



Exact conversion from Bézier tetrahedra to Bézier hexahedra

Gang Xu, Yaoli Jin, Zhoufang Xiao, Qing Wu, Bernard Mourrain, Timon Rabczuk

► To cite this version:

Gang Xu, Yaoli Jin, Zhoufang Xiao, Qing Wu, Bernard Mourrain, et al.. Exact conversion from Bézier tetrahedra to Bézier hexahedra. Computer Aided Geometric Design, 2018, 62, pp.154 - 165. 10.1016/j.cagd.2018.03.022 . hal-01936167

HAL Id: hal-01936167

<https://inria.hal.science/hal-01936167>

Submitted on 27 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exact conversion from Bézier tetrahedra to Bézier hexahedra

Gang Xu^{1,2}, Yaoli Jin¹, Zhoufang Xiao¹, Qing Wu¹, Bernard Mourrain³, Timon Rabczuk⁴

¹ School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

² Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou 310018, China

³ UCA, INRIA Sophia Antipolis Méditerranée, Aromath, 2004 Route des Lucioles, 06902 Sophia Antipolis Cedex, France

⁴ Institute of Structural Mechanics, Bauhaus-University Weimar, Marienstr. 15, D-99423 Weimar, Germany

Abstract

Modeling and computing of trivariate parametric volumes is an important research topic in the field of three-dimensional isogeometric analysis. In this paper, we propose two kinds of exact conversion approaches from Bézier tetrahedra to Bézier hexahedra with the same degree by reparametrization technique. In the first method, a Bézier tetrahedron is converted into a degenerate Bézier hexahedron, and in the second approach, a non-degenerate Bézier tetrahedron is converted into four non-degenerate Bézier hexahedra. For the proposed methods, explicit formulas are given to compute the control points of the resulting tensor-product Bézier hexahedra. Furthermore, in the second method, we prove that tetrahedral spline solids with C^k -continuity can be converted into a set of tensor-product Bézier volumes with G^k -continuity. The proposed methods can be used for the volumetric data exchange problems between different trivariate spline representations in CAD/CAE. Several experimental results are presented to show the effectiveness of the proposed methods.

1 Introduction

Computer Aided Design and Computer Aided Engineering (CAD/CAE for short) play important roles in advanced manufacturing. A main challenge in this area is the seamless integration of CAD/CAE techniques, numeric simulation and shape optimization. Unified mathematical representation of CAD/CAE geometry is one of the ten CAD challenges from the viewpoint of CAD researchers. Isogeometric analysis (IGA for short) approach originally proposed by Hughes et al. [14] provides a possibility to overcome the gap between CAD and finite element analysis by using the same NURBS representation for all design and analysis tasks.

In three-dimensional isogeometric analysis, trivariate spline modeling plays a key role as a geometric foundation for numerical solving. Trivariate NURBS solids and trivariate T-spline solids have been used as modeling and numerical tools in isogeometric modeling and solving. However, because of their tensor-product structure, trivariate NURBS and T-spline solids have some limitations on the construction of analysis-suitable volumetric parameterizations from arbitrary complex geometries. Recently, some kinds of geometry models consisting of Bézier triangles and Bézier tetrahedra [15, 25, 7, 28], which can represent complicated geometry with arbitrary topology, are employed in isogeometric analysis to overcome the limitations of trivariate NURBS and T-splines. On the other hand, since Bézier triangles and Bézier tetrahedra are not the standard forms in CAD/CAE systems, some data exchange operations have to be performed to make the geometry and field data compatible. Data exchange between different kinds of curves and surfaces has been studied in some previous works [5, 12, 13]. However, there are few works on the exact conversion between different kinds of trivariate solid representations.

Motivated by the issue on volumetric data exchange, the exact conversion approach from Bézier tetrahedra to Bézier hexahedra is investigated in this paper. Our main contributions can be summarized as follows:

- Two kinds of exact conversion approach from Bézier tetrahedra to Bézier hexahedra with the same degree are proposed based on reparametrization techniques. In the first method, a Bézier tetrahedron is converted into a degenerate Bézier hexahedron, and in the second approach, a non-degenerate Bézier tetrahedron is converted into four non-degenerate Bézier hexahedra.
- Explicit formula is given to compute the control points of the resulting tensor-product Bézier hexahedra, and the properties of the exact conversion approach are also described.

The rest of the paper is structured as follows. Related works are reviewed in Section 2. Preliminary knowledges of tetrahedral Bézier volume and tensor-product Bézier volumes are given in Section 3. The exact conversion method from a Bézier tetrahedron to a degenerate Bézier hexahedron is described in 4. Then the exact conversion from a non-degenerate Bézier tetrahedron to four non-degenerate Bézier hexahedra is studied in Section 5. To demonstrate the effectiveness of the proposed method, the experimental results are also presented in Section 4 and Section 5. Finally, we conclude this paper and outline future work in Section 6.

2 Related work

In this section, some related works on conversion between Bézier forms, volumetric modeling and tetrahedral splines in isogeometric analysis will be reviewed.

Conversion between Bézier forms In CAD systems, data exchange between different CAD formats plays an important role. A method of converting a triangular B-B surface into a trimmed patch of a tensor-product Bézier surface is proposed in [3]. Goldman and Filip proposed an explicit formula from a tensor-product Bézier surface of degree $m \times n$ into two triangular B-B surfaces of degree $m + n$ [9]. DeRose et al. [5] proposed a conversion method between different Bézier forms by functional composition algorithm with blossoming technique [6, 18]. Exact conversions from triangular B-B surfaces to tensor-product Bézier surfaces with the same degree are investigated in [12, 13].

Volumetric Modeling in IGA For three-dimensional isogeometric analysis, volumetric modeling from input boundary is a key issue to construct the parameterization of computational domain [29, 30]. Martin et al. [19] proposed a B-spline volume fitting method for triangular meshes based on volumetric harmonic functions. M. Aigner et al proposed a variational approach for

constructing NURBS parameterizations of swept volumes [1]. A method is proposed to construct trivariate T-spline parameterizations by tetrahedral meshing and mesh untangling techniques [8]. Xu et al. proposed a constrained optimization approach for analysis-suitable volume parameterizations [33]. Pettersen and Skytt proposed the spline volume faring method to achieve high-quality volume parameterization [22]. An efficient approach to construct injective solid T-splines for genus-zero geometries from a boundary triangulation is proposed in [36]. In [34], A variational harmonic method is proposed to construct analysis-suitable parameterizations of computational domain . Constructing conformal solid T-splines from boundary T-spline representations is investigated by using octree structure and boundary offset operation [37]. Wang and Qian proposed an efficient method by combining divide-and-conquer, constraint aggregation and the hierarchical optimization techniques to construct valid trivariate B-spline solids from six boundary B-spline surfaces [27]. Analysis-suitable trivariate NURBS representations of composite panels is constructed with a new curve/surface offset algorithm [21]. Xu et al. proposed a two-stages scheme to construct analysis-suitable volumetric parameterization by a uniformity-improved boundary reparameterization approach [35]. A volumetric parameterization method with PHT-splines from the level-set boundary representation is proposed in [4]. For a set of models with similar semantic features, B-spline based consistent volumetric parameterization from a given template domain is proposed in [31].

