Creating Hybrid B-Reps and Hybrid Volume Completions from Trimmed B-Spline B-Reps

Yang Song, Elaine Cohen
University of Utah

UUCS-19-001

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

18 March 2019

Abstract

This paper proposes a method for converting a non-watertight trimmed B-spline B-rep model to a watertight hybrid model representation that preserves the original model's parameterization, representation, and geometry, except within a banded region along the trimming curve. Also, it addresses the problem of volumetric model completion. Solutions to both of these problems are important to have a conformal representation for analysis, in particular, isogeometric analysis, and for describing material attributes. The approach for the boundary representation modifies the original B-spline bases to take into account the trimming curves and introduces new functions along the trimming curves to preserve independence and the convex hull property. Then a volumetric completion algorithm with the new hybrid boundary is introduced. Outside of this research, volume representation completion has witnessed few advances for trimmed B-spline B-rep models. Building on the approach of [21] that is proposed only for untrimmed B-reps, this research generalizes the methodology and solves related issues to make it appropriate for the ubiquitous trimmed B-rep models. The efficacies of both the hybrid B-rep and the hybrid volumetric method are demonstrated on trimmed B-rep models.

Creating Hybrid B-Reps and Hybrid Volume Completions from Trimmed B-Spline B-Reps

Yang Song and Elaine Cohen

School of Computing, University of Utah

March 18, 2019

Abstract

This paper proposes a method for converting a non-watertight trimmed B-spline B-rep model to a watertight hybrid model representation that preserves the original model's parameterization, representation, and geometry, except within a banded region along the trimming curve. Also, it addresses the problem of volumetric model completion. Solutions to both of these problems are important to have a conformal representation for analysis, in particular, isogeometric analysis, and for describing material attributes. The approach for the boundary representation modifies the original B-spline bases to take into account the trimming curves and introduces new functions along the trimming curves to preserve independence and the convex hull property. Then a volumetric completion algorithm with the new hybrid boundary is introduced. Outside of this research, volume representation completion has witnessed few advances for trimmed B-spline B-rep models. Building on the approach of [21] that is proposed only for untrimmed B-reps, this research generalizes the methodology and solves related issues to make it appropriate for the ubiquitous trimmed B-rep models. The efficacies of both the hybrid B-rep and the hybrid volumetric method are demonstrated on trimmed B-rep models.

1 Introduction

Geometric modeling has focused on Boundary representations (B-rep), but with recent advances in Additive Manufacturing (AM), material representations and higher order analyses, the need for volumetric representations is growing [16]. Isogeometric Analysis (IGA) [7] is aimed at bridging the gap between CAD and FEA. In IGA the same basis functions used in modeling, usually B-splines [6] or a variant [29, 8], are also used to solve analyses. A fundamental notion is to keep the parameterization and representation of the CAD model in creating the *volume completion*. However, there is no general technique for creating a volumetric parameterization of the interior that preserves the boundary parameterization and representation of the original complex multi-surface B-spline model. Trimmed B-spline B-rep models present even more complexity.

A trimmed B-rep consists of trimmed B-spline surfaces and their sets of associated trimming curves. Two trimmed surfaces meet along their shared trimming curves. Commercial CAD systems provide each trimming curve approximation in three co-existent representations: 1 in each corresponding parametric domains and 1 in Euclidean space [34]. None of them exactly represents the true trimming curve, nor is any pair identical [28]. This is fundamentally why trimmed models are not watertight so defining a volumetric representation is more challenging for trimmed models.

Creating Hybrid B-Reps and Hybrid Volume Completions from Trimmed B-Spline B-Reps

Yang Song, Elaine Cohen
University of Utah

UUCS-19-001

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

19 March 2019

Abstract

This paper proposes a method for converting a non-watertight trimmed B-spline B-rep model to a watertight hybrid model representation that preserves the original model's parameterization, representation, and geometry, except within a banded region along the trimming curve. Also, it addresses the problem of volumetric model completion. Solutions to both of these problems are important to have a conformal representation for analysis, in particular, isogeometric analysis, and for describing material attributes. The approach for the boundary representation modifies the original B-spline bases to take into account the trimming curves and introduces new functions along the trimming curves to preserve independence and the convex hull property. Then a volumetric completion algorithm with the new hybrid boundary is introduced. Outside of this research, volume representation completion has witnessed few advances for trimmed B-spline B-rep models. Building on the approach of [21] that is proposed only for untrimmed B-reps, this research generalizes the methodology and solves related issues to make it appropriate for the ubiquitous trimmed B-rep models. The efficacies of both the hybrid B-rep and the hybrid volumetric method are demonstrated on trimmed B-rep models.

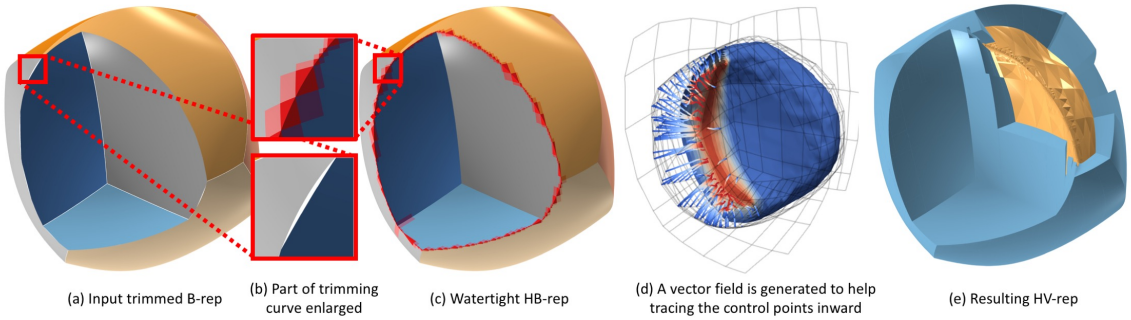


Figure 1: The pipeline: (a) $\mathcal{G}$, a trimmed B-rep. Lower inset (b) a gap between trimmed surfaces. (c) The HB-rep, $\mathcal{B}$. Red area adjacent to the Euclidean trimming curve is to seal the gap, (See Upper inset (b)). (d) Tracing paths for control points from the polyhedron boundary. Control meshes for $\mathcal{G}$ appear in grey. Vertices with no tracings are not on the polyhedron and have no vector field. (e) A cutaway view of the semi-structured trimmed trivariate volume $\mathscr{S}$ (blue) and the outermost boundary of the unstructured region $\mathcal{U}$ (gold) are shown.

Because of the ambiguous representation of trimming curves, alternative representations have been proposed to replace trimmed B-splines for model representation both in the design process and as the representation for IGA simulation. Many methods "re-patch" and approximate the surfaces. Others derive a different geometric representation to perform IGA. We review those ideas in Section 2. Since trimmed NURBS is the predominant technology, according to [22], we focus on dealing with its shortcomings while maintaining the overall representation framework.

The starting point for this research is a trimmed B-spline B-rep model. The goal is to maintain the original B-spline representation and revise the surfaces locally around the trimming curve so that gaps between trimmed surfaces are sealed. Then the problem of generating a volumetric representation is addressed. This document focuses on developing the representations which we term Hybrid B-rep (HB-rep) and the Hybrid Volumetric representation (HV-rep). The contributions of this paper include defining

- The HB-rep that preserves the origin original representation including parameterization, control points, and basis functions in a *banded region* Ω (defined in Section 4) around the trimming curve, and introduces trim functions within Ω for gluing across trimming curves.
- The HV-rep such that it preserves the HB-rep on its boundary, creates trivariate B-splines from the boundary inward and fills the innermost region with Bézier tetrahedra.
- Suitable Bézier pyramids as an interface layer between tensor product trivariate elements and Bézier tetrahedral elements to significantly reduce the DoF.

Figure 1 illustrates methodologies. Related work is discussed in Section 2. B-spline basics are introduced in Section 3. HB-rep and its construction are introduced in Section 4. The volumetric completion is in Section 5. Results are shown in Section 6. Section 7 has concluding remarks.

2 Related work

In dealing with the inexactness of the trimmed model to create watertight representations, surface representations are necessarily modified, and frequently reparameterized in regions around the trimming curves, then remodeled to approximate the original surface with regular patches. These methods are categorized as global methods as surveyed in [22]. For example, [10] preserves the exact geometry (except very close to the trimming curves) by using surface-surface composition to convert trimmed surfaces into regular tensor product NURBS surfaces so that their boundaries match the trimming curves. The method suffers from the high degree of the surface after composition and can result in significantly more knot lines and patches, as well as a modified parameterization.

[28] propose converting NURBS to T-spline surfaces and reconstructing the surface near the trimming curve. This involves parameterization modification and surface approximation near the trimming curve. They argue that error can be made arbitrarily small with sufficient T-spline refinement, each adding degrees of freedom to the representation. [31, 30] propose utilizing Catmull-Clark and non-uniform subdivision surfaces, respectively, to replace the original trimmed surfaces. [31] required approximating the whole B-spline/NURBS surface, as well as the trimming curve. [30] converts the boundary representation to be a non-uniform subdivision surface and represents the trimming curve exactly. However, the geometry and parameterization are changed to emanate from the trimmed region and the surface must be approximated in the new representation. These approaches are aimed at cases when the usage of either T-splines or subdivision surfaces is desirable.

[15] assumes trimmed planar regions, and parameterizes trimmed knot intervals as Bézier triangles that match the trimming curve on one of its edges. [36] uses Bézier triangles and Bézier tetrahedra to construct a watertight representation suitable for IGA. After constructing trimming curves with synchronized knot vectors in corresponding surfaces and model space, the parametric space ones are linearly approximated, adding new knots if necessary, then the surfaces are further refined. Each knot interval of the bi-cubic surface corresponds to two degree 6 Bézier triangles to reproduce the surface. [11] decomposes a model with no trims into an unstructured Bézier discretization. The bounding Bézier tensor product patches are extracted from the boundary B-spline surfaces and are then either left alone or most frequently decomposed into Bézier triangles that represent the surfaces exactly. The volume is then tetrahedralized for IGA.

Other approaches are aimed at using trimmed B-reps for analysis without a watertight representation. Categorized as local methods by [22], for surface analysis, trimming curves separate elements/knot intervals into 3 groups: interior, exterior, and cut element [26, 27, 13]. During integration, exterior elements are excluded, and integration in a cut element is done by subdividing the cut element into equally spaced parametric subcells recursively for integration purposes. They say that as the subcells get smaller, the integral of the partial element gets more accurate. According to [22], no explicit coupling in DoFs across the involved set of surfaces of a trimming curve exposes stability issues.

In general, global approaches reparameterize the surface in an extended region, of at least the trimmed knot intervals, and require surface approximation. On the other hand, while useful in the context of IGA, local methods seem intended to support just analysis.

Most research into transforming boundary representations into volume representations has focused on transformations of triangle meshes. Some is aimed further at creating semi-structured hexahedral representations that can then be approximated as volumetric B-splines, see [19, 12, 42, 2, 3, 18]. [40] solves the problem of completing the representation given a region bounded by a hexahedron whose faces are B-spline surfaces resulting in a trivariate B-spline representation that has been optimized for IGA. The idea was generalized to domains bounded by B-spline surfaces that were already partitioned into hexahedra in [39]. Generalizing the optimization idea to complex planar

regions bounded by B-spline curves, [38] created a semi-structured quadrilateral parameterization preserving the original boundary curve. [23] designs models directly with volumes, eliminating the problem of volume completion, and introduces Booleans on volumes. It raises the dimension of the trimming issues to *trimming surfaces*.

Related hybrid approaches preserve the B-spline boundary representation. Volume completion "grows" the trivariate representation from the original surface to create a trivariate B-spline to a model depth dependent on a mid-structure. [21] fills the rest of the interior with linear tetrahedral elements, while [41] fills it with high degree Bézier tetrahedra that reproduce the interior bounding surface of the trivariate B-spline. These hybrid approaches exploit the advantage high accuracy of B-spline elements in simulation and the freedom in meshing the geometry in complex topology provided by tetrahedral meshes. However, neither approach confronts the issues surrounding trimmed B-rep models.

To our knowledge, there is no methodology that keeps the original B-spline representation in both watertight B-reps and related volumetric representation starting with a trimmed B-spline B-rep.

3 B-spline Basics

Given a non-decreasing sequence $\boldsymbol{\mu} = \{\mu_i\}_i$ called a *knot vector*, a univariate B-spline consists of a non-negative, piecewise degree-d polynomial with minimal local support on $[\mu_i, \mu_{i+d+1}]$. If $\mu_i < \mu_{i+1} \in \boldsymbol{\mu}$ then each nonzero $N_{i,d,\boldsymbol{\mu}}$ is a single polynomial over $[\mu_i, \mu_{i+1})$. Such a single interval is called a *parametric knot interval*. There are $d+1$ nonzero B-splines over each interval. A full set $\{N_{i,d,\boldsymbol{\mu}}(\xi)\}_i$ is a family of linearly independent B-spline functions that form a partition of unity over each knot interval $[\mu_i, \mu_{i+1})$ of the domain, where the parametric domain and the number of functions are determined by the degree and the knot vector. For tensor product surfaces of the form $s(\xi,\eta) = \sum_{i,j} r_{i,j} N_{i,d,\boldsymbol{\mu}}(\xi) N_{j,d,\boldsymbol{\nu}}(\eta)$, a knot interval is a rectangular region $[\mu_i, \mu_{i+1}] \times [\nu_j, \nu_{j+1}]$ that forms the for a single bivariate bi-d polynomial piece.

