Generalized Bézier volumes over simple convex polyhedra

Kaikai Qin, Yajuan Li, Chongyang Deng *

School of Science, Hangzhou Dianzi University, China

ARTICLE INFO

ABSTRACT

In recent years, there has been growing interest in the representation of volumes within the field of geometric modeling (GM). While *polygonal* patches for surface modeling have been extensively studied, there has been little focus on the representation of *polyhedral* volumes. Inspired by the polygonal representation of the *Generalized Bézier (GB)* patch proposed by Várady et al. (2016), this paper introduces a novel method for polyhedral volumetric modeling called the Generalized Bézier (GB) volume.

GB volumes are defined over *simple convex polyhedra* using *generalized barycentric coordinates* (GBCs), with the control nets which are a direct generalization of those of tensor-product Bézier volumes. GB volumes can be smoothly connected to adjacent tensor-product Bézier or GB volumes with G^1 or G^2 continuity. Besides, when the parametric polyhedron becomes a prism, the GB volume also degenerates into a tensor-product form. We provide some practical examples to demonstrate the advantages of GB volumes. Suggestions for future work are also discussed.

1. Introduction

In geometric modeling, a 3D object is usually represented as a collection of surfaces (e.g., trimmed tensor-product surfaces) that bound the object, that is, boundary representation (B-rep) (Cohen et al., 2001). However, accompanied by the development of Isogeometric Analysis (IGA) (Hughes et al., 2005) and additive manufacturing (AM) (Paolini et al., 2019) in recent years, there exists a need for a full volumetric representation (V-rep) of 3D objects. This implies that not only the boundaries but also the interior of the object should be well-defined. In surface modeling, only the continuity between two adjacent surfaces (along a common curve) needs to be considered; but in volumetric modeling, the constraint of continuity between two adjacent volumes (along a common surface) should be satisfied.

Tensor-product volumes (e.g., tensor-product Bézier volumes (Bézier, 1978; Lasser, 1985)) are widely used in volumetric modeling. A tensor-product volume is usually defined over a rectangular prism and its topology is typically that of a hexahedron. Generally speaking, a tensor-product volume has six boundary surfaces and each of these is a quadrilateral patch.

Therefore, it seems that tensor-product volumes may not always be suitable for any case. In surface modeling, irregular regions (i.e., regions that are not four-sided) often occur, and multi-sided surfaces are required to fill holes (Goldman, 2004; Peters, 2019). Until now, the construction of multi-sided patches (Várady et al., 2024) has been an active topic in the field of computer aided geometric design (CAGD). Unfortunately, the irregular regions seem to be more complicated to handle in volumetric modeling. There may be a polyhedral region that has an arbitrary number of boundary surfaces, and not all of these surfaces may be quadrilateral patches, for example, the irregular regions of unstructured hexahedral meshes.

---

* Corresponding author.
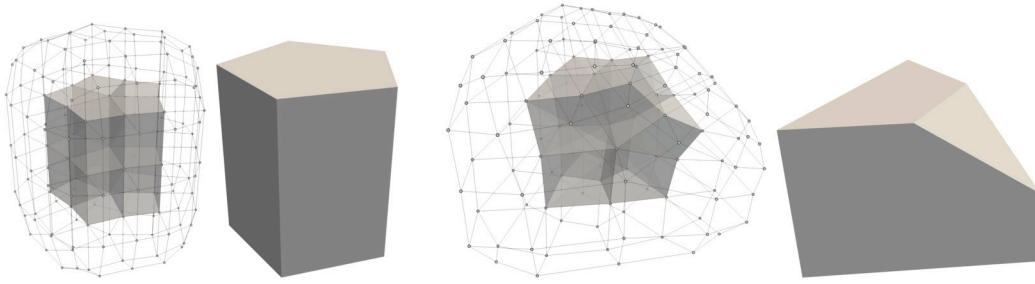*E-mail addresses:* qin-kaikai@hdu.edu.cn (K. Qin), liyajuan@hdu.edu.cn (Y. Li), dcy@hdu.edu.cn (C. Deng).

Fig. 1. Internal irregular regions on unstructured hexahedral meshes are homeomorphic to simple convex polyhedra.

As a discrete volumetric representation, unstructured hexahedral meshes are commonly used in finite element analysis (FEA) and other numerical simulations. Also, subdivision schemes have been extended to hexahedral meshes as a simple and efficient technique for volumetric modeling (e.g., Catmull–Clark volumetric subdivision (MacCracken and Joy, 1996)). Recently, researchers have been utilizing unstructured hexahedral meshes as input control meshes for IGA (Wei et al., 2018). Indeed, irregularities (i.e., extraordinary edges/vertices) of unstructured hexahedral meshes seem to be inevitable and very difficult to deal with. In addition, the internal extraordinary edges are dual to prismatic volumes, while the very extraordinary vertices in the interior are dual to polyhedral volumes (Reif and Sabin, 2019), see Fig. 1.

It can be found that the topology of internal irregular regions on unstructured hexahedral meshes are **simple convex polyhedra** *in which all vertices have three incident faces*.[1] Therefore, we are particularly interested in the construction of simple convex polyhedral volumes in this paper.

Although a considerable amount of literature on volumetric modeling has been published, few have focused on polyhedral volumes. Lasser (1987) defined a *pentahedral Bézier (PB)* volume which is a tensor-product of triangular Bernstein-Bézier surfaces and homeomorphic to a *triangular prism*. Randrianarivony (2011) generalized the conventional transfinite interpolation (Coons, 1967; Charrot and Gregory, 1984) to general convex polytopes via Gregory corner blending (Gregory, 1985). Randrianarivony's approach will generate a simple convex polyhedral volume that interpolates given boundary surfaces (G^0) in 3D case. Haberleitner et al. (2019) proposed a method that divides an irregular region called MS³ (midpoint subdivision suitable solids) which is homeomorphic to a simple convex polyhedron into topological cuboids by midpoint subdivision. This method bears some resemblance to the macro-patch construction, which divides an *n*-sided region into *n* quadrilaterals. These topological cuboids are connected with only G^0 continuity. Peters (2020) constructed refinable globally C^1 continuous splines on unstructured hexahedral meshes with singular corner parameterization and the first-order derivatives will vanish at irregularities. Besides, we should mention that the FEA community has also been researching the construction of appropriate shape functions on polyhedral elements, as demonstrated by recent works such as Schneider et al. (2019) and Bunge et al. (2022).

In surface modeling, a multi-sided patch is required to fulfill smoothness constraints both along the boundaries and in the interior. For example, a multi-sided Bézier patch is usually requested possessing tensor-product Bézier surface boundaries (Várady et al., 2016; Goldman, 2004; Qin et al., 2023). That is, a multi-sided Bézier patch should behave like a tensor-product Bézier patch, both in positional and differential senses, along each boundary curve. Thus its boundary curves are Bézier curves and the corresponding cross-derivatives vector-valued Bézier curves. From this perspective, a polyhedral Bézier volume should

1. behave like a tensor-product Bézier volume along each quadrilateral boundary surface,
2. and a "tensor-product multi-sided Bézier volume" along each non-four-sided boundary surface.

A "tensor-product multi-sided Bézier volume" exactly implies a prismatic volume which is the tensor-product of multi-sided Bézier patches (similar to the PB volume). Thus the boundary surfaces of a polyhedral Bézier volume are tensor-product or multi-sided Bézier surfaces and the corresponding cross-derivatives vector-valued tensor-product or multi-sided Bézier surfaces.

Recently, Várady et al. (2016) proposed a multi-sided Bézier patch — the generalized Bézier (GB) patch which is defined using generalized barycentric coordinates (GBCs) (Floater, 2015) over planar polygons. Inspired by this multi-sided surfacing scheme, we propose a novel volumetric modeling method, the generalized Bézier (GB) volume. A GB volume is a control-point-based volume defined over a simple convex polyhedron and represents a natural extension of tensor-product Bézier volumes. The main advantages of this method are summarized as follows.

- A GB volume behaves as a tensor-product Bézier volume along each quadrilateral boundary surface.
- A GB volume behaves as a tensor-product GB volume (see Section 3) along each multi-sided boundary surface.
- A GB volume can be easily connected to adjacent tensor-product Bézier or another (tensor-product) GB volume with high order geometric continuity (G^1 or G^2).
- A GB volume will reproduce a tensor-product GB volume while the domain polyhedron becomes a prism.

---

[1] Floater (2015) used the term 'simple convex polyhedra' to refer to such polyhedra, and we have adopted the same terminology.
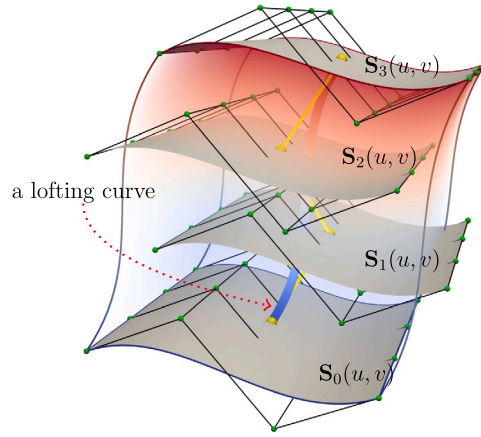
Fig. 2. A lofting curve inside a tri-cubic tensor-product Bézier volume, controlled by the yellow control points lying on the different layer of Bézier surfaces. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

In addition, our contribution goes beyond just proposing a polyhedral volumetric modeling method. We also provide an ***idea*** for extending the concept from polygonal patches to polyhedral volumes. Thus, the existing polygonal patch schemes, such as multi-sided transfinite interpolation surfaces (Várady et al., 2011), might be also extended to volumes by leveraging our approach. We also present a polyhedral Bézier extraction algorithm that enables the construction of a globally smooth volume from any input hexmesh with arbitrary irregularities or from input simple-convex-polyhedral meshes.

The remainder of this paper is organized as follows. In Section 2 we provide some preliminaries on tensor-product Bézier volumes and GB patches. Then we define tensor-product GB volumes in Section 3 before constructing our GB volumes in Section 4. Some examples are shown in Section 5. We discuss the limitations of GB volumes and further work in Section 6 and finally conclude this paper in Section 7.

2. Preliminaries

2.1. Tensor-product Bézier volume

Here we consider a degree d tensor-product Bézier volume $\mathbf{V}(u, v, w)$ defined with ***Bézier*** points $\{\mathbf{C}_{i,j,k}\}_{i,j,k=0}^{d}$.

$$\mathbf{V}(u,v,w) = \sum_{i=0}^{d} \sum_{j=0}^{d} \sum_{k=0}^{d} \mathbf{C}_{i,j,k} \cdot B_{i,j,k}^{d}(u,v,w),$$

where $(u, v, w) \in [0, 1]^3$ and $B_{i,j,k}^{d}(u, v, w) = B_{i}^{d}(u) \cdot B_{j}^{d}(v) \cdot B_{k}^{d}(w)$ is the trivariate Bernstein basis function.

It is well-known that a tensor-product volume can be thought of as being swept out by a moving and deforming isoparametric surface (Hoschek et al., 1993). Generally speaking, the volume $\mathbf{V}(u, v, w)$ is the locus of a Bézier surface that is moving through space and thereby changing its shape. Each point on the surface moves through space along a lofting curve. A lofting curve is a degree d Bézier curve determined by a set of control points. Each control point moves through space on a Bézier surface controlled by the corresponding layer. Thus, $\mathbf{V}(u, v, w)$ can be represented as

$$\mathbf{V}(u,v,w) = \sum_{k=0}^{d} \mathbf{S}_{k}(u,v) \cdot B_{k}^{d}(w),$$

where $\mathbf{S}_{k}(u, v) = \sum_{i=0}^{d} \sum_{j=0}^{d} \mathbf{C}_{i,j,k} \cdot B_{i,j}^{d}(u, v)$ is a degree d tensor-product Bézier surface determined by the k-th layer of control points. Fig. 2 shows a lofting curve inside a tri-cubic tensor-product Bézier volume.

2.2. Generalized Bézier (GB) patch

The generalized Bézier (GB) patch is a multi-sided Bézier patch with a simple control structure (Várady et al., 2016). An n-sided degree d GB patch is defined over a convex n-sided polygon $\mathbf{\Omega}$ with n vertices $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$. Its control net is a multi-sided generalization of those of the tensor-product Bézier patches. The topology of the control net is determined by n and d.

An n-sided GB patch is a composition of n side-Bézier ribbons. Each side-Bézier ribbon consists of half-Bézier control points multiplied by the corresponding weighted Bernstein polynomials. The weighted Bernstein polynomials are determined by local parameters which are computed using Wachspress coordinates (Wachspress, 1975).

Local control net of a side-Bézier ribbon. Várady et al. regarded an n-sided Bézier control net as n overlapping half-Bézier control nets which belong to n side-Bézier ribbons. A local control net, denoted by $\{\mathbf{C}_{j,k}^{i}\}_{j,k=0}^{d,l}$, has $(l + 1) \cdot (d + 1)$ control points
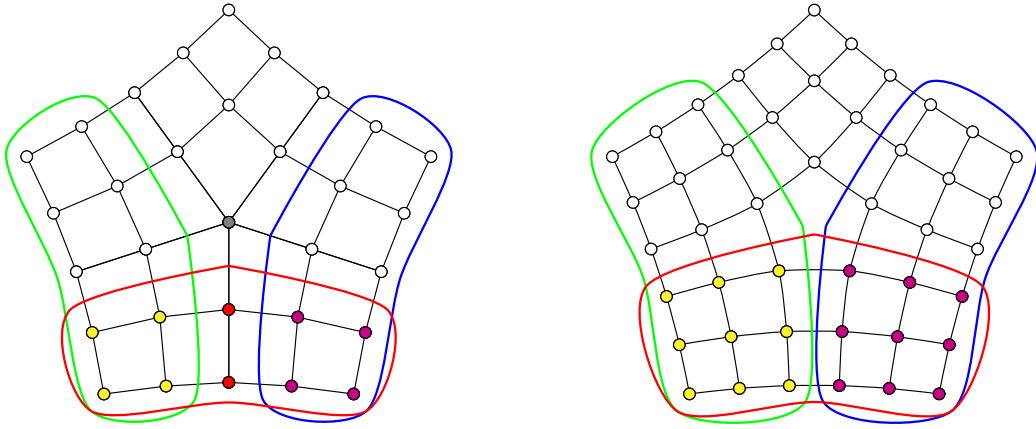
Fig. 3. Local control nets. Left: a 5-sided degree 4 control net. Right: a 5-sided degree 5 control net. The $(i-1)$-th, i-th and $(i+1)$-th local control nets fall into green, red and blue frames, with respect. The control points of i-th local control net are colored differently according to their weights (○: α_i; ◉: β_i; ●: 1; and ◉: $B_{l+1,l+1}$).
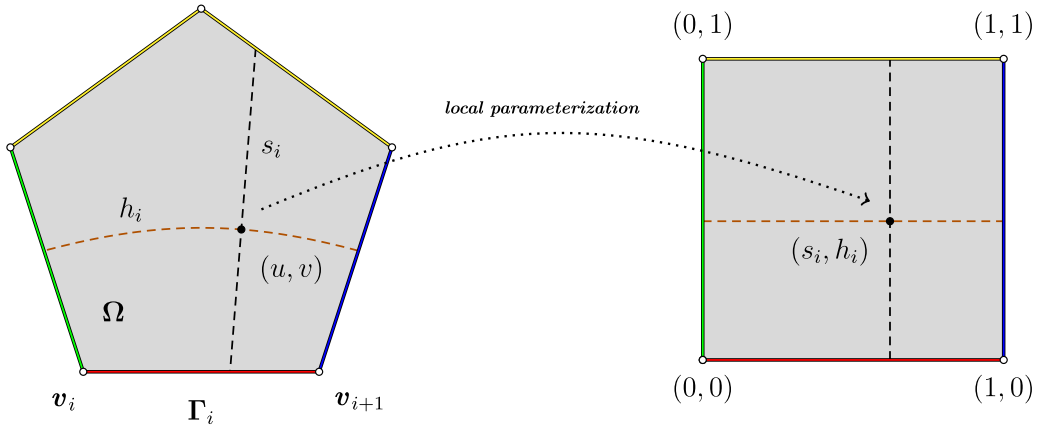

Fig. 4. Local parameterization. A pentagonal domain (left) is reparameterized into a unit square (right).

where $l = \lfloor (d-1)/2 \rfloor$. Two adjacent local control nets share $(l+1) \cdot (l+1)$ control points around the corner. Remark, when degree d is even, there exists a central control point $\mathbf{C}_{l+1,l+1}$ besides local control nets (see Fig. 3).

