Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn
Wegelerstraße 6 • 53115 Bonn • Germany
phone +49 228 73-3427 • fax +49 228 73-7527
www.ins.uni-bonn.de

Maharavo Randrianarivony

Tetrahedral transfinite interpolation with B-patch faces: construction and regularity

INS Preprint No. 0803

June 2008

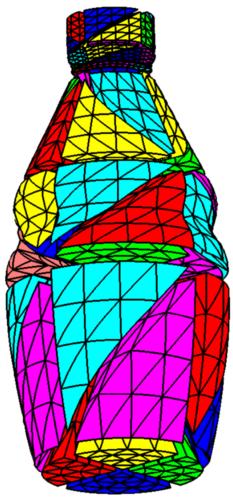
Abstract: We provide a methodology for efficiently checking whether a tetrahedral transfinite interpolation is regular. From given four triangular surfaces in form of B-patches fulfilling some compatibility conditions, we generate a transfinite interpolation defined on the unit 3D simplex. An efficient subdivision scheme is provided for a B-patch in order to obtain criteria which are verifiable in a discrete manner. That yields an adaptive method where only some parts of the simplex need to be subdivided. In order to reduce the computational cost, we make use of degree reductions. That is achieved by utilizing products of scaled Jacobi polynomials. After each subdivision process, one performs degree reductions provided that the induced error is sufficiently small. This work is important for application in refineable functions and for generation of multiscale or multiresolution bases functions from 3D CAD models.
Key Words: Tetrahedron, subdivision, B-patch, transfinite interpolation, degree reduction.

1 Introduction

Methods based upon refineable structures [12] are usually very efficient in practice [9] because they give rise to subdivision algorithms which can be used for the construction of hierarchical bases [6]. Such a hierarchical setting produces in general good accuracy with low computational cost [6]. For instance, in multigrid for finite element method, one needs $\mathcal{O}(n)$ complexity which is optimal to solve a problem having n degrees of freedom. In the case of multiscale methods, the rate between cost and accuracy has been demonstrated to be optimal [5] as specified by N-term approximation. While the theoretical advantages have been completely proved [5, 6, 20], applications to real-world CAD data do not seem to have attained a stage of maturity. Such problems are referred [2] to as *curse of geometry* because of lack of geometric data. In this document, we want to contribute in the CAD modeling for application in 3D hierarchical models. More precisely, we want to check whether a tetrahedral transfinite interpolation which is defined on the unit tetrahedron is regular. A tetrahedral transfinite interpolation is useful for generation of hierarchical mesh such as that in Fig. 1. Regularity is important when generating a mesh by computing the image of a uniform mesh on a reference tetrahedron. We use B-patches to represent the triangular non-planar faces of the tetrahedron. Generally, B-patch is the extension of Bézier or B-spline curve to triangular surfaces as discussed in [4]. In terms of parametrizations, hexahedra are more attractive than tetrahedra because hexahedral mappings have tensor product structure. But a tetrahedral decomposition is simpler to generate than a hexahedral one [19, 14]. That is due to the fact that locally improving the quality of a tetrahedral decomposition can be done without using hanging nodes by using bistellar flippings [7]. Similarly, for inserting a new node in a tetrahedral decomposition, local operations can be applied. Such local operations do not completely exist for hexahedral decompositions. A local rectification in a hexahedral decomposition can affect hexahedra which are very far from the hexahedron to be modified.

Related works about CAD preparation and triangular patch representations are as follows. Brunnett and Randrianarivony have invested a lot to develop a method which is appropriate for surfaces in integral equations. Their methods have already been successfully implemented to CAD and molecular surfaces [17, 18]. Harbrecht and Randrianarivony [8] have used those surface CAD models for applications in Wavelet BEM. On the other hand, B-patches have been introduced by H. Seidel, W. Dahmen, C. Micchelli around the early 1990's. Later on, B-patches have been further developed or generalized by many other scientists in CAGD and graphics. Here, we want to extend the surface methods in [17, 18] to 3D solid models where

(a)



(b)

Figure 1: (a) CAD model (b) Tetrahedral decomposition

each cell is topologically a tetrahedron.

The structure of this paper is as follows. In the next section, we will formulate the problem accurately and we will introduce various notions including tetrahedral transfinite interpolation. In section 3, we will treat the representation of the triangular faces as B-patches. Afterwards, the correlation between transfinite interpolation and blossomings will be treated in section 5. In section 4, we will apply recursive subdivision in order to obtain an adaptive algorithm. Section 6 will contain degree reductions which can be used to reduce the computational cost of the problem. We do not give any numerical results from CAD for that we want to concentrate on theoretical matter in this paper.