Because a B-spline restricted to any knot-free bivariate interval $[\xi_1, \xi_2] \times [\eta_1, \eta_2]$ is purely polynomial, it can be written in terms of a tensor product Bézier basis over that interval. The Bézier coefficients depend on the knot vectors, $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, the degree d, and the values ξ_1, ξ_2, η_1, η_2. For example, let $\{B_k^{\boldsymbol{\xi}}\}_{k=0}^d$ designate the degree d univariate Bézier basis functions over a particular interval $\boldsymbol{\xi} = [\xi_1, \xi_2]$. Then,

$$N_{i,d,\boldsymbol{\mu}}(\xi) = \sum_{k=0}^{d} \alpha_{i,k}^{\boldsymbol{\mu},\boldsymbol{\xi}} B_k^{\boldsymbol{\xi}}(\xi) \qquad \text{for } \xi \in [\xi_1, \xi_2] \tag{1}$$

where $B_k^{\boldsymbol{\xi}}(\xi) = \binom{d}{k} \frac{(\xi_2 - \xi)^{d-k} (\xi - \xi_1)^k}{(\xi_2 - \xi_1)^d}$. An analogous result holds for the η direction. Thus, for each tensor product B-spline basis function restricted to $J^{\text{rect}} = [\xi_1, \xi_2] \times [\eta_1, \eta_2]$, we have,

$$\begin{aligned} N_{i_1,i_2}(\xi,\eta) &= N_{i_1,d,\boldsymbol{\mu}}(\xi) N_{i_2,d,\boldsymbol{\nu}}(\eta) \\ &= \sum_{k_1=0}^{d} \sum_{k_2=0}^{d} \alpha_{i_1,k_1}^{\boldsymbol{\mu},\boldsymbol{\xi}} \alpha_{i_2,k_2}^{\boldsymbol{\nu},\boldsymbol{\eta}} B_{k_1}^{\boldsymbol{\xi}}(\xi) B_{k_2}^{\boldsymbol{\eta}}(\eta) \\ &= \sum_{k_1=0}^{d} \sum_{k_2=0}^{d} \alpha_{i_1,k_1}^{\boldsymbol{\mu},\boldsymbol{\xi}} \alpha_{i_2,k_2}^{\boldsymbol{\nu},\boldsymbol{\eta}} B_{k_1,k_2}^{\boldsymbol{\xi},\boldsymbol{\eta}}(\xi,\eta) \quad \text{for } (\xi,\eta) \in J^{\text{rect}}. \end{aligned} \tag{2}$$

So the surface piece over $J^{\text{rect}} = [\xi_1, \xi_2] \times [\eta_1, \eta_2]$ can be written in terms of a local Bézier

representation,

$$s(\xi,\eta)=\sum_{k_1=0}^{d}\sum_{k_2=0}^{d}\rho_{k_1,k_2}B^{\xi,\eta}_{k_1,k_2}(\xi,\eta)\quad\text{for }(\xi,\eta)\in J^{\text{rect}},\tag{3}$$

and,

$$\rho_{k_1,k_2}=\sum_{i_1}\sum_{i_2}\alpha^{\mu,\xi}_{i_1,k_1}\alpha^{\nu,\eta}_{i_2,k_2}r_{i_1,i_2}.\tag{4}$$

Since there are only $(d+1)$ nonvanishing B-spline basis functions over J^{rect}, each double summation in Equation (3) has just $(d+1)^2$ terms.

If J^{rect} does not correspond to a full knot interval, it is referred to as a *sub-knot interval* to distinguish it from the full knot intervals in $\boldsymbol{\mu}\times\boldsymbol{\nu}$. Thus, over any bivariate interval J^{rect}, it is possible to compute interchangeably with the original B-spline or with the corresponding Bézier representation. This duality is particularly useful when evaluating a model's integral properties such as mass or moments, or solving PDEs, as proposed in [4].

Over a canonical triangle with vertices at $(0,0)$, $(1,0)$, and $(0,1)$, each Bézier triangle basis function of degree d is written $G_{i,j,k}(u,v)=\frac{d!}{i!j!k!}u^iv^j(1-u-v)^k$ where $i,j,k\geq 0$, $i+j+k=d$, $0\leq u,v$ and $u+v\leq 1$. A surface s^{T} with the canonical triangle as its parametric space has representation

$$s^{\text{T}}(u,v)=\sum_{\substack{0\leq i,j,k,\\ i+j+k=d}}b_{i,j,k}G_{i,j,k}(u,v)\tag{5}$$

The domain for the Bézier triangle can have an arbitrary triangle $\triangle$ as the domain in parametric space by mapping vertex P_0 to $(0,0)$, P_1 to $(1,0)$ and P_2 to $(0,1)$ with an affine transformation [6]. Typically if the parametric triangle is a right triangle, the right angle vertex is mapped to $(0,0)$. If a single Bézier segment of a trimming curve forms a curvilinear diagonal of a sub-knot interval in the parametric space of the surface, the curvilinear triangle part of the sub-knot interval that is kept as part of the trimmed surface is called a *curvilinear sub-knot interval* and is denoted J^{tri}.

4 Watertight HB-rep $\mathcal{B}$

In this section, we introduce related schemes for transforming the ubiquitous trimmed B-spline B-rep model into a B-rep with watertight boundaries. The representations all create a parametric region Ω bounded between rectilinear curves and parametric trimming curves. Outside Ω, the schemes preserve the original B-spline representation. Within Ω the parameter spaces differ but lead to the same geometry. The difference is related to whether the trimmed parametric domain is fully utilized.

Let $\mathcal{G}$ represent a fully detailed trimmed B-spline B-rep model, including all surfaces, trimming curves, and topology information. For each surface in $\mathcal{G}$, its *trimmed control mesh* contains the control points (denoted $\mathcal{R}$) and structure necessary to define the part of the surface that remains in the model after Boolean operations. Suppose two trimmed surfaces s_1 and s_2 meet along their shared trimming curve. Commercial CAD systems provide each trimming curve approximation in three co-existent representations: c_1 and c_2 are mappings from the unit interval into each respective parametric domain, and c_{m} maps the unit interval into model space [34]. None of them exactly represents the true trimming curve, nor is any pair of them identical [28]. A parametric curve can be composed with its respective surface mapping to create a model space curve that also represents the intersection curve (s_1c_1 and s_2c_2, respectively). Both parametric trimming curves and the model space curve are typically represented as either piecewise cubic or piecewise linear curves. Without

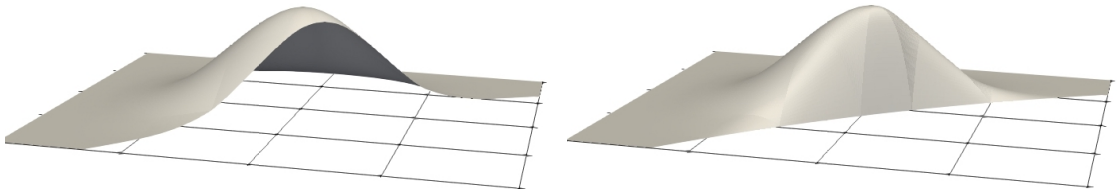


Figure 2: left: a B-spline trimmed along a parametric trimming curve; right: its corresponding revised basis function.

loss of generality, the work reported herein matches the model space trimming curve c_{m} rather than making one parametric space representation dominant.

Recall the term *knot interval* is used herein to denote a tensor product of two nontrivial intervals between knots in each parametric direction of an original surface, respectively. The term *sub-knot interval* is used to denote a tensor product of two knot-free intervals that may not span a full possible parametric knot interval. For each surface in $\mathcal{G}$, a knot interval is classified into 3 groups, *interior*, *trimmed*, and *exterior*, based on its relative position with respect to trimming curves. Each *trimmed knot interval* can intersect a parametric trimming curve by having a trim be a part of the knot interval's isoparametric boundary or by the trimming curve crossing in its interior.

Now we give a high-level outline of the detailed procedure that follows. For each trimmed surface $s \in \mathcal{G}$,

1. For each trimmed knot interval K, construct a rectilinear approximation to the trimming curve that does not touch it and is completely interior to the trimmed surface. The region Ω is bounded by the rectilinear approximation and the trimming curve illustrated by the orange polyline in Figure 3d. Changes from the original surface to seal the representation can occur only inside Ω.

2. Create basis functions for the new representation.

   - The *revised basis functions* are identical to B-splines outside Ω and continuously decrease to evaluate to 0 on the trimming curve inside Ω.
   - Define *trim functions* with support in Ω that form a partition of unity on the trimming curve and are linearly independent.

3. Determine coefficients for the new basis functions to complete the watertight representation and match the model space trimming curve. The representation is called a Hybrid Boundary Representation (HB-rep).

4. The region in each trimmed knot interval minus Ω, $K-\Omega$, is a rectilinear region, see Figure 3c and 11 left. It is represented by the original surface. Since it is easier to evaluate B-spline surfaces over rectangular sub-knot intervals, $K-\Omega$ is partitioned into interior sub-knot intervals with locally optimal shape properties. See Figure 11.

Thus each trimmed surface $s \in \mathcal{G}$ becomes a surface $\tilde{s} \in \mathcal{B}$ that is identical to s except in Ω. Each B-spline function $N_{i,j}$ whose support overlaps the trimmed region is modified to a *revised basis function* $\tilde{N}_{i,j}$ whose support is the trimmed version of its original support, and that continuously decreases to evaluate to 0 on the trimming curve. New trim functions have support over Ω to enable exact representation of the model space trimming curve and to help approximate the surface s in Ω.

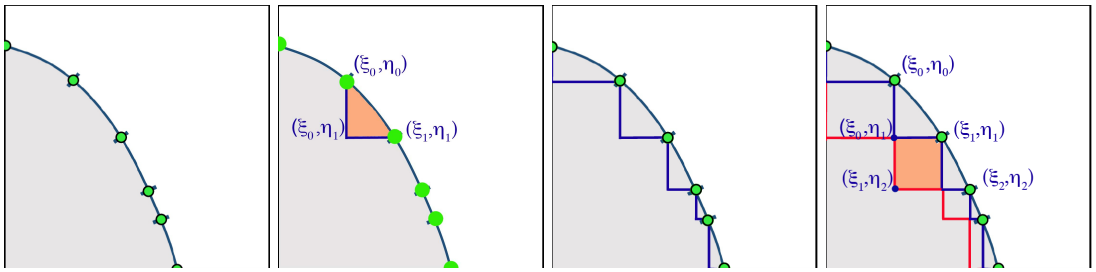
In the following, it is assumed that all three of its representation for each trimming curve are compatible. That is, their knot vectors are the same and the model space images of each knot value represent the same trimming curve point, even though they are not identically the same. Knot insertion and reparameterization may be necessary to ensure this. Supposing c_1 and c_2 are the parametric trimming curves corresponding to c_{m}, if c_1 crosses a knot line in the parameter space of s_1 at some value t^*, or if a local extremum occurs there, then a knot is inserted at t^* in c_2 and c_{m}, as well as in c_1. Analogously, knots are inserted in c_1 and c_{m} wherever c_2 crosses a knot line in the parametric space of s_2 or at a local extremum. Finally, extra knot multiplicities are inserted into B-spline representations of the trimming curves to cause the representation to turn into piecewise Bézier. Each Bézier curve segment is monotonic in its embedded surface's parameter space. If the trimmed representation input does not have these prerequisite characteristics, the technique in [36] can be used to produce it.

4.1 Creating the Rectilinear Approximation

When a parametric trimming curve touches a knot interval, either by crossing it or sharing part of its parametric boundary, that interval is called a *trimmed knot interval*. The trimming curve may have multiple Bézier segments within a single trimmed knot interval. (See Figure 3a.) A rectilinear approximation is built for each segment. We will present several approaches for dealing with the curvilinear triangles that result from constructing Ω. The approach to creating the rectilinear approximation and the region Ω is the same for whichever alternative is chosen.

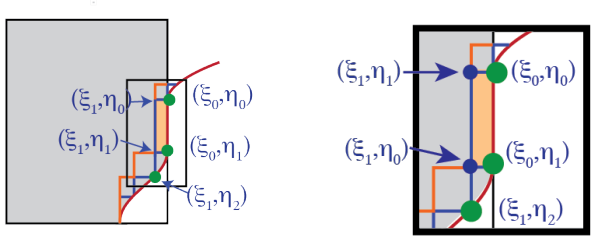
Given a Bézier segment with endpoints at (ξ_0, η_0), and (ξ_1, η_1), we assume without loss of generality that (ξ_0, η_1) is in the interior of the trimmed domain, as in Figure 3b. The rectilinear approximation to the Bézier segment is found by connecting the two segment endpoints to (ξ_0, η_1). The orange region between the trimming Bézier segment and its rectilinear approximation defines a curvilinear triangle that is the interior part of the sub-knot interval with vertices (ξ_0, η_1), (ξ_0, η_0), (ξ_1, η_0), and (ξ_1, η_1). It is called a *curvilinear triangular sub-knot interval*. In Figure 3c, the rectilinear approximations and their corresponding curvilinear triangular sub-knot intervals are shown for all the Bézier segments in the trimmed knot interval. Figure 3d shows (ξ_1, η_1) is a shared endpoint of two Bézier trimming segments and is on the rectilinear approximation that is not separated from the interior. It must have an added rectangle whose boundary will form a boundary to a modified rectilinear approximation. It is constructed to have edges identical with its adjacent curvilinear triangle sub-knot intervals. This specifies all vertices of the rectangle, as the vertex (ξ_1, η_2) is implied by the other three, as shown in Figure 3d (the orange rectangle). In Figure 3d, all such rectangles constructed to separate points on the trimming curve from the rest of the interval have orange and blue sides. For the trimmed knot interval shown, Ω is defined as the region bounded by the orange rectilinear piecewise linear curve and the trimming curve.

