}$ and $\mathbf{I}_{i}^{u} = (i_{i}^{u}, j_{i}^{u}, k_{i}^{u})$, with $i_{i}^{u}, j_{i}^{u}, k_{i}^{u} \in \{0, 1\}, |\mathbf{I}^{u}| = |\mathbf{I}_{1}^{u}| + \dots + |\mathbf{I}_{p}^{u}| = p$ where $|\mathbf{I}_{i}^{u}| = i_{i}^{u} + j_{i}^{u} + k_{i}^{u}$, similarly for \mathbf{I}_{i}^{v} , and $|\mathbf{I}| = |\mathbf{I}^{u}| + |\mathbf{I}^{v}| = p + q$. The construction points $\mathbf{P}_{0,0}^{p,q}(u_{\mathbf{I}^{u}}^{p}, v_{\mathbf{I}^{v}}^{q})$ can be calculated recursively using the de Casteljau algorithm. In u direction it can be computed as

$$\mathbf{P}_{i,j}^{a,b}(u_{\mathbf{I}^{u}}^{a}, v_{\mathbf{I}^{v}}^{b}) = (1 - u_{\mathbf{I}^{u}_{a}})\mathbf{P}_{i,j}^{a-1,b}(u_{\mathbf{I}^{u}}^{a-1}, v_{\mathbf{I}^{v}}^{b}) + u_{\mathbf{I}^{u}_{a}}\mathbf{P}_{i+1,j}^{a-1,b}(u_{\mathbf{I}^{u}}^{a-1}, v_{\mathbf{I}^{v}}^{b}),$$
(22)

where $\mathbf{P}_{i,j}^{0,0} = \mathbf{P}_{ij}$, and the argument $(u_{\mathbf{I}^u}^a, v_{\mathbf{I}^v}^b)$ means $\mathbf{P}_{i,j}^{a,b}$ has to be evaluated by performing a de Casteljau calculations in u direction for the u parameter values $u_{\mathbf{I}_1^u}, u_{\mathbf{I}_2^u}, \cdots, u_{\mathbf{I}_a^u}$ and b de Casteljau calculations for the v parameter values $v_{\mathbf{I}_1^v}, v_{\mathbf{I}_2^v}, \cdots, v_{\mathbf{I}_b^v}$. And a similar recursion in the v direction can also be computed.

An example is given in Figure 7, where a Bézier triangle of degree 6 is extracted from a bicubic tensor-product Bézier patch. As can be seen, the previous conversion of a tensor-product Bézier patch to two Bézier triangles is just a special case of this problem.



Bézier patch.

(c) The composition of the linear triangle and bicubic tensorproduct Bézier patch is a Bézier triangle of degree 6.

Figure 7: Triangular Bézier subpatch on a tensor-product Bézier patch.

Now we describe the detailed steps of our method to fix the gap shown in Figure 6.

- 1. Bézier extraction of intact triangles. In the parametric domains of the NURBS surfaces $\mathbf{S}_1, \mathbf{S}_2$, we first find the triangles that are intact from the trimming operation. As shown in Figure 8(a) and 8(b), the intact triangles are shaded. For each intact triangle in the parametric domain, there is a corresponding Bézier triangle in the physical space that can be extracted using Eq.(19). The resulting rTBS patches of the intact triangles of $\mathbf{S}_1, \mathbf{S}_2$ are shown in Figure 8(c).
- 2. Compatible subdivision of physical trimming curves and a pre-selected trimming boundary curve. This step can be further explained in the following sub-steps.
 - (a) Select a curve from $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2$ as the trimming boundary curve that will be used in the converted rTBS geometry. In this example, we choose \mathbf{C}_0 since it is relatively closer to both \mathbf{S}_1 and \mathbf{S}_2 than the other two curves. However, if we want the trimming boundary curve lies exactly on, e.g. \mathbf{S}_1 , then we can choose \mathbf{C}_1 as the trimming boundary curve.
 - (b) Subdivide the parametric trimming curves $\mathbf{c}_1, \mathbf{c}_2$ and the selected trimming boundary curves \mathbf{C}_0 into Bézier curves. Since the composited curves are calculated in terms of the Bézier form, we subdivide $\mathbf{c}_1, \mathbf{c}_2$ into Bézier curves at



Figure 8: Extraction of triangular Bézier patches from intact triangles. (a-b) Identification of intact triangles (shaded) of \mathbf{S}_1 , \mathbf{S}_2 . (c) Extracted rTBS patches from the intact triangles.

their knot points and the intersections with the knot lines of $\mathbf{S}_1, \mathbf{S}_2$, as marked by red and green squares respectively in Figure 9(a) and 9(b). In this example, \mathbf{c}_1 and \mathbf{c}_2 are subdivided into 11 and 10 Bézier curves respectively. Using the point inversion algorithm [38], we also find the knot values of $\mathbf{c}_1, \mathbf{c}_2$ corresponding to these intersections. The computed knot values for all subdivision points are denoted as Ξ_1, Ξ_2 for $\mathbf{c}_1, \mathbf{c}_2$ respectively.



Figure 9: Subdivision of \mathbf{c}_1 (red curve in (a)), \mathbf{c}_2 (blue curve in (b)) into Bézier curves. The subdivision points (marked as squares) include the knot points of $\mathbf{c}_1, \mathbf{c}_2$ and their intersections with the knot lines of $\mathbf{S}_1, \mathbf{S}_2$ respectively.

We also subdivide the selected trimming boundary curve C_0 into 9 Bézier curves at the values in its knot vector Ξ_0 . Figure 10 shows the subdivided curves C_0 and the composited curves C_1, C_2 . As can be seen, although the three curves C_0, C_1, C_2 are almost the same, the subdivided curve segments do not match at the subdivision points.

(c) Compatible subdivision of $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2$ such that the distances between corresponding end points of the subdivided segments are smaller than a prescribed tolerance δ_1 . The value δ_1 is used to control the approximation error caused by replacing the subdivided segments of $\mathbf{C}_1, \mathbf{C}_2$ with the corresponding segments of \mathbf{C}_0 . Since the difference among the three curves $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2$ depends on the specific CAD geometry, the value δ_1 can be determined by trial-and-



Figure 10: Initial subdivision of the selected trimming boundary curve \mathbf{C}_0 (blue) and the composited curves \mathbf{C}_1 (red), \mathbf{C}_2 (green). The subdivision points (marked as squares) of the three curves usually do not match with each other. (a) Overlapped view. (b) Side-by-side view.

error. Another alternative is to use a minimization algorithm to make the difference between corresponding subdivision points as small as possible. For the examples in this paper, we choose $\delta_1 = 10^{-5}$. Now suppose we have three knot values $\xi_{1,i} \in \Xi_1, \xi_{2,j} \in \Xi_2$, and $\xi_{0,k} \in \Xi_0$, as shown in Figure 11(a), we explain this idea using the following steps. To simplify the notation, we define $\mathbf{C}_1(\xi_{1,i}) = \mathbf{S}_1(\mathbf{c}_1(\xi_{1,i}))$ and $\mathbf{C}_2(\xi_{2,j}) = \mathbf{S}_2(\mathbf{c}_2(\xi_{2,j}))$.