Local parameterization. For each side $\mathbf{\Gamma}_i = \{(1-\lambda)\mathbf{v}_i + \lambda\mathbf{v}_{i+1} \mid 0 \le \lambda \le 1\}$ of the polygon $\mathbf{\Omega}$, two local parameters are introduced as the side parameter s_i and the distance parameter h_i. For an arbitrary point $(u,v) \in \mathbf{\Omega}$, $\{\lambda_i = \lambda_i(u,v)\}_{i=1}^n$ are its Wachspress coordinates with respect to n vertices $\{\mathbf{v}_i\}_{i=1}^n$. Then, the two local parameters are defined as

$$s_i = \frac{\lambda_{i+1}}{\lambda_i + \lambda_{i+1}}, \quad h_i = 1 - (\lambda_i + \lambda_{i+1}). \tag{1}$$

By this local parameterization, the whole polygonal domain $\mathbf{\Omega}$ is reparameterized into a rectangular domain $[0,1]^2$ along side $\mathbf{\Gamma}_i$ (see Fig. 4).

Weighted Bernstein polynomials. A local control net $\{\mathbf{C}_{j,k}^i\}_{j,k=0}^{d,l}$ is associated with side $\mathbf{\Gamma}_i$ by weighted Bernstein polynomials $\{\mu_{j,k}^i B_{j,k}^d(s_i, h_i)\}_{j,k=0}^{d,l}$. The weighting function $\mu_{j,k}^i$ is defined as

$$\mu_{j,k}^i = \begin{cases} \alpha_i = \dfrac{h_{i-1}^3}{h_{i-1}^3 + h_i^3}, & 0 \le j \le l, \\[2mm] \beta_i = \dfrac{h_{i+1}^3}{h_{i+1}^3 + h_i^3}, & d - l \le j \le d, \\[2mm] 1, & otherwise. \end{cases} \tag{2}$$

As for the central control point $\mathbf{C}_{l+1,l+1}$, the blending function is $B_{l+1,l+1}(u,v) = \frac{1}{n} \sum_{i=1}^n B_{l+1,l+1}^d(s_i, h_i)$ (see Fig. 3).
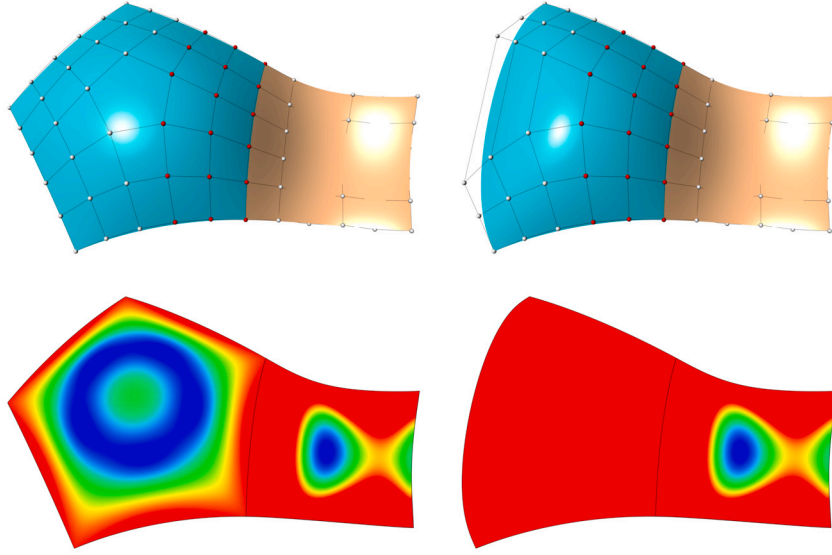
Fig. 5. A 5-sided degree 5 GB patch (left, colored in cyan) behaves along a boundary as a degree 5 tensor-product Bézier patch (right, colored in cyan) having the same 3 rows of control points (colored in red). Both the GB and tensor-product Bézier patches (colored in cyan) are smoothly connected to the same ordinary Bézier patch (colored in fleshcolor). Top: Shading and control nets. Bottom: Mean curvature (from red via green to blue).

Side-Bézier ribbons. Now the i-th side-Bézier ribbon $\mathbf{R}^i(u,v)$ can be obtained

$$\mathbf{R}^i(u,v) = \sum_{j=0}^{d} \sum_{k=0}^{l} \mathbf{C}_{j,k}^i \cdot \mu_{j,k}^i B_{j,k}^d(s_i, h_i). \tag{3}$$

Each side-Bézier ribbon is treated as a side interpolator and only the i-th ribbon affects the i-th side $\mathbf{\Gamma}_i$.

The patch equation. Then the final n-sided degree d GB patch $\mathbf{S}(u,v)$ is defined as:

$$\mathbf{S}(u,v) = \frac{1}{B_\Sigma(u,v)} \Big[\sum_{i=1}^{n} \mathbf{R}^i(u,v) + \mathbf{C}_{l+1,l+1} \cdot B_{l+1,l+1}(u,v) \Big], \tag{4}$$

where

$$B_\Sigma(u,v) = \sum_{i=1}^{n} \sum_{j=0}^{d} \sum_{k=0}^{l} \mu_{j,k}^i B_{j,k}^d(s_i, h_i) + B_{l+1,l+1}(u,v).$$

Note that, the last term $\mathbf{C}_{l+1,l+1} \cdot B_{l+1,l+1}(u,v)$ exists only if the degree is even.

A GB patch possesses tensor-product Bézier boundaries. That is, along the boundary, a GB patch behaves as an ordinary tensor-product Bézier patch which has the same three rows (assuming degree $d \geq 5$) of control points (see Fig. 5).[2] For example, on side $\mathbf{\Gamma}_i$, we have

$$\mathbf{S}(u,v)\Big|_{\mathbf{\Gamma}_i} = \mathbf{R}^i(u,v)\Big|_{\mathbf{\Gamma}_i} = \sum_{j=0}^{d} \mathbf{C}_{j,0}^i \cdot B_{j,0}^d(s_i, 0),$$

$$\partial(\mathbf{S}(u,v))\Big|_{\mathbf{\Gamma}_i} = \partial(\mathbf{R}^i(u,v))\Big|_{\mathbf{\Gamma}_i} = \sum_{j=0}^{d} \sum_{k=0}^{1} \mathbf{C}_{j,k}^i \cdot \partial(B_{j,k}^d(s_i, 0)), \tag{5}$$

$$\partial^2(\mathbf{S}(u,v))\Big|_{\mathbf{\Gamma}_i} = \partial^2(\mathbf{R}^i(u,v))\Big|_{\mathbf{\Gamma}_i} = \sum_{j=0}^{d} \sum_{k=0}^{2} \mathbf{C}_{j,k}^i \cdot \partial^2(B_{j,k}^d(s_i, 0)),$$

where the notation ∂ indicates directional derivatives in an arbitrary direction in the domain.

3. Tensor-product GB volume

Before delving into GB volumes, it is advisable to first explore the tensor-product GB volumes.

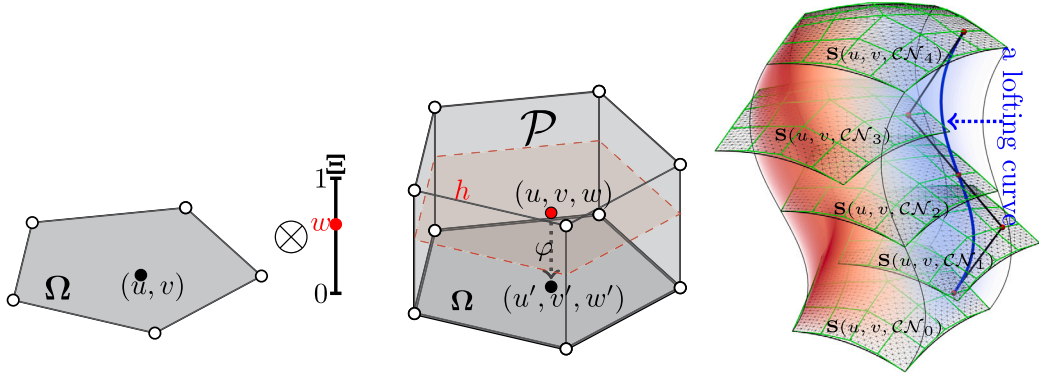[2] Two rows of control points (with degree $d \geq 3$) for G^1 continuity.

Fig. 6. Left: Parametric space spanning by $\boldsymbol{\Omega}$ and $\boldsymbol{\Xi}$. Middle: The polyhedral domain $\mathcal{P}$. Right: A 5-prismatic degree 4 tensor-product GB volume; and a lofting curve inside the tensor-product GB volume, controlled by the red control points lying on the different layer of GB patches.

3.1. Definition of a tensor-product GB volume

By combining the definitions of tensor-product Bézier volumes and GB patches, we can now define a tensor-product GB volume. An *n-sided polygonal prismatic* (*n-prismatic* for short) degree d tensor-product GB volume $\mathbf{V}(u,v,w)$ is defined by

$$\mathbf{V}(u,v,w) = \sum_{m=0}^{d} \mathbf{S}(u,v,\mathcal{CN}_m) \cdot B_m^d(w), \tag{6}$$

where $\mathbf{S}(u,v,\mathcal{CN}_m)$ is an *n*-sided degree d GB patch defined over an *n*-sided polygon $\boldsymbol{\Omega}$, with the *m*-th layer control net $\mathcal{CN}_m$, and $w \in \boldsymbol{\Xi} := [0,1]$ (see Fig. 6).

$\mathcal{CN}_m$ is defined as

$$\mathcal{CN}_m = \bigcup_{i=1}^{n} \{\mathbf{C}_{j,k,m}^i\}_{j,k=0}^{d,l} \cup \mathbf{C}_{l+1,l+1,m},$$

where an additional subscript m is used to distinct different layers. Please note, as with the aforementioned and the following text, the central control point $\mathbf{C}_{l+1,l+1,m}$ only exists when degree d is even. We use the notation Δ^r to denote the *iterated forward difference operator*

$$\Delta^0 \mathcal{CN}_m = \mathcal{CN}_m,$$
$$\Delta^r \mathcal{CN}_m = \Delta^{r-1} \mathcal{CN}_{m+1} - \Delta^{r-1} \mathcal{CN}_m,$$

where

$$\mathcal{CN}_{m_1} - \mathcal{CN}_{m_2} = \bigcup_{i=1}^{n} \{\mathbf{C}_{j,k,m_1}^i - \mathbf{C}_{j,k,m_2}^i\}_{j,k=0}^{d,l} \cup \{\mathbf{C}_{l+1,l+1,m_1} - \mathbf{C}_{l+1,l+1,m_2}\}.$$

3.2. Reparameterization using GBCs

The parametric space of the tensor-product GB volume can be regarded as an *n*-sided prism $\mathcal{P}$ spanning by $\boldsymbol{\Omega}$ and $\boldsymbol{\Xi}$, that is

$$\mathcal{P} = \boldsymbol{\Omega} \otimes \boldsymbol{\Xi}.$$

Due to the parametric domain $\mathcal{P}$ is exactly a polyhedron, we would prefer to represent a tensor-product GB volume $\mathbf{V}(u,v,w)$ via using 3D Wachspress coordinates (Warren et al., 2007).

Let $\{\boldsymbol{v}_i\}_{i=1}^{2n}$ be $2n$ vertices of an *n*-prism $\mathcal{P}$ and $\boldsymbol{\Omega}$ the bottom polygonal base with vertices $\{\boldsymbol{v}_i\}_{i=1}^{n}$. For an arbitrary point $(u,v,w) \in \mathcal{P}$, $\{\lambda_i\}_{i=1}^{2n}$ are its Wachspress coordinates of the $2n$ vertices. To reproduce the tensor-product GB volume, we now introduce two local parameterization systems, which are similar to the local parameterization of the GB patch scheme.

The first is *face-distance* parameter h, which is defined as

$$h = 1 - \sum_{i=1}^{n} \lambda_i. \tag{7}$$

Another is *face* parameterization φ, by which point $(u,v,w) \in \mathcal{P}$ is mapped to a point $(u',v',w') \in \boldsymbol{\Omega}$. The face parameterization is defined as
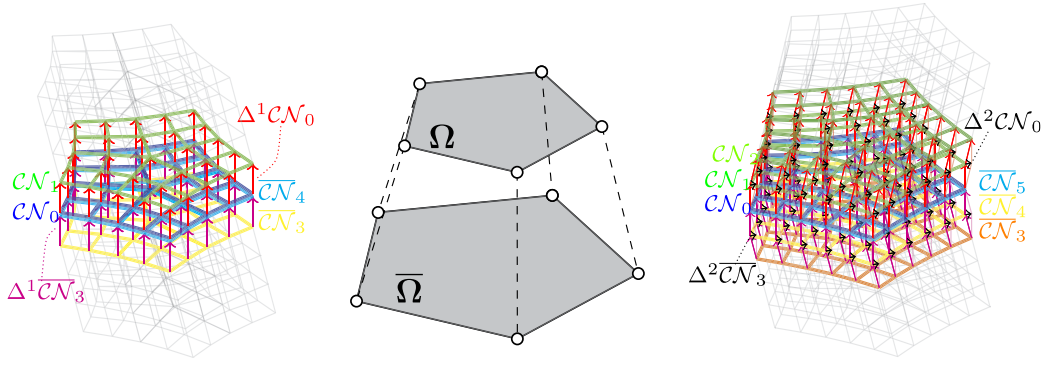
Fig. 7. Geometric condition. Left: The control nets of two adjacent 5-prismatic, degree 4 tensor-product GB volumes satisfy Equations (11) and (12). Right: The control nets of two adjacent 5-prismatic, degree 5 tensor-product GB volumes satisfy Equations (11), (12), and (14). Middle: Their parametric domains satisfy Equation (13).
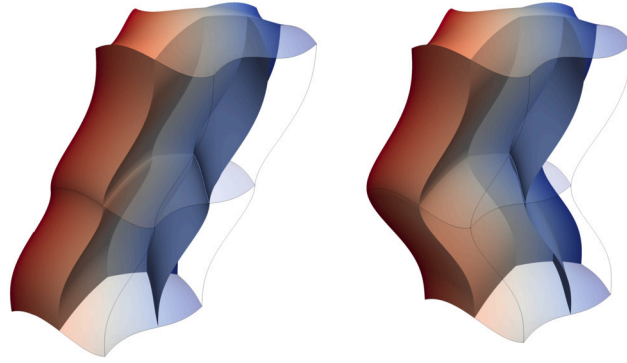

Fig. 8. Two 5-prismatic tensor-product GB volumes are smoothly connected. Left: C^1 connection. Right: C^2 connection.

$$\varphi : \mathcal{P} \to \mathbf{\Omega}, \quad (u', v', w') = \sum_{i=1}^{n} \lambda_i' \cdot \mathbf{v}_i, \tag{8}$$

where $\lambda_i' = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}$. The face-distance parameter h is 0 on the bottom base and 1 on the top base and increases linearly from 0 to 1 between the bottom and top bases. The constant surface of h is exactly parallel to the bottom base (see Fig. 6). The face parameterization φ is exactly a parallel projection that is parallel to the lateral edges. When the parametric domain polyhedron $\mathcal{P}$ is a right prism with a height of 1 and its base $\mathbf{\Omega}$ lies in the $z = 0$ plane, we indeed have $h = w$ and $(u', v', w') = (u, v, 0)$. The process of spanning the prism $\mathcal{P}$ can be viewed as lofting polygons, while local parameterizations h and φ act as its inverse transformation, resembling a slicing process.