If instead of the configuration depicted in Figure 3, a Bézier segment is isoparametric as in Figure 4a with endpoints (ξ_0, η_0) and (ξ_0, η_1), a suitable rectilinear approximation is found with corners (ξ_0, η_0), (ξ_1, η_0), (ξ_1, η_1), (ξ_0, η_1) where (ξ_1, η_1) follows from the endpoint of the adjacent curvilinear triangle and of the isoparametric segment. In this case, the region between the Bézier curve and its rectilinear approximation is a rectangular sub-knot interval, shown in orange in Figure 4. Note that rectilinear approximations should not have intersections with rectilinear approximations for other trimming curves nor with themselves. If it occurs, the related trimming curve segments should be subdivided before a rectilinear approximation is recomputed.



(a) (b) (c) (d)

Figure 3: A progressive construction for generating Ω. The retained portion of a single knot interval is in gray. (a) a trimming curve with 5 Bézier segments (breaks shown with green markers.) (b) the rectilinear approximation and curvilinear triangle for one trimming Bézier segment (in orange). Similarly, this is repeated for each Bézier trimming curve segment with results shown in (c). In (d) rectangular sub-knot intervals are created that have one corner point on the trimming curve. See the orange rectangle. The orange rectangle's fourth corner (ξ_1, η_2) aligns with the right angle vertices of adjacent curvilinear triangles.



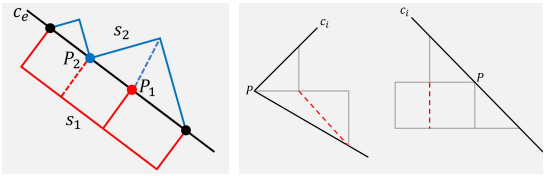
(a) (b)

Figure 4: (a) A trimmed knot interval with a trimming curve touching its boundary. (b) Ω near an isoparametric trimming curve segment.

4.2 Consistency Constraints

In order to keep the resulting new basis functions C^0 *valid* sub-knot intervals in Ω requires that (i) curvilinear triangular sub-knot intervals exist only in Ω; (ii) the initial and end control points of each Bézier trimming curve segment corresponds to corners of sub-knot intervals; and (iii) a shared edge between two sub-knot intervals in Ω must be the full edge of the two involved sub-knot intervals. Two types of inconsistencies may result in invalid Ω and must be resolved to ensure the new basis functions are C^0 over each trimmed surface and across multiple surfaces. This section describes the inconsistencies and presents methods to resolve them.

A *Type 1* inconsistency can occur across the trimming curve of two surfaces. Since Bézier segments in a surface may have been subdivided during Section 4.1, representations of the trimming curves segments on each side of the intersection may no longer be synchronized. The piecewise Bézier trimming curves c_1 and c_2 from their respective surfaces, now augmented through the results of Section 4.1, are compared. c_1 is subdivided at a value if c_2 had been subdivided, and vice versa, and c_m is made consistent. Corresponding sub-knot interval partitioning then takes place in s_1 and s_2 to accommodate the added trimming curve segments, also modifying the rectilinear partitioning. See Figure 5a for illustration.



(a) *Type 1* (b) *Type 2*

Figure 5: (a) *Type 1* inconsistency: Section 4.1 introduced a subdivision to the trimming curve at P_1 on surface s_1 and at P_2 on s_2 causing inconsistent trimming curve representations. Corresponding sub-knot intervals must be partitioned (dashed lines). (b) Two examples of *Type 2* inconsistency: T-junctions exist along an isoparametric edge inside Ω in the parametric domain of trimmed surfaces. They must be eliminated by further partitioning sub-knot intervals (red dashed line).

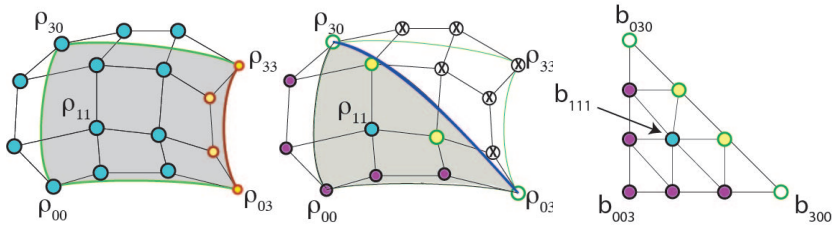
A *Type 2* inconsistency arises within Ω of a single trimmed surface when T-junctions were created during partitioning. This is a potential problem only within Ω since basis functions are defined across sub-knot intervals. If two adjacent sub-knot intervals share one edge i.e., have the same endpoints on that edge, both tensor product Bézier basis function and Bézier curvilinear triangle basis function reduce to 1-D Bézier basis functions on that edge, forming a C^0 basis functions across that edge. When an edge is only partially shared at a T-junction, it breaks the continuity when the control point associated with the trimming curve is moved. So the relevant sub-knot interval is subdivided, as shown in Figure 5b.

Satisfaction of each constraint process on one surface might introduce new sub-knot intervals, breaking the consistency constraint of the other, and so require repetition. Since only partitioning of rectangular sub-knot intervals can lead to new T-junctions, the process termination condition is that no rectangular sub-knot intervals is allowed to have two opposite edges that both contain trimming curves. Before constraint satisfaction, the algorithm checks and bisects any such rectangular sub-knot interval.

4.3 Revised and Trim Basis Functions

Ω is used in defining the revised and trim basis functions. A sub-knot interval in Ω is either an isoparametric rectangular sub-knot interval, denoted J^{rect}, or a curvilinear triangular sub-knot interval, denoted J^{tri}. The sub-knot interval has been trimmed by the trimming curve according to one of the following configurations: (i) at least one isoparametric edge of J^{rect} resides on a trimming curve, (ii) a corner of J^{rect} is on the trimming curve, or (iii) the curvilinear edge of J^{tri} is part of trimming curve, or (iv) a curvilinear triangle that has been split into multiple triangles to avoid an inconsistency.

Using the multi-index $\boldsymbol{i}$, let $N_{\boldsymbol{i}} = N_{i_1,i_2}$. Each B-spline $N_{\boldsymbol{i}}$ and its corresponding revised basis function $\tilde{N}_{\boldsymbol{i}}$ are identical, except in Ω. Thus, for $(\xi,\eta) \in J^{\text{rect}} \not\subseteq \Omega$, then $\tilde{N}_{\boldsymbol{i}}(\xi,\eta) \equiv N_{\boldsymbol{i}}(\xi,\eta)$, and it can be evaluated using Equation (2). For $J^{\text{rect}} \subseteq \Omega$, where the support of $N_{\boldsymbol{i}}$ overlaps J^{rect}, using Equation (2), the tensor product Bézier functions defining $N_{\boldsymbol{i}}$ over J^{rect} are partitioned into two sets: those basis functions that vanish on the trimming curve and those that are nonzero. Letting $d = 3$, Figure 6 illustrates this for bicubic surfaces. Figure 6a shows a red trimming curve segment coincident with a sub-knot interval's isoparametric boundary. The basis functions that vanish on the trimming curve, whose coefficients in Figure 6a have blue dots, are used in Equation (2) to define $\tilde{N}_{\boldsymbol{i}}$. Then $\tilde{N}_{\boldsymbol{i}}$ is defined over J^{rect} by substituting the 0 functions in Equation (2) for those Bézier functions that are nonzero on the trimming curve. Over J^{rect}, $\tilde{N}_{\boldsymbol{i}}$ now has been modified to



(a) (b) (c)

Figure 6: Trimmed sub-knot interval with local representations. In (a) Colors show the groupings of control points associated with basis functions for revised basis functions and trim basis functions. In (b) the groupings show which tensor product basis functions, signified by their control points, are replaced by triangular Bézier basis functions. The X-ed vertices indication subscripts of functions that are discarded. Along with (c) the locations in the summation for the Bézier functions are specified.

decrease continuously to have value 0 on the parametric trimming curve. For $(\xi, \eta) \in J^{\text{rect}} \subseteq \Omega$, as in Figure 6a, the revised basis functions and the trim functions are now,

$$\tilde{N}_{i_1,i_2}(\xi,\eta) = \sum_{k_1=0}^{3}\sum_{k_2=0}^{2} \alpha_{i_1,k_1}^{\boldsymbol{\mu},\xi}\alpha_{i_2,k_2}^{\boldsymbol{\nu},\eta}B_{k_1,k_2}^{\xi,\eta}(\xi,\eta) \tag{6}$$

$$\tilde{T}_{k_1}(\xi,\eta) = B_{k_1,3}^{\xi,\eta}(\xi,\eta) \quad \text{for } i = 0,1,\ldots,3. \tag{7}$$

$$s^{\text{R}}(\xi,\eta) = \sum_{i_1}\sum_{i_2} r_{i_1,i_2}\tilde{N}_{i_1,i_2} + \sum_{k_1=0}^{3}\rho_{k_1,3}\tilde{T}_{k_1}(\xi,\eta) \tag{8}$$
$$\text{for } (\xi,\eta) \in J^{\text{rect}} \in \Omega$$

Each Bézier function for J^{rect} that is nonzero on the trimming curve is used to define a trim function(s) over that element. Using its coefficient from Equation (4) and keeping its part of the summation under its new name still permits the surface to be represented exactly over J^{rect}, so $s \equiv s^{\text{R}}$, as in Equation (8). When the coefficients of the trim functions are modified to match the model space trimming curve, thus sealing the model, s^{R} is altered to be $\tilde{s}$, an approximation of s that matches c_{m} exactly over the shared model space Bézier trimming segment.

Given a $J^{\text{tri}} \subseteq \Omega$, there are two main approaches to approximating the model space surface. They can result in related representations. Without loss of generality, suppose the curvilinear triangle has 3 corners at (ξ_0, η_0), (ξ_1, η_0), and (ξ_0, η_1) with 2 isoparametric edges and 1 Bézier trimming curve segment edge, as in Figure 7b. Note that these vertices are labeled differently than those in Figure 3b to simplify the explanation. Assume the model space untrimmed surface in Figure 6b is over the sub-knot interval $[\xi_0, \xi_1] \times [\eta_0, \eta_1]$ with the basis functions and control mesh generated by Equation (2) and (4), as shown in Figure 6b. Let $\triangle$ be the triangle with the same corners as J^{tri}, and let $\{G_{k_1,k_2,k_3}\}_{\substack{0\le k_1,k_2,k_3 \\ k_1+k_2+k_3=d}}$ be the Bézier triangle basis functions over $\triangle$.

Figure 7a shows a trimmed sub-knot interval for a single Bézier segment of the parametric trimming curve. One approach, call it *Method I* approximates the surface over a straight-sided triangular subregion of Ω, shown in Figure 7c as the green triangle. A Bézier triangle representation is constructed to approximate the surface and then the control points along the diagonal edge that is supposed to correspond to the trimming curve are adjusted to be the control points for the relevant

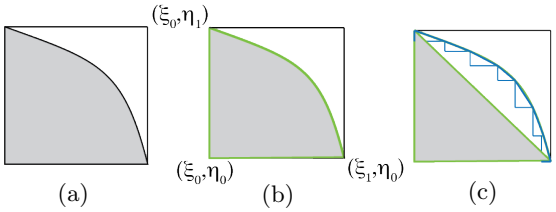


Figure 7: Possible triangle regions. (a): A single cubic Bézier trimming segment and its sub-knot interval. (b) The curvilinear triangle with the Bézier segment forms the rectilinear approximation (green). (c) The Bézier segment is approximated by either a single edge (green triangle) that can result in poor coverage of the parametric domain. A good fit to the parametric trimming curve (blue) has a good parametric approximation and more triangular regions are used.

Bézier segment of c_{m}. The parameter space for Method I is either not the full curvilinear triangle J^{tri} (for example in Figure 7c shaded) or the rectilinear approach must be recomputed and the approximation must occur over multiple straight-sided triangles. It has the advantage that by using a degree $2d$ Bézier triangle, the surface over that straight-sided triangle can be computed exactly. However, the approach presented here has the goal of using the full curvilinear parametric triangle in its surface representation. To do this, the single Bézier segment must be approximated to get better coverage of the curvilinear triangle, and then the rectilinear approximation must be reconstructed and Ω recomputed. Typically, the triangle size becomes relatively small and the polynomial surface over it can be approximated with a low degree surface. In what follows, we approximate the surface over $\triangle$ by constructing revised and trim basis functions using cubic Bézier triangle basis functions and name the approximation s^{T}. We discuss why this is appropriate at the end of the presentation of the construction.