i. For every knot point $\mathbf{C}_1(\xi_{1,i})$, $\mathbf{C}_2(\xi_{2,j})$ on \mathbf{C}_1 and \mathbf{C}_2 respectively, find the corresponding point $\mathbf{C}_0(\xi_{01,i})$, $\mathbf{C}_0(\xi_{02,j})$ on \mathbf{C}_0 such that their distance is smaller than the prescribed value δ_1 , i.e. $\forall \xi_{1,i} \in \Xi_1$, find $\xi_{01,i}$ such that

$$|\mathbf{C}_0(\xi_{01,i}) - \mathbf{C}_1(\xi_{1,i})| < \delta_1,$$

and $\forall \xi_{2,j} \in \Xi_2$, find $\xi_{02,j}$ such that

$$|\mathbf{C}_0(\xi_{02,j}) - \mathbf{C}_2(\xi_{2,j})| < \delta_1$$

The set of values $\{\xi_{01,i}\}$, $\{\xi_{02,j}\}$ are denoted as knot vectors $\Xi_{01} = \{\xi_{01,i}\}$ and $\Xi_{02} = \{\xi_{02,j}\}$ respectively. The illustration of this step is shown in Figure 11(b).

ii. For every knot point $\mathbf{C}_0(\xi_{0,k})$ and $\mathbf{C}_0(\xi_{02,j})$ (obtained in Step 2(c)i) on \mathbf{C}_0 , find the corresponding point $\mathbf{C}_1(\xi_{10,k})$ and $\mathbf{C}_1(\xi_{10,j})$ on \mathbf{C}_1 such that their distance is smaller than the prescribed value δ_1 , i.e. $\forall \xi_{0,k} \in \Xi_0$ and $\xi_{02,j}$ in Step 2(c)i, find $\xi_{10,k}$ and $\xi_{10,j}$ such that

$$|\mathbf{C}_1(\xi_{10,k}) - \mathbf{C}_0(\xi_{0,k})| < \delta_1, \quad |\mathbf{C}_1(\xi_{10,j}) - \mathbf{C}_0(\xi_{02,j})| < \delta_1.$$

The set of values $\{\xi_{10,j}, \xi_{10,k}\}$ is denoted as $\Xi_{10} = \{\xi_{10,j}, \xi_{10,k}\}$. The illustration is shown in Figure 11(c).

iii. For every knot point $\mathbf{C}_0(\xi_{0,k})$ and $\mathbf{C}_0(\xi_{01,i})$ (obtained in Step 2(c)i) on \mathbf{C}_0 , find the corresponding point $\mathbf{C}_2(\xi_{20,k})$ and $\mathbf{C}_2(\xi_{20,i})$ on \mathbf{C}_2 such that their distance is smaller than the prescribed value δ_1 , i.e. $\forall \xi_{0,k} \in \Xi_0$ and $\xi_{01,i}$ in Step 2(c)i, find $\xi_{20,k}$ and $\xi_{20,i}$ such that

$$|\mathbf{C}_2(\xi_{20,k}) - \mathbf{C}_0(\xi_{0,k})| < \delta_1, \quad |\mathbf{C}_2(\xi_{20,i}) - \mathbf{C}_0(\xi_{01,i})| < \delta_1.$$

The set of values $\{\xi_{20,i}, \xi_{20,k}\}$ is denoted as $\Xi_{20} = \{\xi_{20,i}, \xi_{20,k}\}$. The illustration is shown in Figure 11(d).



Figure 11: Compatible subdivision of the three curves C_0 (blue), C_1 (red) and C_2 (green).

Following the above steps, we obtain three new knot vectors $\Xi'_0 = \Xi_0 \cup \Xi_{01} \cup \Xi_{02}$, $\Xi'_1 = \Xi_1 \cup \Xi_{10}$ and $\Xi'_2 = \Xi_2 \cup \Xi_{20}$. Now if we subdivide \mathbf{C}_0 and $\mathbf{C}_1, \mathbf{C}_2$ at the parametric values in Ξ'_0, Ξ'_1 and Ξ'_2 respectively, for every $\xi'_0 \in \Xi'_0$, there are corresponding values $\xi'_1 \in \Xi'_1$ and $\xi'_2 \in \Xi'_2$ that

$$|\mathbf{C}_0(\xi_0') - \mathbf{C}_1(\xi_1')| < \delta_1 \quad \& \quad |\mathbf{C}_0(\xi_0') - \mathbf{C}_2(\xi_2')| < \delta_1.$$

There are a few remarks for this compatible subdivision scheme.

i. To avoid very thin elements, if there are two values in any knot vector Ξ'_0 , Ξ'_1 and Ξ'_2 that are closer than δ_2 ($\delta_2 = 0.05$ in this paper), we remove one of them, and also remove the corresponding values in the other two knot vectors. If one of the two close values belongs to Ξ_0 , Ξ_1 or Ξ_2 , then it is kept and the other one is removed. Otherwise, it is up to the user to decide which one of the two is removed. This is to make sure the original knot vectors Ξ_0 , Ξ_1 and Ξ_2 of the three curves \mathbf{C}_0 and \mathbf{c}_1 , \mathbf{c}_2 remain unchanged after subdivision.

ii. According to [37], the composition of tensor-product Bézier patches of degree p, q and Bézier curve of degree n are Bézier curves of degree n(p+q). Although we can compute such high order Bézier curves exactly, to reduce the complexity of the problem, we approximate c₁, c₂ using piecewise linear segments c
₁ and c
₂ by connecting the knot points. This will reduce the degree of the composited Bézier curves C
₁ = S₁(c
₁) and C
₂ = S₂(c
₂) to 6, compared to degree 12 of S₁(c
₁) and S₂(c
₂). To ensure a good approximation, we check the length of each segment of c
₁ and c
₂ defined by each knot span of E
₁, E
₂, and the length of the line in c
₁ and c
₂ that approximates it. If the normalized difference of the lengths is larger than a prescribed threshold (3% in this paper), a new knot is inserted in E
₁ (or C
₂) is replaced with two new line segments. At the same time, the knot vectors E
₀', E
₁' and E
₂' are also updated using step 2c. This process continues until it convergences.

Figure 12 shows the compatible physical trimming curves $\overline{\mathbf{C}}_1, \overline{\mathbf{C}}_2$ and the selected trimming boundary curve \mathbf{C}_0 after subdivision. The corresponding segments approximating the subdivided parametric trimming curves $\overline{\mathbf{c}}_1$ and $\overline{\mathbf{c}}_2$ are also shown.