Thus the tensor-product GB volume $\mathbf{V}(u, v, w)$ can be represented as

$$\mathbf{V}(u, v, w) = \sum_{m=0}^{d} \mathbf{S}(\varphi(u, v, w), \mathcal{CN}_m) \cdot B_m^d(h(u, v, w)), \tag{9}$$

where $(u, v, w) \in \mathcal{P}$.

3.3. Tensor-product GB volume boundary

Now we investigate the boundary behavior of a tensor-product GB volume along its boundary surfaces. First, due to its tensor-product construction, a tensor-product GB volume behaves along each lateral boundary surface as an ordinary tensor-product Bézier volume. Thus a tensor-product GB volume can be easily smoothly connected to adjacent tensor-product Bézier volumes with high-order geometric continuity (G^1 or G^2) along its lateral boundary surfaces.

Especially, we are more interested in its boundary behavior on the bases. It is clear that the basic boundary surfaces are GB patches and the corresponding cross-derivatives are vector-valued GB patches. For example, while on the bottom base $\mathbf{\Omega}$ ($w = 0$), we have that

$$\mathbf{V}(u,v,w)\big|_{w=0} = \mathbf{S}(u,v,\Delta^0 C\mathcal{N}_0),$$

$$\frac{\partial}{\partial w}\mathbf{V}(u,v,w)\big|_{w=0} = d \cdot \mathbf{S}(u,v,\Delta^1 C\mathcal{N}_0), \tag{10}$$

$$\frac{\partial^2}{\partial w^2}\mathbf{V}(u,v,w)\big|_{w=0} = d(d-1) \cdot \mathbf{S}(u,v,\Delta^2 C\mathcal{N}_0).$$

High-order derivatives are similar.

3.4. Geometric conditions for smooth joint

Hence, we can obtain some **sufficient** geometric conditions for joining two adjacent tensor-product GB volumes smoothly of up to second-order parametric or geometric continuity.

Condition 3.1. (C^1-condition) *Two adjacent n-prismatic degree d tensor-product GB volumes $\mathbf{V}$ and $\overline{\mathbf{V}}$ join C^1 if*

1. *their control nets $\{C\mathcal{N}_m\}_{m=0}^d$ and $\{\overline{C\mathcal{N}}_m\}_{m=0}^d$ satisfy*

$$C\mathcal{N}_0 = \overline{C\mathcal{N}}_d, \tag{11}$$

$$\Delta^1 C\mathcal{N}_0 = \Delta^1 \overline{C\mathcal{N}}_{d-1}, \tag{12}$$

2. *and the n-sided polygonal bases Ω and $\overline{\Omega}$ of their parametric domains must be similar, that is*

$$\Omega \sim \overline{\Omega}. \tag{13}$$

Condition 3.2. (C^2-condition) *Two adjacent n-prismatic degree d tensor-product GB volumes $\mathbf{V}$ and $\overline{\mathbf{V}}$ join C^2 if they fulfill **Condition 3.1** and their control nets $\{C\mathcal{N}_m\}_{m=0}^d$ and $\{\overline{C\mathcal{N}}_m\}_{m=0}^d$ additionally satisfy*

$$\Delta^2 C\mathcal{N}_0 = \Delta^2 \overline{C\mathcal{N}}_{d-2}. \tag{14}$$

Condition 3.3. (G^1-condition) *Two adjacent n-prismatic degree d tensor-product GB volumes $\mathbf{V}$ and $\overline{\mathbf{V}}$ join C^1 if*

1. *their control nets $\{C\mathcal{N}_m\}_{m=0}^d$ and $\{\overline{C\mathcal{N}}_m\}_{m=0}^d$ satisfy (11) and*

$$\Delta^1 C\mathcal{N}_0 = \alpha \cdot \Delta^1 \overline{C\mathcal{N}}_{d-1}, \ \alpha > 0, \tag{15}$$

2. *and the n-sided polygonal bases Ω and $\overline{\Omega}$ of their parametric domains must be similar.*

Condition 3.4. (G^2-condition) *Two adjacent n-prismatic degree d tensor-product GB volumes $\mathbf{V}$ and $\overline{\mathbf{V}}$ join G^2 if they fulfill **Condition 3.3** and their control nets $\{C\mathcal{N}_m\}_{m=0}^d$ and $\{\overline{C\mathcal{N}}_m\}_{m=0}^d$ additionally satisfy*

$$\Delta^2 C\mathcal{N}_0 = \alpha^2 \cdot \Delta^2 \overline{C\mathcal{N}}_{d-2} \tag{16}$$

with the same α as in Equation (15).

Fig. 7 illustrates the geometric conditions for C^1 and C^2 joint, while Fig. 8 provides examples of C^1 and C^2 joining. Please note that the smoothness conditions we have chosen are rather specific, as we only consider derivatives in one direction.

4. Generalized Bézier volume

Now we consider a volume, called generalized Bézier (GB) volume, with a general topology. An n-faced degree d GB volume is defined over an n-faced simple convex polyhedron $\mathcal{P}$ with n faces $\Omega_1, \Omega_2, \ldots, \Omega_n$ and n_v vertices $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n_v}$ (see the top of Fig. 9). Its control net is an n-faced degree d control net (see the bottom of Fig. 9) which is a natural and straightforward generalization of those of degree d tensor-product GB volumes.

The construction of a GB volume is also a natural extension of GB patches. A GB patch consists of multiple side-Bézier ribbons and each side-Bézier ribbon interpolates one tensor-product Bézier surface boundary. Likewise, a GB volume consists of multiple face-Bézier ribbons and each face-Bézier ribbon interpolates one tensor-product GB volume boundary.

4.1. Multi-faced control net and polyhedral domain

As mentioned above, a multi-sided control net consists of multiple overlapping half-Bézier control nets. Similarly, an n-faced control net consists of n overlapping half tensor-product GB control nets. The bottom left of Fig. 9 shows a 6-faced (345-hexahedral)
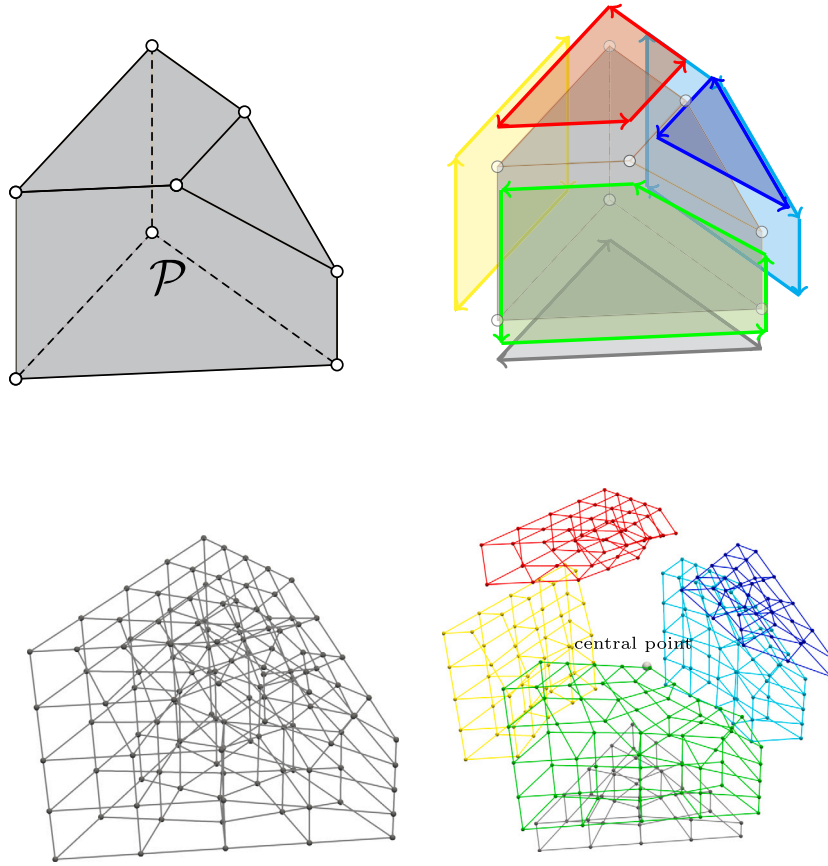
Fig. 9. A 6-faced degree 4 control net and its corresponding simple convex polyhedral domain. Top left: the simple convex polyhedron $\mathcal{P}$ called 345-hexahedron (exactly a hexahedron that has two three-sided, two four-sided, and two five-sided faces). Top right: the polygonal faces of $\mathcal{P}$ in an exploded view. The vertices of each face are oriented consistently in counterclockwise order. Bottom left: the whole multi-faced control net. Bottom right: FLCNs and the central control point in an exploded view.
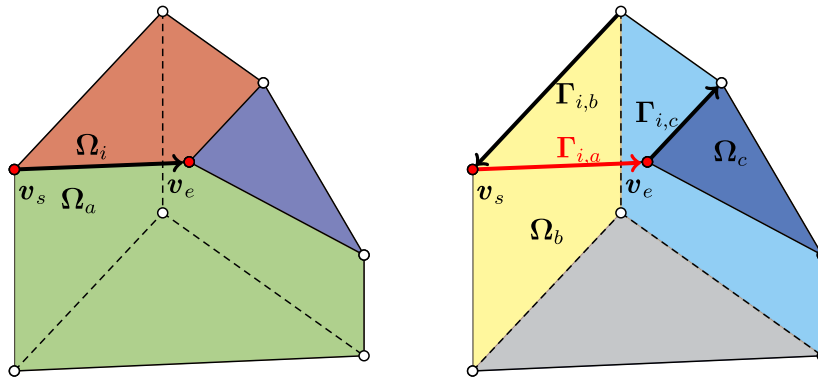

Fig. 10. Explanations about some notation for the polyhedral domain $\mathcal{P}$ as shown in Fig. 9. Side or halfedge $\Gamma_{i,a}$ belongs to Ω_i (red face) and is adjacent to Ω_a (green face). Ω_b (yellow face) and Ω_c (blue face) are the faces adjacent to the previous (left) and next (right) sides of $\Gamma_{i,a}$, respectively.

degree 4 control net. It is seen that the three adjacent half tensor-product GB control nets (see the bottom right of Fig. 9) share one common "corner". This is why we choose a simple convex polyhedron, in which all vertices have three incident faces, as the parametric domain. What's more, the multi-faced control net and the parametric domain polyhedron have a corresponding relationship. For example, each polygonal face of the polyhedron is associated with a half tensor-product GB control net by *weighted Bernstein polynomials* (refer to Section 4.4). In addition, the polyhedral domain is regarded as a closed polygonal mesh and represented via *halfedge* method (Muller and Preparata, 1978; Weiler, 1985) (see the top right of Fig. 9). Appendix A provides a simple heuristic
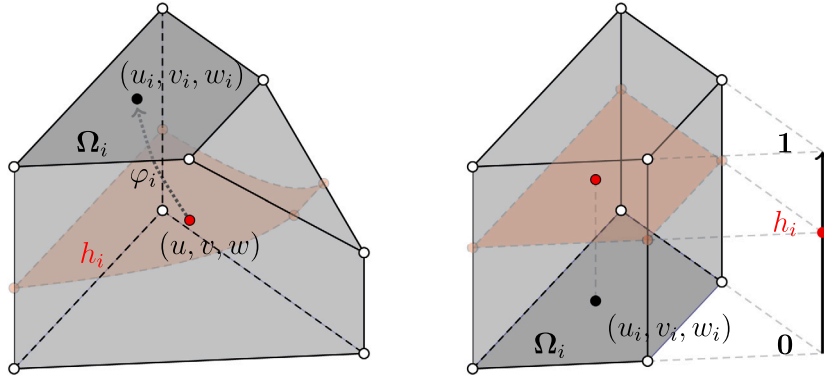
Fig. 11. Local parameterization along $\mathbf{\Omega}_i$. Left: the point (u, v, w) located on an isosurface of h_i is mapped to the point $(u_i, v_i, w_i) \in \mathbf{\Omega}_i$ using φ_i. Right: the prismatic domain spanning by $(u_i, v_i, w_i) \in \mathbf{\Omega}_i$ and $h_i \in [0, 1]$.

domain polyhedron generation algorithm, through which we can construct a simple domain polyhedron from the input multi-faced control net.

4.2. Local control net of a face-Bézier ribbon

Similarly, local control nets of an n-faced degree d control net indicate these half tensor-product GB control nets. For the sake of convenience, we use SLCNs to refer to side local control nets and FLCNs to refer to face local control nets. Two adjacent FLCNs share $(d + 1) \cdot (l + 1)^2$ control points, which are exactly the tensor product of $l + 1$ SLCNs. Three adjacent FLCNs share $(l + 1)^3$ common control points. Remark, there also exists one central control point $\mathbf{C}_{l+1,l+1,l+1}$ besides all FLCNs when d is even.

We adopt a method that is similar to halfedges to encode the control points of FLCNs. The FLCN of the i-th face-Bézier ribbon (also the i-th polygonal face $\mathbf{\Omega}_i$) is denoted by $\{C\mathcal{N}_m^i\}_{m=0}^l$ where $C\mathcal{N}_m^i$ is the m-th layer control net. Each $C\mathcal{N}_m^i$ consists of n_i SLCNs (and one additional central point $\mathbf{C}_{l+1,l+1,m}^i$ when d is even) while $\mathbf{\Omega}_i$ has n_i vertices. Each SLCN of $C\mathcal{N}_m^i$ is represented by $\{\mathbf{C}_{j,k,m}^{i,a}\}_{j,k=0}^{d,l}$, where the first superscript i indicates that the SLCN belongs to the i-th FLCN, and the second superscript a indicates that it is "adjacent" to the a-th FLCN. Moreover, its corresponding side or halfedge $\mathbf{\Gamma}_{i,a} := (\mathbf{v}_s, \mathbf{v}_e)$ (assuming its starting and ending vertices are $\mathbf{v}_s$ and $\mathbf{v}_e$, respectively) belongs to $\mathbf{\Omega}_i$ and is adjacent to $\mathbf{\Omega}_a$ (see Fig. 10).

4.3. Local parameterization

In the GB patch scheme, a polygon is reparameterized into a rectangle using local parameterization, which serves as a parametric domain of a tensor-product Bézier patch. Similarly, a polyhedron can be reparameterized into a prism using local parameterization mentioned above (Equation (7) and (8)), which serves as a parametric domain of a tensor-product GB volume. Hence, for each face, we define two local parameterization systems.