The second approach, say *Method II* first creates a Bézier triangle mapping $\phi : \triangle \to J^{\mathrm{tri}}$ that is the identity on the two isoparametric sides and maps the diagonal to the parametric space trimming curve. Then, each revised and trim basis function over $\triangle$ is composed with ϕ^{-1} to create a mapping $s^{\mathrm{c}} = s^{\mathrm{T}} \circ \phi^{-1}$ that maps J^{tri} to a model space approximation of s. If s is a bicubic B-spline surface, then each revised basis function is exactly the same curve as its corresponding B-spline basis function on the isoparametric boundaries of J^{tri}, and s^{c} matches s. In that way, J^{tri} remains the parametric space for that piece of the surface. However, ϕ^{-1} is not rational, so s^{c} is not rational. Numerical integration methods can take advantage of the curvilinear parametric domain in Method II when the mapping from the integration domain to J^{tri} uses the curvilinear mapping in [33].

The idea for Method I is to substitute Bézier triangle basis functions for tensor product basis functions in Equation (2), and to determine trim functions from the Bézier triangle basis functions. Since J^{tri} is only a subset of the rectangle, all the tensor product basis functions are not used. Without loss of generality, the substitution rules are presented for just one triangle because the relative relationships between the triangle and the tensor product basis functions subscripts (control net) are rotated and analogously applied for the other types of triangles. For the surface triangle shown in Figure 6b, the Bézier triangle basis functions are substituted for the tensor product Bézier basis functions whose subscripts are in the lower triangular control mesh of tensor products subscripts, and split into those that are zero on the trimming curve and used to define $\tilde{N}$'s ,and those that are not and used for $\tilde{T}$'s. For the case illustrated in Figure 6b, the basis functions with subscripts above

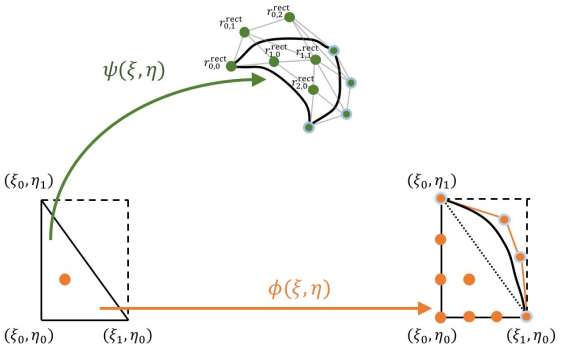


Figure 8: ϕ maps from a related parametric triangle to the parametric domain of the curvilinear triangle, while ψ maps from the parametric triangle to the physical domain. Coefficients of ψ and ϕ with light blue circles are from Euclidean/parametric trimming curves, respectively; the rest of those for ψ are from $\mathbf{r}^{\text{rect}}$ (green) and its corresponding nodal values (orange). The original surface mapping is approximated by $\psi \circ \phi^{-1}$.

the diagonal are discarded. Using $d = 3$ as indicated in Figure 6b,

$$\text{for } (\xi,\eta) \in \triangle^{\xi,\eta} \in \Omega$$
$$\text{the following equations hold:}$$

$$\tilde{N}_{i_1,i_2}(\xi,\eta) = \sum_{\substack{0\le k_1,k_2;0<k_3,\\ k_1+k_2+k_3=3}} \alpha^{\boldsymbol{\mu},\xi}_{i_1,k_1}\alpha^{\boldsymbol{\nu},\eta}_{i_2,k_2} G_{k_1,k_2,3-k_1-k_2}(\xi,\eta) \tag{9}$$

$$\tilde{T}_{k_1}(\xi,\eta) = G_{3-k_1,k_1,0}(\xi,\eta) \quad \text{for } k_1 = 0,1,\ldots,3. \tag{10}$$

$$s^{\mathrm{T}}(\xi,\eta) = \sum_{i_1}\sum_{i_2} r_{i_1,i_2}\tilde{N}_{i_1,i_2}(\xi,\eta) + \sum_{k_1=0}^{3} \rho_{3-k_1,k_1}\tilde{T}_{k_1}(\xi,\eta) \tag{11}$$

The coefficients for the trim functions in Equation (11) are modified to be the coefficients for the Bézier segment of c_{m} over J^{tri} and form a final approximation in Section 4.4. The set of all control points that are coefficients of trim functions is Υ. There are four possible orientations of the retained curvilinear triangle (and its straight-sided pairing). In addition to substituting for the lower triangular subscripted tensor product basis function, the upper triangular ones, and the two using the other diagonal are the corresponding orientations. The subscripted triangular Bézier functions must be inserted into the summations with the correct orientations and parts of the sub-knot interval domain.

Method II maintains the parameterization of the original surface s as closely as possible, as well as the geometric representation, using a C^1 mapping from J^{tri} to $\triangle$. Define $\phi : \triangle \to J^{\mathrm{tri}}$ to be a degree 3 triangular Bézier mapping that is the identity on the isoparametric edges and reproduces the degree 3 parametric Bézier trimming curve segment boundary exactly and without folds or degeneracies. The simplest general initializes $\phi = \sum_{\substack{0\le i,j,k,\\ i+j+k=3}} f_{i,j,k}G_{i,j,k}(u,v) : \triangle \to \triangle$ as the identity. Then, coefficients $f_{2,1,0}$ and $f_{1,2,0}$ are modified to the interior coefficients for the cubic parametric space Bézier trimming segment, and ϕ now maps $\triangle$ to J^{tri}. (The segment end coefficients already match the coefficients for the identity function.) In the case that $f_{1,1,1}$ is not in the interior of J^{tri},then a point along the line between $f_{0,0,3}$ and $f_{1,1,1}$ that is interior to J^{tri} is chosen to replace

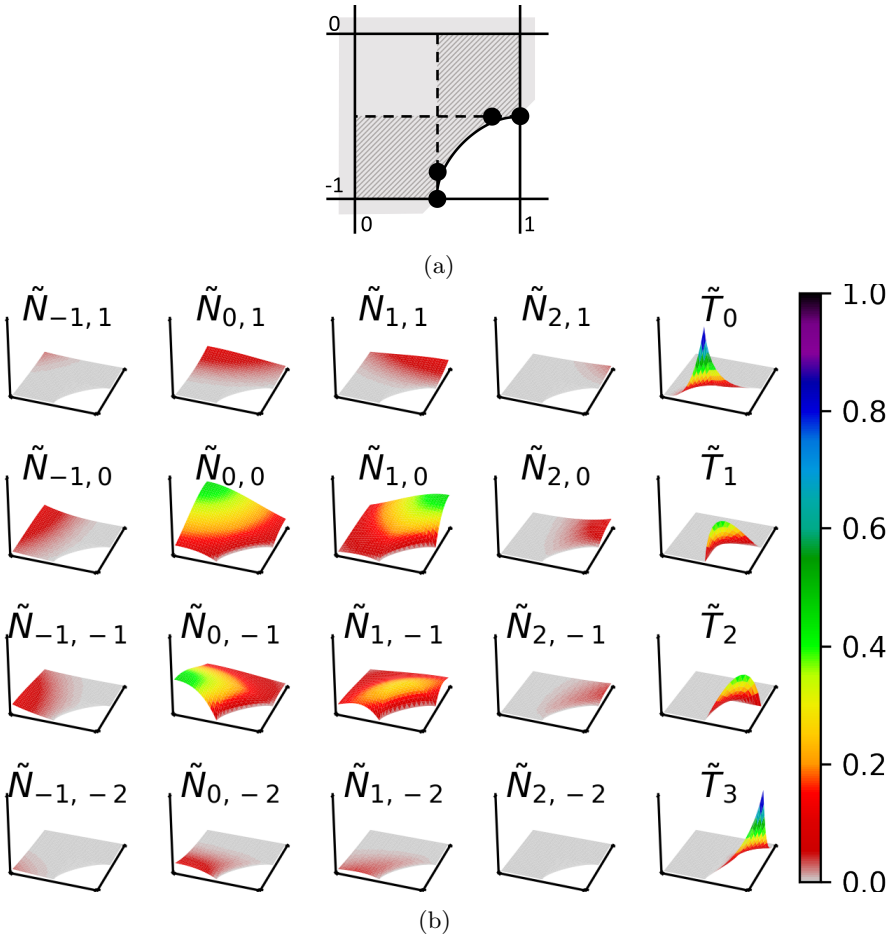


Figure 9: In (a), the partition of knot interval $[0,1] \times [-1,0]$ consists of 4 sub-knot intervals: 3 rectangular and 1 curvilinear triangular. Ω is represented by the hatched region, and the 4 black dots represent the control points of the trimming curve. (b) shows all of the 16 revised B-spline basis functions $\tilde{N}_{i,j}$ and 4 trim basis functions $\tilde{T}_i$ corresponding to the 4 dots.

the value of the identity $f_{1,1,1}$.

ϕ^{-1} is C^1 inside J^{tri} and maps onto $\triangle$. Each $\tilde{N}_{\boldsymbol{i}}$ is modified over J^{tri}, so that instead of substituting Bézier triangle basis functions $G_{\boldsymbol{i}}$ into Equation (2), $G_{\boldsymbol{i}} \circ \phi^{-1}$ is substituted. The Bézier triangle functions that are used for the trim functions are composed with ϕ^{-1} similarly. This new mapping, $s^{\text{c}} = s^{\text{T}} \circ \phi^{-1}$, is a reparameterization of s^{T} so all of J^{tri}, the parametric domain of the trimmed surface can be used. (See Figure 8.)

Figure 9a shows a partition of trimmed knot interval. Originally, 16 B-spline basis functions $N_{i,j}$ for $i \in \{-1,0,1,2\}, j \in \{-2,-1,0,1\}$ exist over this domain corresponding to 16 original control points. Two types of new basis functions exist: (i) 16 revised B-spline basis functions $\tilde{N}_{i,j}$ that are different from original ones only in Ω, and forced to be 0 on the trimming curve; (ii) 4 trim basis functions $\tilde{T}_i$ for $i \in \{0,1,2,3\}$ with coefficients that correspond to 4 control points of the Euclidean cubic trimming curve (Figure 9b). $\tilde{T}_i$ serve as C^0 interface basis functions for consistency across trimming curves. Outside of Ω, all $\tilde{T}_i = 0$, and $\tilde{N}_{i,j} = N_{i,j}$. The extraction relationship between

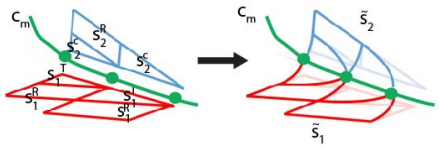


Figure 10: Making the representation watertight. Two trimmed surfaces that intersect at c_{m} (green) in model space. Representations of each surface over its trimmed J^{rect} and J^{tri} are shown to not actually coincide. Then the appropriate coefficients of the trim basis functions are modified so that the model space trimming curve bounds on both the red and blue surfaces.

the original surface and the sub-knot interval is maintained so there are no new degrees of freedom, except those in Υ.

4.4 Sealing the Representation

Each trim function from s^{R} and s^{c} has associated with it a cubic parametric space Bézier segment and its corresponding cubic model space Bézier segment from c_{m}. Since c_{m} is compatible with the parametric space trimming curves, global subscripts are coordinated, so the control points of s^{R} and s^{c} corresponding to the trimming curve can be replaced by the corresponding control point for c_{m}, as shown in Figure 10. All individual trim basis functions with the same control point coming from different sub-knot intervals and different surfaces are combined to form global trim basis function. The global $\tilde{N}_{\boldsymbol{i}}$ is completely defined by its behavior for parametric points Ω and those not in Ω, with two different types of $\tilde{N}$ depending on whether the full parametric domain is desirable or whether just the geometry is important. The control point for a revised basis function is set to be the control point for the corresponding B-spline basis function. The new basis functions are thereby constructed, and with the specified coefficients, the new representation is watertight, as illustrated in Figure 10. The set of all control points that are coefficients of trim basis functions is denoted Υ. The total number of independent basis functions in the watertight representation is the sum of the number of control points in $\mathcal{G}$ and the number of control points in Υ.

By careful construction, the error in the new representation $\tilde{s}$ occurs only over intervals in Ω. If the revised functions are constructed using the straight-sided triangles, the error is difficult to measure because of the lack of agreement in the parametric domains of s and $\tilde{s}$. While it is possible to measure error between the same parameter value on the surfaces, that leaves some part of s that has no comparison in $\tilde{s}$, or vice versa, depending on the particular curvilinear triangles. Otherwise, it is possible to measure only orthogonal distance between points on each surface and the other surface. Instead, by using the curvilinear triangle intervals as the domain with the non-polynomial result, it is possible to measure the error as the distances between model space values of the same parameter under s and $\tilde{s}$. Effectively, this also measures the error in the straight-sided triangle approximation.

The choice was made to use degree d Bézier triangles instead of degree $2d$, despite having more accurate representational properties with a higher degree [41]. The main representation is the tensor product B-spline from the original surface, so in creating the HB-rep, one aim was to keep the curvilinear triangle regions relatively small where the surface is not rapidly changing. Using degree d Bézier triangle basis functions has added triangular regions, but overall, the error has been small to start and decreased rapidly with any trimming curve subdivision. We discuss examples in Section 6.