2 Problem Setting and Motivation

Let us first introduce the notion of transfinite interpolation where we consider the following reference domains:

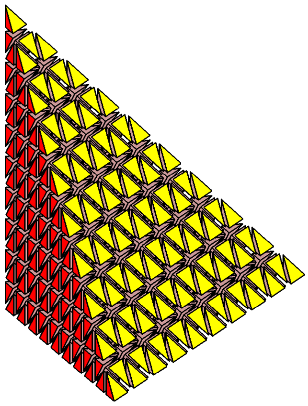
$$\Delta^2_{\rm ref} := \{\boldsymbol{\sigma} = (s,t) \in \mathbb{R}^2 : \quad s \geq 0, \quad t \geq 0, \quad s+t \leq 1\}, \tag{1}$$
$$\Delta^3_{\rm ref} := \{\mathbf{u} = (u,v,w) \in \mathbb{R}^3 : u \geq 0,\ v \geq 0,\ w \geq 0,\ u+v+w \leq 1\}. \tag{2}$$

Suppose that we have four triangular surfaces $\mathbf{F}_i : \Delta^2_{\rm ref} \longrightarrow \mathbb{R}^3$ where $i = 1, \cdots, 4$. A transfinite interpolant is a function $\mathbf{X} : \Delta^3_{\rm ref} \longrightarrow \mathbb{R}^3$ which verifies the following boundary conditions:

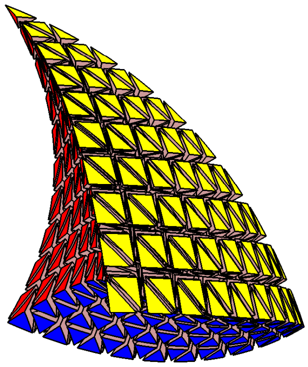
$$\begin{array}{rcll} \mathbf{X}(u,v,0) & = & \mathbf{F}_1(v,u) & \\ \mathbf{X}(u,0,w) & = & \mathbf{F}_2(u,w) & \\ \mathbf{X}(0,v,w) & = & \mathbf{F}_4(w,v) & \\ \mathbf{X}(u,v,w) & = & \mathbf{F}_3(v,w) & \text{if} \quad u+v+w=1. \end{array} \tag{3}$$

In order that these conditions can be fulfilled, it is necessary to assume the following compatibility conditions at the four corners

$$\mathbf{F}_1(0,0) = \mathbf{F}_2(0,0) = \mathbf{F}_4(0,0), \qquad \mathbf{F}_2(1,0) = \mathbf{F}_1(0,1) = \mathbf{F}_3(0,0),$$
$$\mathbf{F}_4(0,1) = \mathbf{F}_1(1,0) = \mathbf{F}_3(1,0), \qquad \mathbf{F}_2(0,1) = \mathbf{F}_3(0,1) = \mathbf{F}_4(1,0),$$



(a)



(b)

Figure 2: (a)Tetrahedralization of the reference tetrahedron Δ^3_{ref} (b)Image by a tetrahedral transfinite interpolation of the left-hand mesh.

and the following six compatibility conditions at the interfaces

$$\begin{aligned} \mathbf{F}_1(0,\mu) =& \mathbf{F}_2(\mu,0), & \mathbf{F}_1(1-\mu,\mu) =& \mathbf{F}_3(1-\mu,0),\\ \mathbf{F}_1(\mu,0) =& \mathbf{F}_4(0,\mu), & \mathbf{F}_2(1-\mu,\mu) =& \mathbf{F}_3(1-\mu,0),\\ \mathbf{F}_2(0,\mu) =& \mathbf{F}_4(\mu,0), & \mathbf{F}_3(1-\mu,\mu) =& \mathbf{F}_4(\mu,1-\mu). \end{aligned}$$

For the local node and face numbering, we follow the CGNS convention [3] because our theory will eventually be used in practical computer implementations. Therefore, it is advantageous to follow standard indexation though an arbitrary indexation would work as well. Consider four blending functions $p_i : \mathbb{R}^3 \to \mathbb{R}$ which are entities verifying

$$\begin{array}{llll} (C1) & p_1(0,v,w)=0 & \text{for all} & v,w\in\mathbb{R} \\ (C2) & p_2(u,0,w)=0 & \text{for all} & u,w\in\mathbb{R} \\ (C3) & p_3(u,v,0)=0 & \text{for all} & u,v\in\mathbb{R} \\ (C4) & p_4(u,v,w)=0 & \text{for all} & u,v,w\in\mathbb{R} \quad \text{with} \quad u+v+w=1, \end{array}$$

and whose sum is such that

$$\sum_{i=1}^{4} p_i(\mathbf{u}) = 1 \qquad \forall\, \mathbf{u} = (u,v,w) \in \mathbb{R}^3. \tag{4}$$