- 3. Bézier extraction of the interface layer triangles. We define the interface layer as the domain sandwiched between the boundary of the intact triangles and the parametric trimming curve, as shown in the shaded color in Figure 13. With the approximated parametric trimming curves $\bar{\mathbf{c}}_1$ and $\bar{\mathbf{c}}_2$ in previous steps, we discretize the interface layers into triangles. The composition of each triangle with its underlying tensor-product Bézier patch results in a Bézier triangle of degree 6. The converted rTBS geometry preserves the original NURBS surfaces and their continuities exactly. However, they generally do not match at the trimming boundary where they are supposed to meet. As shown in Figure 13(d), the control points of the Bézier triangles along the trimmed boundaries do not match with each other. So we need another step to fix the gap.
- 4. Replace both physical trimming curves $\overline{\mathbf{C}}_1, \overline{\mathbf{C}}_2$ with \mathbf{C}_0 . To make sure the trimming boundaries on the two trimmed NURBS surfaces match with each other exactly, we replace both $\overline{\mathbf{C}}_1, \overline{\mathbf{C}}_2$ with \mathbf{C}_0 obtained in previous steps. This results in a seamless join of the rTBS patches converted from the two trimmed NURBS surfaces, as shown in Figure 14. In addition, the resulting rTBS geometry preserves exactly the input NURBS surfaces except for the interface layer triangles (dark shaded elements in Figure 14). Moreover, this interface layer can be further refined to improve the accuracy.

Note that, in this example the interface layer is triangulated simply using the points on the boundaries only. When the number of points on the boundary becomes excessive, elements can become skewed. The quality of the elements can be improved by inserting additional points in the interior of the interface layer.



Figure 12: (a-b) Approximation of the final subdivided trimming curves \mathbf{c}_1 and \mathbf{c}_2 using piecewise linear segments $\mathbf{\bar{c}}_1$ and $\mathbf{\bar{c}}_2$ respectively. (c-e) Compatible Bézier curves $\mathbf{\overline{C}}_1$ (red), \mathbf{C}_0 (blue), $\mathbf{\overline{C}}_2$ (green) after subdivision. The red, blue, and green squares indicate the subdivision points corresponding to points on $\mathbf{\overline{C}}_1$, \mathbf{C}_0 , $\mathbf{\overline{C}}_2$ respectively.



Figure 13: Extraction of rTBS patches (dark shaded) from the interface layer triangles. (a-b) Triangulation of the interface layer of \mathbf{S}_1 and \mathbf{S}_2 . (c) Extracted rTBS patches with control points. (d) zoom-in view inside the white box in (c) shows the edges along the trimming boundary do not match since they are evaluated on $\mathbf{S}_1, \mathbf{S}_2$ respectively.



Figure 14: Replacement of the trimmed boundary edges with the selected trimming boundary curve \mathbf{C}_0 . (a) The blue points are the control points of the subdivided Bézier segments of \mathbf{C}_0 . (b) Zoom-in view inside the white box in (a) shows a seamless join of the rTBS patches.

We would like to remark that, the entire process of generating watertight rTBS geometry from trimmed NURBS geometry can be fully automatic with the given tolerance values such as δ_1, δ_2 . In our examples, given the IGES files of a trimmed NURBS geometry and the tolerance values, the watertight rTBS geometries can be generated in real time.

4 Volumetric parametrization with rational Bézier tetrahedra

After converting the NURBS surfaces to watertight rTBS patches, a tetrahedral partition can be generated using the linearized version of the triangular Bézier boundary surface. Then we can elevate the degree of the linear tetrahedral elements and move the control points on the boundary surface to recover the high order triangular Bézier boundary surface. To improve the mesh quality, the interior control points are also moved by solving an elasticity problem [39]. In the end, a high order Bézier tetrahedral partition is generated from the rTBS geometry, where the boundary surfaces are preserved exactly. Meanwhile, the generated linear tetrahedral mesh can be used as the parametric mesh after degree elevation. A set of basis functions is then constructed in the parametric space to be used for isogeometric analysis.

4.1 Tetrahedralization of watertight rTBS patches

Once we have a watertight geometry represented by rTBS patches, we can parametrize it into Bézier tetrahedral elements. We explain the process with an example shown in Figure 15. Given a rTBS geometry **b** shown in Figure 15(a), by connecting the vertices of the Bézier triangles in **b**, a linearized version $\overline{\mathbf{b}}$ of the triangular Bézier boundary surface is obtained (Figure 15(b)). From $\overline{\mathbf{b}}$, a tetrahedral partition \overline{T} is generated by a tetrahedral mesh generator (see Figure 15(b), we use Tetgen [40] for the examples in this paper.). The degree of the linear tetrahedral partition \overline{T} is then elevated to 6, yielding the parametric mesh \hat{T}_0 (Figure 15(c)). Then by moving the control points on the boundary surface of \hat{T}_0 to the position on the triangular Bézier boundary surface **b**, we create a high order Bézier tetrahedral partition T_0 of the geometry, as shown in Figure 15(d). To alleviate possible element distortion in the mesh, we move the interior control points in T_0 by solving an elastic equation and smoothing their weights by solving a Laplacian equation. Optimization technique such as [21, 7] can also be used. To show the interior of the mesh, some elements are removed, as shown in Figure 15(e) and 15(f).

Note that, while the parametric and physical mesh \hat{T}_0 and T_0 are ready to be used for isogeometric analysis, the geometric map $\mathbf{G}_0(\boldsymbol{\xi}): \hat{T}_0 \mapsto T_0$ is only C^0 , including the map of the boundary surface. When generating the parametric mesh \hat{T}_0 , the linearization process of the triangular Bézier patches may create many sharp corners and edges in \hat{T}_0 that correspond to continuous points and edges in T_0 . Therefore when imposing C^r constraints to obtain a C^r geometric map, these sharp corners and edges will create singularities in the corresponding physical mesh. In order to preserve the mapping continuity without introducing singularities, ideally a new parametric mesh \hat{T} with no additional sharp corners or edges as in the input NURBS geometry is needed. While such \hat{T} can be generated easily for simple geometries such as shown in Figure 15, for complex geometries, singularities may be unavoidable to preserve the C^r geometric map. In Figure 15, the geometric map of the boundary surfaces $\mathbf{G}_{\Gamma} : \widehat{T}_{\Gamma} \mapsto T_{0,\Gamma}$ is continuous up to the same order as the input NURBS surfaces.



Figure 15: Tetrahedralization of a geometry represented by watertight rTBS patches. The red points are control points, the blue dashed and black solid lines represent control nets and element edges respectively. (a) A watertight geometry **b** represented by rTBS patches. (b) Linearized surface triangulation $\overline{\mathbf{b}}$. (c) Generated parametric mesh \hat{T}_0 after degree elevation of the linear tetrahedral partition \overline{T}_0 . (d) Physical mesh T_0 after moving the boundary and smoothing the interior control points. (e) Interior mesh after removing the elements in (d). (f) Removed elements in (d).