Tetrahedral splines in IGA Compared with the tensor-product splines, tetrahedral splines can represent complex geometry with arbitrary topology. Alfeld derived a C^1 tetrahedral Clough-Tocher spline which yields an interpolant that is piecewise quintic on each tetrahedron [2]. A C^1 quadratic trivariate macro-element space defined over arbitrary tetrahedral partitions is studied in [23]. Hecklin et al. described an algorithm for constructing a Lagrange interpolation pair based on C^1 cubic splines defined on tetrahedral partitions [11]. Speleers [24] proposed a method of constructing a normalized basis for the multivariate quadratic spline space defined over a generalized Powell-Sabin refinement of a triangulation in \mathbb{R}^s , $s \geq 1$. Lai and Matt proposed an approach to construct a C^1 trivariate macro-element based on the Alfeld split of tetrahedra [16]. To overcome the limitations of trivariate NURBS and T-splines, Engvall and Evans proposed a new IGA framework by using unstructured Bernstein-Bézier discretizations and trivariate mixed-elements [7]. Given a NURBS represented geometry, Xia and Qian proposed a general framework for automatic volumetric parameterization with Bézier tetrahedral partition [28]. Furthermore, optimal convergence in C^r spaces can be achieved and complicated geometries can be handled. To achieve volumetric data-exchange between different CAD formats, in this paper we propose two kinds of exact conversion approaches from Bézier tetrahedra to Bézier hexahedra.

3 Tetrahedral Bézier volume and tensor-product Bézier volume

3.1 Tensor-product Bézier volume

A tensor-product trivariate Bézier volume of degree $n \times n \times n$ with control points $P_{i,j,k}$ is defined by

$$P(r, s, t) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n P_{i,j,k} B_i^n(r) B_j^n(s) B_k^n(t) \quad (1)$$

where

$$B_i^n(r) = \binom{n}{i} r^i (1-r)^{n-i}, B_j^n(s) = \binom{n}{j} s^j (1-s)^{n-j}, B_k^n(t) = \binom{n}{k} t^k (1-t)^{n-k}$$

with

$$0 \leq r \leq 1, 0 \leq s \leq 1, 0 \leq t \leq 1.$$

3.2 Tetrahedral Bézier volume

A tetrahedral Bézier volume of degree n with control points $T_{i,j,k,l}$ is defined by

$$T(u, v, w, h) = \sum_{i+j+k+l=n} T_{i,j,k,l} B_{i,j,k,l}^n(u, v, w, h) \quad (2)$$

in which

$$B_{i,j,k,l}^n(u, v, w, h) = \binom{n}{i,j,k,l} u^i v^j w^k h^l,$$

and

$$u + v + w + h = 1, 0 \leq u, v, w, h \leq 1.$$

Since $u + v + w + h = 1$, we can rewrite the definition of a tetrahedral Bézier volume as follows

$$T(u, v, w) = \sum_{i+j+k=0}^n T_{i,j,k,n-i-j-k} B_{i,j,k,n-i-j-k}^n(u, v, w, 1-u-v-w) \quad (3)$$

Similar with the triangular Bézier surface [12, 13], we introduce some operators for tetrahedral Bézier volume as follows:

- The invariant operator $I : I T_{i,j,k} = T_{i,j,k}$.
- The shifting operator $E_i : E_1 T_{i,j,k} = T_{i+1,j,k}, E_2 T_{i,j,k} = T_{i,j+1,k}, E_3 T_{i,j,k} = T_{i,j,k+1}$.
- The difference operator $\Delta_i = E_i - I : \Delta_1 T_{i,j,k} = T_{i+1,j,k} - T_{i,j,k}, \Delta_2 T_{i,j,k} = T_{i,j+1,k} - T_{i,j,k}, \Delta_3 T_{i,j,k} = T_{i,j,k+1} - T_{i,j,k}$.

Using the above operators, the Bézier tetrahedron in (3) can be represented by

$$T(u, v, w) = (I + u\Delta_1 + v\Delta_2 + w\Delta_3)^n T_{000} \quad (4)$$

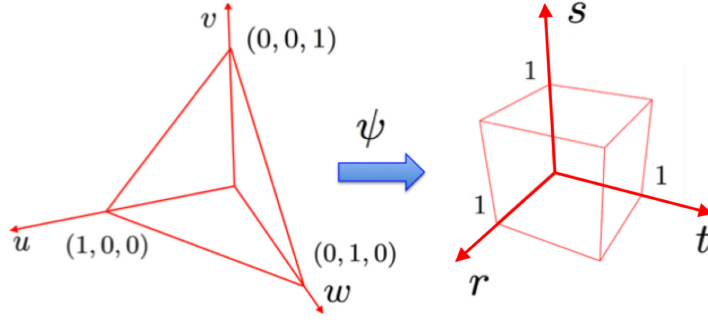


Figure 1: Parameter transformation ψ .

4 Converting a Bézier tetrahedron into a degenerate Bézier hexahedron

In this section, we will present the exact conversion method from a Bézier tetrahedron into a degenerate Bézier hexahedron.

4.1 Conversion method

Theorem 1. A Bézier tetrahedron $T(u, v, w, h)$ of degree n can be converted into a degenerate Bézier hexahedron of degree $n \times n \times n$, in which the control points $P_{i,j,k}$ of the resulting Bézier hexahedron can be computed as follows

$$\begin{bmatrix} P_{i,0,k} \\ P_{i,1,k} \\ \vdots \\ P_{i,n,k} \end{bmatrix} = A_n A_{n-1} \cdots A_{n-i+1} \begin{bmatrix} Q_{i,0,k} \\ Q_{i,1,k} \\ \vdots \\ Q_{i,n-i,k} \end{bmatrix}, \quad i = 1, \dots, n, \quad k = 0, \dots, n, \quad (5)$$

where

$$\begin{bmatrix} Q_{i,j,0} \\ Q_{i,j,1} \\ \vdots \\ Q_{i,j,n} \end{bmatrix} = A_n A_{n-1} \cdots A_{n-i-j+1} \begin{bmatrix} T_{i,j,0} \\ T_{i,j,1} \\ \vdots \\ T_{i,j,n-i-j} \end{bmatrix}, \quad i + j = 1, \dots, n, \quad (6)$$

and A_{n-m} ($m = 0, 1, \dots, n-1$) is defined as follows

$$A_{n-m} = \begin{bmatrix} \frac{1}{m+1} & \frac{0}{m+1} & 0 & \cdots & 0 & 0 \\ \frac{m}{m+1} & \frac{m+1}{m+1} & 0 & \cdots & 0 & 0 \\ 0 & \frac{2}{m+1} & \frac{m-1}{m+1} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{m}{m+1} & \frac{1}{m+1} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}_{(m+2) \times (m+1)}. \quad (7)$$