For an arbitrary point $(u, v, w) \in \mathcal{P}$, $\lambda_1, \lambda_2, \ldots, \lambda_{n_v}$ are its 3D Wachspress coordinates of the n_v vertices.[3] Then for each face we define the *face-distance* parameter h_i as

$$h_i = 1 - \sum_{\mathbf{v}_t \in \mathbf{\Omega}_i} \lambda_t. \tag{17}$$

And the *face* parameterization φ_i is defined as

$$\varphi_i : \mathcal{P} \to \mathbf{\Omega}_i, \quad (u_i, v_i, w_i) = \sum_{\mathbf{v}_t \in \mathbf{\Omega}_i} \lambda_{i,t} \cdot \mathbf{v}_t, \tag{18}$$

where

$$\lambda_{i,t} = \frac{\lambda_t}{\sum_{\mathbf{v}_j \in \mathbf{\Omega}_i} \lambda_j}. \tag{19}$$

By φ_i a point $(u, v, w) \in \mathcal{P}$ is mapped to a point $(u_i, v_i, w_i) \in \mathbf{\Omega}_i$ (see Fig. 11, 12). Remark, when the point (u, v, w) lies on the distant faces, Equation (19) is not defined due to the denominator being 0. However, this does not matter because the corresponding blend will vanish (see Section 4.7).

Then we can obtain the side and distance parameters of each side of $\mathbf{\Omega}_i$. For example, for a side $\mathbf{\Gamma}_{i,a}$, the side and distance parameters, denoted by $s_{i,a}$ and $h_{i,a}$, are obtained

---

[3] Coincidentally, the original 3D Wachspress coordinates were also defined over simple convex polyhedra (Floater, 2015; Warren et al., 2007).
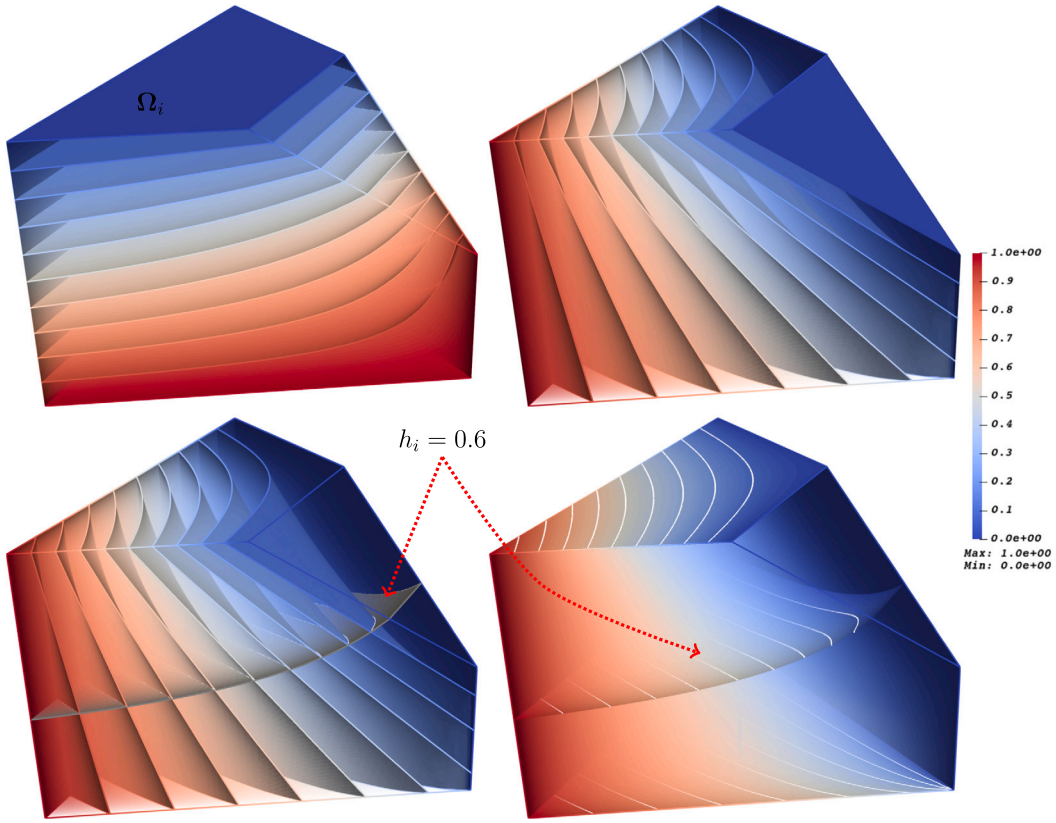
Fig. 12. Isosurfaces of local parameterizations over the polyhedral domain $\mathcal{P}$ as shown in Fig. 9. Top left: isosurfaces of h_i. Top right: Isosurfaces of φ_i. Bottom: the isosurface of $h_i = 0.6$ is mapped the quadrilateral face $\mathbf{\Omega}_i$ by φ_i.
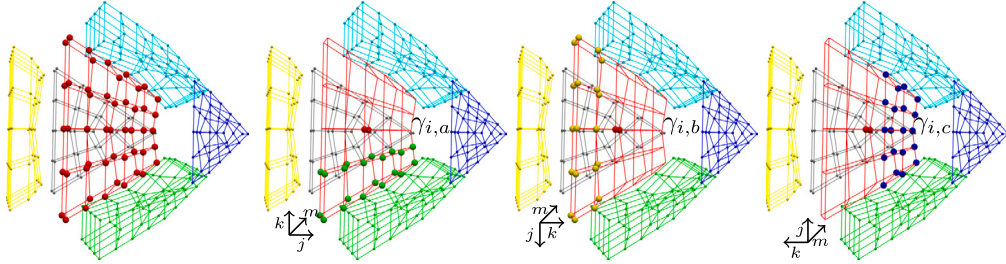

Fig. 13. Distribution of weighting functions for the i-th FLCN of the 6-faced degree 4 grid as shown in Fig. 9. Left: the i-th FLCN (red points locating at red frames). Middle left: control points shared by the i-th and a-th FLCNs (green points locating at red frames) with weighting function $\gamma_{i,a}$. Middle right: control points shared by the i-th and b-th FLCNs (yellow points locating at red frames) with weighting function $\gamma_{i,b}$. Ringht: control points shared by the i-th and c-th FLCNs (blue points locating at red frames) with weighting function $\gamma_{i,c}$.

$$s_{i,a} = \frac{\lambda'_{i,e}}{\lambda'_{i,s} + \lambda'_{i,e}}, \; h_{i,a} = 1 - (\lambda'_{i,s} + \lambda'_{i,e}), \tag{20}$$

where $\{\lambda'_{i,t} \mid \boldsymbol{v}_t \in \mathbf{\Omega}_i\}$ are 2D Wachspress coordinates of the point (u_i, v_i, w_i) with respect to the n_i vertices of $\mathbf{\Omega}_i$.

4.4. Weighted Bernstein polynomials

For each control point $\mathbf{C}^{i,a}_{j,k,m}$, a weighted Bernstein polynomial $\gamma_{i,a} \mu^{i,a}_{j,k,m} B^d_{j,k,m}(s_{i,a}, h_{i,a}, h_i)$ is defined as the blending function, which is similar to the GB patch scheme. The weighting function $\mu^{i,a}_{j,k,m}$ is defined the same way as Equation (2)
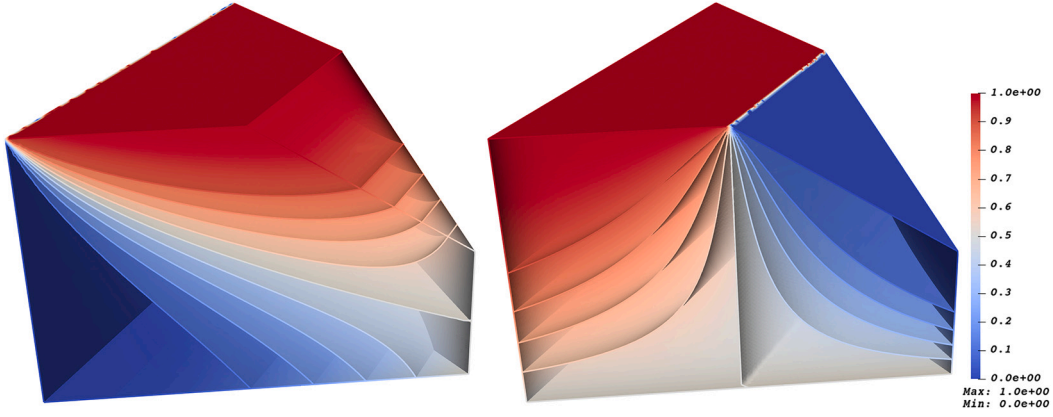
Fig. 14. Some weighting functions of the polyhedral domain $\mathcal{P}$ as shown in Fig. 9. Left: the weighting function $\gamma_{i,b}$. Right: the weighting function $\gamma_{i,c}$.

$$
\mu_{j,k,m}^{i,a} = \begin{cases} \alpha_{i,a} = \dfrac{h_{i,b}^3}{h_{i,b}^3 + h_{i,a}^3}, & 0 \leq j \leq l, \\[2mm] \beta_{i,a} = \dfrac{h_{i,c}^3}{h_{i,c}^3 + h_{i,a}^3}, & d - l \leq j \leq d, \\[2mm] 1, & otherwise, \end{cases}
\tag{21}
$$

where $\mathbf{\Gamma}_{i,b}$ and $\mathbf{\Gamma}_{i,c}$ are the previous and next sides of $\mathbf{\Gamma}_{i,a}$ (see Fig. 10), respectively. And $\gamma_{i,a}$ is defined as

$$
\gamma_{i,a} = \frac{h_a^3}{h_a^3 + h_i^3}.
\tag{22}
$$

Fig. 13 and 14 show the distributions and performances of some weighting functions $\gamma_{i,*}$-s. As for the central point $\mathbf{C}_{l+1,l+1,m}^i$ of $C\mathcal{N}_m^i$ (when d is even), the blending function is

$$
\begin{aligned}
\mathcal{B}_{l+1,l+1,m}^i &= \frac{1}{n_i} \sum_{\mathbf{\Omega}_a \uparrow\downarrow \mathbf{\Omega}_i} B_{l+1,l+1,m}^d(s_{i,a}, h_{i,a}, h_i) \\
&= \frac{1}{n_i} \sum_{\mathbf{\Omega}_a \uparrow\downarrow \mathbf{\Omega}_i} B_{l+1,l+1}^d(s_{i,a}, h_{i,a}) \cdot B_m^d(h_i) \\
&= \mathcal{B}_{l+1,l+1}^i \cdot B_m^d(h_i),
\end{aligned}
$$

where the notation $\uparrow\downarrow$ indicates the two faces are adjacent. And the blending function for the central point $\mathbf{C}_{l+1,l+1,l+1}$ of the n-faced control net is

$$
\mathcal{B}_{l+1,l+1,l+1} = \frac{1}{n} \sum_{i=1}^n \frac{1}{n_i} \sum_{\mathbf{\Omega}_a \uparrow\downarrow \mathbf{\Omega}_i} B_{l+1,l+1,l+1}^d(s_{i,a}, h_{i,a}, h_i).
$$

Fig. 16 shows some blending functions over the polyhedral domain.

4.5. Face-Bézier ribbons

Combining these control points and the corresponding blending functions the face-Bézier ribbon $\mathcal{R}^i$ can be obtained:

$$
\begin{aligned}
\mathcal{R}^i &= \sum_{\mathbf{\Omega}_a \uparrow\downarrow \mathbf{\Omega}_i} \sum_{j=0}^d \sum_{k=0}^l \sum_{m=0}^l \mathbf{C}_{j,k,m}^{i,a} \cdot \gamma_{i,a} \mu_{j,k,m}^{i,a} B_{j,k,m}^d(s_{i,a}, h_{i,a}, h_i) \\
&+ \sum_{m=0}^l \mathbf{C}_{l+1,l+1,m}^i \cdot \mathcal{B}_{l+1,l+1,m}^i.
\end{aligned}
\tag{23}
$$

Let $\mathbf{R}_m^{i,a}$ be a side-Bézier ribbon. That is

$$
\mathbf{R}_m^{i,a} = \sum_{j=0}^d \sum_{k=0}^l \mathbf{C}_{j,k,m}^{i,a} \cdot \mu_{j,k,m}^{i,a} B_{j,k}^d(s_{i,a}, h_{i,a}).
\tag{24}
$$

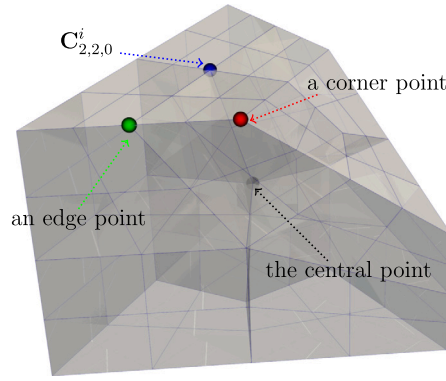Then $\mathcal{R}^i$ can be represented as

Fig. 15. Some control points on the 6-faced degree 4 grid as shown in Fig. 9. Red point: a corner point lying on a three- (c-th), a four- (i-th), and a five-sided (a-th) FLCN and also six SLCNs. Green point: an edge point lying on two FLCNs (i-th and a-th) and also two SLCNs. Blue point: the central point $\mathbf{C}_{2,2,0}^i$ of $C\mathcal{N}_0^i$. Black point: the central point $\mathbf{C}_{2,2,2}$.
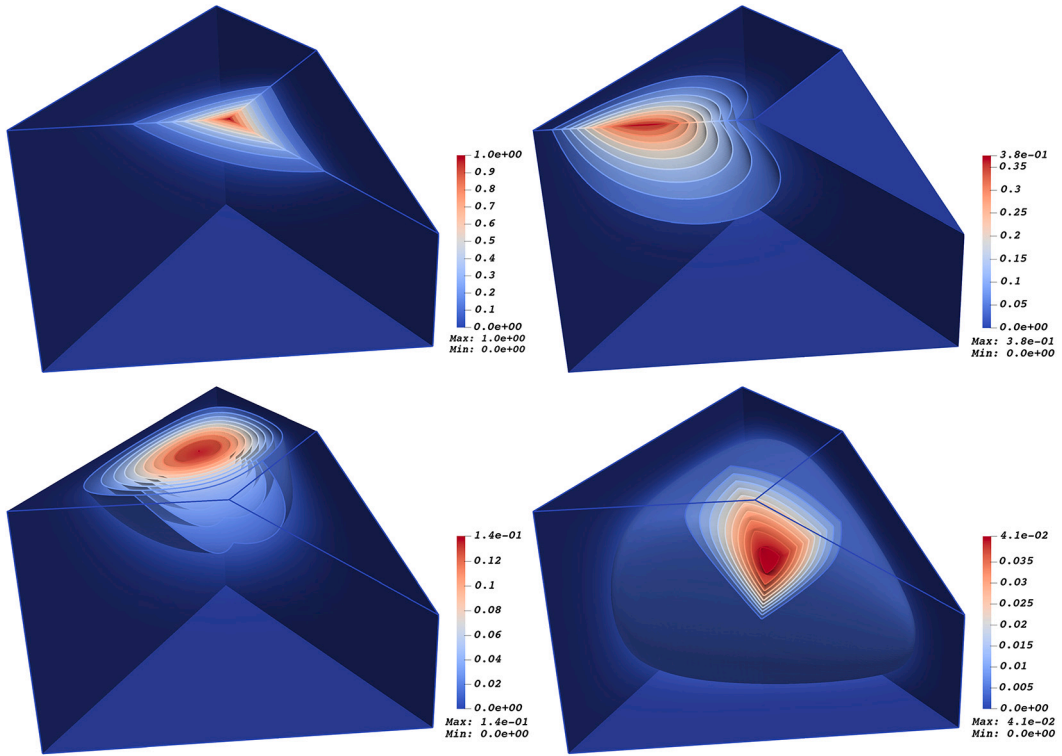