4.2 Construction of C^r basis

Once the geometry is parametrized, we can construct a set of basis functions over the tetrahedral mesh in the parametric space. We first extend the definition of domain points in Eq. (11) from a tetrahedron τ to a tetrahedral partition \hat{T} , which is defined as $\mathcal{D}_{d,\hat{T}} := \bigcup_{\tau \in \hat{T}} \mathcal{D}_{d,\tau}$. Note that domain points shared by multiple tetrahedra are included only once in $\mathcal{D}_{d,\hat{T}}$. The set $\mathcal{D}_{d,\hat{T}}$ can be used to parametrize the C^0 space \mathcal{S}_d^0 . That is, each domain point $\boldsymbol{\xi} \in \mathcal{D}_{d,\hat{T}}$ is associated with a rational C^0 basis $\phi_i(\boldsymbol{\xi})$ and for any $f \in \mathcal{S}_d^0$ we can write

$$f = \sum_{i} t_{i} \phi_{i}(\boldsymbol{\xi}) = \mathbf{t}_{\mathcal{D}}^{T} \boldsymbol{\phi}(\boldsymbol{\xi}), \qquad (23)$$

where \sum_{i} has the meaning of summation over all domain points in $\mathcal{D}_{d,\hat{T}}$, and $\mathbf{t}_{\mathcal{D}_{d,\hat{T}}}$ contains the corresponding Bézier ordinates in the physical space. As shown in Eq. (5), the basis ϕ_i is the extension of the Bernstein basis $B_{ijk,d}$ in rational space. It has the property of linear independence, nonnegativity and partition of unity, which make it ready to be used in analysis.

By imposing the smoothness conditions (12) on the Bézier ordinates, Eq. (23) can also be used to represent a C^r continuous polynomial $f \in S_d^r$. However, in this case we cannot assign arbitrary values to every coefficient of f. Instead, only the coefficients corresponding to a reduced determining set of the domain points $\mathcal{M}_{d,\hat{T}} \subset \mathcal{D}_{d,\hat{T}}$ can be assigned, and all remaining coefficients will be determined by the smoothness conditions. When $\mathcal{M}_{d,\hat{T}}$ is the smallest set among all possible determining sets, it is called a minimal determining set (MDS) [32, 41]. The domain points in $\mathcal{M}_{d,\hat{T}}$ are referred as free nodes and those in $\mathcal{D}_{d,\hat{T}}$ but not in $\mathcal{M}_{d,\hat{T}}$ are referred as dependent nodes.

The application of smoothness conditions (12) on a tetrahedral partition essentially forms a homogeneous linear system

$$\mathbf{At}_{\mathcal{D}_{d\,\widehat{T}}} = \mathbf{0},\tag{24}$$

where **A** is a coefficient matrix depending on the position of the domain points and $\mathbf{t}_{\mathcal{D}_{d,\hat{T}}}$ are the *n* Bézier ordinates for the domain points in $\mathcal{D}_{d,\hat{T}}$. The dimension of the space $\mathcal{S}_{d}^{r}(\hat{T})$ thus is

$$\dim \mathcal{S}_d^r(\widehat{T}) = \dim \mathcal{S}_d^0(\widehat{T}) - \operatorname{rank} \mathbf{A}.$$
 (25)

After some matrix operations of \mathbf{A} , Eq.(24) can be written in the form

$$\mathbf{t}_{\mathcal{D}_{d,\widehat{T}}} = \mathbf{C}^T \mathbf{t}_{\mathcal{M}_{d,\widehat{T}}},\tag{26}$$

where \mathbf{C} is called the continuity matrix. Substituting Eq.(26) into Eq.(23) gives

$$f(\boldsymbol{\xi}) = \mathbf{t}_{\mathcal{D}_{d,\hat{T}}}^{T} \boldsymbol{\phi}(\boldsymbol{\xi}) = \mathbf{t}_{\mathcal{M}_{d,\hat{T}}}^{T} \mathbf{C} \boldsymbol{\phi}(\boldsymbol{\xi}) = \mathbf{t}_{\mathcal{M}_{d,\hat{T}}}^{T} \boldsymbol{\psi}(\boldsymbol{\xi})$$
(27)

where

$$\boldsymbol{\psi}(\boldsymbol{\xi}) = \mathbf{C}\boldsymbol{\phi}(\boldsymbol{\xi}) \tag{28}$$

is a set of global C^r basis functions composed as the linear combinations of the C^0 basis $\phi(\boldsymbol{\xi})$. Now we have represented a C^r continuous function $f \in \mathcal{S}_d^r$ in terms of the C^r basis and the coefficients corresponding to the free nodes $\mathbf{t}_{\mathcal{M}_{d,\hat{T}}}$.

In order to construct a set of C^r basis that can be used for analysis, we use the so-called macro-element spaces [32] only. In macro-element spaces, the basis corresponding to its MDS is stable and locally supported. Moreover, the basis has full approximation power, thus optimal convergence can be achieved in analysis [24]. Many C^r macro-element spaces defined over tetrahedral partitions can be found in [32, 33] and the papers listed therein. In this paper, we choose a quintic C^1 space $\mathcal{S}_5^{1,2,4}(\widehat{T}_A)$ defined over the Alfeld split of a tetrahedral partition to demonstrate our analysis approach. The space is defined as

$$\mathcal{S}_5^{1,2,4}(\widehat{T}_A) := \{ f \in \mathcal{S}_5^1(\widehat{T}_A) : f \in C^2(v), \forall v \in \mathcal{V}, f \in C^4(v_\tau), \forall T \in \widehat{T} \},$$
(29)

where $\mathcal{V}, v_{\tau}, \tau$ are the vertices, barycenters and tetrahedra in \hat{T} .

In Section 4.1, we have constructed a C^0 geometric map $\mathbf{G}_0(\boldsymbol{\xi}) : \widehat{T} \mapsto T_0$. With the C^r basis $\boldsymbol{\psi}$ and MDS $\mathcal{M}_{d,\widehat{T}}$, we can replace the control points in the C^0 mesh T_0 with a new set of control points that satisfy the C^r continuity constraints

$$\mathbf{p} = \mathbf{C}^T \mathbf{p}^f,\tag{30}$$

where \mathbf{p} are all the control points in the new physical mesh T, and \mathbf{p}^f are the control points in T that corresponds to the free domain points in $\mathcal{M}_{d,\hat{T}}$. Note that, the control points on the boundary surface of T_0 will not be replaced, because the boundary map $\mathbf{G}_{\Gamma}: \hat{T}_{\Gamma} \mapsto T_{0,\Gamma}$ created in Section 4.1 is already C^r continuous. Therefore, the exact boundary is preserved in the C^r map. It should be mentioned that, for geometries with complicated shapes, replacing the control points may lead to distortion or even self-intersection in the C^r map. However, as concluded in [42], basis functions with high order and high continuity were able to lessen the impact of the distortions in IGA. Nonetheless, further study on improving the mesh quality using optimization technique such as presented in [21] will be helpful.

The C^r geometric mapping $\mathbf{G}(\boldsymbol{\xi}) : T \mapsto T$ now can be expressed in terms of rational C^r basis $\psi_i(\boldsymbol{\xi})$, or equivalently, the rational C^0 basis $\phi_j(\boldsymbol{\xi})$ as

$$\mathbf{G}(\boldsymbol{\xi}) = \sum_{i}^{m} \mathbf{p}_{i}^{f} \psi_{i}(\boldsymbol{\xi}) = \sum_{j}^{n} \mathbf{p}_{j} \phi_{j}(\boldsymbol{\xi}), \qquad (31)$$

where m and n are the dimension of the space S_d^r and S_d^0 respectively, and $\mathbf{p}_i^f, \mathbf{p}_j$ satisfy the continuity constraints in Eq.(30).