Proof. The basis function $B_{i,j,k,l}^n(u, v, w, h)$ of $T(u, v, w, h)$ in (3) can be rewritten as follows

$$\begin{aligned} B_{i,j,k,l}^n(u, v, w, h) &= \binom{n}{i, j, k, l} u^i v^j w^k h^l = \binom{n}{i, j, k, l} u^i v^j w^k h^l \frac{(1-u)^{n-i}}{(v+w+h)^{j+k+l}} \\ &= \binom{n}{i} u^i (1-u)^{n-i} \binom{n-i}{j, k, l} \left(\frac{v}{v+w+h} \right)^j \left(\frac{w}{v+w+h} \right)^k \left(\frac{h}{v+w+h} \right)^l \\ &= \binom{n}{i} u^i (1-u)^{n-i} \binom{n-i}{j, k, l} \left(\frac{v}{v+w+h} \right)^j \left(\frac{w}{v+w+h} \right)^k \left(\frac{h}{v+w+h} \right)^l \frac{\left(1 - \frac{v}{v+w+h} \right)^{n-i-j}}{\left(\frac{w+h}{v+w+h} \right)^{k+l}} \\ &= \binom{n}{i} u^i (1-u)^{n-i} \binom{n-i}{j} \left(\frac{v}{v+w+h} \right)^j \left(1 - \frac{v}{v+w+h} \right)^{n-i-j} \binom{n-i-j}{k} \left(\frac{w}{w+h} \right)^k \left(1 - \frac{w}{w+h} \right)^{n-i-j-k} \end{aligned} \quad (8)$$

Motivated from (8), we define the following parameter transformation

$$\psi : r = u, \quad s = \frac{v}{v+w+h}, \quad t = \frac{w}{w+h}. \quad (9)$$

which converts the tetrahedral domain $\{(u, v, w) | 0 \leq u, v, w, u+v+w \leq 1\}$ into the hexahedral domain $\{(r, s, t) | 0 \leq r \leq 1, 0 \leq s \leq 1, 0 \leq t \leq 1\}$ as shown in Figure 1.

Substituting (9) into (8), we have

$$B_{i,j,k,l}^n(u, v, w, h) = B_i^n(r) B_j^{n-i}(s) B_k^{n-i-j}(t)$$

Hence, from (3), we have

$$T(u, v, w, h) = \sum_{i=0}^n \sum_{j=0}^{n-i} \sum_{k=0}^{n-i-j} T_{i,j,k,n-i-j-k} B_i^n(r) B_j^{n-i}(s) B_k^{n-i-j}(t) \quad (10)$$

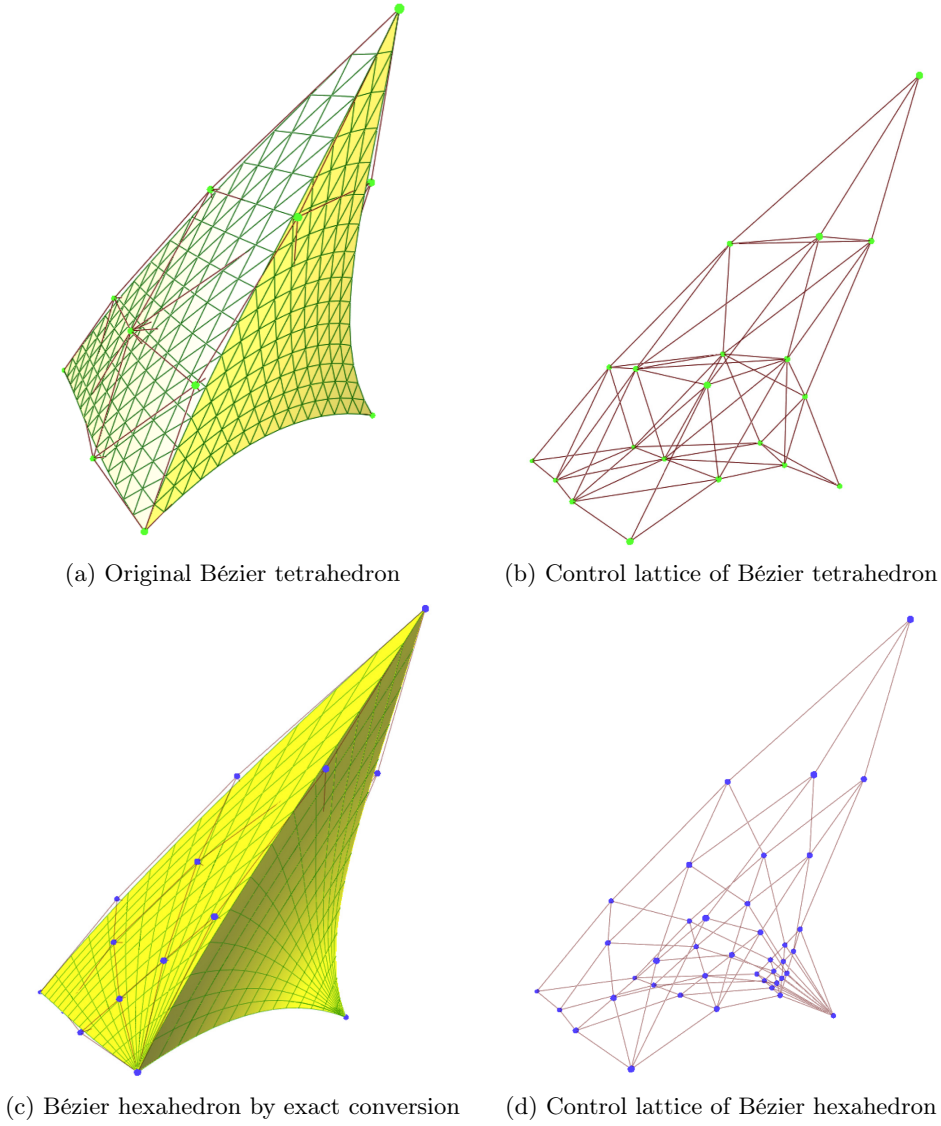


Figure 2: Exact conversion from a Bézier tetrahedron to a degenerate Bézier hexahedron.

In order to represent $T(u, v, w, h)$ in (10) as a Bézier hexahedron of degree $n \times n \times n$, we need to raise the degree of the following Bézier surfaces

$$R_i(s, t) = \sum_{j=0}^{n-i} \sum_{k=0}^{n-i-j} T_{i,j,k,n-i-j-k} B_j^{n-i}(s) B_k^{n-i-j}(t)$$

into degree $n \times n$. From the degree-elevation matrix for the tensor product Bézier surfaces, we can get the explicit formula for the control points of the Bézier hexahedron as shown in (5) and (6). This completes the proof. \square

Remark. Another proof of Theorem 1 can be given with the operator representation of $T(u, v, w)$ in (4). Firstly, the corresponding inverse transformation of ψ can be derived as follows

$$\psi^{-1} : u = r, \quad v = (1-r)s, \quad w = (1-r)(1-s)t, \quad (11)$$

then by substituting (11) into (4), a similar conversion result can be obtained.

From Theorem 1, a Bézier tetrahedron can be considered as a special case of Bézier hexahedron. Furthermore, we can obtain the sufficient and necessary condition for a Bézier hexahedron degenerating to a Bézier tetrahedron from Theorem 1.

Theorem 2. A Bézier hexahedron of degree $n \times n \times n$ can be degenerated into a Bézier tetrahedron of degree n if and only if the tensor-product Bézier surfaces of degree $n \times n$ determined by the control points on i -th column along two parametric directions (i.e., u -direction and v -direction) degenerate to triangular Bézier surfaces of degree $n-i$, $i = 1, 2, \dots, n$.

4.2 Examples

The proposed conversion method described in Theorem 1 can be considered as a corner-cutting process with explicit geometric meaning. That is, the control points of the Bézier hexahedron can be obtained by the corner-cutting operation on the control points of Bézier tetrahedron in v and w direction. In the following, we will present an example with $n = 3$ to show the conversion process.