Fig. 16. Blending functions for some control points as shown in Fig. 15. Top left: the blending function for the corner point is the sum of $\gamma_{i,c}\mathring{\mu}_{0,0,0}^{i,c}B_{0,0,0}^4(h_{i,a},h_{i,c},h_i)$, $\gamma_{i,a}\mathring{\mu}_{4,0,0}^{i,a}B_{4,0,0}^4(h_{i,b},h_{i,a},h_i)$, $\gamma_{a,i}\mu_{0,0,0}^{a,i}B_{0,0,0}^4(s_{a,i},h_{a,i},h_i)$, $\gamma_{a,c}\mu_{4,0,0}^{a,c}B_{4,0,0}^4(s_{a,c},h_{a,c},h_i)$, $\gamma_{c,a}\mu_{0,0,0}^{c,a}B_{0,0,0}^4(s_{c,a},h_{c,a},h_i)$, and $\gamma_{c,i}\mu_{4,0,0}^{c,i}B_{4,0,0}^4(s_{c,i},h_{c,i},h_i)$. Top right: the blending function for the edge point is the sum of $\gamma_{i,a}\mathring{\mu}_{2,0,0}^{i,a}B_{2,0,0}^4(h_{i,b},h_{i,a},h_i)$ and $\gamma_{a,i}\mu_{2,0,0}^{a,i}B_{2,0,0}^4(s_{a,i},h_{a,i},h_i)$. Bottom left: the blending function $\mathring{\mathcal{B}}_{2,2,0}^i$ for the point $\mathbf{C}_{2,2,0}^i$ (intersection view). Bottom right: the blending function $\mathcal{B}_{2,2,2}$ for the central point (intersection view). As for the alternative weighting function such as $\mathring{\mu}_{0,0,0}^{i,c}$, please refer to Section 4.8.

$$\mathcal{R}^i = \sum_{m=0}^{l} B_m^d(h_i)[\sum_{\boldsymbol{\Omega}_a \uparrow \downarrow \boldsymbol{\Omega}_i} \gamma_{i,a}\mathbf{R}_m^{i,a} + \mathbf{C}_{l+1,l+1,m}^i \cdot \mathcal{B}_{l+1,l+1}^i], \tag{25}$$

Remark, the central control point $\mathbf{C}_{l+1,l+1,m}^i$ only exists when degree d is even.

The construction of face-Bézier ribbons is a straightforward generalization of side-Bézier ribbons. A side-Bézier ribbon is a partial tensor-product Bézier patch in which weighting functions are assigned at the left and right corners. Likewise, a face-Bézier ribbon is a partial tensor-product GB volume in which weighting functions are assigned at the different sides. A side-Bézier ribbon interpolates a tensor-product Bézier patch boundary determined by its SLCN. A face-Bézier ribbon also interpolates a tensor-product GB volume boundary determined by its FLCN.
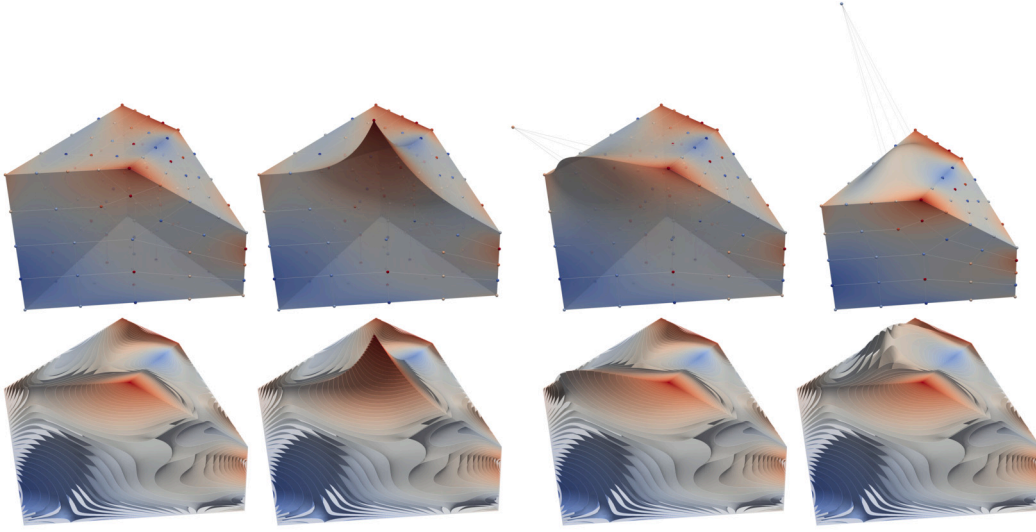
Fig. 17. Editing a GB volume. Top: the GB volumes with control nets. Bottom: isosurfaces of GB volumes. From left to right: the original GB volume; moving a corner control point; moving an edge point; moving a face point.

4.6. The volume equation

Now we are ready to formulate the equation of the GB volume. An n-faced degree d GB volume $\mathbf{V}(u, v, w)$ is defined as

$$\mathbf{V}(u, v, w) = \frac{1}{B_\Sigma} [\sum_{i=1}^{n} \mathcal{R}^i + \mathbf{C}_{l+1,l+1,l+1} \cdot \mathcal{B}_{l+1,l+1,l+1}], \tag{26}$$

where B_Σ is the sum of all blending functions and $\mathbf{C}_{l+1,l+1,l+1}$ only exists when degree d is even.

4.7. Boundary behavior

A GB volume behaves like a tensor-product GB volume on the boundary surface. This conclusion can be easily reached because the construction of GB volumes is a direct extension of the construction of GB patches. Here we provide a simple and intuitive explanation.

Firstly, it is clear that the i-th face-Bézier ribbon $\mathcal{R}^i$ does not affect the distant faces $\mathbf{\Omega}_d$ ($\mathbf{\Omega}_d \notin \{\mathbf{\Omega}_a \mid \mathbf{\Omega}_a \uparrow\downarrow \mathbf{\Omega}_i\}$ and $\mathbf{\Omega}_d \neq \mathbf{\Omega}_i$). This is because the *face-distance* parameter $h_i = 1$ on the distant faces and all the blending functions of $\mathcal{R}^i$ contain a $(1 - h_i)^{d-m}$ term. Thus, $\mathcal{R}^i$ does not affect distant faces, either in terms of position or differential properties.

Secondly, with the *face* parameterization φ_i, all points on adjacent faces are mapped to the corresponding sides of $\mathbf{\Omega}_i$. For example, all points on the adjacent face $\mathbf{\Omega}_a$ will be mapped to the corresponding side $\mathbf{\Gamma}_{i,a}$. Thus we can see that only the term $\sum_{m=0}^{l} \gamma_{i,a} \mathbf{R}_m^{i,a}$ of $\mathcal{R}^i$ contributes to the adjacent face $\mathbf{\Omega}_a$.

In addition, $\gamma_{i,a}$ and its up to second-order derivatives vanish on the adjacent face $\mathbf{\Omega}_a$ since that $\gamma_{i,a}$ contains a term h_a^3 and $h_a \equiv 0$ on $\mathbf{\Omega}_a$. Thus, only the i-th face-Bézier ribbon $\mathcal{R}^i$ affects the i-th face $\mathbf{\Omega}_i$.

Then we consider the performance of $\mathcal{R}^i$ on the i-th face $\mathbf{\Omega}_i$. It can be easily seen that $\gamma_{i,a} = 1$ and $\partial(\gamma_{i,a}) = \partial^2(\gamma_{i,a}) = 0$ on $\mathbf{\Omega}_i$. By the observation, we can conclude that (assuming degree $d \geq 5$)

$$\mathcal{R}^i\Big|_{\mathbf{\Omega}_i} = \sum_{\mathbf{\Omega}_a \uparrow\downarrow \mathbf{\Omega}_i} \mathbf{R}_0^{i,a}\Big|_{\mathbf{\Omega}_i} + \mathbf{C}_{l+1,l+1,0}^i \cdot \mathcal{B}_{l+1,l+1}^i\Big|_{\mathbf{\Omega}_i},$$

$$\partial(\mathcal{R}^i)\Big|_{\mathbf{\Omega}_i} = \sum_{m=0}^{1} \partial[B_m^d(h_i)(\sum_{\mathbf{\Omega}_a \uparrow\downarrow \mathbf{\Omega}_i} \mathbf{R}_m^{i,a} + \mathbf{C}_{l+1,l+1,m}^i \cdot \mathcal{B}_{l+1,l+1}^i)]\Big|_{\mathbf{\Omega}_i}, \tag{27}$$

$$\partial^2(\mathcal{R}^i)\Big|_{\mathbf{\Omega}_i} = \sum_{m=0}^{2} \partial^2[B_m^d(h_i)(\sum_{\mathbf{\Omega}_a \uparrow\downarrow \mathbf{\Omega}_i} \mathbf{R}_m^{i,a} + \mathbf{C}_{l+1,l+1,m}^i \cdot \mathcal{B}_{l+1,l+1}^i)]\Big|_{\mathbf{\Omega}_i},$$

where the right term(s) only exists when degree d is even.

While this may still be some distance away from the conclusion we desire, it is not far off. It is clearly that $\partial^p(\mathcal{R}^i)\Big|_{\mathbf{\Omega}_i}$ differs $\partial^p \mathbf{V}\Big|_{\mathbf{\Omega}_i}$ by only one denominator, B_Σ (for $p = 0, 1, 2$). Notably, B_Σ is exactly the sum of all blending functions of $\mathcal{R}^i$ on face $\mathbf{\Omega}_i$. Thus, the final GB volume $\mathbf{V}(u, v, w)$ will become a tensor-product GB volume on $\mathbf{\Omega}_i$.
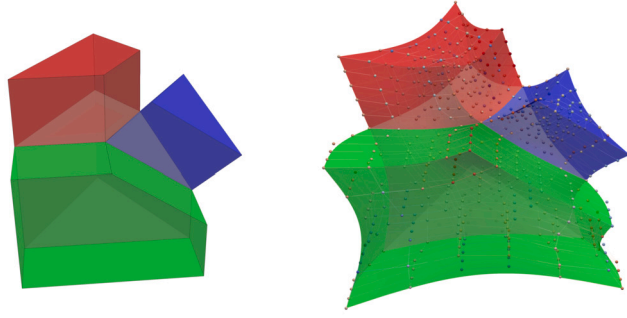
Fig. 18. Parametric and physical domains. Left: a 345-hexahedron with three adjacent prisms. Right: the corresponding degree 5 GB volumes with control nets.
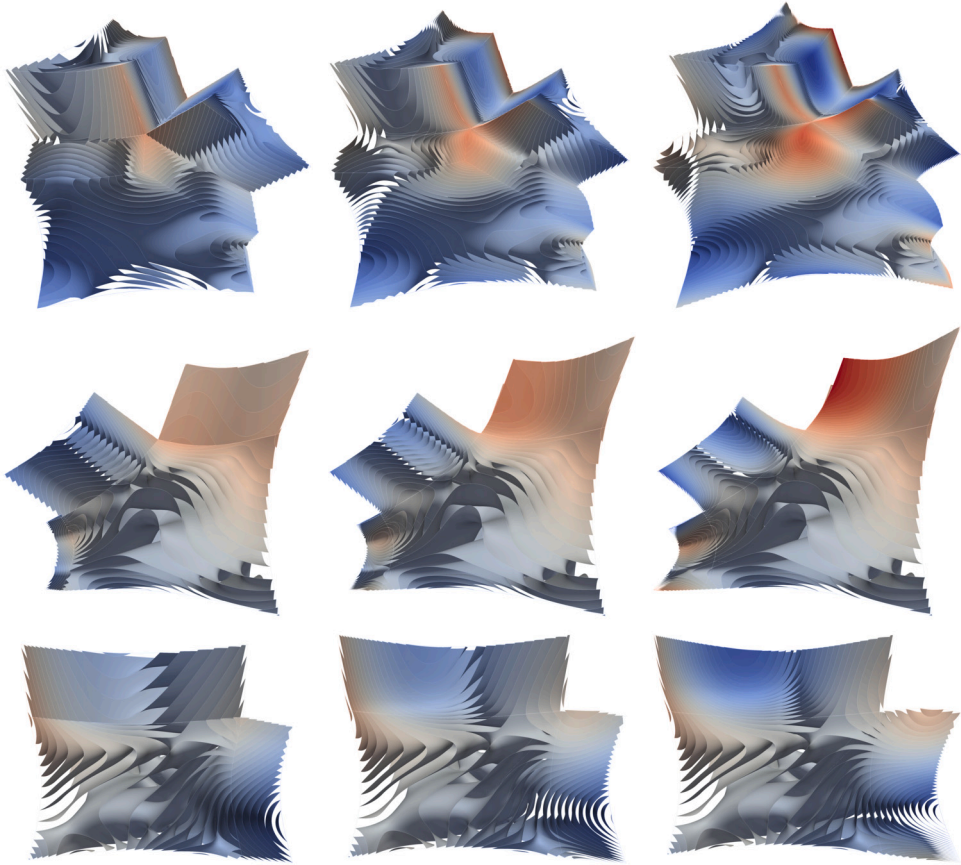

Fig. 19. Isosurfaces. A 345-hexahedral GB volume is assembled with a tensor-product Bézier, a 3-prismatic tensor-product GB, and a 5-prismatic tensor-product GB volume. From left to right: G^0 connection; G^1 connection; and G^2 connection. From top to bottom: front view; back view; and left view.

In conclusion, the above provides sufficient evidence that the GB volume possesses tensor-product GB volume boundaries. For instance, on the i-th face Ω_i, the GB volume behaves like a tensor-product GB volume that shares the same three layers of control points $\{C\mathcal{N}_m^i\}_{m=0}^2$, and whose polygonal base Ω is similar to Ω_i. Thus GB volumes can be smoothly connected to other GB volumes or tensor-product GB volumes with geometric continuity by using **Condition 3.1**, **Condition 3.2**, **Condition 3.3**, or **Condition 3.4**.

4.8. Adjustment to a 4-sided face-Bézier ribbon

Although the GB volume possesses tensor-product GB volume boundaries, a 4-prismatic tensor-product GB volume is usually not a tensor-product Bézier volume. This is because a 4-sided GB patch cannot be a tensor-product Bézier patch unless its parametric domain is a parallelogram. However, we desire a GB volume to have a tensor-product Bézier volume boundary if the corresponding polygonal face of the parametric polyhedron is an arbitrary quadrilateral.