As discussed in [24], in order to achieve optimal convergence rates in C^r spaces, the smooth-refine-smooth scheme must be used. A pre-refinement geometric map with sufficient smoothness need to be constructed before applying any refinements. This smooth map is different from the C^r geometric map (31) in that it does not need a set of stable basis in the macro-element spaces. Therefore the pre-refinement map can be constructed simply by enforcing the smoothness constraints and solving the system using Gaussian elimination.

5 IGA with Bézier tetrahedra

In this section we describe the method of isogeometric analysis using rational Bézier tetrahedral elements where the classical Galerkin formulation is applied as in [24]. The problems considered in this paper include linear elasticity and Poisson problems. The governing equation for the linear elasticity is

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0} & \text{on } \Omega \\ \boldsymbol{\sigma} = \mathbf{D} \nabla_s \mathbf{u} & \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t} & \text{on } \Gamma_t \\ \boldsymbol{u} = \bar{\boldsymbol{u}} & \text{on } \Gamma_u, \end{cases}$$
(32)

where **D** is the elasticity matrix, **b** and **t** refer to body force and traction respectively, **u** is the displacement, Γ_t and Γ_u are the portions of the boundary where traction and displacement are specified respectively. The Poisson problem is defined as

$$\begin{cases} -\nabla^2 u = f & \text{in } \Omega, \\ u = \bar{u} & \text{on } \Gamma, \end{cases}$$
(33)

where $f: \Omega \to \mathbb{R}$ is a given function and \bar{u} denotes prescribed boundary values.

Using the basis constructed in the previous section, we approximate the solution in the corresponding parametric domain as

$$\hat{u}(\boldsymbol{\xi}) = \sum_{i} u_{i} \psi_{i}(\boldsymbol{\xi}) = \boldsymbol{u}^{T} \boldsymbol{\psi}$$
(34)

where u_i corresponds to the approximate solution's Bézier ordinate at the *i*-th domain point in the parametric domain $\widehat{\Omega}_{\widehat{T}}$. The solution $u(\boldsymbol{x})$ over the domain Ω_T in the physical space is obtained by composing $\hat{u}(\boldsymbol{\xi})$ with the inverse of the geometric mapping \mathbf{G}^{-1} such that $u(\boldsymbol{x}) : \Omega \mapsto \mathbb{R}^2$,

$$u(\boldsymbol{x}) = \hat{u}(\boldsymbol{\xi}) \circ \mathbf{G}^{-1}(\boldsymbol{x}).$$
(35)

After inserting the approximate solution and basis functions into the corresponding weak form of the PDE, we obtain the following mass and stiffness matrices respectively as

$$\mathbf{M}_0 = \int_{\Omega} \boldsymbol{\phi} \cdot \boldsymbol{\phi} \, d\Omega, \tag{36}$$

$$\mathbf{K}_{0} = \int_{\Omega} \nabla \boldsymbol{\phi} \cdot \nabla \boldsymbol{\phi} \ d\Omega, \tag{37}$$

for C^0 elements. For C^r elements, we use the same Bézier extraction technique [43, 44, 45] that has been used in IGA with Bézier triangles [24] to calculate the mass and stiffness matrices as

$$\mathbf{M}_{r} = \int_{\Omega} \boldsymbol{\psi} \cdot \boldsymbol{\psi} \ d\Omega = \int_{\Omega} (\boldsymbol{C}\boldsymbol{\phi}) \cdot (\boldsymbol{C}\boldsymbol{\phi}) \ d\Omega = \boldsymbol{C}^{T} \widetilde{\mathbf{M}}_{0} \boldsymbol{C}, \tag{38}$$

$$\mathbf{K}_{r} = \int_{\Omega} \nabla \boldsymbol{\psi} \cdot \nabla \boldsymbol{\psi} \ d\Omega = \int_{\Omega} (\boldsymbol{C} \nabla \boldsymbol{\phi}) \cdot (\boldsymbol{C} \nabla \boldsymbol{\phi}) \ d\Omega = \boldsymbol{C}^{T} \widetilde{\mathbf{K}}_{0} \boldsymbol{C}, \tag{39}$$

where $\widetilde{\mathbf{M}}_0$ and $\widetilde{\mathbf{K}}_0$ are the mass and stiffness matrices respectively for the same C^r elements in terms of the C^0 basis $\boldsymbol{\phi}$. The difference between $\widetilde{\mathbf{M}}_0$ and \mathbf{M}_0 , $\widetilde{\mathbf{K}}_0$ and \mathbf{K}_0 is due to the potential relocation of the control points to satisfy the C^r continuity constraints for the C^r elements. Using the Bézier extraction technique allows the implementation to be applied in any existing FEM routine without changing the assembly process. The numerical integration is performed in each element (micro-element if split is used) by using standard and collapsed [46] Gaussian quadrature rules on the boundaries and element interiors respectively. The least square method is used to apply the Dirichlet boundary conditions.

6 Numerical examples

In this section, we demonstrate IGA with Bézier tetrahedra on linear elasticity and Poisson problems. In the first example, we demonstrate that optimal convergence rate can be achieved for IGA with Bézier tetrahedra in C^1 space. In the second example, we show that IGA with Bézier tetrahedra can be applied on geometries with very complicated shape. In the third and fourth examples, we demonstrate that our method can be used to handle complex trimmed geometries. At last we demonstrate the analysis of a complex geometry using C^1 smooth splines. For the analysis procedure, we follow the same framework for IGA with Bézier triangles in [24]. The element matrices for C^r elements are computed in terms of C^0 elements using the Bézier extraction technique. Boundary MDS is used to apply Dirichlet boundary conditions. The numerical integration in each element is implemented using the collapsed Gaussian quadrature rules [46].

6.1 Optimal convergence in C^r space

The first example demonstrate how optimal convergence can be achieved using IGA with Bézier tetrahedra in C^1 space.

We solve a Poisson problem

$$-\nabla^2 u = \sin(x)\sin(y)$$

on the geometry shown in Figure 16. The exact solution are imposed as Dirichlet boundary conditions on all boundary surfaces. We use the C^1 elements in the superspline space $S_5^{1,2,4}$ with Alfeld split. As discussed in [24], in order to obtain optimal convergence in a superspline space, the smooth-refine-smooth scheme is needed to obtain a pre-refinement geometric map with sufficient smoothness. In this example, we need to obtain a C^2 map since the C^4 smoothness only occurs at the barycenters and is always satisfied. We apply the smoothness conditions on the control points of the mesh in Figure 16(a) to obtain a C^2 mesh as shown in Figure 16(c) and 16(d). Then we can construct a set of C^1 basis in the space $S_5^{1,2,4}$. A pair of parametric and physical mesh with free and dependent nodes is shown in Figure 17. The initial mesh with Alfeld split consists of 100 elements and 2,301 control points. Figure 18 shows a comparison of the basis function and their gradients of C^0 and C^1 basis. As can be seen, the gradient of C^0 basis is discontinuous across some of the element edges, while the gradient of C^1 basis is continuous across all element edges.