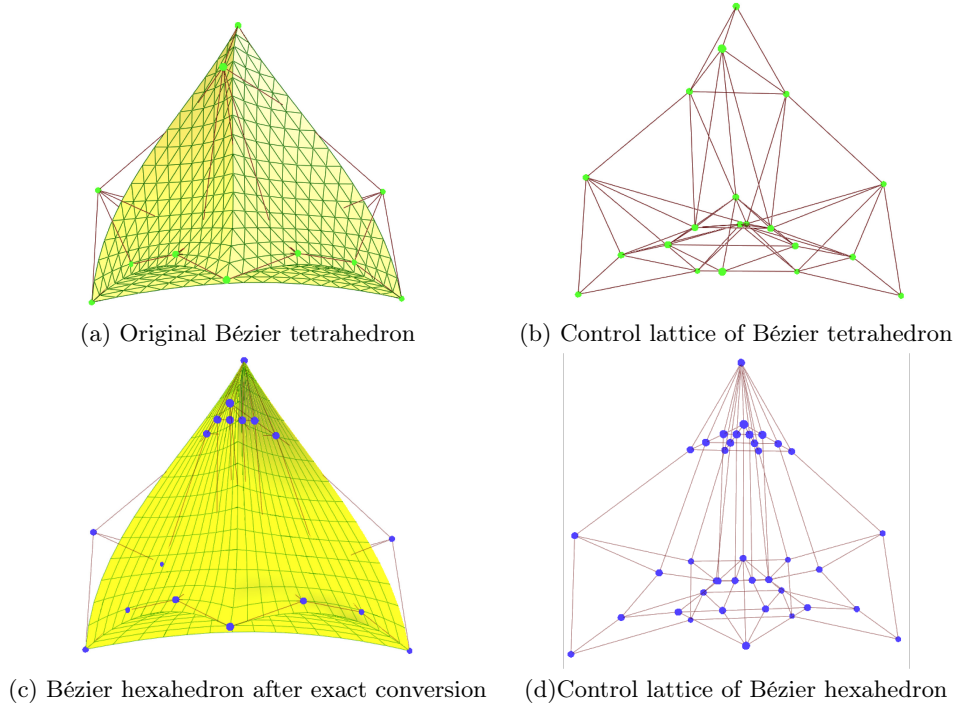


Figure 3: Exact conversion from a Bézier tetrahedron to a degenerate Bézier hexahedron.

For $n = 3$, we have

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & \frac{2}{3} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

By using the above formulas for $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$, we can compute the control points as follows

$$\begin{aligned} \begin{bmatrix} Q_{000} & Q_{001} & Q_{002} & Q_{003} \\ Q_{010} & Q_{011} & Q_{012} & Q_{013} \\ Q_{020} & Q_{021} & Q_{022} & Q_{023} \\ Q_{030} & Q_{031} & Q_{032} & Q_{033} \end{bmatrix} &= \begin{bmatrix} T_{000} \\ T_{001} \\ T_{002} \\ T_{003} \end{bmatrix} \mathbf{A}_1 \begin{bmatrix} T_{010} \\ T_{011} \\ T_{012} \end{bmatrix} \mathbf{A}_1 \mathbf{A}_2 \begin{bmatrix} T_{020} \\ T_{021} \end{bmatrix} \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 T_{030}^T, \\ \begin{bmatrix} P_{000} & P_{001} & P_{002} & P_{003} \\ P_{010} & P_{011} & P_{012} & P_{013} \\ P_{020} & P_{021} & P_{022} & P_{023} \\ P_{030} & P_{031} & P_{032} & P_{033} \end{bmatrix} &= \begin{bmatrix} Q_{000} & Q_{001} & Q_{002} & Q_{003} \\ Q_{010} & Q_{011} & Q_{012} & Q_{013} \\ Q_{020} & Q_{021} & Q_{022} & Q_{023} \\ Q_{030} & Q_{031} & Q_{032} & Q_{033} \end{bmatrix}, \\ \begin{bmatrix} Q_{100} & Q_{101} & Q_{102} & Q_{103} \\ Q_{110} & Q_{111} & Q_{112} & Q_{113} \\ Q_{120} & Q_{121} & Q_{122} & Q_{123} \end{bmatrix} &= \begin{bmatrix} T_{100} \\ T_{101} \\ T_{102} \end{bmatrix} \mathbf{A}_1 \mathbf{A}_2 \begin{bmatrix} T_{110} \\ T_{111} \end{bmatrix} \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 T_{120}^T, \\ \begin{bmatrix} P_{100} & P_{110} & P_{102} & P_{103} \\ P_{110} & P_{111} & P_{112} & P_{113} \\ P_{120} & P_{121} & P_{122} & P_{123} \\ P_{130} & P_{131} & P_{132} & P_{133} \end{bmatrix} &= \begin{bmatrix} Q_{100} \\ Q_{110} \\ Q_{120} \end{bmatrix} \mathbf{A}_1 \begin{bmatrix} Q_{101} \\ Q_{111} \\ Q_{121} \end{bmatrix} \mathbf{A}_1 \begin{bmatrix} Q_{102} \\ Q_{112} \\ Q_{122} \end{bmatrix} \mathbf{A}_1 \begin{bmatrix} Q_{103} \\ Q_{113} \\ Q_{123} \end{bmatrix}^T, \\ \begin{bmatrix} Q_{200} & Q_{201} & Q_{202} & Q_{203} \\ Q_{210} & Q_{211} & Q_{212} & Q_{213} \end{bmatrix} &= \begin{bmatrix} T_{200} \\ T_{201} \end{bmatrix} \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 T_{210}^T, \\ \begin{bmatrix} P_{200} & P_{201} & P_{202} & P_{203} \\ P_{210} & P_{211} & P_{212} & P_{213} \\ P_{220} & P_{221} & P_{222} & P_{223} \\ P_{230} & P_{231} & P_{232} & P_{233} \end{bmatrix} &= \begin{bmatrix} Q_{200} \\ Q_{210} \end{bmatrix} \mathbf{A}_1 \mathbf{A}_2 \begin{bmatrix} Q_{201} \\ Q_{211} \end{bmatrix} \mathbf{A}_1 \mathbf{A}_2 \begin{bmatrix} Q_{202} \\ Q_{212} \end{bmatrix} \mathbf{A}_1 \mathbf{A}_2 \begin{bmatrix} Q_{203} \\ Q_{213} \end{bmatrix}^T, \\ [Q_{300} & Q_{301} & Q_{302} & Q_{303}] &= [\mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 T_{300}]^T = [T_{300} \quad T_{300} \quad T_{300} \quad T_{300}], \\ \begin{bmatrix} P_{300} & P_{301} & P_{302} & P_{303} \\ P_{310} & P_{311} & P_{312} & P_{313} \\ P_{320} & P_{321} & P_{322} & P_{323} \\ P_{330} & P_{331} & P_{332} & P_{333} \end{bmatrix} &= [\mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 Q_{300} \quad \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 Q_{301} \quad \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 Q_{302} \quad \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 Q_{303}]^T \\ &= \begin{bmatrix} T_{300} & T_{300} & T_{300} & T_{300} \\ T_{300} & T_{300} & T_{300} & T_{300} \\ T_{300} & T_{300} & T_{300} & T_{300} \\ T_{300} & T_{300} & T_{300} & T_{300} \end{bmatrix}. \end{aligned}$$

Figure 2 and Figure 3 illustrate two examples for the exact conversion from a Bézier tetrahedron to a degenerate Bézier hexahedron.