There are two types of error that can easily be avoided in the stage of creating the rectilinear

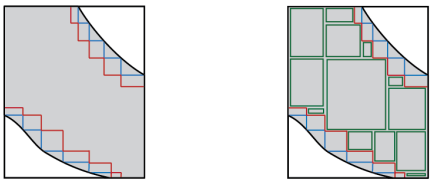


Figure 11: left: The trimmed knot interval minus Ω results in a rectilinear region, shown bounded by orange rectilinear curves. right: The inset partitioning into rectangular sub-knot intervals by the greedy algorithm

approximation and Ω. In one type, the triangular region may be too big a part of the surface, even if it is an excellent approximation. This can happen with relatively flat surfaces. Or the approximation s^{c} may not be a good approximation to the surface. In both cases, these attributes can be tested in the early stage of Ω creation by creating each local surface $\tilde{s^{\mathrm{c}}}$ and $\tilde{s^{\mathrm{R}}}$ and testing it. If either error problem arises, the triple of curves representing a single trimming curve can be further subdivided. With each subdivision, the size of the new curvilinear triangles is approximately 1/4 the parent triangle size. With that subdivision, new rectangular elements may be added to Ω, and some rectangular sub-knot intervals must have their sides shortened to match the length of the sides of the new adjacent curvilinear triangles. Since the representation is approximated for the smaller curvilinear triangles, if the parametric trimming curve segment is short enough then it will be almost a straight line, so ϕ is close to the identity. Little error occurs in assuming ϕ and ϕ^{-1} are the identities.

4.5 Partitioning each $K-\Omega$

After Ω is created in a trimmed knot interval K, the residual part of the trimmed knot interval $K-\Omega$ is a rectilinear region. All evaluations and properties can be computed over this region using the original surface representation. While rectilinear, $K-\Omega$ is not a rectangle, so, for the purpose of computation, it is partitioned into interior rectangular sub-knot intervals whose local surface representations are defined as in Equation (2) and (3). A greedy divide-and-conquer recursive approach was introduced to find locally *as square as possible* rectangles (See Figure 11). These partitions enable fast, simple computation over $\tilde{s}$.

One goal is that the resulting sub-knot intervals should not have a relatively very short edge. However, finding an optimal solution of such a problem is considered to be NP-hard [25]. Therefore, a divide-and-conquer approach is introduced to solve the problem recursively. Starting from any rectilinear domain the algorithm iterates through all vertex pairs, for example, (ξ_i, η_i) and (ξ_j, η_j), and tests if the axis-aligned rectangle defined by two points (with 4 vertices (ξ_i, η_i), (ξ_i, η_j), (ξ_j, η_j) and (ξ_j, η_i)) is valid, i.e., non-degenerate and interior to the rectilinear region without intersections with its boundary. Valid rectangles are considered candidates. Edge length ratio of all candidates is evaluated; the one with ratio closest to 1 (with its shape closest to a square) is chosen and removed from the rectilinear domain. The removal partitions the residual rectilinear domain into 1 or more pieces. The scheme is then recursively applied to the remaining pieces until the rectilinear is partitioned into rectangular sub-knot intervals.

The final collection of sub-knot intervals in the segmentation of the trimmed domains of all the surfaces in the B-rep is named Ξ. All interior knot intervals of the original trimmed surfaces are in Ξ. After being converted into watertight HB-rep, the revised version of surfaces in $\mathcal{G}$ along with their partitioning Ξ is denoted $\mathcal{B}$. The set of B-spline control points needed to define $\mathcal{B}$ is $\mathcal{R} \cup \Upsilon$,

where $\mathcal{R}$ is the set of control points needed for the trimmed surfaces in $\mathcal{G}$ and Υ is the set of control points of the introduced basis functions. Therefore, the total number of control points is equal to $|\mathcal{R}|+|\Upsilon|$. We note that $|\Upsilon|$ can be larger than the number of control points on the original trimming curve since some Bézier segments may have been subdivided in Section 4.2.

5 HV-rep $\mathcal{V}$

B-spline B-rep models with no trimming curves have been completed to volumes [21, 41]. With no trimming curves and consistent open knot vectors and surface degrees, the model control meshes quad faces converted to triangles. The result is manifold and forms a polyhedron. Then, a point set mid-structure was input and a discrete harmonic function created with increasing values from the boundary to the mid-structure. Each of the surface control points was traced in the harmonic gradient field and used to compute the control mesh for a trivariate B-spline representation that had one boundary surface on the original model boundary and the other boundary surface in the interior. The remaining interior was filled with Bézier tetrahedra. While related to the approach in [41], the approach presented here deals with the difficult issues occasioned by having *trimmed surfaces*.

A trimmed model (denoted $\mathcal{G}$) contains trimmed surfaces. Trimming curves go through the parametric domain of the surface, representing the boundary of trimmed surfaces. An HB-rep (denoted $\mathcal{B}$) of $\mathcal{G}$ is identical to $\mathcal{G}$ except in a banded region near the trimming curves where control points from the trimming curve along with introduced trim basis functions help seal the seam. Control points from trimmed surfaces and trimming curves that contribute to $\mathcal{B}$ are denoted $\mathcal{R}$ and Υ, respectively.

That process of volume completion cannot be directly applied to the HB-rep model for several reasons. In particular,

1. The collection of control meshes inherited from the trimmed B-rep and used as the control meshes of $\mathcal{B}$ do not form a manifold, as shown in Fig. 13. This requires determining a new approach to creating a vector field through which to trace control points.

2. The construction of trivariate region in volume completion (as did in [21, 41]) require that the traced control mesh results in manifold topologically identical to $\mathcal{B}$ in each layer so that the basis functions defined in Section 4 can be used. However, the surfaces move inward independently and the match at the parametric trimming curves is lost because there is a lack of control mesh consistency.

3. A discrete harmonic field may have negative [35] or 0 value around the boundary. These regions can lead to gradient fields pointing outside the model or gradient values of 0.

4. In efforts to generate Bézier tetrahedra from tensor product B-reps such as in [41, 36], surfaces are often reproduced, which lead to degree 6 Bézier tetrahedron from a bi-cubic B-spline. This lead to high DoF in tetrahedra.

The methodology in creating the Hybrid Volume Representation (HV-rep) goes according to the following steps:

1. Input an HB-rep $\mathcal{B}$ and a mid-structure $\mathcal{M}$, as well as several parameters, where one of them is the number of control levels in the final trivariate spline, say n. Note that the information in $\mathcal{G}$ is contained in $\mathcal{B}$.

2. Create a vector field for tracing the control points in $\mathcal{R}$ from the given boundary surface to the mid-structure $\mathcal{M}$. Set $\mathcal{R}^0 = \mathcal{R}$.

3. Proceed through the following process, for i from 1 to n,

   (a) Trace each control point in $\mathcal{R}^{i-1}$ from its current location to the mid-structure and compute its length.

   (b) Normalize the newly computed length to $\frac{n-i+1}{n}$, and move along the trace a scaled distance of $\frac{1}{n}$. These new points form the initial i^{th} control mesh layer $\mathcal{R}^i_{\text{tr}}$. Since it is likely that when the trimming curves are applied, the surfaces do not match along their trims, they must be adjusted.

   (c) Apply a least squares process to adjust the surfaces so the trimmed surfaces edges are close to each other, forming $\mathcal{R}^i_{\text{lstsq}}$.

   (d) Move the surfaces apart so that there is a small separation gap around each trimming curve (i.e., the surfaces do not overlap), forming $\mathcal{R}^i_{\text{sep}}$

   (e) Form $\mathcal{R}^i$ by applying mesh smoothing (as in [21, 41]) so the spacing of the tracings is more uniform.

   (f) Using Ω, the revised and trim basis functions, seal the layer and create $\mathcal{B}^i$.

4. Combine all surface layers $\mathcal{B}^i$ to generate a trivariate structure

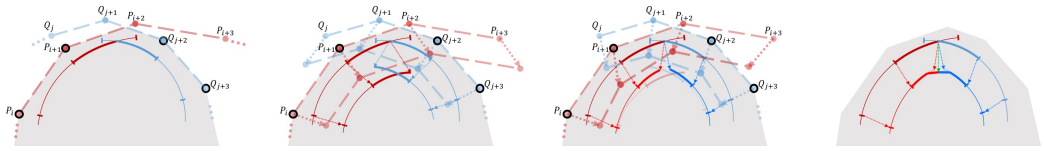
5. Create a Bézier pyramid element layer using Ξ in the innermost region of the trivariate B-spline

6. The interior volume without representation is filled with Bézier tetrahedral elements

The input is the HB-rep $\mathcal{B}$, a mid-structure $\mathcal{M}$ and parameters, including n, the number of inward trajectory layers, the parameter of the innermost layer ω, offsetting distance ϵ, blending factor β of Laplacian smoothing and pyramid thickness parameter θ. The naming convention has the superscript designate the corresponding layer, and the subscript means the stage in each layer such as $\mathcal{R}^i_{\text{tr}}$ and $\mathcal{G}^i$. By using the naming convention, Step 3 is simply,

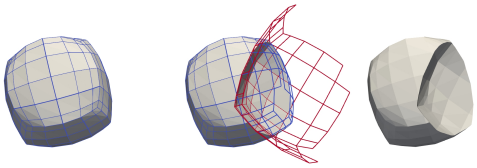
$$\ldots \mathcal{R}^{i-1} \xrightarrow{\text{tracing}} \mathcal{R}^i_{\text{tr}} \xrightarrow{\text{least squares}} \mathcal{R}^i_{\text{lstsq}} \xrightarrow{\text{separation}} \mathcal{R}^i_{\text{sep}} \xrightarrow{\text{relaxation}} \mathcal{R}^i \xrightarrow{\text{tracing}} \mathcal{R}^{i+1}_{\text{tr}} \ldots$$

Fig. 12 demonstrates the issues as well as the solution approach in a 2D scenario. Section 5.1 (corresponding to Step 3a and 3b) presents a methodology to find an *inward trajectory* of $n > d$ points for each control point in $\mathcal{R}$, where n is an input parameter. The i^{th} points in all trajectories form a layer $\mathcal{R}^i_{\text{tr}}$ of meshes. A surface layer $\mathcal{G}^i_{\text{tr}}$ can be evaluated using B-spline functions and corresponding control points as specified in $\mathcal{R}^i_{\text{tr}}$. This treats the trajectories of $\mathcal{R}$ as families of trimmed surfaces, one family for each surface in $\mathcal{G}$. The original B-spline functions are used because for each collection of surfaces $\mathcal{G}^i$, appropriate new trimming curves must be computed. Each surface in a family has the basis functions and parametric trimming curves. It is desirable that the edges of the two surfaces across a trimming curve are near each other, so the method in Section 4 can be applied to glue them. However, since each surface in $\mathcal{G}^i$ has been computed individually, it is possible for two of them to intersect in other places or have a large gap (Fig. 12b). Section 5.2 (Step 3c-3f) presents the 3-step method used to adjust i^{th} trajectory control point locations to address this issue with $\mathcal{R}^i_{\text{lstsq}}$, $\mathcal{R}^i_{\text{sep}}$ and $\mathcal{R}^i$ as the resulting control points for each step. To create a C^0 match from a bi-degree d to triangular surfaces at the interface between the unstructured tetrahedral region and the structured tensor product region, it was necessary in [41] to use degree $2d$ triangles, and hence degree $2d$ tetrahedra. In Section 5.3 (Step 4-6), a Bézier pyramid interface layer is generated that enables use of degree d tetrahedra instead.



(a) (b) (c) (d)

Figure 12: A 2D example of our approach. (a) An intersection of two curves with control polygons ($P_i \dots P_{i+3}$ and $Q_i \dots Q_{i+3}$) uses only part of the curves (thick strokes). Control points are separated into two groups: $\mathcal{E}$ (colored dots in black circles) and $\mathcal{R} - \mathcal{E}$. (b) $\mathcal{E}$ are traced appropriately through the vector field while inward trajectories for $\mathcal{R} - \mathcal{E}$ are generated using the trajectories from their mesh neighbors (Section 5.1). Resulting curves can intersect with each other and are resolved in Section 5.2 (c). In (d), endpoints of traced trimmed curve are moved to their midpoint sealing the gap.



(a) (b) (c)

Figure 13: Generation of polyhedron $\mathcal{P}$: (a) shows the control mesh (blue) and resulting triangle mesh (grey) of an untrimmed model. The triangle mesh forms a manifold. (b) Boolean subtraction is taken places on two models. The parts of the control meshes that affects the resulting trimmed surfaces are drawn in blue and red. (c) shows the resulting polyhedron $\mathcal{P}$ that is a manifold with the same topology with $\mathcal{B}$.

5.1 Tracing $\mathcal{R}$

This subsection presents the construction of a sequence of n copies of each control point in $\mathcal{R}$. The control meshes $\mathcal{R}$ are used to form a polyhedron, that, along with a mid-structure $\mathcal{M}$, help create a vector field that points inward from its boundary. While points on the polyhedral boundary can be traced, not all of the points in $\mathcal{R}$ are vertices of the polyhedron, but still must be traced. Ensuring the result leads to manifold trimmed inward surfaces is addressed in Section 5.2. Each sequence of traced n copies of a control point is an inward trajectory of the control point, starting from the original and moving each subsequent copy closer to the center of the body (as defined by the $\mathcal{M}$) than the previous.