To evaluate the convergence, we refine the mesh using the quasi-uniform refinement method. As shown in Figure 19, optimal convergence rate is achieved when using the smooth-refine-smooth scheme [24]. As a comparison, the non-optimal convergence when using the refine-then-smooth scheme is also presented.



(a) The original geometry represented by Bézier tetrahedra of degree 5 with control points.



(c) C^2 mesh of the geometry. The red and blue points are free and dependent control points respectively in the C^2 smoothness constraints.



(b) The wireframe view of the original geometry.



(d) The wireframe view of the C^2 geometry.

Figure 16: A 3D model represented by exact Bézier tetrahedral partition with control points.



Figure 17: Meshes in space $S_5^{1,2,4}$ with Alfeld split. The initial mesh with Alfeld split consists of 100 elements and 2,301 control points. The red and blue points represent free and dependent points respectively.



tion.

-10

-12

Figure 18: Comparison of C^0 and C^1 basis functions and their gradients.



Figure 19: Convergence rates in $\mathcal{S}_5^1(\widehat{T}_A)$.

6.2 Analysis of an untrimmed geometry

In the second example, we illustrate the ability of our method to handle untrimmed complex geometries, where the linear elasticity of a propeller is analyzed.

The model shown in Figure 20 is a simplified version of the model obtained from [47]. The geometry is represented by 24 untrimmed bicubic NURBS surfaces. Using the parametrization method described in Section 3.1, the NURBS surfaces are exactly converted to 3,870 Bézier triangles of degree 6, from which a tetrahedral mesh with 6,431 elements are constructed. As can be seen in Figure 20(c) and 20(d), the triangular Bézier surface of the generated Bézier tetrahedral partition preserves the original NURBS surfaces and continuity exactly. The final mesh consists of 798,366 dofs. A wind loading is simulated by setting all displacement components on the interior cylindrical surface to zero, the traction on all other surfaces is set to $\mathbf{t} = [0, 0, -n_z]$ if $n_z > 0$ and zero otherwise. The Young's modulus is $E = 10^5$, and Poisson's ratio $\nu = 0.3$. The linear elasticity equation is solved using C^0 basis of degree 6. The simulated displacement profile is shown in Figure 21 with the original Bézier tetrahedral mesh superimposed.



(a) Input: bicubic NURBS surfaces.



(c) Bézier tetrahedral mesh with converted triangular Bézier surface.



(b) Zoom-in view of the NURBS surface and control points. The blue dashed lines are control nets.



(d) Zoom-in view of the triangular Bézier surface and control points.



(e) Tetrahedral mesh in parameter space.



(f) Zoom-in view of the polygonal surface in (e).



(g) Parametric and physical element samples.

Figure 20: Parametrization of the propeller geometry represented by 24 bicubic NURBS surfaces.



Figure 21: The original Bézier tetrahedral mesh for the propeller model and an exaggerated displacement profile with displacement magnitude superimposed.

6.3 Analysis of trimmed geometries

To demonstrate the ability of IGA with Bézier tetrahedra to handle trimmed geometries, we analyzed two geometries represented by multiple trimmed NURBS surfaces.

The first geometry is a simple computer mouse represented by six bicubic NURBS surfaces, as shown in Figure 22. The four side surfaces $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_4$ and top surface \mathbf{S}_5 are trimmed by their intersection curves. The four side surfaces and the bottom surface \mathbf{S}_6 join with each other with C^0 smoothness. The intersection of the top and four side surfaces results in four intersection curves. As shown in Figure 22(c), a gap can be seen in the zoom-in view inside the white box in Figure 22(a). The gap needs to be fixed before the CAD geometry is discretized for analysis.



Figure 22: A trimmed CAD geometry represented by six bicubic NURBS surfaces. (a) The untrimmed NURBS surfaces with control nets. The thick black curves are the boundaries of the NURBS surfaces. The four side surfaces and the bottom surface join with each other with C^0 smoothness. (b) The four side surfaces and the top surface are trimmed by their intersection curves. (c) Zoom-in view inside the white box in (a) showing a gap.

Using our method described in Section 3.2, we first identify the intact triangles and interface layers in the parametric domains. As shown in Figure 23, the intact triangles and interface layers are separated by the thick black lines. For the intact triangles, each can be extracted as a rTBS patch. For the interface layers, we triangulate them after compatible subdivision. The resulting triangulations are shown in Figure 24. The four selected trimming boundary curves are also subdivided into compatible Bézier segments. Figure 25 shows the extracted rTBS patches, which preserve the original NURBS surfaces exactly. However, since the rTBS patches are extracted from different NURBS surfaces, their interfaces usually do not match with each other and results in smalll gaps. After replacing the Bézier edges along the trimmed boundaries with the compatible curves, a watertight representation of the trimmed geometry is obtained, as shown in Figure 26. Then a Bézier tetrahedral mesh is created from the rTBS patches, as shown in Figure 27 where a corner of the model has been cut off to show the interior of the generated tetrahedral elements. The Bézier tetrahedral mesh consists of 513 elements and 22,384 control points.



Figure 23: (a-d) Trimming curves in the parameter space of the four side surfaces. (e) Trimming curves in the parameter space of the top surface. The dashed lines represent domains that are trimmed off. (f) The four selected trimming boundary curves in the physical space. The triangles represent the end points of the curves.

A linear elasticity problem is solved using the created Bézier tetrahedral mesh. The geometry is fixed at two nodes on the left side and a unit displacement is applied at the right side. The Young's modulus is 10^3 with the Poisson's ratio 0.3. The resulting displacement contour is shown in Figure 28, which is expected.



Figure 24: Compatible subdivision of the trimming curves. (a-e) Triangulations of the interface layers of the four side surfaces and top surface after compatible subdivision. (f) Subdivided trimming boundary curves in physical space. The blue squares are the subdivision points.



Figure 25: Conversion of trimmed NURBS surfaces to rTBS patches. (a) The converted rTBS patches and control points. (b) Zoom-in view inside the white box in (a) showing the control points do not match at the trimming boundary. Different colors are used for the side and top surfaces. The dark shaded patches are extracted from the interface layers.



Figure 26: Watertight representation after replacing the boundary edges with the subdivided trimming boundary curves. (a) The blue points are the control points of the Bézier segments of the subdivided trimming boundary curves. (b) Zoom-in view inside the white box in (a) showing a seamless join of the rTBS patches.



Figure 27: Generated Bézier tetrahedral elements with some elements removed to show the interior mesh. (a) Bézier tetrahedral mesh in physical space. (b) Tetrahedral mesh in parametric space.



Figure 28: Displacement contours from numerical result.