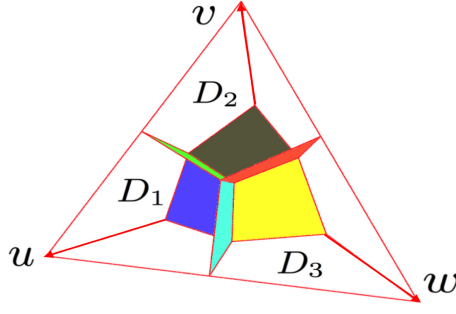


Figure 4: Subdivide a tetrahedron into four hexahedra.

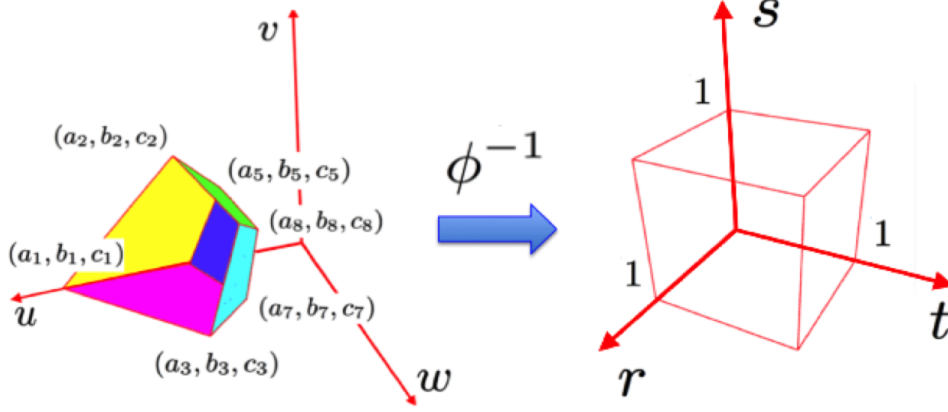


Figure 5: Parameter transformation ϕ .

5 Converting a Bézier tetrahedra into four Bézier hexahedras

In practice, degenerate representation has some limitations for geometric modeling and numerical analysis. Hence, it is more interesting to achieve exact data conversion without degenerate representation. In this section, we will propose an exact conversion method from a non-degenerate Bézier tetrahedron to non-degenerate four Bézier hexahedra.

5.1 Conversion method

Suppose that the non-degenerate Bézier tetrahedron $T(u, v, w)$ is defined on a domain $D = \{(u, v, w) | 0 \leq u, v, w, u + v + w \leq 1\}$ as shown in (3). The parametric domain D can be subdivided into four hexahedral sub-domains D_1, D_2, D_3 and D_4 as shown in Figure 4. The parametric surfaces with respect to the six planar faces with different colors in Figure 4 divide the Bézier tetrahedron $T(u, v, w)$ into four Bézier hexahedra.

In this section, we will present an explicit formula to compute the control points for each Bézier hexahedron.

Theorem 3. The trimmed volume of a Bézier tetrahedron $T(u, v, w)$ defined on the sub-domain D_i ($i = 1, 2, 3, 4$) with eight vertexes $(a_1, b_1, c_1), (a_7, b_7, c_7), \dots, (a_8, b_8, c_8)$ as shown in Figure 5, can be represented as a Bézier hexahedron with degree $n \times n \times n$, and its control points are determined as follows,

$$P_{i,j,k} = \sum_{x_1=0}^i \sum_{x_2=0}^{n-i} \frac{\binom{i}{x_1} \binom{n-i}{x_2}}{\binom{n}{j}} \sum_{y_1=0}^{i-x_1} \sum_{z_1=0}^{x_2} \sum_{y_2=0}^{n-i-x_2} \sum_{z_2=0}^{n-i-x_2} \frac{\binom{x_1}{y_1} \binom{i-x_1}{z_1} \binom{x_2}{y_2} \binom{n-i-x_2}{z_2}}{\binom{n}{k}} Q_{x_1, x_2, y_1, y_2, z_1, z_2}^i, \quad (12)$$

where

$$Q_{x_1, x_2, y_1, y_2, z_1, z_2}^i = (I + a_3\Delta_1 + b_3\Delta_2 + c_3\Delta_3)^{y_1} (I + a_4\Delta_1 + b_4\Delta_2 + c_4\Delta_3)^{x_1-y_1} \cdot (I + a_1\Delta_1 + b_1\Delta_2 + c_1\Delta_3)^{z_1} (I + a_2\Delta_1 + b_2\Delta_2 + c_2\Delta_3)^{i-x_1-z_1} \cdot (I + a_7\Delta_1 + b_7\Delta_2 + c_7\Delta_3)^{y_2} (I + a_8\Delta_1 + b_8\Delta_2 + c_8\Delta_3)^{x_2-y_2} \cdot (I + a_5\Delta_1 + b_5\Delta_2 + c_5\Delta_3)^{z_2} (I + a_6\Delta_1 + b_6\Delta_2 + c_6\Delta_3)^{n-i-x_2-z_2} T_{000} \quad (13)$$

Proof. Firstly, we construct the following parameter transformation ϕ

$$\begin{cases} u = r\{s[a_3t + a_4(1-t)] + (1-s)[a_1t + a_2(1-t)]\} + (1-r)\{s[a_7t + a_8(1-t)] + (1-s)[a_5t + a_6(1-t)]\} \\ v = r\{s[b_3t + b_4(1-t)] + (1-s)[b_1t + b_2(1-t)]\} + (1-r)\{s[b_7t + b_8(1-t)] + (1-s)[b_5t + b_6(1-t)]\} \\ w = r\{s[c_3t + c_4(1-t)] + (1-s)[c_1t + c_2(1-t)]\} + (1-r)\{s[c_7t + c_8(1-t)] + (1-s)[c_5t + c_6(1-t)]\} \end{cases} \quad (14)$$

The corresponding inverse transformation ϕ^{-1} transforms the sub-domain D_i ($i = 1, 2, 3, 4$) with eight vertexes $(a_1, b_1, c_1), (a_2, b_2, c_2), \dots, (a_8, b_8, c_8)$ into a unit cube $\{r, s, t | 0 \leq r, s, t \leq 1\}$ as shown in Figure 5.