First, the control meshes in $\mathcal{R}$ each quadrilateral of a control mesh is divided into two triangles, creating faceted triangle meshes. The same Boolean operations that were used to create $\mathcal{G}$ are then applied to corresponding triangle meshes to generate a polyhedron $\mathcal{P}$ that should have the same topology as $\mathcal{B}$. If the control meshes are too coarse, it may be necessary to refine individual surfaces until this requirement is satisfied. This is always possible since, under refinement, control meshes converge to the surfaces. Then $\mathcal{R}$ is separated into subsets: those control points that are on $\mathcal{P}$ and those that are not (denoted $\mathcal{E}$ and $\mathcal{R} - \mathcal{E}$, respectively). Define $\mathcal{O}_\mathcal{E} = \{r \in \mathcal{E} :$ there is an edge in $\mathcal{R}$ joining r to a point in $\mathcal{R} - \mathcal{E}\}$. Define $\mathcal{O}_{\mathcal{R}-\mathcal{E}}$ analogously, interchanging the roles of $\mathcal{E}$ and $\mathcal{R} - \mathcal{E}$. Define $\mathcal{O} = \mathcal{O}_\mathcal{E} \cup \mathcal{O}_{\mathcal{R}-\mathcal{E}}$. Geometrically, the control points in $\mathcal{O}$ are close to

the trimming curves.

A mid-structure $\mathcal{M}$ is an input to the tracing process. It can be computed from $\mathcal{P}$ using methods such as those that create polyhedral medial axes or simplified versions (farther away from the boundary) such as [24] and [17] or by sampling the orthogonal inward offset of $\mathcal{P}$. A simplified mid-structure farther from the boundary can lead to a region $\mathscr{S}$ that penetrates deeper into the interior. An exact medial axis is likely to extend down to the polyhedral vertices and edges and so must be simplified. The user decides the appropriate mid-structure as in [21, 41].

The control points of $\mathcal{E}$ are traced inward towards $\mathcal{M}$ by following a vector field. In [20, 35], the points in $\mathcal{P}$ along with the points in $\mathcal{M}$ are tetrahedralized using TetGen [32]. By setting $\mathcal{P}$ (value 0) and $\mathcal{M}$ (value 1) as the Dirichlet boundary conditions and solving a discrete Laplacian equation over the tetrahedral mesh, a discrete harmonic function is created [35]. Ideally, by following the gradient field of the harmonic function, an inward trajectory for each vertex on the boundary can be generated. However, that gradient may be 0, i.e., when all four corners of a tetrahedron have the same harmonic value, or, may point outward if the harmonic function has negative value around the boundary [35] as a property of the cotangent weight used to form the Laplacian operator. To avoid this, the vector field $\boldsymbol{f}$ used is a blend of the gradient of the harmonic function, h, and the gradient of a signed distance function, g, that always points to the interior of the boundary. Both of the harmonic field and the signed distance function are computed using LibIGL [14]. A simple blend is

$$\boldsymbol{f} = \begin{cases} h\frac{\nabla h}{\|\nabla h\|} + (1-h)\nabla g & \text{if } 0 < h \leq 1, \\ \nabla g & \text{if } h \leq 0, \end{cases} \tag{12}$$

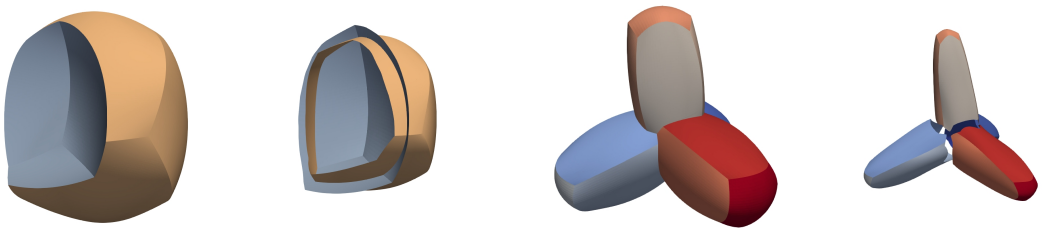
As the tracing follows an inward trajectory, the influence of the harmonic field dominates, so eventually, the trace terminates at the mid-structure $\mathcal{M}$. Because the harmonic gradient field is generated from a discrete representation it is possible that trajectories could tend to get very close or merge, also a problem with the distance vector field. We prevent this possibility by correcting the tracing trajectories in a relaxation step in Section 5.2, related to that in [21].

For each vertex in $\mathcal{E}^i$, the tracing curve to find $\mathcal{E}^{i+1}_{\text{tr}}$ terminates at harmonic value 1 (on $\mathcal{M}$). Normalized arc-length parameterization of the tracing curve are reparameterized to parametric domain $[\frac{i}{n}\omega, 1]$; and the point at parametric value $\frac{i+1}{n}\omega$ is assigned to $\mathcal{E}^{i+1}_{\text{tr}}$. As control points in $\mathcal{R}-\mathcal{E}$ are not on $\mathcal{P}$, they cannot be traced. Instead, for $p \in \mathcal{R}-\mathcal{E}$, a breadth-first search is applied on the control mesh to find $q \in \mathcal{E}$ closest to p in Euclidean distance. A translated copy of the tracing of q is used.

Finally, $\mathcal{R}$ itself and its n layers of inward trajectories, $\{\mathcal{R}^0_{\text{tr}}, \mathcal{R}^1_{\text{tr}}, \ldots, \mathcal{R}^n_{\text{tr}}\}$, with $\mathcal{R}^0_{\text{tr}} = \mathcal{R}$, form the initial estimate of the control lattice of the semi-structured volume. Control mesh $\mathcal{R}^i_{\text{tr}}$, along with the original basis functions of $\mathcal{G}$ and parametric trimming curves result in a trimmed surfaces layer $\mathcal{G}^i$. $\mathcal{G}^i$, in general, is not a manifold at this stage and its control mesh positions, $\mathcal{R}^i_{\text{tr}}$, must be adjusted as each layer is generated.

5.2 Adjusting $\mathcal{R}^i_{\text{tr}}$

By following the tracing, the resulting $\mathcal{G}^i_{\text{tr}}$ is not a manifold as illustrated in Figures 14b and 14d. $\mathcal{R}^i_{\text{tr}}$ are adjusted so that the resulting trimmed surfaces do not intersect and corresponding trimming curve pairs on different surface pieces are close to each other. In this section, control points in $\mathcal{R}^i_{\text{tr}}, 1 \leq i \leq n$ are adjusted so that trimmed surfaces in $\mathcal{G}^i$ do not intersect and corresponding trimming curve pairs on different surface pieces are close to each other. A three-step process is proposed that adjusts $\mathcal{R}^i_{\text{tr}}$. The basic idea first fits a least squares to move the surfaces on the two sides of the trimming curves close to each other, forming control meshes $\mathcal{R}^i_{\text{lstsq}}$. Then adjacent



(a) (b) (c) (d)

Figure 14: HB-rep models are shown in (a) and (c). When only tracing is applied, self-intersections are typically found in models that involve either Boolean intersection or subtractions (b). Tracing on model that only has union operations leads to gaps around trimming curves (d).

trimmed surfaces are pushed away from each other leaving a small gap around the shared trimming curve to form $\mathcal{R}^i_{\text{sep}}$. Finally, relaxation is applied to generate a better-spaced control mesh, $\mathcal{R}^i$. Tracing (Section 5.1) is applied again to generate $\mathcal{R}^{i+1}_{\text{tr}}$ from $\mathcal{R}^i$ for the next layer.

The least squares fit adjusts the control points in $\mathcal{O}$. Along a trimming curve, c_1, c_2 exist in the parametric domains of two trimmed surfaces $s^i_{1,\text{tr}}$, $s^i_{2,\text{tr}}$, respectively. We start by sampling over the trimming curve with parameter t. For each sample t on the parametric trimming curves from both sides, the reference position of the Euclidean trimming curve is computed as their midpoint,

$$c^i_{\text{m,tr}}(t) = \frac{s^i_{1,\text{tr}}c_1(t) + s^i_{2,\text{tr}}c_2(t)}{2} \tag{13}$$

To simplify the explanation, the notation from Section 4 is used, and we adopt subscript k to designate the ordering of each tensor product B-spline basis function and its corresponding control point. Thus, let s^i_{tr} be one of the surfaces resulting from tracing, and, with parametric trimming curve c,

$$s^i_{\text{tr}}(\xi,\eta) = \sum_k N_k(\xi,\eta) r^i_{k,\text{tr}}, \tag{14}$$

where $r^i_{k,\text{tr}} \in \mathcal{R}^i_{\text{tr}}$, and its basis function is N_k. The least squares system is formulated as,

$$\underset{\Delta r^i_k}{\text{minimize}} \sum_u \left\| \sum_k N_k(c(t))(r^i_{k,\text{tr}} + \Delta r^i_{k,\text{tr}}) - c^i_e(t) \right\|^2 \tag{15}$$

where $\Delta r^i_{k,\text{tr}}$ is the unknown displacement vector of control point $r^i_{k,\text{tr}}$. The geometric rationale is moving each control point in $\mathcal{R}^i_{\text{tr}}$ by its Δ solution modifies the surface so that the resulting trimming curve is close to c_{m}. As stated, only one trimming curve between $s^i_{1,\text{tr}}$ and $s^i_{2,\text{tr}}$ is considered, and each surface can be considered separately. To solve the global problem, a global least squares system is formed and solved in the same fashion. The system is formulated to solve for displacements of those coefficients in $\mathcal{O}^i_{\text{tr}}$, since they affect the surface the most. Solving a broader system whose basis functions exert much less effect can cause the system to be ill-conditioned. Hence, $\Delta r^i_{k,\text{tr}} = 0$ if $r_k \notin \mathcal{O}^i_{\text{tr}}$. We denote the set of updated control points $\mathcal{R}^i_{\text{lstsq}} = \{r^i_{k,\text{lstsq}} = r^i_{k,\text{tr}} + \Delta r^i_{k,\text{tr}} : r^i_{k,\text{tr}} \in \mathcal{R}^i_{\text{tr}}\}$.

After corresponding trimming curve pairs are moved closer to each other with the least squares solutions, the control points in $\mathcal{R}^i_{\text{lstsq}}$ are pushed away from each other across the trimming curves to avoid intersections. For that purpose, we create a direction vector in parametric space for $\mathcal{R}^i_{\text{lstsq}}$.

Each mesh of parametric direction vectors is initialized to the 0 vector. Then the parametric trimming curve is sampled and for each sample, planar curve normal direction pointing towards the interior of the trimmed domain is computed. The basis function for each control point in $\mathcal{O}^i_{\text{lstsq}}$, is evaluated at the parametric samples. The normal direction of the sample(s) with the largest basis function value is assigned to the control point and denoted $v^i_{k,\text{sep}}$. The direction vectors of $\mathcal{O}^i_{\text{lstsq}}$ are then propagated to all direction vectors in each mesh by applying 50 iterations of Laplacian smoothing over the control meshes of parametric direction vectors. Finally, we update the control mesh by moving the surface layer using these vectors assigned to each control point for a short distance (times with ϵ) to leave a gap around the trimming curve, creating $\mathcal{R}^i_{\text{sep}} = \{r^i_{k,\text{lstsq}} + \epsilon \hat{v}^i_{k,\text{sep}} | \forall r^i_{k,\text{lstsq}} \in \mathcal{R}^i_{\text{lstsq}}\}$ where $\hat{v}^i_{k,\text{sep}}$ is $v^i_{k,\text{sep}}$ projected into model space using the fitted plane of $r^i_{k,\text{lstsq}}$'s local control mesh.

Next, a relaxation step adjusts the control mesh $\mathcal{R}^i_{\text{sep}}$ to separate control points that are close to merging. This step is necessary because both the signed distance function and the harmonic field shrink distances between control points non-uniformly as they move inward. The relaxation applies Laplacian smoothing to the control mesh to get a target positions $\tilde{p}$ for each control point p and move p to $(1-\beta)p+\beta\tilde{p}$ where β is a blending input parameter in $[0, 1]$, thus creating a better-spaced control mesh $\mathcal{R}^i$.

Finally, the trimmed surface layer $\mathcal{G}^i$ (corresponding to control meshes $\mathcal{R}^i$) is transformed to $\mathcal{B}^i$ that is topologically identical to $\mathcal{B}$ by replacing the B-splines with revised B-splines and trim basis functions with coefficients Υ^i. Again, points in Υ^i are obtained from curve fitting midpoint samples similar to equation 13, using knot vectors from corresponding $\mathcal{B}$ trimming curves. This glues the two sides of the trimming curve as in Section 4.4. The resulting control lattice in all layers $\mathcal{B}^i$ along with the same degree d as the surface and uniform open knot vector (0 on $\mathcal{B}$ and 1 on $\mathcal{B}^n$) a hybrid trivariate B-spline region $\mathscr{T}$ is defined.