The second trimmed geometry is a piston head represented by 31 trimmed NURBS surfaces with 64 intersection curves, as shown in Figure 29. The control points for one selected surface in blue color are also shown. The 31 NURBS surfaces are either cylindrical surfaces or planar surfaces with degree up to 3 in each direction. Most surfaces have only 1 knot interval in both directions, i.e. they are actually tensor-product Bézier patches, as shown in Figure 30 and 31.



Figure 29: The piston head model represented by 31 trimmed NURBS surfaces with 64 intersection curves. (a) The trimmed geometry represented in CAD system. The blue curves are the intersection curves. (b) The original NURBS surfaces before trimming. The blue curves are the intersection curves between surfaces. The control points are shown for one of the surfaces indicated as blue.

Figure 30 is a bilinear planar surface with 8 trimming curves. The original surface is a Bézier patch and the whole domain enclosed by the trimming curves is the interface layer, as indicated by the shaded color in Figure 30(c). The parametric domain is then triangulated after compatible subdivision. The extracted rTBS patches are shown in Figure 30(d), where the degree is elevated to 6. Figure 31 is a cylindrical surface with 6 trimming curves. The original surface is also a Bézier patch of degree 1 and 2 in each direction. After compatible subdivision, we triangulate the domain, as shown in Figure 31(c). The extracted rTBS patches are shown in Figure 31(d), where the degree is elevated to 6. Only the rTBS patches extracted from the interface layer (shaded in Figure 31(c)) will be affected by the replacement of the trimming boundary curves. The interface layer can be further refined so more extracted rTBS patches will become exact with respect to the original NURBS surface, as shown in Figure 31(d).

Finally, we successfully convert the trimmed NURBS surfaces to watertight rTBS patches with 1308 Bézier triangles of degree 6. Then we parametrize the watertight rTBS patches volumetrically using 1965 Bézier tetrahedra elements, with a total number of 82, 682 control points, as shown in Figure 32.

We solve the linear elastic equation over the Bézier tetrahedral mesh. We applied uniform pressure at the bottom surface while fixing the inner surfaces of the two horizontal holes near the top. The Young's modulus is set as 10^3 with the Poisson's ratio 0.3. Figure 33 shows the displacement contours of the analysis results.



(a) Parametric doamin with trimming curves (red).



(c) Triangulation of parametric doamin after compatible subdivision.



(b) Original NURBS surface with physical trimming curves (blue).



(d) Extracted rTBS patches with control points (degree elevated to 6).

Figure 30: Conversion of a selected trimmed surface 1 to rTBS patches.



(a) Parametric doamin with trimming curves (red).



(c) Triangulation of parametric doamin after compatible subdivision. The interface layer triangles are shaded.



(b) Original NURBS surface with physical trimming curves (blue).



(d) Extracted rTBS patches with control points (degree elevated to 6).



(e) Refinement of mesh in (c). The size of the interface layer is reduced.



(f) Extracted rTBS patches correponding to (d). The size of rTBS patches extracted from the interface layer is reduced.

Figure 31: Conversion of a selected trimmed surface 2 to rTBS patches. The size of the interface layer can be reduced by mesh refinement, leading to more rTBS patches that are exact with respect to the original trimmed geometry.



Figure 32: The piston head model represented by watertight rTBS patches, which is then parametrized into 1965 Bézier tetrahedra elements. (a) The trimmed NURBS surfaces are converted to 1308 rTBS patches. (b) Zoom-in view inside the white box in (a) showing the smooth surface. (c) Generated Bézier tetrahedral mesh with some elements removed to show the interior mesh. (d) Another view of the mesh in (c).



Figure 33: Displacement contours of the analysis results.

6.4 Stress analysis using C^1 Bézier tetrahedra

In this last example, we demonstrate our approach of analyzing a connecting rod model (Fig. 34(a)) using C^1 rational Bézier tetrahedra. We first parametrize the initial geometry into 2405 C^0 Bézier tetrahedral elements of degree 5, as shown in Figure 34. In this tetrahedral model, the three circular holes are exactly represented due to the use of rational form of Bézier tetrahedra. Then we construct C^1 smooth basis functions by using the Alfeld split and imposing the corresponding smoothness constraints. The resulting C^1 physical and parametric elements are shown in Figure 35(c). For the ease of domain parameterization, singularities of the mapping from the parametric domain to the physical domain are introduced at the sharp vertical edges of the parametric domain corresponding to the cylindrical surfaces. At these vertical edges, the smoothness is reduced to C^0 . The final C^1 meshes are shown in Figure 35. The interior of the smooth Bézier tetrahedra is also shown in Fig. 35(b).

For the stress analysis, we fix the half-cylindrical surface on the left, and apply a unit force pointing to the left on the inner surface of the hole on the right. We compare the analysis results from the C^0 and C^1 elements in Figure 36. For the C^1 elements, the stress at the singular edges on the cylindrical surfaces are evaluated by averaging the stress of a small neighborhood.. As can be seen, the overall stress distribution is about the same for C^0 and C^1 elements. However, a close-in view shows that C^0 elements give discontinuous stress contours across element edges while C^1 elements give continuous stress contours even across element edges.



Figure 34: (a) A connecting rod model. (b) The model is parametrized with 2405 rational Bézier tetrahedral elements of degree 5. The red points are the control points.



Figure 35: C^1 parametrization of the connecting rod model in Fig. 34. The red and blue points are free and dependent points respectively. (a) Physical mesh. (b) Some elements are removed to show the interior of the mesh. (c) Parametric mesh.



Figure 36: Comparison of stress contours from the analysis results with C^0 and C^1 elements. (a) stress from C^0 elements. (b) Zoom-in view showing the discontinuous stress using C^0 elements. (c) stress from C^1 elements. (b) Zoom-in view showing the continuous stress using C^1 elements.

7 Conclusion

In this paper, we present an isogeometric analysis approach with Bézier tetrahedra. We convert any complex NURBS geometries (either trimmed or untrimmed) into watertight geometries represented by rTBS patches. A compatible subdivision scheme is developed to guarantee the watertightness when converting trimmed geometries. The converted rTBS geometry preserves the original NURBS surfaces exactly, except for an interface layer for trimmed geometries. The approximation accuracy of this interface layer can be improved by further refinement. From the rTBS geometry surface, a high order Bézier tetrahedral partition is created automatically, discretizing the geometry into Bézier tetrahedral elements.

The same trivariate Bernstein polynomials representing the Bézier tetrahedra are used as basis for analysis. In addition, by imposing continuity constraints on the Bézier ordinates, we can construct a set of C^r basis that can be used for analysis. Using the smooth-refine-smooth scheme, optimal convergence rate has been demonstrated using such C^r basis. Numerical results show that our approach can handle any geometry with complex trimmed NURBS surfaces.

Compared with NURBS based IGA, our approach inherits the advantage of representing CAD geometries exactly and efficient analysis using high order continuous basis. In addition, our approach can represent complex geometries, including both trimmed and untrimmed.

8 Acknowledgement

The authors acknowledge the financial support from National Science Foundation grant #1435072.