Then by substituting (14) to (4), we have

$$\begin{aligned}
\mathbf{T}(u, v, w) &= (I + u\Delta_1 + v\Delta_2 + w\Delta_3)^n \mathbf{T}_{000} \\
&= \{I[(1-r) + r][(1-s) + s][(1-t) + t] + r\{s[a_3t + a_4(1-t)] + (1-s)[a_1t + a_2(1-t)]\}\Delta_1 \\
&\quad + (1-r)\{s[a_7t + a_8(1-t)] + (1-s)[a_5t + a_6(1-t)]\}\Delta_1 \\
&\quad + r\{s[b_3t + b_4(1-t)] + (1-s)[b_1t + b_2(1-t)]\}\Delta_2 \\
&\quad + (1-r)\{s[b_7t + b_8(1-t)] + (1-s)[b_5t + b_6(1-t)]\}\Delta_2 \\
&\quad + r\{s[c_3t + c_4(1-t)] + (1-s)[c_1t + c_2(1-t)]\}\Delta_3 \\
&\quad + (1-r)\{s[c_7t + c_8(1-t)] + (1-s)[c_5t + c_6(1-t)]\}\Delta_3\}^n \mathbf{T}_{000} \\
&= \{r\{s[t(I + a_3\Delta_1 + b_3\Delta_2 + c_3\Delta_3) + (1-t)(I + a_4\Delta_1 + b_4\Delta_2 + c_4\Delta_3)] \\
&\quad (1-s)[t(I + a_1\Delta_1 + b_1\Delta_2 + c_1\Delta_3) + (1-t)(I + a_2\Delta_1 + b_2\Delta_2 + c_2\Delta_3)]\} \\
&\quad + (1-r)\{s[t(I + a_7\Delta_1 + b_7\Delta_2 + c_7\Delta_3) + (1-t)(I + a_8\Delta_1 + b_8\Delta_2 + c_8\Delta_3)] \\
&\quad (1-s)[t(I + a_5\Delta_1 + b_5\Delta_2 + c_5\Delta_3) + (1-t)(I + a_6\Delta_1 + b_6\Delta_2 + c_6\Delta_3)]\}\}^n \mathbf{T}_{000} \\
&= \sum_{i=0}^n B_i^n(r) \sum_{x_1=0}^i \sum_{x_2=0}^{n-i} B_{x_1}^i(s) B_{x_2}^{n-i}(s) \sum_{y_1=0}^{x_1=0} \sum_{z_1=0}^{i-x_1} \sum_{y_2=0}^{x_2} \sum_{z_2=0}^{n-i-x_2} B_{y_1}^{x_1}(t) B_{z_1}^{i-x_1}(t) B_{y_2}^{x_2}(t) B_{z_2}^{n-i-x_2}(t) \\
&\quad \cdot (I + a_3\Delta_1 + b_3\Delta_2 + c_3\Delta_3)^{y_1} (I + a_4\Delta_1 + b_4\Delta_2 + c_4\Delta_3)^{x_1-y_1} \\
&\quad \cdot (I + a_1\Delta_1 + b_1\Delta_2 + c_1\Delta_3)^{z_1} (I + a_2\Delta_1 + b_2\Delta_2 + c_2\Delta_3)^{i-x_1-z_1} \\
&\quad \cdot (I + a_7\Delta_1 + b_7\Delta_2 + c_7\Delta_3)^{y_2} (I + a_8\Delta_1 + b_8\Delta_2 + c_8\Delta_3)^{x_2-y_2} \\
&\quad \cdot (I + a_5\Delta_1 + b_5\Delta_2 + c_5\Delta_3)^{z_2} (I + a_6\Delta_1 + b_6\Delta_2 + c_6\Delta_3)^{n-i-x_2-z_2} \mathbf{T}_{000} \\
&= \sum_{i=0}^n B_i^n(r) \sum_{x_1=0}^i \sum_{x_2=0}^{n-i} \frac{\binom{i}{x_1} \binom{n-i}{x_2}}{\binom{x_1+x_2}{x_1+x_2}} B_{x_1+x_2}^n(s) \sum_{y_1=0}^{x_1} \sum_{z_1=0}^{i-x_1} \sum_{y_2=0}^{x_2} \sum_{z_2=0}^{n-i-x_2} \\
&\quad \frac{\binom{x_1}{y_1} \binom{i-x_1}{z_1} \binom{x_2}{y_2} \binom{n-i-x_2}{z_2}}{\binom{y_1+z_1+y_2+z_2}{y_1+z_1+y_2+z_2}} B_{y_1+y_2+z_1+z_2}^n(t) \mathbf{Q}_{x_1, x_2, y_1, y_2, z_1, z_2}^i \\
&= \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n B_i^n(r) B_j^n(s) B_k^n(t) \mathbf{P}_{ijk},
\end{aligned}$$

in which \mathbf{P}_{ijk} and $\mathbf{Q}_{x_1, x_2, y_1, y_2, z_1, z_2}^i$ are defined in (12) and (13) respectively, and

$$x_1 + x_2 = j, \quad y_1 + y_2 + z_1 + z_2 = k.$$

This completes the proof. \square

From the following identity,

$$\binom{i}{x_1} = \binom{i-1}{x_1} + \binom{i-1}{x_1-1}, \quad (15)$$

we can prove that for an arbitrary sequence $\mathbf{I}_0, \mathbf{I}_1, \dots, \mathbf{I}_i$,

$$\sum_{x_1=0}^i \binom{i}{x_1} \mathbf{I}_{x_1} = \sum_{x_1=0}^{i-1} \binom{i-1}{x_1} \mathbf{I}_{x_1} + \sum_{x_1=0}^{i-1} \binom{i-1}{x_1-1} \mathbf{I}_{x_1+1}. \quad (16)$$

Suppose that $\mathbf{H}_{x_1, j-x_1}^{i, j, k}$ is defined as

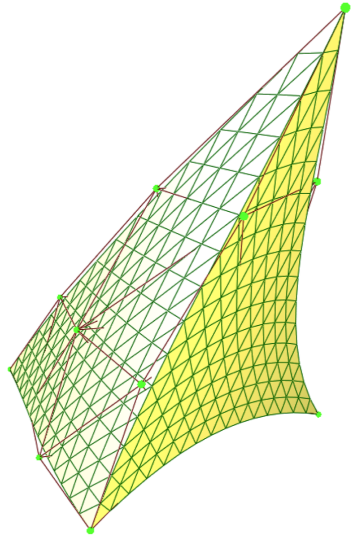
$$\mathbf{H}_{x_1, j-x_1}^{i, j, k} = \sum_{y_1=0}^{x_1} \sum_{z_1=0}^{i-x_1} \sum_{y_2=0}^{x_2} \sum_{z_2=0}^{n-i-x_2} \left[\binom{x_1}{y_1} \binom{i-x_1}{z_1} \binom{x_2}{y_2} \binom{n-i-x_2}{z_2} \right] \mathbf{Q}_{x_1, x_2, y_1, y_2, z_1, z_2}^i. \quad (17)$$

Then from (16), the control point \mathbf{P}_{ijk} in (12) can be rewritten as follows,

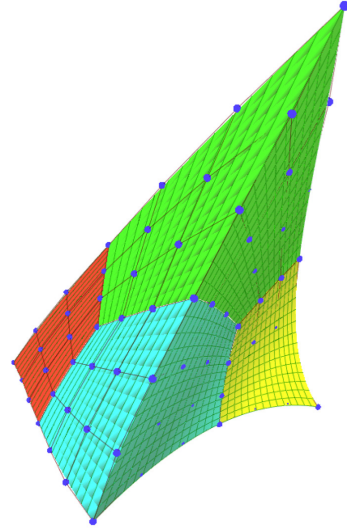
$$\begin{aligned}
\mathbf{P}_{i, j, k} &= \sum_{x_1=0}^i \binom{i}{x_1} \left[\binom{n-i}{j-x_1} \right] \mathbf{H}_{x_1, j-x_1}^{i, j, k} \\
&= \sum_{x_1=0}^{i-1} \binom{i-1}{x_1} \left[\binom{n-i}{j-x_1} \right] \mathbf{H}_{x_1, j-x_1}^{i, j, k} + \sum_{x_1=0}^{i-1} \binom{i-1}{x_1} \left[\binom{n-i}{j-(x_1+1)} \right] \mathbf{H}_{x_1+1, j-(x_1+1)}^{i, j, k} \\
&= \frac{n-j}{n} \sum_{x_1=0}^{i-1} \left[\binom{i-1}{x_1} \binom{n-i}{j-x_1} \right] \mathbf{H}_{x_1, j-x_1}^{i, j, k} + \frac{j}{n} \sum_{x_1=0}^{i-1} \left[\binom{i-1}{x_1} \binom{n-i}{j-(x_1+1)} \right] \mathbf{H}_{x_1+1, j-(x_1+1)}^{i, j, k}
\end{aligned}$$

That is, \mathbf{P}_{ijk} can be computed as a convex linear combination of $\mathbf{H}_{x_1, j-x_1}^{i, j, k}$ in a recursive way. Analogously, it can be proved that $\mathbf{H}_{x_1, j-x_1}^{i, j, k}$ can be also evaluated recursively. Furthermore, from (13), we can find that $\mathbf{Q}_{x_1, x_2, y_1, y_2, z_1, z_2}^i$ can be computed as a convex linear combination of \mathbf{T}_{ijk} in a similar recursive way as de Casteljau algorithm. Hence, the proposed conversion approach is stable and efficient.