Therefore, we need to make some adjustments to the weighted Bernstein polynomials of a 4-sided face-Bézier ribbon. That is, for each control point $\mathbf{C}_{j,k,m}^{i,a}$ of a 4-sided face-Bézier ribbon, an alternative weighted Bernstein polynomial $\gamma_{i,a}\mathring{\mu}_{j,k,m}^{i,a}B_{j,k,m}^{d}(h_{i,b},h_{i,a},h_i)$ is defined as the blending function. Please note that the original trivariate Bernstein basis function $B_{j,k,m}^{d}(s_{i,a},h_{i,a},h_i)$ has been replaced by $B_{j,k,m}^{d}(h_{i,b},h_{i,a},h_i)$, which is somewhat similar to the Overlap GB patch (Salvi, 2022). The weighting function $\mathring{\mu}_{j,k,m}^{i,a}$ is defined as

$$\mathring{\mu}_{j,k,m}^{i,a} = \begin{cases} \frac{1}{2}\gamma_{i,b}, & 0 \leq j \leq l, \\ \frac{1}{2}\gamma_{i,c}, & d - l \leq j \leq d, \\ 1, & otherwise. \end{cases} \tag{28}$$

Similarly, the blending functions for the central point $\mathbf{C}_{l+1,l+1,m}^{i}$ of $C\mathcal{N}_m^i$ (when d is even) is

$$\mathring{B}_{l+1,l+1,m}^{i} = \frac{1}{4} \sum_{\mathbf{\Omega}_a \uparrow \mathbf{\Omega}_i} B_{l+1,l+1,m}^{d}(h_{i,b},h_{i,a},h_i).$$

And the blending function for the central point $\mathbf{C}_{l+1,l+1,l+1}$ of the n-faced control net is

$$B_{l+1,l+1,l+1} = \frac{1}{n}[\sum_{n_i \neq 4} \frac{1}{n_i} \sum_{\mathbf{\Omega}_a \uparrow \mathbf{\Omega}_i} B_{l+1,l+1,l+1}^{d}(s_{i,a},h_{i,a},h_i)$$
$$+ \sum_{n_i = 4} \frac{1}{4} \sum_{\mathbf{\Omega}_a \uparrow \mathbf{\Omega}_i} B_{l+1,l+1,l+1}^{d}(h_{i,b},h_{i,a},h_i)].$$

Through this adjustment, not only does the boundary of the GB volume on the quadrilateral faces become a tensor-product Bézier volume boundary, *but also when the parametric polyhedron degenerates into a prism, the GB volume also degenerates into a tensor-product GB volume*. Hence, when a GB volume is joined along a quadrilateral face with other volumes, only the relationships between control points need to be considered, without the need to consider the shape of the corresponding quadrilateral faces in the parametric domain. Thus GB volumes can be easily integrated into existing workflows for constructing and manipulating tensor-product Bézier volumes.

5. Examples

In this section, some examples are presented to show the features of GB volumes. For visualization purposes, each control point of the GB volumes in the examples consists of four components, namely, coordinate values (x, y, z) and an attribute value g, where the attribute value is randomly assigned. Furthermore, when two volumes are joined together, their control points, which are four-dimensional (i.e., (x, y, z, g)), should satisfy the corresponding geometric conditions. The generation of isoparametric surfaces on the volumes is based on the attribute value g for rendering.

5.1. Editing

Fig. 17 illustrates the process of editing a 345-hexahedral degree 4 GB volume. Just like Bézier curves and surfaces, we can easily modify the shape of GB volumes by moving their control points. In addition, from the figure, we can also observe that GB volumes exhibit a certain degree of locality. Therefore, GB volumes are relatively easy to design in terms of shape.

5.2. Smooth joining

Fig. 19 and 20 illustrate the integration of a 345-hexahedral degree 5 GB volume with surrounding tensor-product volumes. A 345-hexahedral GB volume is joined with a tensor-product Bézier, a 3-prismatic tensor-product GB, and a 5-prismatic tensor-product GB volume around a corner. Their parametric domains are shown in Fig. 18. The cases of G^0, G^1 and G^2 joining have all been demonstrated. Here we utilized the **Condition 3.3** and **Condition 3.4** with $\alpha = 0.5$. Fig. 19 shows the isosurfaces inside these volumes and Fig. 20 the details of partial isosurfaces. From the isosurfaces, it is evident that the GB volume can be smoothly joined with surrounding tensor-product volumes with a certain degree of continuity.

In Fig. 21, we present two simple examples: a tetrahedral and a dodecahedral GB volume. Both the two GB volumes are smoothly connected with adjacent tensor-product GB volumes with G^2 continuity.

5.3. Complex volumetric models

Here we will present some complex volumetric models solved with tensor-product GB and GB volumes. All these volumetric models have a global G^1 continuity. For more details about the construction of control nets please refer to Appendix B. To represent polyhedral meshes, we utilized the *half-face* data structure proposed by Kremer et al. (2013).

Fig. 22 shows a volumetric model that is controlled by a simple input polyhedral mesh comprising a 345-hexahedron and a tetrahedron. The final smooth volumetric model consists of 38 tensor-product Bézier, 9 tensor-product GB and 2 GB volumes. Fig. 23
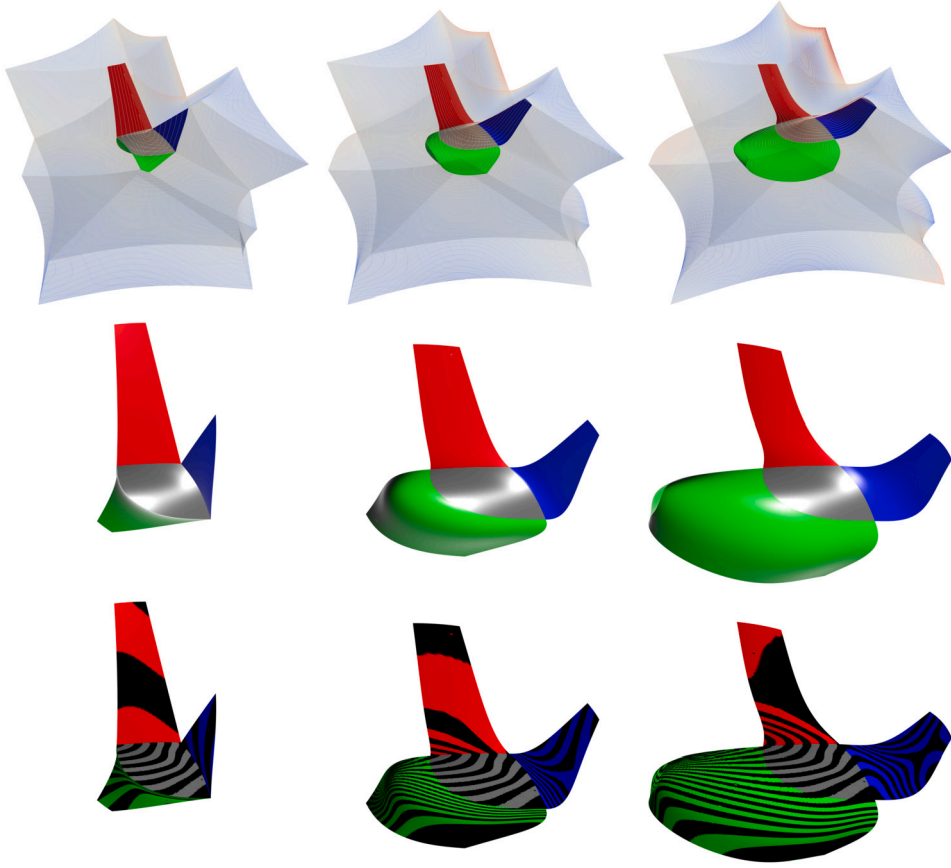
Fig. 20. Partial isosurfaces in Fig. 19. The isosurfaces which take the same value of g and belong to different volumes are colored differently. From left to right: G^0 connection; G^1 connection; and G^2 connection. From top to bottom: isosurfaces inside volumes; close-up of shaded isosurfaces; and isophotes on the isosurfaces.
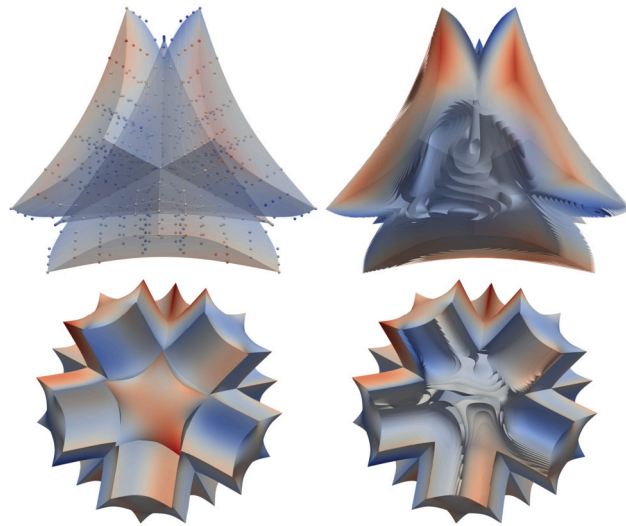

Fig. 21. G^2 connection. Top: a tetrahedral GB volume is surrounded by four 3-prismatic tensor-product GB volumes. Bottom: a dodecahedral GB volume is surrounded by twelve 5-prismatic tensor-product GB volumes. Left: GB volumes. Right: interior view of isosurfaces.
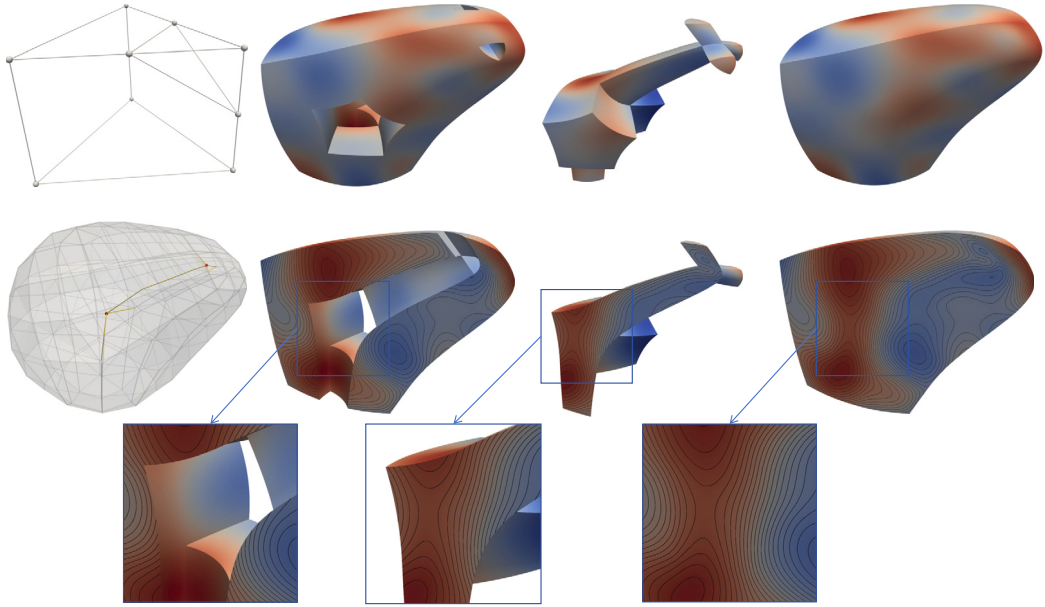
Fig. 22. 345-**hexahedron & tetrahedron**. Top left: input polyhedral mesh. Bottom left: refined mesh with red internal irregular points and yellow internal irregular edges (by applying Catmull–Clark volumetric subdivision twice). Top middle left: volumetric model with holes (consisting of tensor-product Bézier volumes). Bottom middle left: contours in a slice plane of the volumetric model with holes. Top middle right: irregular regions (consisting of tensor-product GB and GB volumes). Bottom middle right: contours in a slice plane of the irregular regions. Top right: hole-filled volumetric model. Bottom right: contours in a slice plane of the hole-filled volumetric model.
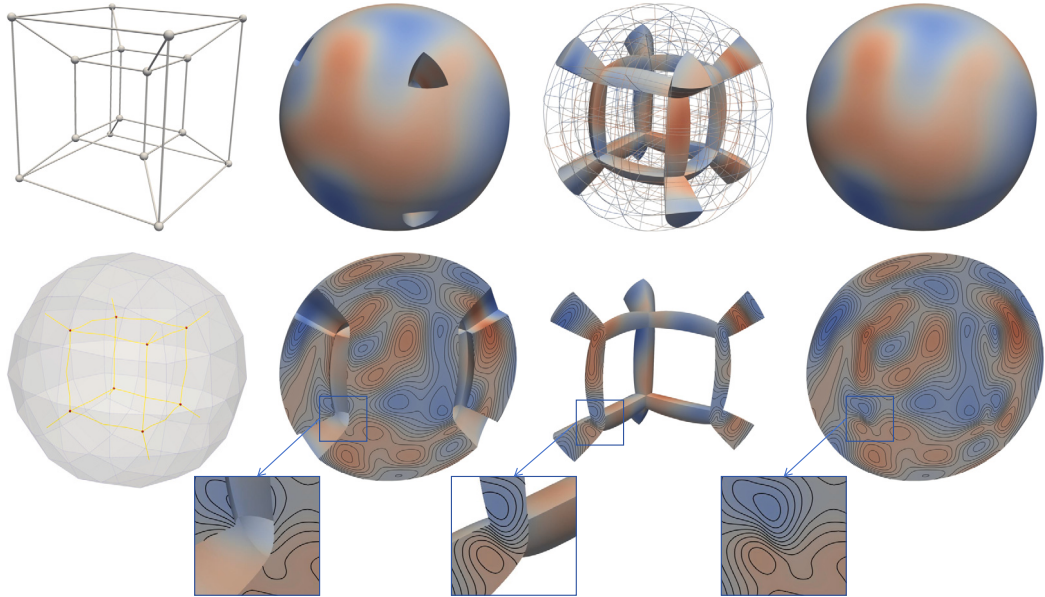

Fig. 23. Cube-Sphere. Top left: input hexmesh. Bottom left: refined mesh with red internal irregular points and yellow internal irregular edges (by applying Catmull–Clark volumetric subdivision twice). Top middle left: volumetric model with holes (consisting of tensor-product Bézier volumes). Bottom middle left: contours in a slice plane of the volumetric model with holes. Top middle right: irregular regions (consisting of tensor-product GB and GB volumes). Bottom middle right: contours in a slice plane of the irregular regions. Top right: hole-filled volumetric model. Bottom right: contours in a slice plane of the hole-filled volumetric model.

illustrates a commonly used test model — the input is a hexmesh consisting of a cube and six square frustums surrounding it. The final smooth sphere-like model consists of 351 tensor-product Bézier, 60 tensor-product GB, and 8 GB volumes.