For our purpose, we will consider only blending functions p_i which are polynomials. Note that conditions (C1) until (C4) ensure that each blending function p_i has zero values on respective face of the unit tetrahedron Δ^3_{ref}. In order to describe the definitions in this document well, let us introduce the next notations. We will use $[\![a,b]\!]$ to denote the set of integers between $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$ (a and b included). Like in many documents of CAGD, multi-indices will be denoted by bold Greek letters such as $\boldsymbol{\beta} \in \mathbb{N}_0^{d+1}$ where $\boldsymbol{\beta} = (\beta_0, \cdots, \beta_d)$ in which we have $|\boldsymbol{\beta}| := \sum_{p=0}^{d} \beta_p$. We introduce the following set

$$\Omega^d_n := \{\boldsymbol{\beta} = (\beta_0, ..., \beta_d) \in \mathbb{N}_0^{d+1} \,:\, |\boldsymbol{\beta}| = n\}. \tag{5}$$

The tetrahedral transfinite interpolation which is used in this document will be given in the form

$$\mathbf{X}(\mathbf{u}) = \sum_{i=0}^{M} p_{r(i)}(\mathbf{u})\mathbf{F}_{s(i)}[\psi_i(\mathbf{u})] \in \mathbb{R}^3, \tag{6}$$

where r and s are certain functions defined on $[\![1, M]\!]$ and taking value in $[\![0, 4]\!]$ while $\psi_i : \mathbb{R}^3 \to \mathbb{R}^2$ are some affine functions. The following expanded expression of (6) will not be used here but it is important for verification and practical implementation. It is immediate to verify that the next expression fulfils the conditions in (3) by using the properties of the blending functions and the compatibility conditions.

$$\begin{aligned}
\mathbf{X}(\mathbf{u}) \quad & := p_3(\mathbf{u})\mathbf{F}_4(u+w,v) + p_2(\mathbf{u})\mathbf{F}_4(w,u+v) + p_1(\mathbf{u})\mathbf{F}_1(v,u+w) \\
& +p_2(\mathbf{u})\mathbf{F}_1(v+w,u) + p_4(\mathbf{u})\mathbf{F}_1(v,u) + p_4(\mathbf{u})\mathbf{F}_4(w,v) + p_4(\mathbf{u})\mathbf{F}_2(u,w) \\
& +p_1(\mathbf{u})\mathbf{F}_1(u+v,w) - p_4(\mathbf{u})\mathbf{F}_2(0,w) + p_1(\mathbf{u})\mathbf{F}_3(v,w) + p_1(\mathbf{u})\mathbf{F}_3(0,0) \\
& -p_1(\mathbf{u})\mathbf{F}_3(0,0) - p_3(\mathbf{u})\mathbf{F}_4(1-v,v) - p_2(\mathbf{u})\mathbf{F}_4(w,1-w) - p_4(\mathbf{u})\mathbf{F}_4(0,v) \\
& -p_2(\mathbf{u})\mathbf{F}_4(0,u+v+w) + p_3(\mathbf{u})\mathbf{F}_1(u,v+w) - p_3(\mathbf{u})\mathbf{F}_2(0,u+v+w) \\
& -p_1(\mathbf{u})\mathbf{F}_3(0,w) - p_2(\mathbf{u})\mathbf{F}_1(1-u,u) - p_1(\mathbf{u})\mathbf{F}_1(v,1-v) - p_4(\mathbf{u})\mathbf{F}_1(0,u) \\
& +p_3(\mathbf{u})\mathbf{F}_2(0,1) + p_2(\mathbf{u})\mathbf{F}_1(1,0) + p_4(\mathbf{u})\mathbf{F}_4(0,0) + p_3(\mathbf{u})\mathbf{F}_3(v,1-v-u) \\
& +p_2(\mathbf{u})\mathbf{F}_1(1-w-u,w) - p_1(\mathbf{u})\mathbf{F}_1(0,u+v+w) - p_3(\mathbf{u})\mathbf{F}_3(0,1-u).
\end{aligned}$$

In practical hierarchical mesh generation, it is important that the transfinite mapping $\mathbf{X}$ is regular