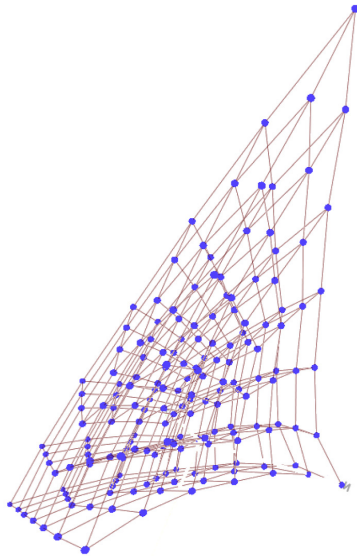
Starting from the same Bézier tetrahedron in Figure 2 and Figure 3, Figure 6 and Figure 7 present two examples for the exact conversion from a Bézier tetrahedron to four Bézier hexahedra.



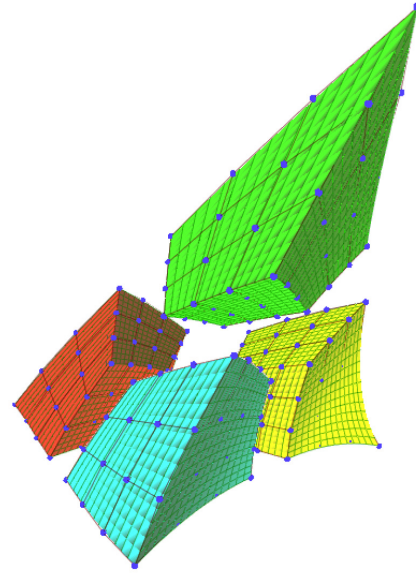
(a) Original Bézier tetrahedron



(b) Bézier hexahedra after exact conversion



(c) Control lattice of Bézier hexahedra



(d) Exploded view of four Bézier hexahedra

Figure 6: Exact conversion from a Bézier tetrahedron to four Bézier hexahedra.

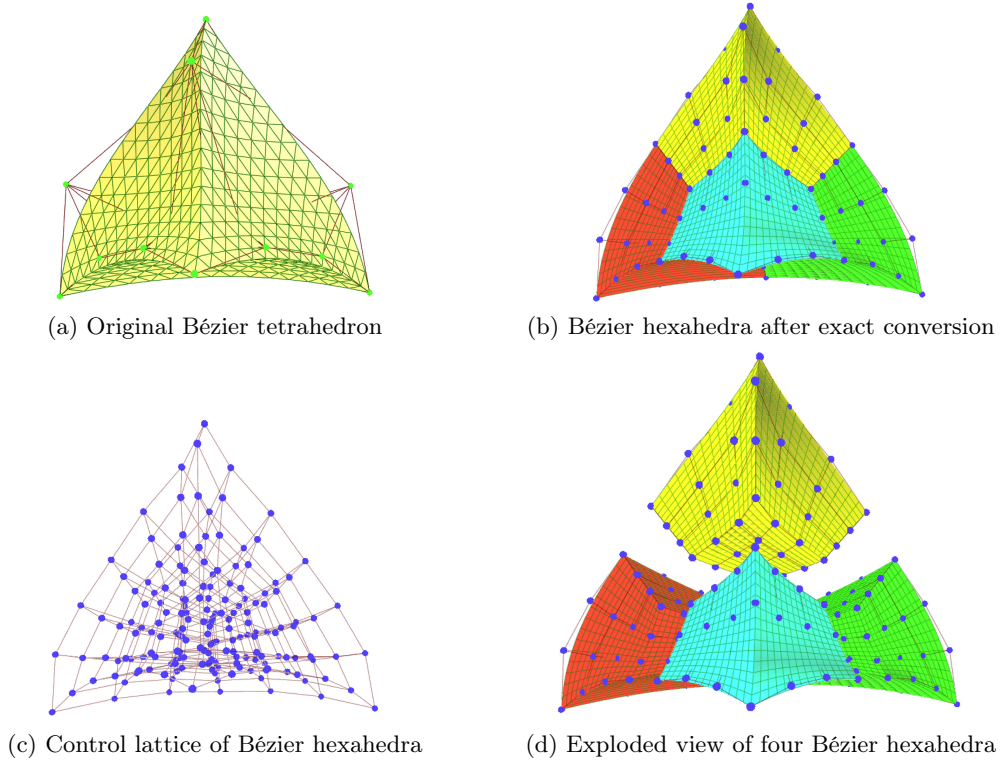


Figure 7: Exact conversion from a Bézier tetrahedron to four Bézier hexahedra.

5.2 Properties of the conversion method

From the parameter transformation in (14), we can obtain some interesting properties of the proposed conversion approach. Firstly, after some straightforward computation, we can prove that the Jacobian of the parameter transformation ϕ in (14) is non-zero. Hence, we have the following proposition,

Proposition 4. *If the original tetrahedral Bézier volume is non-degenerate, then the four tensor-product Bézier volumes obtained by the proposed conversion method are also non-degenerate.*

On the other hand, Bézier tetrahedra with continuity constraints have been used as powerful tools in isogeometric analysis [7, 28, 24] to represent complex geometry with arbitrary topology. By the proposed conversion method, we can convert the Bézier tetrahedra into a set of Bézier hexahedra, which are the standard data format in CAD systems. Furthermore, we have the following proposition for the smoothness between the resulting Bézier hexahedral elements,

Proposition 5. *A C^k tetrahedral spline solid $\mathcal{S}(u, v, w)$ with N tetrahedral Bézier volumes can be converted into an unstructured G^k trivariate spline representation $\mathbf{X}(r, s, t)$ with $4N$ tensor-product Bézier volumes.*

Proof. From Theorem 3, we can convert the trivariate spline volume $\mathcal{S}(u, v, w)$ into a set of tensor-product Bézier volumes by the parameter transformation ϕ_i for each sub-volume \mathbf{X}_i . Without loss of the generality, suppose that $\mathbf{X}_1 = \mathbf{S}_1 \circ \phi_1$, $\mathbf{X}_2 = \mathbf{S}_2 \circ \phi_2$, hence along the common interface \mathbf{P}_{12} between \mathbf{S}_1 and \mathbf{S}_2 , we have $\mathbf{X}_1 \circ \phi_1^{-1} = \mathbf{X}_2 \circ \phi_2^{-1}$. Furthermore, since \mathbf{S}_1 and \mathbf{S}_2 are C^k -continuous along their common interface \mathbf{P}_{12} , according to the definition of geometric-continuity in [20], \mathbf{X}_1 and \mathbf{X}_2 are G^k -continuous along the common interface. \square

6 Conclusion

In this paper, motivated by volume modeling in isogeometric analysis, we propose two kinds of exact conversion approach from a tetrahedral Bézier volume to tensor-product trivariate Bézier representation with the same degree. The explicit formula is given to compute the control points of the resulting tensor-product Bézier volumes. We also prove that if we apply this approach on each tetrahedral Bézier elements of the tetrahedral spline volume with C^k -continuity, then an unstructured set of tensor-product Bézier volumes with G^k -continuity can be achieved. Experimental results show the effectiveness of the proposed methods.