Fig. 24 shows a volumetric model **Rabbit** which has 1232 tensor-product Bézier, 287 tensor-product GB, and 347 GB volumes. The input is a pure hexmesh. Fig. 25 illustrates a more general scenario. The input is a triangular mesh **Lego** which has 14849 vertices and 10744 triangles. We first transformed it into a hexmesh by the CE-PolyCubeMap algorithm proposed by Guo et al. (2020). Other
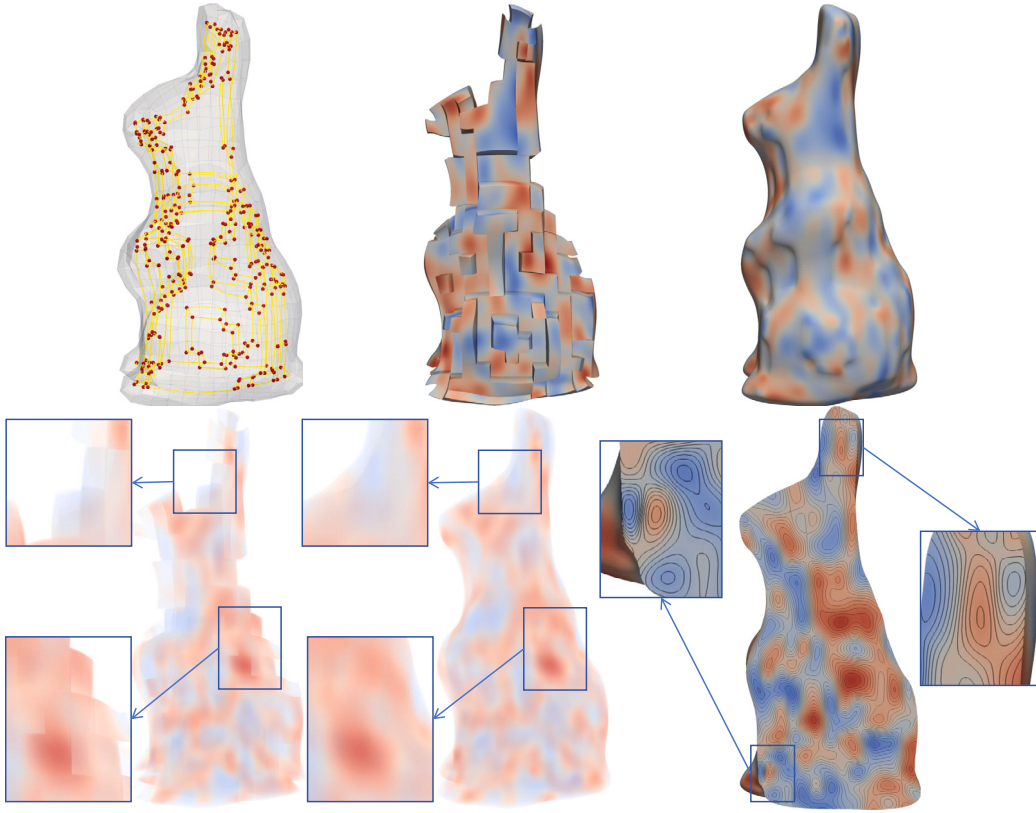
Fig. 24. Rabbit. From left to right in the top row: input hexmesh with irregularities (red internal irregular points and yellow internal irregular edges); volumetric model with holes (consisting of tensor-product Bézier volumes); and hole-filled volumetric model. From left to right in the bottom row: volumetric model with holes — volume rendering; hole-filled volumetric model — volume rendering; and contours in a slice plane of the hole-filled volumetric model.
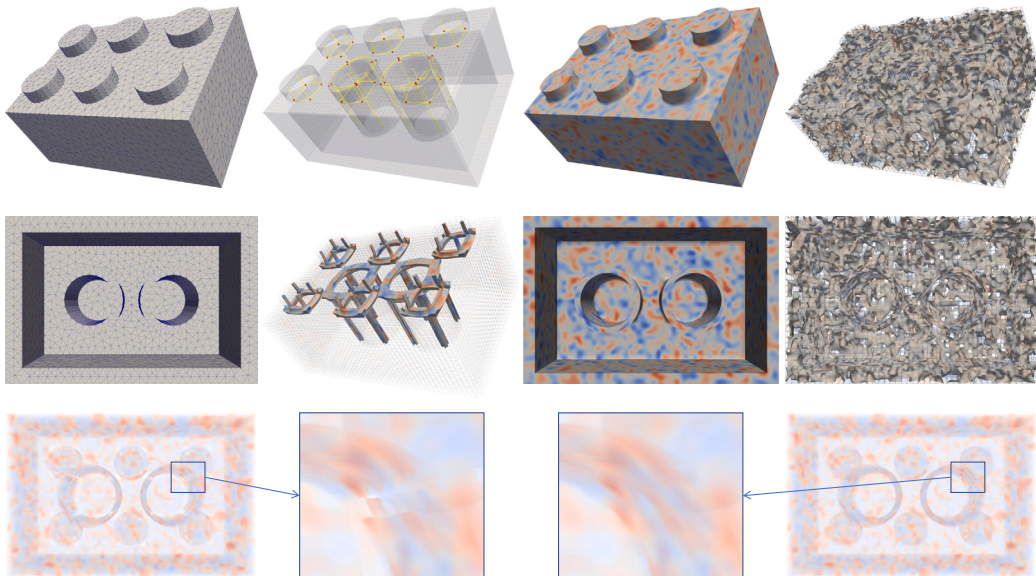

Fig. 25. Lego. From left to right in the first row: input triangular mesh; generated hexmesh with irregularities; final volumetric model; and isosurface with the median value of g. From left to right in the second row: input triangular mesh — back view; irregular regions (consisting of tensor-product GB and GB volumes); final volumetric model — back view; and isosurface with the median value of g — back view. From left to right in the third row: volumetric model with holes and hole-filled volumetric model (volume rendering).
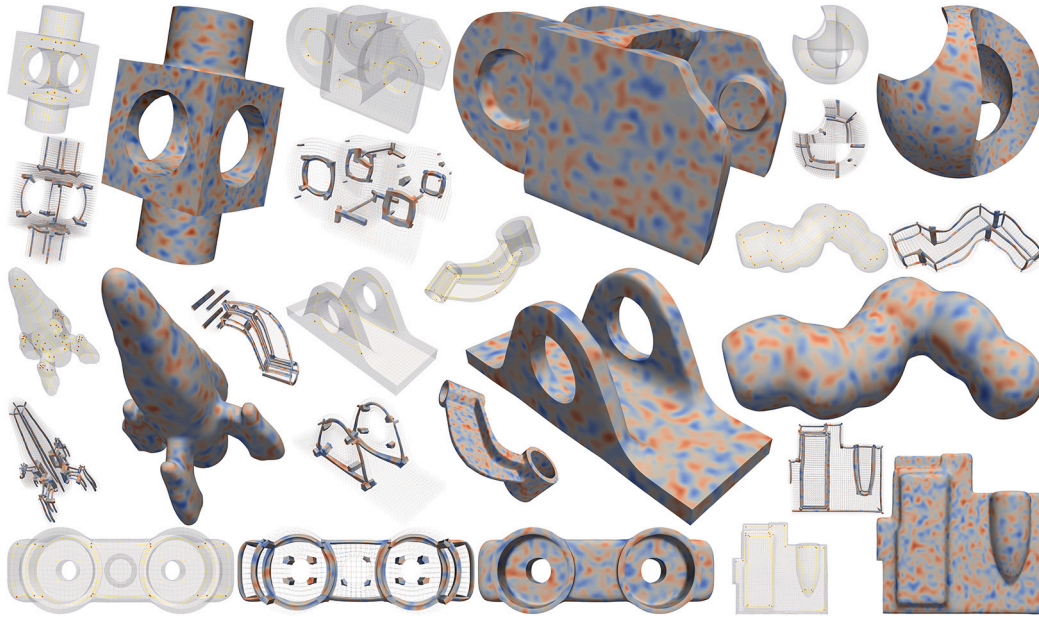
Fig. 26. Practical CAD models. All the data used in this figure are from Guo et al. (2020).

hexmesh generation algorithms such as CUBECOVER (Nieser et al., 2011) can also be adopted. The final smooth volumetric model consists of 11460 tensor-product Bézier, 560 tensor-product GB, and 40 GB volumes. More practical CAD models are shown in Fig. 26.

6. Discussion

In this section, we will discuss the limitations of the GB volume and outline potential areas for future research.

Local parameterization. There is an issue that needs to be addressed, and that is the effectiveness of local parameterization. As mentioned above, the isosurfaces of h_i are mapped to the polygonal face by φ_i (cf. Fig. 11). However, we are not yet certain if this process is injective. For example, we are unsure if there could be a scenario where two points on the isosurface ($h_i < 1$) map to the same point on the polygonal face after local parameterization. Based on our experience, the local parameterization appears to be injective, but it still needs to be theoretically proven. In addition, the side parameterization of GB patch scheme (s_i, cf. Equation (1) and Fig. 4) is a *radial sweeping parameterization*, which seems to have first appeared in Charrot and Gregory's pentagonal surface construction (Charrot and Gregory, 1984), that all isolines of s_i are straight lines. And we wonder if our face parameterization φ_i has the similar property that the isosurfaces of φ_i are ruled surfaces. To address these issues, a profound understanding of the 3D generalized barycentric coordinates might be necessary.

Extensions. Although we have provided geometric conditions for smooth joining between GB volumes, similar to the conditions for joining tensor-product Bézier volumes, achieving a smooth connection between volumes in complicated geometric modeling can be challenging. In 2018, Salvi and Várady (2018) extended their GB patch scheme so that the input side-Bézier ribbons can be independent. For example, the adjacent ribbons can have different degrees. We believe that this technique is also suitable for the GB volume and by using independent face-Bézier ribbons smooth joining between volumes may be easier. Moreover, independent ribbons are highly likely to lead to twist incompatibility and how the twist will influence the shape of a volume is still a terra incognita. Additionally, since the shapes of GB volumes are directly influenced by the polyhedral parametric domains, constructing suitable assembled polyhedral parametric domains is also an important task. Another valuable work is the extension of GB volumes to B-spline. As mentioned above, irregularities over unstructured hexahedral meshes are very difficult to deal with. GB patches have been extended to B-spline to handle the irregularities over unstructured quadrilateral meshes (Hettinga and Kosinka, 2020). Therefore, extending GB volumes to B-spline to handle the irregularities over unstructured hexahedral meshes is also feasible. Although the topology of a GB volume is currently limited to a simple convex polyhedron, we firmly believe that this restriction can be lifted in the future, just as the GB patch was extended from a convex polygonal domain to an arbitrary domain (Várady and Salvi, 2020). Therefore, generalized barycentric coordinates that can be defined over concave polyhedra, such as mean value coordinates (Floater et al., 2005), harmonic coordinates (Joshi et al., 2007), and maximum likelihood coordinates (Chang et al., 2023), need to be utilized.

How to use? Most people might be more interested in applications of GB volumes and this is also what we are concerned about. Nevertheless, our work has not delved into such depths yet. Two possible applications of the GB volume may be as volumetric parameterization in IGA and as shape functions in FEA, but constructing analysis-suitable GB volumes may be not easy work. This is because we use generalized barycentric coordinates as parameters, resulting in the composite function having a potential relatively high rational degree. Therefore, whether it is computing its derivatives or integrals, or converting it into NURBS representation, it is all very challenging. The same issue also arises in those polygonal surfaces that use generalized barycentric coordinates as

parameters (Peters, 2019). It is worth mentioning that recently, Wang et al. (2023) successfully applied GB patches in IGA, which serves as an inspiring reference for future work. In 2016, Massarwi and Elber (2016) proposed a volumetric modeling framework based on trimmed B-spline. They suggest designing or modeling a 3D object by using volumetric cells which are trimmed B-spline trivariates. We are wondering if the GB volume can also be used for the same task. In addition, GB volumes have the potential to be applied in microstructure design. A microstructure model is typically composed of multiple cells stacked together with the same shape. Most microstructure cells are irregular in shape. Thus, using polyhedral GB volumes to represent microstructure cells seems to be a good fit.

7. Conclusion

We have introduced the GB volume, a novel polyhedral volumetric representation, which is a natural generalization of the GB patch from 2D manifold to 3D manifold. A GB volume is defined over a simple convex polyhedral domain by using 3D Wachspress coordinates, with a multi-faced (polyhedral) control net. It possesses tensor-product borders that its boundary surfaces are GB patches and corresponding cross-derivatives vector-valued GB patches. Thus GB volumes connect to adjacent tensor-product Bézier/GB volumes with G^1 or G^2 continuity by using some simple and intuitive geometric conditions. When the parametric polyhedron becomes a prism, the GB volume also becomes a tensor-product GB volume.

Also, we have to point out that our work is still in an exploratory phase and is at a preliminary stage. There are still many issues here, both theoretically and in terms of application, that await resolution in the future.

CRediT authorship contribution statement

Kaikai Qin: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yajuan Li:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Conceptualization. **Chongyang Deng:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

Appendix A. Domain generation

Given a general polyhedral Bézier control net, a domain polyhedron can be created heuristically by following these steps (see Fig. A.27).

1. ***Initialization.*** First, we can obtain an initial polyhedron $\mathcal{P}'$ by setting the corner points on the polyhedral Bézier control net as the polyhedron vertices $\{v_i'\}$. The initial polyhedron $\mathcal{P}'$ could be invalid because not all of the vertices belonging to each polygon are necessarily planar.
2. ***Fitting planes.*** For each polygon $\mathbf{\Omega}_j'$ of $\mathcal{P}'$, with k vertices $\{v_{j_i}'\}_{i=1}^k$, a plane $\vec{n}_j \cdot (x - x_j, y - y_j, z - z_j) = 0$ can be defined, where $(x_j, y_j, z_j) = \frac{\sum_{i=0}^k v_{j_i}'}{k}$ the centroid c_j' of $\mathbf{\Omega}_j'$ and $\vec{n}_j$ a normal vector of the plane. $\vec{n}_j = \frac{\sum_{i=1}^k \vec{n}_{j_i}}{k}$ where $\vec{n}_{j_i}$ is the unit normal of $\triangle(v_{j_i}', v_{j_{(i+1)\%k}}', c_j')$.
3. ***Vertices generation.*** Then the domain polyhedron $\mathcal{P}$ is naturally defined by these planes. Each vertex v_i can be obtained by solving a linear system

$$\begin{cases} \vec{n}_a \cdot (x - x_a, y - y_a, z - z_a) = 0 \\ \vec{n}_b \cdot (x - x_b, y - y_b, z - z_b) = 0 \\ \vec{n}_c \cdot (x - x_c, y - y_c, z - z_c) = 0 \end{cases},$$

where $\mathbf{\Omega}_a$, $\mathbf{\Omega}_b$ and $\mathbf{\Omega}_c$ are three incident faces of v_i.
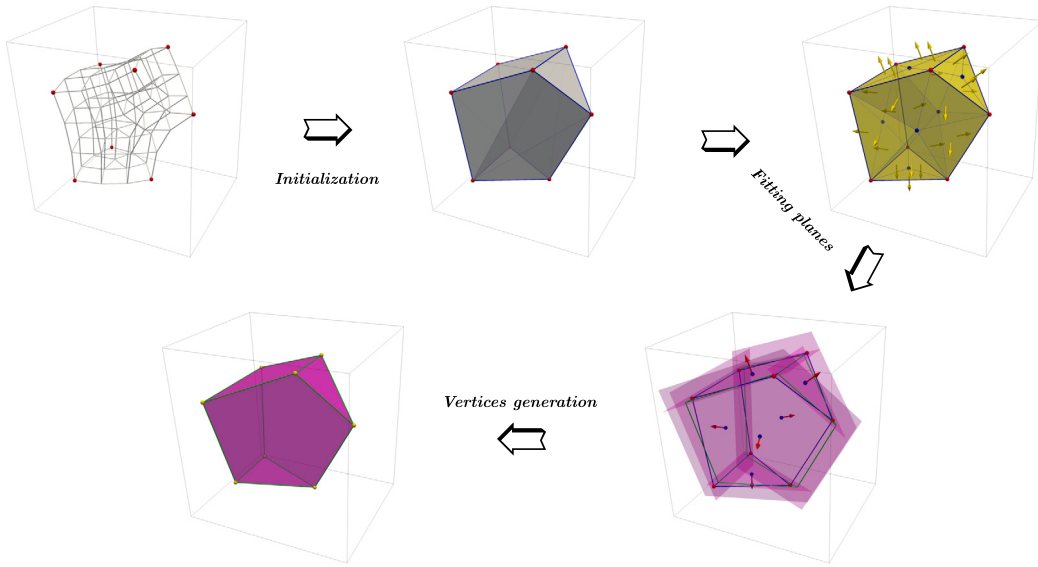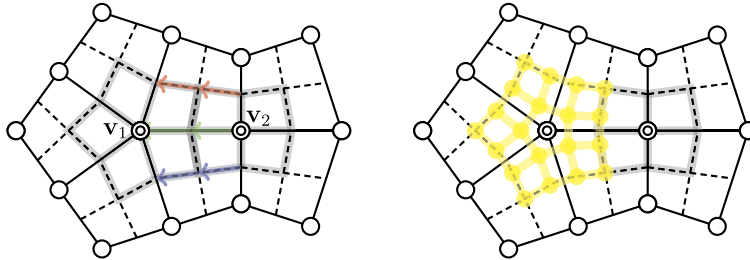
Fig. A.27. Domain generation process.