References

- 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] Xiaoping Qian. Full analytical sensitivities in NURBS based isogeometric shape optimization. Computer Methods in Applied Mechanics and Engineering, 199(29):2059– 2071, 2010.
- [3] 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.
- [4] Kang Li and Xiaoping Qian. Isogeometric analysis and shape optimization via boundary integral. Computer-Aided Design, 43(11):1427–1437, 2011.
- [5] 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.

- [6] 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.
- [7] Xilu Wang and Xiaoping Qian. An optimization approach for constructing trivariate B-spline solids. *Computer-Aided Design*, 46:179–191, 2014.
- [8] Thomas W Sederberg, G Thomas Finnigan, Xin Li, Hongwei Lin, and Heather Ipson. Watertight trimmed NURBS. ACM Transactions on Graphics (TOG), 27(3):79, 2008.
- [9] 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.
- [10] Thomas W Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. T-splines and T-NURCCs. In ACM Transactions on Graphics (TOG), volume 22, pages 477– 484. ACM, 2003.
- [11] Lei Liu, Yongjie Jessica Zhang, and Xiaodong Wei. Weighted T-splines with application in reparameterizing trimmed NURBS surfaces. Computer Methods in Applied Mechanics and Engineering, 295:108–126, 2015.
- [12] Jingjing Shen, Jiří Kosinka, Malcolm A Sabin, and Neil A Dodgson. Conversion of trimmed NURBS surfaces to Catmull–Clark subdivision surfaces. *Computer Aided Geometric Design*, 31(7):486–498, 2014.
- [13] Yuri Bazilevs, Victor M Calo, John A Cottrell, John A Evans, Thomas Jr R Hughes, S Lipton, Michael A Scott, and Thomas W Sederberg. Isogeometric analysis using Tsplines. Computer Methods in Applied Mechanics and Engineering, 199(5):229–263, 2010.
- [14] Michael Andrew Scott. T-splines as a Design-Through-Analysis technology. Ph.D. Thesis, The University of Texas at Austin, 2011.
- [15] Daniel Burkhart, Bernd Hamann, and Georg Umlauf. Isogeometric finite element analysis based on Catmull-Clark: subdivision solids. In *Computer Graphics Forum*, volume 29, pages 1575–1584. Wiley Online Library, 2010.
- [16] Xiaodong Wei, Yongjie Zhang, Thomas JR Hughes, and Michael A Scott. Truncated hierarchical Catmull–Clark subdivision with local refinement. *Computer Methods in Applied Mechanics and Engineering*, 291:1–20, 2015.
- [17] 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.
- [18] 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.

- [19] 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.
- [20] Hendrik Speleers, Carla Manni, Francesca Pelosi, and M Lucia Sampoli. Isogeometric analysis with Powell-Sabin splines for advection-diffusion-reaction problems. *Computer Methods in Applied Mechanics and Engineering*, 221:132–148, 2012.
- [21] 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.
- [22] Yue Jia, Yongjie Zhang, Gang Xu, Xiaoying Zhuang, and Timon Rabczuk. Reproducing kernel triangular B-spline-based FEM for solving PDEs. Computer Methods in Applied Mechanics and Engineering, 267:342–358, 2013.
- [23] Noah Jaxon and Xiaoping Qian. Isogeometric analysis on triangulations. Computer-Aided Design, 46:45–57, 2014.
- [24] 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.
- [25] Hendrik Speleers. A normalized basis for quintic Powell–Sabin splines. Computer Aided Geometric Design, 27(6):438–457, 2010.
- [26] Hendrik Speleers. Construction of normalized B-splines for a family of smooth spline spaces over Powell–Sabin triangulations. *Constructive Approximation*, 37(1):41–72, 2013.
- [27] Hendrik Speleers. Multivariate normalized Powell–Sabin B-splines and quasiinterpolants. Computer Aided Geometric Design, 30(1):2–19, 2013.
- [28] Hendrik Speleers, Carla Manni, and Francesca Pelosi. From NURBS to NURPS geometries. Computer Methods in Applied Mechanics and Engineering, 255:238–254, 2013.
- [29] Suqin Zeng and Elaine Cohen. Hybrid volume completion with higher-order Bézier elements. Computer Aided Geometric Design, 35:180–191, 2015.
- [30] Gerard Awanou, Ming-Jun Lai, and Paul Wenston. The multivariate spline method for scattered data fitting and numerical solutions of partial differential equations. *Wavelets and splines: Athens*, pages 24–74, 2005.
- [31] Gerard Awanou and Ming-Jun Lai. Trivariate spline approximations of 3D Navier-Stokes equations. *Mathematics of computation*, 74(250):585–601, 2005.
- [32] Ming-Jun Lai and Larry L Schumaker. Spline Functions on Triangulations, volume 110. Cambridge University Press, 2007.
- [33] Michael Andreas Matt. Trivariate local lagrange interpolation and macro elements of arbitrary smoothness. Springer Science & Business Media, 2012.

- [34] Rida T Farouki, Chang Yong Han, Joel Hass, and Thomas W Sederberg. Topologically consistent trimmed surface approximations based on triangular patches. *Computer Aided Geometric Design*, 21(5):459–478, 2004.
- [35] Jiří Kosinka and Thomas J Cashman. Watertight conversion of trimmed CAD surfaces to Clough–Tocher splines. Computer Aided Geometric Design, 37:25–41, 2015.
- [36] Ronald N Goldman and Daniel J Filip. Conversion from Bézier rectangles to Bézier triangles. Computer-Aided Design, 19(1):25–27, 1987.
- [37] Dieter Lasser. Triangular subpatches of rectangular Bézier surfaces. Computers & Mathematics with Applications, 55(8):1706–1719, 2008.
- [38] Les Piegl and Wayne Tiller. The NURBS book. Springer Science & Business Media, 2012.
- [39] K Stein, T Tezduyar, and R Benney. Mesh moving techniques for fluid-structure interactions with large displacements. *Journal of Applied Mechanics*, 70(1):58–63, 2003.
- [40] Hang Si and A TetGen. A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. Weierstrass Institute for Applied Analysis and Stochastic, Berlin, Germany, 2006.
- [41] Peter Alfeld. Bivariate spline spaces and minimal determining sets. Journal of Computational and Applied Mathematics, 119(1):13–27, 2000.
- [42] 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.
- [43] Larry L Schumaker. Computing bivariate splines in scattered data fitting and the finite-element method. Numerical Algorithms, 48(1-3):237–260, 2008.
- [44] Michael J Borden, Michael A Scott, John A Evans, and Thomas JR Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. International Journal for Numerical Methods in Engineering, 87(1-5):15–47, 2011.
- [45] Michael A Scott, Michael J Borden, Clemens V Verhoosel, Thomas W Sederberg, and Thomas JR Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88(2):126–156, 2011.
- [46] James N Lyness and Ronald Cools. A survey of numerical cubature over triangles. In Proceedings of Symposia in Applied Mathematics, volume 48, pages 127–150, 1994.
- [47] Juan Santocono. 51 free T-splines models. [http://www.tsplines.com/m/models/ modelpack2.zip, Online; accessed 2-February-2016].