In the field of high-order mesh generation, a Bézier tetrahedron is usually used as a high-order element. As part of future work, we will investigate the application of the proposed methods (especially the second approach) in the field of high-order mesh generation.

Acknowledgment

This research was supported by the National Nature Science Foundation of China under Grant Nos. 61772163, 61761136010, 61472111, Zhejiang Provincial Natural Science Foundation of China under Grant Nos. LQ16F020005, LR16F020003.

References

- [1] M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon and A.-V. Vuong. Swept volume parametrization for isogeometric analysis. In E. Hancock and R. Martin (eds.), *The Mathematics of Surfaces (MoS XIII 2009)*, LNCS 5654(2009) 19-44.

- [2] P. Alfeld. A trivariate Clough-Tocher scheme for tetrahedral data. *Computer Aided Geometric Design*, 1(1984): 169-181
- [3] I. Brueckner. Construction of Bézier points of quadrilaterals from Bézier triangles. *Computer-Aided Design* 12(1980) 21-24.
- [4] C. Chan, C. Anitescu, T. Rabczuk. Volumetric parametrization from a level set boundary representation with PHT-splines. *Computer-Aided Design* 82 (2017) 29-41.
- [5] T.D. DeRose, R.N. Goldman, H. Hagen, S. Mann. Functional composition algorithms via blossoming. *ACM Transactions on Graphics* 12(1993) 113-135.
- [6] T.D. DeRose. Compositing Bézier simplexes. *ACM transactions on Graphics* 7(1998): 198-221.
- [7] Luke Engvall, John A. Evans. Isogeometric unstructured tetrahedral and mixed-element Bernstein-Bézier discretizations. *Computer Methods in Applied Mechanics and Engineering* 319(2017) 83-123
- [8] J.M. Escobara, J.M. Cascónb, E. Rodríguez, R. Montenegro. A new approach to solid modeling with trivariate T-spline based on mesh optimization. *Computer Methods in Applied Mechanics and Engineering* 200(2011) 3210-3222.
- [9] R.N. Goldman, D. Filip. Conversion from Bézier rectangles to Bézier triangles. *Computer-Aided Design*, 19(1): 25-28, 1987.
- [10] D. Groisser, J. Peters. Matched G^k -constructions always yield C^k -continuous isogeometric elements. *Computer Aided Geometric Design* 34(2015) 67-72.
- [11] G. Hecklin, G. Nürnberger, L. L. Schumaker, F. Zeilfelder. Local lagrange interpolation with cubic C^1 splines on tetrahedral partitions. *Journal of Approximation Theory* 160(2009) 89-102.
- [12] Hu, S.-M. Conversion of a triangular Bézier patch into three rectangular Bézier patches. *Computer Aided Geometric Design* 13(1996) 219-226.
- [13] Hu, S.-M. Conversion between triangular and rectangular Bézier patches. *Computer Aided Geometric Design* 18(2001) 667-671.
- [14] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194(2005) 4135-4195.
- [15] Noah Jaxon, Xiaoping Qian. Isogeometric analysis on triangulations. *Computer-Aided Design* 46(2014) 45-57.
- [16] M.J. Lai, M. A. Matt. A C^r trivariate macro-element based on the Alfeld split of tetrahedra. *Journal of Approximation Theory* 175(2013) 114-131.
- [17] L. Liu, Y. Zhang, T. J.R. Hughes, M. A. Scott, T. W. Sederberg. Volumetric T-Spline construction using Boolean operations. *Engineering with Computers*, 30(4)(2014) 425-439.
- [18] W. Liu, S. Mann. An optimal algorithm for expanding the composition of polynomials. *ACM Transaction on Graphics* 16(1997) 155-178
- [19] T. Martin, E. Cohen, R.M. Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided Geometric Design* 26(2009) 648-664.
- [20] B. Mourrain, R. Vidunas, N. Villamizar. Dimension and bases for geometrically continuous splines on surfaces of arbitrary topology. *Computer Aided Geometric Design* 45(2016) 108-133.
- [21] V.P. Nguyen, P. Kerfriden, S.P.A. Bordas, T. Rabczuk. Isogeometric analysis suitable trivariate NURBS representation of composite panels with a new offset algorithm. *Computer-Aided Design* 55(2014) 49-63.
- [22] K.F. Pettersen, V. Skytt. Spline volume fairing. *Lecture Notes in Computer Science* 6920(2012) 553-561.
- [23] L. L. Schumaker, T. Sorokina, A. J. Worsey. A C^1 quadratic trivariate macro-element space defined over arbitrary tetrahedral partitions. *Journal of Approximation Theory* 158(2009) 126-142.
- [24] H. Speleers. Multivariate normalized Powell-Sabin B-splines and quasi-interpolants. *Computer Aided Geometric Design* 30(2013) 2-19.
- [25] H. Speleers, C. Manni. Optimizing domain parameterization in isogeometric analysis based on Powell-Sabin splines. *Journal of Computational and Applied Mathematics* 289(2015) 68-86.
- [26] W. Wang, Y. Zhang, L. Liu, T. J.R. Hughes. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design* 45(2013) 351-360.
- [27] X. Wang, X. Qian. An optimization approach for constructing trivariate B-spline solids. *Computer-Aided Design* 46 (2014) 179 - 191.
- [28] S. Xia, X. Qian. Isogeometric analysis with Bézier tetrahedra. *Computer Methods in Applied Mechanics and Engineering* 316(2017) 782-816.
- [29] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. Parameterization of computational domain in isogeometric analysis: methods and comparison. *Computer Methods in Applied Mechanics and Engineering* 200(2011) 2021-2031.
- [30] G. Xu, M. Li, B. Mourrain, T. Rabczuk, J. Xu, S.P.A. 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) 175-200.
- [31] G. Xu, T. H. Kwok, C. C. Wang. Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization. *Computer-Aided Design* 91(2017) 1-13.
- [32] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. *Computer-Aided Design* 45(2013) 812-821.
- [33] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design* 45(2013) 395-404.
- [34] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method. *Journal of Computational Physics* 252(2013) 275-289.
- [35] G. Xu, B. Mourrain, A. Galligo, T. Rabczuk. High-quality construction of analysis-suitable trivariate NURBS solids by reparameterization methods. *Computational Mechanics* 54 (2014) 1303-1313.
- [36] Y. Zhang, W. Wang, T. J.R. Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering*, 201(2012) 185-197.
- [37] Y. Zhang, W. Wang, T. J.R. Hughes. Conformal solid T-spline construction from boundary T-spline representations. *Computational Mechanics* 51(2013) 1051-1059.