Fig. B.28. Construct Bézier control nets from quadmeshes. Solid black wireframe: input quadmesh $\mathcal{M}$. Dashed black wireframe: refined mesh $\mathcal{M}'$. Left: Quadratic Bézier control nets in Zheng et al. (2005) (gray wireframe). Right: Degree elevated polygonal cubic Bézier control net (yellow wireframe).

We should mention that, in certain extreme cases, the vertices we obtain may not precisely form a convex polyhedron, causing our algorithm to fail. Additionally, previous work (Várady and Salvi, 2020) on polygonal surfaces suggests that the generated domain polygon should closely mimic the input boundary to minimize parameterization distortion. Nevertheless, our domain polyhedron generation algorithm relies solely on the information of corner points, which may result in the generated domain polyhedron not closely mimicking the input configuration.

Appendix B. Polyhedral Bézier extraction

Here we provide an effective algorithm for polyhedral Bézier control nets construction from input hexmeshes or simple-convex-polyhedral meshes, which is a generalization of the patch generation algorithm proposed by Zheng et al. (2005).

First, let us review the patch generation algorithm in Zheng et al. (2005). Given a 2-manifold quadmesh $\mathcal{M}$ with internal vertices $\{\mathbf{v}_i\}$, a new refined mesh $\mathcal{M}'$ can be created by carrying out midpoint subdivision once. If the input 2-manifold mesh includes polygonal faces, first applying Catmull–Clark subdivision once or more times will cause all polygonal faces to disappear. Obviously, vertices $\{\mathbf{v}_i\}$ also lie on the refined mesh $\mathcal{M}'$. Each internal vertex $\mathbf{v}_i$ and its 1-ring neighboring vertices in $\mathcal{M}'$ form a quadratic Bézier control net then adjacent quadratic Bézier control nets naturally satisfy C^1 condition (see the left of Fig. B.28). Zheng et al. construct a Zheng–Ball patch for each irregular vertex and a tensor-product Bézier patch for each regular vertex.[4]

But these quadratic polygonal Bézier control nets are not what we desire. Because for GB patches to satisfy the G^1 condition, each side requires at least two layers of control points, meaning at least cubic grids. Indeed, we can easily obtain the required polygonal cubic Bézier control nets by the degree elevation algorithm proposed by Várady et al. (2016) (see the right of Fig. B.28). Although the geometry of the surface has changed due to the degree elevation, its boundary information has been preserved. Thus we construct quadratic Bézier patches at internal regular vertices and cubic GB patches at internal irregular vertices, resulting in a globally G^1 smooth surface.

---

[4] Zheng et al. did consider the case of boundary vertices; however, for the sake of convenience, we will not consider it here.
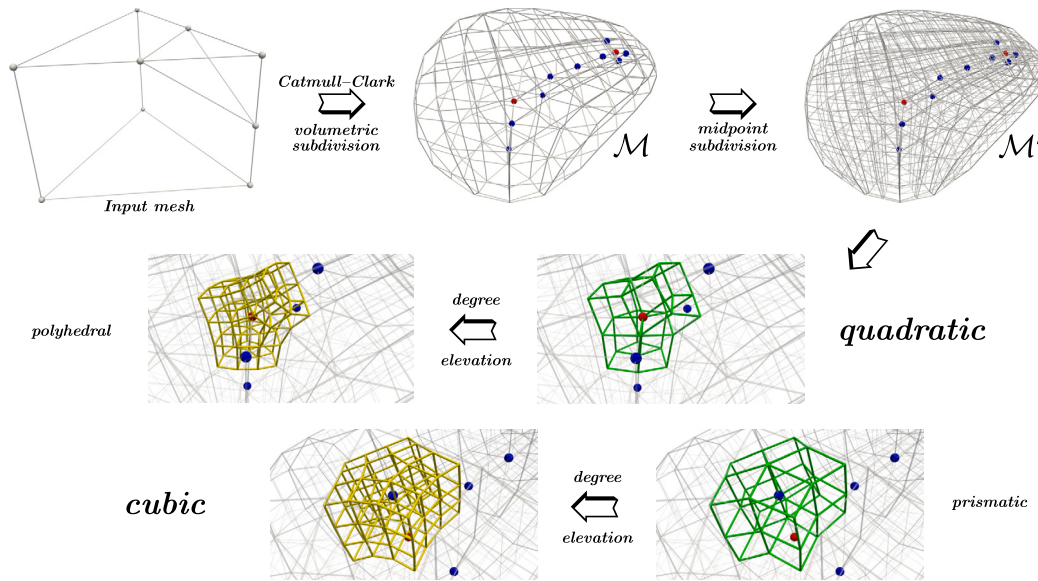
Fig. B.29. Polyhedral Bézier extraction. The input mesh is a simple-convex-polyhedral mesh comprising a 345-hexahedron and a tetrahedron. After applying Catmull–Clark volumetric subdivision twice, a pure hexmesh $\mathcal{M}$ is generated, and the original 345-hexahedron and tetrahedron are transformed into internal irregular vertices (depicted as red points). Subsequently, a refined mesh $\mathcal{M}'$ is generated through a single midpoint subdivision. Each internal irregular or semi-regular vertex (shown in blue) of $\mathcal{M}$ and its 1-ring neighboring vertices in $\mathcal{M}'$ constitute a quadratic polyhedral or prismatic Bézier control net (represented by green wireframes). By performing degree elevation, the desired cubic grids (represented by yellow wireframes) can be created.

This algorithm can be easily extended to higher dimensions. Given an initial hexmesh or simple-convex-polyhedral mesh, applying Catmull–Clark volumetric subdivision (MacCracken and Joy, 1996) once or more times will produce a pure hexmesh $\mathcal{M}$ with internal vertices $\{\mathbf{v}_i\}$. Although the original work (MacCracken and Joy, 1996) only carried out Catmull–Clark volumetric subdivision on hexmeshes, we find that it could be suitable for simple-convex-polyhedral meshes. Bajaj et al. (2002) also pointed out that the subdivision scheme for hexmeshes could be applied for such polyhedral meshes. They suggested constructing simple-convex-polyhedral meshes using 3D Voronoi tessellation because, in general, the cells of a 3D Voronoi diagram are simple convex polyhedra. Similarly, by performing midpoint subdivision once on $\mathcal{M}$, we can obtain a refined mesh $\mathcal{M}'$ then each internal vertex $\mathbf{v}_i$ and its 1-ring neighboring vertices in $\mathcal{M}'$ form a quadratic Bézier control net. There exists a polyhedral quadratic grid for each internal irregular vertex and a prismatic quadratic grid for each internal semi-regular vertex.[5] By degree elevation these polyhedral or prismatic quadratic Bézier control net will be cubic. Hence we can generate a globally G^1 smooth volume consisting of quadratic tensor-product Bézier, cubic tensor-product GB, and cubic GB volumes. Fig. B.29 shows the process of polyhedral Bézier extraction.

References

Ahrens, James P., Geveci, Berk, Law, C. Charles, 2005. ParaView: an end-user tool for large-data visualization. In: Hansen, Charles D., Johnson, Christopher R. (Eds.), The Visualization Handbook. Academic Press / Elsevier, pp. 717–731.

Bajaj, Chandrajit L., Schaefer, Scott, Warren, Joe D., Xu, Guoliang, 2002. A subdivision scheme for hexahedral meshes. Vis. Comput. 18, 343–356.

Bézier, P., 1978. General distortion of an ensemble of biparametric surfaces. Comput. Aided Des. 10 (2), 116–120.

Bunge, Astrid, Herholz, Philipp, Sorkine-Hornung, Olga, Botsch, Mario, Kazhdan, Michael, 2022. Variational quadratic shape functions for polygons and polyhedra. ACM Trans. Graph. 41 (4).

Chang, Qingjun, Deng, Chongyang, Hormann, Kai, 2023. Maximum likelihood coordinates. Comput. Graph. Forum 42 (5), e14908.

Charrot, Peter, Gregory, John A., 1984. A pentagonal surface patch for computer aided geometric design. Comput. Aided Geom. Des. 1 (1), 87–94.

Cohen, E., Riesenfeld, R.F., Elber, G., 2001. Geometric Modeling with Splines: An Introduction, 1st. edition. A K Peters/CRC Press, New York, NY.

Coons, S.A., 1967. Surfaces for computer-aided design of space forms. Technical report, USA.

Floater, Michael S., Kós, Géza, Reimers, Martin, 2005. Mean value coordinates in 3D. Comput. Aided Geom. Des. 22 (7), 623–631.

Floater, Michael S., 2015. Generalized barycentric coordinates and applications. Acta Numer. 24, 161–214.

Goldman, Ron, 2004. Multisided arrays of control points for multisided Bézier patches. Comput. Aided Geom. Des. 21 (3), 243–261.

Gregory, John A., 1985. Interpolation to boundary data on the simplex. Comput. Aided Geom. Des. 2 (1), 43–52.

Guo, Hao-Xiang, Liu, Xiaohan, Yan, Dong-Ming, Liu, Yang, 2020. Cut-enhanced PolyCube-Maps for feature-aware all-hex meshing. ACM Trans. Graph. 39 (4).

Haberleitner, Michael, Jüttler, Bert, Masson, Yannick, 2019. Isogeometric segmentation via midpoint subdivision suitable solids. Comput. Aided Des. 114, 179–190.

Hettinga, Gerben J., Kosinka, Jiří, 2020. A multisided C2 B-spline patch over extraordinary vertices in quadrilateral meshes. Comput. Aided Des. 127, 102855.

Hoschek, Josef, Lasser, Dieter, Schumaker, Larry L., 1993. Fundamentals of Computer Aided Geometric Design. A. K. Peters, Ltd., USA.

Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y., 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. 194 (39), 4135–4195.

Joshi, Pushkar, Meyer, Mark, DeRose, Tony, Green, Brian, Sanocki, Tom, 2007. Harmonic coordinates for character articulation. ACM Trans. Graph. 26 (3) 71–es.

---

[5] Please refer to Peters (2020) for detailed definitions of irregular and semi-regular vertices.

Kremer, Michael, Bommes, David, Kobbelt, Leif, 2013. Openvolumemesh – a versatile index-based data structure for 3d polytopal complexes. In: Jiao, Xiangmin, Weill, Jean-Christophe (Eds.), Proceedings of the 21st International Meshing Roundtable. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 531–548.

Lasser, Dieter, 1985. Bernstein-Bézier representation of volumes. Comput. Aided Geom. Des. 2 (1), 145–149.

Lasser, Dieter, 1987. Bernstein Bézier-Darstellung trivariater Splines. PhD thesis, Darmstadt.

MacCracken, Ron, Joy, Kenneth I., 1996. Free-form deformations with lattices of arbitrary topology. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'96. Association for Computing Machinery, New York, NY, USA, pp. 181–188.

Massarwi, Fady, Elber, Gershon, 2016. A B-spline based framework for volumetric object modeling. Comput. Aided Des. 78, 36–47. SPM 2016.

Muller, D.E., Preparata, F.P., 1978. Finding the intersection of two convex polyhedra. Theor. Comput. Sci. 7 (2), 217–236.

Nieser, Matthias, Reitebuch, Ulrich, Polthier, Konrad, 2011. CubeCover– parameterization of 3D volumes. Comput. Graph. Forum 30.

Paolini, Alexander, Kollmannsberger, Stefan, Rank, Ernst, 2019. Additive manufacturing in construction: a review on processes, applications, and digital planning methods. Addit. Manuf. 30, 100894.

Peters, Jörg, 2019. Splines for meshes with irregularities. SMAI J. Comput. Math. S5, 161–183.

Peters, Jörg, 2020. Refinable tri-variate C1 splines for box-complexes including irregular points and irregular edges. Comput. Aided Geom. Des. 80, 101877.

Qin, Kaikai, Li, Yajuan, Deng, Chongyang, 2023. Blending Bézier patch for multi-sided surface modeling. Comput. Aided Geom. Des. 105, 102222.

Randrianarivony, Maharavo, 2011. On transfinite interpolations with respect to convex domains. Comput. Aided Geom. Des. 28 (2), 135–149.

Reif, Ulrich, Sabin, Malcolm A., 2019. Old problems and new challenges in subdivision. J. Comput. Appl. Math. 349, 523–531.

Salvi, Péter, 2022. A circular parameterization for multi-sided patches. In: Proceedings of the Tenth Hungarian Conference on Computer Graphics and Geometry. NJSZT, pp. 22–26.

Salvi, Péter, Várady, Tamás, 2018. Multi-sided Bézier surfaces over concave polygonal domains. Comput. Graph. 74, 56–65.

Schneider, Teseo, Dumas, Jérémie, Gao, Xifeng, Botsch, Mario, Panozzo, Daniele, Zorin, Denis, 2019. Poly-spline finite-element method. ACM Trans. Graph. 38 (3).

Várady, Tamás, Rockwood, Alyn, Salvi, Péter, 2011. Transfinite surface interpolation over irregular n-sided domains. In: Solid and Physical Modeling 2011. Comput. Aided Des. 43 (11), 1330–1340.

Várady, Tamás, Salvi, Péter, Karikó, György, 2016. A multi-sided Bézier patch with a simple control structure. Comput. Graph. Forum 35 (2), 307–317.

Várady, Tamás, Salvi, Péter, 2020, Vaitkus, Márton, Sipos, Ágoston Multi-sided Bézier surfaces over curved, multi-connected domains. Comput. Aided Geom. Des. 78, 101828.

Várady, Tamás, Salvi, Péter, Vaitkus, Márton, 2024. Genuine multi-sided parametric surface patches – a survey. Comput. Aided Geom. Des. 110, 102286.

Wachspress, Eugene L., 1975. A Rational Finite Element Basis. Springer, Berlin.

Wang, Mengyun, Ji, Ye, Zhu, Chungang, 2023. Degree elevation and knot insertion for generalized Bézier surfaces and their application to isogeometric analysis. J. Comput. Math.

Warren, Joe D., Schaefer, Scott, Hirani, Anil N., Desbrun, Mathieu, 2007. Barycentric coordinates for convex sets. Adv. Comput. Math. 27, 319–338.

Wei, Xiaodong, Zhang, Yongjie Jessica, Toshniwal, Deepesh, Speleers, Hendrik, Li, Xin, Manni, Carla, Evans, John A., Hughes, Thomas J.R., 2018. Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis. Comput. Methods Appl. Mech. Eng. 341, 609–639.

Weiler, Kevin, 1985. Edge-based data structures for solid modeling in curved-surface environments. IEEE Comput. Graph. Appl. 5 (1), 21–40.

Zheng, Jin Jin, Zhang, Jian J., Zhou, Hong Jun, Shen, L.G., 2005. Smooth spline surface generation over meshes of irregular topology. Vis. Comput. 21 (8–10), 858–864.