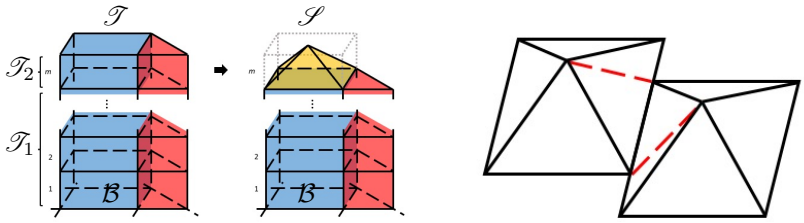
5.3 Construction of Pyramid Elements

The interior boundary surface of $\mathscr{T}$ must transition to the unstructured remaining interior volume. For all interior knot intervals (as defined in [26]), the problem is simplified to transition from one knot interval to unstructured volume. With the partitioning Ξ, the same problem on the trimmed (partial) knot intervals can be similarly described as transition from one sub-knot interval to unstructured volume, because except for the curvilinear triangular sub-knot intervals in Ξ, the sub-knot intervals of Ξ are all rectangular. By adding Bézier pyramid elements [5] on top of rectangular (sub-)knot intervals, it leaves the rest of the volume a Bézier triangle boundary, on which, the strategy in [41] can be applied to fill the volume with Bézier tetrahedra representation $\mathcal{U}$.

The benefit of having Bézier pyramid is that it has the boundary geometry of degree d Bézier triangles on each of the 4 faces, and a tensor product bi-degree d Bézier on its base surface. Thus, for $\mathcal{B}$ that is bi-degree d, the resulting Bézier triangles are degree d. Alternatively, if the interior boundary surface of $\mathscr{T}$ were to match the unstructured region, there would need to be two tetrahedra over each rectangular sub-knot interval surface, each one of degree $2d$ as in [41, 36].

In creating the pyramid interface, it is necessary that pyramids not intersect each other. Since $\mathscr{T}$ is known to have no self-intersections, it is subdivided into two trivariate splines in the inward direction to enforce this. The subdivision occurs at a user input parametric value $\theta < 1$ that can be greater or equal than the greatest interior knot, creating a trivariate spline $\mathscr{T}_2$ with one knot interval in the inward direction, that will become the pyramid interface layer. The remaining part of $\mathscr{T}$ is $\mathscr{T}_1$.

$\mathscr{T}_2$ undergoes Bézier extractions according to Ξ. The resulting elements are either Bézier wedges, corresponding to the curvilinear triangular sub-knot intervals of Ξ, or tensor-product Bézier volumes. The wedges in $\mathscr{T}_2$ are discarded, exposing the innermost curvilinear triangular sub-knot interval



(a) (b)

Figure 15: (a) $\mathscr{T}$ is converted to $\mathscr{S}$. Left: inward extensions of rectangular (blue) and triangular (red) sub-knot intervals. $\mathscr{T}$ is divided into $\mathscr{T}_1$ and $\mathscr{T}_2$. Right: The wedge in $\mathscr{T}_2$ are removed, and the tensor product Béziers in $\mathscr{T}_2$ converted to a pyramid (yellow). The semi-structured region $\mathscr{S}$ is the union of three types of elements. The inner boundary of $\mathscr{S}$ consists only of Bézier triangles. (b) T-junctions between adjacent pyramids are shown. The red dashed curves connecting the top points and the T-junctions subdivide each Bézier triangle into two.

boundary surfaces of $\mathscr{T}_1$ as inner bounding surfaces of the semi-structured region, illustrated in Figure 15a. Each of the tensor-product Bézier volumes in $\mathscr{T}_2$ is transformed to a Bézier pyramid whose coefficients are derived from those of its tensor-product Bézier volume. Consider each tensor-product Bézier volumes resulting from the hierarchical partitioning of $\mathscr{T}_2$. The tensor product Bézier volume is tri-d degree and has $d+1$ layers of control mesh. For the i^{th} $(0 \leq i \leq d)$ layer of the control mesh, counting from the base, the i^{th} recursion of the tensor product pyramid algorithm/de Casteljau algorithm [1] is applied for a value 1/2, in both ξ and η directions. The result is $(d+1-i) \times (d+1-i)$ control points on level i of the pyramid. The control points with corresponding pyramid basis functions [5] define a pyramid element.

The semi-structured region consists of $\mathscr{T}_1$ and the pyramids, where each pyramid is a C^0 match to the inner bounding surface of $\mathscr{T}_1$ over its rectangular sub-knot interval from Ξ, and whose coefficients are completely determined by coefficients from $\mathscr{T}_2$. Thus, the only DoF are the ones from $\mathcal{B}^i$ for $i \in \{0, 1, \ldots, n\}$. The region is denoted $\mathscr{S}$. The number of DoF of $\mathscr{S}$ is $(n+1)(|\mathcal{R}| + |\Upsilon|)$. $\mathcal{U}$ is an unstructured region with tetrahedral elements. Its boundary coincides with the inner surface of $\mathscr{S}$. A surface composed of Bézier triangles is formed by the inner boundary surface of $\mathscr{S}$. However, the surface contains T-junctions so it should not be tetrahedralized directly. To remove undesirable T-junctions between neighboring pyramids, the triangular de Casteljau subdivision algorithm is applied on the edge/face of Bézier triangles that have T-junctions to divide the triangle into two dependent triangle faces. Figure 15b illustrates this idea.

The unstructured region $\mathcal{U}$ is constructed following [41]. The resulting unstructured Bézier tetrahedral mesh is denoted $\mathcal{U}$. If needed, $\mathcal{U}$ can be transformed into unstructured Bézier hexahedra using reparameterization technique described in [37]. The HV-rep is complete as $\mathcal{V} = (\mathscr{S}, \mathcal{U})$.

6 Results

Results of applying the HB-rep are shown and analyzed in this section. Figure 16 shows a single trimmed surface with a sculptured "L" shaped hole. The inset of a typical trimmed knot interval is shown on its right. The model space isoparametric boundary of the trimmed knot interval is colored in blue. Sub-knot intervals within Ω are rendered with yellow boundaries. In particular, there are 3 *Curvilinear triangular sub-knot intervals* and 4 *rectangular sub-knot intervals* (indicated with red and yellow dots respectively, in their interior). The partitioning of the residual is also shown with black lines.

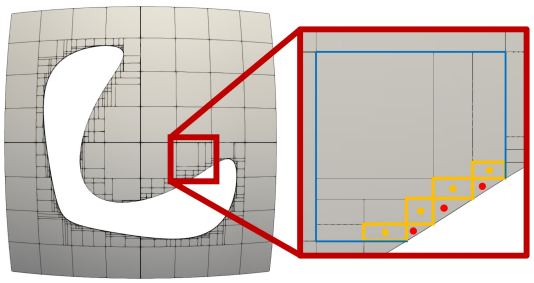


Figure 16: Ω, as well as the partition of the residual, is shown on a single trimmed surface. The inset of a typical trimmed knot interval is shown on the right.

Error in the HB-rep over Ω is now computed on the model in Figure 17. Figure 17a shows a coarse Ω region, with large Bézier triangle-based representations. The surface is flat so the geometry is exact over the triangles except for the error caused by the trimming curve itself. As shown in Figure 17b, the error seems to reach the white level of approximately 0.0024. After refining each surface in both parametric directions, the refined images of the triangular regions are shown in Figure 17c with the corresponding error shown in Figure 17d. Ω's diameter has shrunk significantly. The inset demonstrates the largest error in all the cases occurred on the trimming curve, that was the output of the CAD system. Hence the error in the trimming curve is a lower bound that the representation can achieve. The same behavior for trimming curve subdivision can also be observed in Figure 18.

More examples of HB-rep are shown in Figure 19. The trimmed models are constructed by using a Boolean operation feature of IRIT [9]. B-spline surfaces are trimmed by cubic trimming curves. Only Ω around the trimming curves is modified and is shown in red (with curvilinear triangular and rectangular sub-knot intervals in darker and lighter shades of red, respectively). Most rectangular sub-knot intervals only have one control point modified. C^0 continuity is maintained across the trimming curve. Figures 19a and 19b show the teapot model. Figure 19c used the straight sided method with a finer approximation to create the turbine.

Based on the HB-rep, the method for generating a volumetric representation is also tested on several models. Figure 1 shows the model constructed by subtracting one curved box shape from another with parameters: $n = 5$, $\omega = 0.75$, $\epsilon = 0.15$, $\beta = 0.5$ and $\theta = 0.75$. In the current unoptimized python software, the HB-rep takes 12.41 seconds to complete while the HV-rep takes 126.68 seconds. The mid-structure used in this example is generated using Q-MAT [17]. Near trimming curves from subtraction and intersection, tracing inward frequently causes surface layers to intersect in undesirable curves without the adjustment process. Figure 1 demonstrates the effectiveness of the adjustment phase to correctly seal the gap. On the right, a cutaway of the HV-rep $\mathcal{V}$ is shown near the trimming curve. $\mathscr{S}$ and $\mathcal{U}$ are shown in blue and gold.

Tetrapod in Figure 20a and 20b is constructed by applying union operation to 4 capsule shapes. 3 trimming curves emanate from 4 saddle points where 3 capsules touch. The Euclidean positions of those points are determined by going through all adjacent surfaces and using the average of their point evaluations. The model also illustrates the capability of our method to deal with branched shapes. Sweep based methods can have difficulties with such scenarios. Parameters used are: $n = 5$, $\omega = 0.5$, $\epsilon = 0.005$, $\beta = 0.5$ and $\theta = 0.75$.

Figures 20c, 20d and 20e demonstrate that our algorithm can handle transitions between different

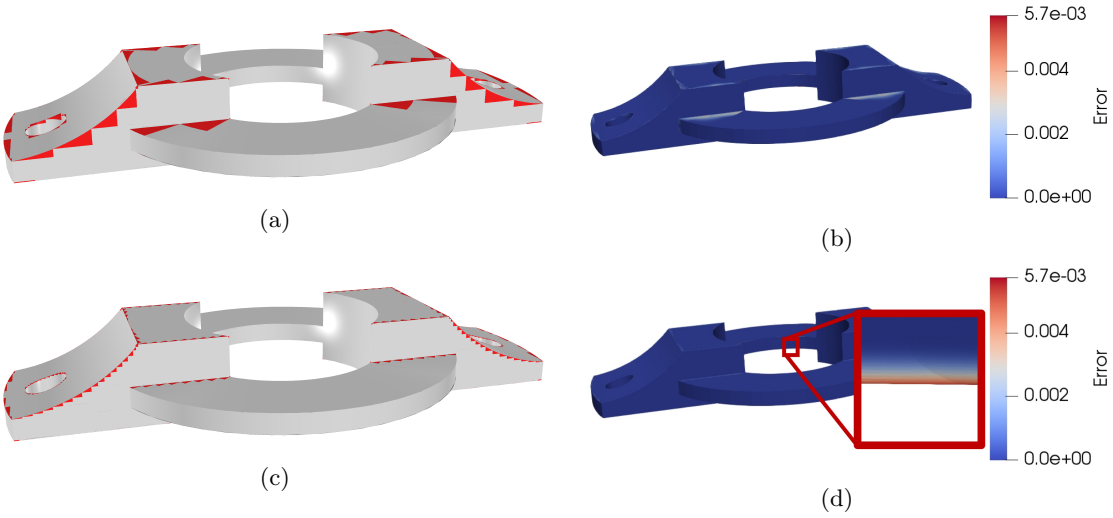


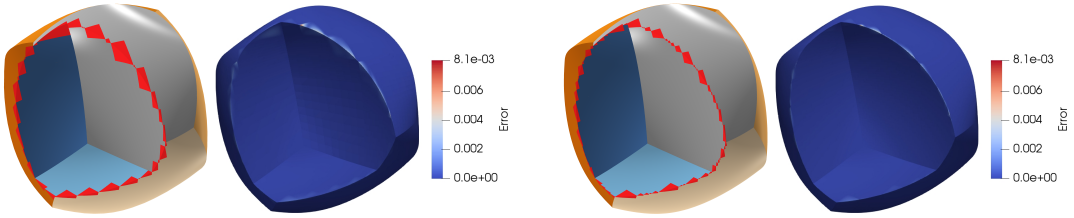
Figure 17: (a) a coarse HB-rep of a mechanical part is constructed with 8 Boolean operations. Curvilinear triangular sub-knot intervals are drawn in red. Its error from original B-spline B-rep is shown in (b). After refining the surface representation in both parametric directions, the HB-rep and smaller triangular based surfaces are (c), with the corresponding error in (d). For both cases, largest errors occur along the trimming curve.

geometries within one connected component. Parameters are $n = 5$, $\omega = 0.6$, $\epsilon = 0$, $\beta = 0.85$ and $\theta = 0.6$.

As is discussed in Section 5.3, using Bézier pyramids as an interface layer to the unstructured region, instead of directly using the higher degree tetrahedra needed to match the tensor product region, reduces the number of DoF needed. We compare DoF on only the unstructured region $\mathcal{U}$ of the tetrapod that has $10,928$ elements, $25,232$ faces, $17,681$ edges, and $3,378$ points. Since the tetrapod surfaces are bi-cubic, $\mathcal{U}$ would be degree 6 [41], where the independent DoF for each element, face, edge, and point are 20, 10, 5 and 1, respectively, resulting in $562,663$ DoF. However, for degree 3 Bézier tetrahedra made possible by the pyramid interface, the DoF for each element, face, edge, point are 0, 1, 2 and 1, respectively. The total DoF is reduced to $63,972$, a significant 89% reduction.

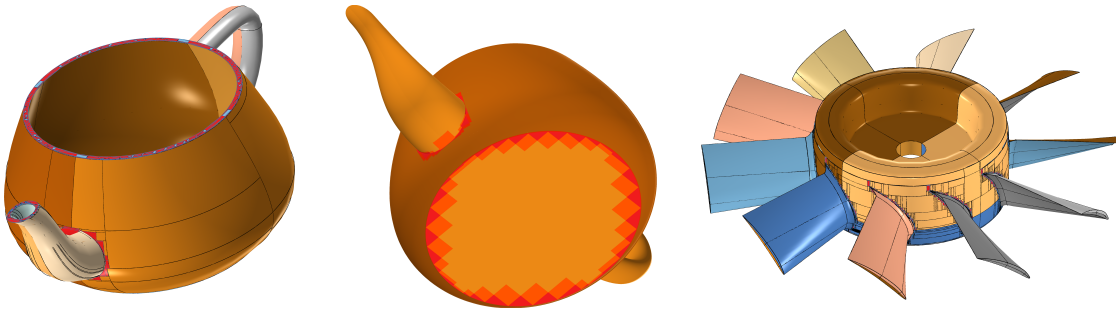
7 Conclusion

We introduce a novel hybrid boundary representation to create watertight trimmed CAD models. Based on trimmed B-spline B-reps, we modify the vicinity of trimming curves and seal the gap between trimming curves. In order to achieve that, we introduce two types of new suitably modified basis functions that we refer to as revised and trim functions. The new basis functions utilize both tensor product and triangular Bézier, which commonly exist in commercial CAD systems. Their corresponding coefficients are linear combinations of the original B-spline coefficients so they are easy to compute. Unlike many methods that rely on reparameterizing and reapproximating the surfaces, methods herein preserve nearly all of the B-rep. They respect both geometry and parameterization of original B-spline, except for a narrow region arbitrarily close to the trimming curve. When the original parameterization is kept over larger triangles, the resulting representation in Ω has the



(a) (b) (c) (d)

Figure 18: A model made by subtracting one curved box from another. Different levels of refinement are shown in (a) and (c) with errors shown in (b) and (d), respectively.

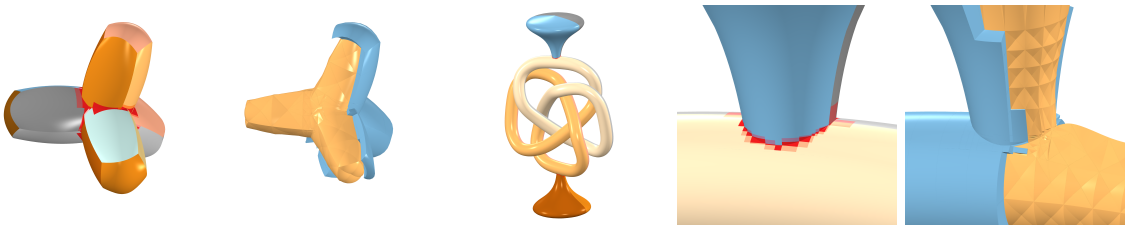


(a) (b) (c)

Figure 19: The results of the HB-reps are shown. Each trimmed surface is assigned a color. Only Ω in red is different from the original. (a) shows only a small banded region at the base of the spout. Also, on the rim of both the body and the spout, our algorithm correctly generate a rectilinear approximation on these narrow regions. In (b) trimming curves and surfaces had few knots. Although Ω is exact for the flat bottom, the rectilinear partitioning was refined to ensure that Ω is tight to the trimming curve, and the original surface representation is kept in most of the surface.

drawback that it is non-polynomial, even though computationally it is simple to use and useful when integrating over trimmed regions.

Based on the HB-rep, we introduce volume completion which was not possible because of the lack of a watertight representation. The resulting HV-rep has both a semi-structured trivariate B-spline region and an unstructured Bézier tetrahedral region near the core of the volume. The representation preserves the original HB-rep and directly extends that into a semi-structured trivariate region without needing further approximation. The extension of the revised and trim basis functions in that region allow explicit coupling across trims for both geometry and material attribute representations. High order analysis such as IGA can be applied to the HV-rep. Without the need of being remodeled, the design-and-simulation loop for trimmed geometry is further streamlined. By using Bézier pyramids, Bézier tetrahedra with significantly lower DoFs are used in the unstructured region. The mid-structure choice remains a crucial factor of the methodology. We would like to do further analysis on different choices of mid-structures to find out what are the preferred characteristic for a trimmed model's mid-structure. The explicit coupling across the trimming curves can theoretically allow some shape modifications directly on the HV-rep, without requiring recomputation. We plan to investigate such operations. For now, parameters are specified for each model. We plan to



(a) (b) (c) (d) (e)

Figure 20: (a) HB-rep of a tetrapod. The corresponding HV-rep is shown in (b). (c) The HB-rep of union of a knot and two ends is shown. (d) and (e) shows enlarged views of the region around an intersection curve of the model.

automate this process so that the process depends less on human intervention.

8 Acknowledgements

This research was supported in part with funding from the Defense Advanced Research Projects Agency (DARPA), under contract HR0011-17-2-0028. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] M. Ainsworth, *Pyramid algorithms for bernstein–bézier finite elements of high, nonuniform order in any dimension*, SIAM Journal on Scientific Computing, 36 (2014), pp. A543–A569.

[2] H. A. Akhras, T. Elguedj, A. Gravouil, and M. Rochette, *Isogeometric analysis-suitable trivariate NURBS models from standard B-rep models*, Computer Methods in Applied Mechanics and Engineering, 307 (2016), pp. 256 – 274.

[3] ——, *Towards an automatic isogeometric analysis suitable trivariate models generation - application to geometric parametric analysis*, Computer Methods in Applied Mechanics and Engineering, 316 (2017), pp. 623 – 645.

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

[5] J. Chan and T. Warburton, *A short note on a Bernstein–Bézier basis for the pyramid*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2162–A2172.

[6] E. Cohen, R. F. Riesenfeld, and G. Elber, *Geometric modeling with splines : an introduction*, A. K. Peters, Natick (Mass.), 2001.

[7] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, Wiley Publishing, 1st ed., 2009.

[8] T. Dokken, T. Lyche, and K. F. Pettersen, *Polynomial splines over locally refined box-partitions*, Computer Aided Geometric Design, 30 (2013), pp. 331 – 356.

[9] G. Elber, *Irit modeling environment*.

[10] G. Elber and M.-S. Kim, *Modeling by composition*, Computer-Aided Design, 46 (2014), pp. 200 – 204. 2013 SIAM Conference on Geometric and Physical Modeling.

[11] L. Engvall and J. A. Evans, *Isogeometric unstructured tetrahedral and mixed-element Bernstein-Bézier discretizations*, Computer Methods in Applied Mechanics and Engineering, 319 (2017), pp. 83 – 123.

[12] X. Gao, T. Martin, S. Deng, E. Cohen, Z. Deng, and G. Chen, *Structured volume decomposition via generalized sweeping*, Visualization and Computer Graphics, IEEE Transactions on, PP (2015), pp. 1–1.

[13] Y. Guo, M. Ruess, and D. Schillinger, *A parameter-free variational coupling approach for trimmed isogeometric thin shells*, Computational Mechanics, 59 (2017), pp. 693–715.

[14] A. Jacobson, D. Panozzo, et al., *libigl: A simple C++ geometry processing library*, 2018. http://libigl.github.io/libigl/.

[15] H.-J. Kim, Y.-D. Seo, and S.-K. Youn, *Isogeometric analysis with trimming technique for problems of arbitrary complex topology*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 2796 – 2812.

[16] V. Kumar, D. Burns, D. Dutta, and C. Hoffmann, *A framework for object modeling*, Computer-Aided Design, 31 (1999), pp. 541 – 556.

[17] P. Li, B. Wang, F. Sun, X. Guo, C. Zhang, and W. Wang, *Q-MAT: computing medial axis transform by quadratic error minimization*, ACM Trans. Graph., 35 (2015), pp. 8:1–8:16.

[18] L. Liu, Y. Zhang, T. J. R. Hughes, M. A. Scott, and T. W. Sederberg, *Volumetric T-spline construction using Boolean operations*, Engineering with Computers, 30 (2014), pp. 425–439.

[19] T. Martin and E. Cohen, *Volumetric parameterization of complex objects by respecting multiple materials*, Computers & Graphics, 34 (2010), pp. 187 – 197. Shape Modelling International (SMI) Conference 2010.

[20] T. Martin, E. Cohen, and R. Kirby, *Volumetric parameterization and trivariate b-spline fitting using harmonic functions*, Computer Aided Geometric Design, 26 (2009), pp. 648 – 664. Solid and Physical Modeling 2008ACM Symposium on Solid and Physical Modeling and Applications.

[21] T. Martin, E. Cohen, and R. M. Kirby, *Mixed-element volume completion from NURBS surfaces*, Computers & Graphics, 36 (2012), pp. 548 – 554. Shape Modeling International (SMI) Conference 2012.

[22] B. Marussig and T. J. R. Hughes, *A review of trimming in isogeometric analysis: Challenges, data exchange and simulation aspects*, Archives of Computational Methods in Engineering, 25 (2018), pp. 1059–1127.

[23] F. Massarwi and G. Elber, *A b-spline based framework for volumetric object modeling*, Computer-Aided Design, 78 (2016), pp. 36 – 47. {SPM} 2016.

[24] B. Miklos, J. Giesen, and M. Pauly, *Discrete scale axis representations for 3d geometry*, in ACM SIGGRAPH 2010 Papers, SIGGRAPH '10, New York, NY, USA, 2010, ACM, pp. 101:1–101:10.

[25] J. O'Rourke and G. Tewari, *The structure of optimal partitions of orthogonal polygons into fat rectangles*, Computational Geometry, 28 (2004), pp. 49 – 71. 14th Canadian Conference on Computational Geometry - CCCG02.

[26] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank, *Weakly enforced essential boundary conditions for nurbs-embedded and trimmed nurbs geometries on the basis of the finite cell method*, International Journal for Numerical Methods in Engineering, 95, pp. 811–846.

[27] M. Ruess, D. Schillinger, A. I. Özcan, and E. Rank, *Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries*, Computer Methods in Applied Mechanics and Engineering, 269 (2014), pp. 46 – 71.

[28] T. W. Sederberg, G. T. Finnigan, X. Li, H. Lin, and H. Ipson, *Watertight trimmed NURBS*, in ACM SIGGRAPH 2008 Papers, SIGGRAPH '08, New York, NY, USA, 2008, ACM, pp. 79:1–79:8.

[29] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, *T-splines and t-nurccs*, ACM Trans. Graph., 22 (2003), pp. 477–484.

[30] J. Shen, J. Kosinka, M. Sabin, and N. Dodgson, *Converting a CAD model into a non-uniform subdivision surface*, Computer Aided Geometric Design, 48 (2016), pp. 17 – 35.

[31] J. Shen, J. Kosinka, M. A. Sabin, and N. A. Dodgson, *Conversion of trimmed NURBS surfaces to Catmull–Clark subdivision surfaces*, Computer Aided Geometric Design, 31 (2014), pp. 486 – 498. Recent Trends in Theoretical and Applied Geometry.

[32] H. Si, *Tetgen, a delaunay-based quality tetrahedral mesh generator*, ACM Trans. Math. Softw., 41 (2015), pp. 11:1–11:36.

[33] S. Soghrati and R. A. Merel, *NURBS enhanced HIFEM: A fully mesh-independent method with zero geometric discretization error*, Finite Elements in Analysis and Design, 120 (2016), pp. 68 – 79.

[34] Q. Song and J. Wang, *Generating parametric blending surfaces based on partial reparameterization of base surfaces*, Computer-Aided Design, 39 (2007), pp. 953 – 963.

[35] O. Sorkine, *Differential representations for mesh processing*, Computer Graphics Forum, 25 (2006), pp. 789–807.

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

[37] G. Xu, Y. Jin, Z. Xiao, Q. Wu, B. Mourrain, and T. Rabczuk, *Exact conversion from Bézier tetrahedra to Bézier hexahedra*, Computer Aided Geometric Design, 62 (2018), pp. 154 – 165.

[38] G. Xu, M. Li, B. Mourrain, T. Rabczuk, J. Xu, and S. P. Bordas, *Constructing iga-suitable planar parameterization from complex cad boundary by domain partition and global/local optimization*, Computer Methods in Applied Mechanics and Engineering, 328 (2018), pp. 175 – 200.

[39] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo, *Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications*, Computer-Aided Design, (2013), pp. 395–404.

[40] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo, *Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method*, Journal of Computational Physics, 252 (2013), pp. 275 – 289.

[41] S. Zeng and E. Cohen, *Hybrid volume completion with higher-order Bézier elements*, Comput. Aided Geom. Des., 35 (2015), pp. 180–191.

[42] Y. Zhang, W. Wang, and T. J. Hughes, *Solid T-spline construction from boundary representations for genus-zero geometry*, Computer Methods in Applied Mechanics and Engineering, 249–252 (2012), pp. 185 – 197. Higher Order Finite Element and Isogeometric Methods.