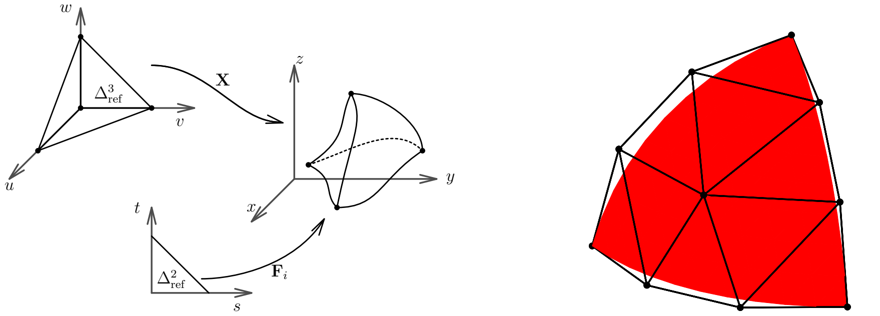
$$\mathcal{J}(u,v,w) = \det[\mathbf{X}(u,v,w)] \neq 0 \qquad \forall\,(u,v,w) \in \Delta^3_{\mathrm{ref}}. \tag{7}$$

Regularity is essential because finding a hierarchical splitting on the image mesh $\mathbf{X}(\Delta^3_{\mathrm{ref}})$ will be done in the following way. One first generates a hierarchical mesh $\mathcal{M}$ on the reference tetrahedron Δ^3_{ref}. Afterwards, one computes its image $\mathbf{X}(\mathcal{M})$ by $\mathbf{X}$ as shown in Fig. 2. If the transfinite interpolation $\mathbf{X}$ is not regular, then the image mesh $\mathbf{X}(\mathcal{M})$ will have some foldings which are not appropriate for the subsequent numerical simulations. Note that although a triangular face is completely inside the whole CAD model, it is possible that it is nonplanar. As a consequence, it is important to consider this problem in its full generality and not only for cases where some triangular faces are planar. It is possible that the four surfaces are quite smooth while the internal cells present irregularity which causes conflict in subsequent mesh refinement. The purpose of this paper is not to give a remedy of irregularity. Our main objective here is exclusively to recognize if a tetrahedral transfinite interpolation $\mathbf{X}$ determines a regular functions. Since it is not possible to check the sign of the determinant in (7) for all points inside the tetrahedron, we need discrete information which is easy to check. We will search for criteria related to the control points (see Fig. 3) of the faces $\mathbf{F}_i$ to verify regularity.

3 Multivariate B-patch representation

Since the triangular faces $\mathbf{F}_i$ are represented in B-patch form, we will introduce the definition and certain properties of B-patch in this section. We will see only those topics which are immediately related to our needs and complete details can be found in [21, 22, 4]. First, let us utilize the barycentric coordinates with respect to $W = (\mathbf{w}_0, \cdots, \mathbf{w}_d)$ where $\mathbf{w}_i \in \mathbb{R}^d$ by using

$$D(W) := \det \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \mathbf{w}_0 & \mathbf{w}_1 & \ldots & \mathbf{w}_d \end{bmatrix}. \tag{8}$$



(a) (b)

Figure 3: (a)Tetrahedral transfinite interpolation (b)B-patch for $n = 3$

Additionally, we define for any $\mathbf{u} \in \mathbb{R}^d$

$$D_i(W|\mathbf{u}) := \det \begin{bmatrix} 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ \mathbf{w}_0 & \dots & \mathbf{w}_{i-1} & \mathbf{u} & \mathbf{w}_{i+1} & \dots & \mathbf{w}_d \end{bmatrix}. \tag{9}$$

The barycentric coordinates of $\mathbf{u}$ with respect to W are defined by

$$\lambda_i(\mathbf{u}) := D_i(W|\mathbf{u})/D(W). \tag{10}$$

For a set $\mathcal{A} := \{\mathbf{t}_{p,i} \in \mathbb{R}^d : \quad p \in [\![0, d]\!], \quad i \in [\![0, n-1]\!]\}$ and a multi-index $\boldsymbol{\delta} = (\delta_0, \cdots, \delta_d)$, we define $\mathcal{A}_{\boldsymbol{\delta}} := \{\mathbf{t}_{0,\delta_0}, \cdots, \mathbf{t}_{d,\delta_d}\}$. The set $\mathcal{A}$ is called a knot arrangement if $\mathcal{A}_{\boldsymbol{\delta}}$ is affinely independent for all $|\boldsymbol{\delta}| \leq n$. Let $\lambda_{\boldsymbol{\delta},p}(\mathbf{u})$ be the barycentric coordinates of $\mathbf{u} \in \mathbb{R}^d$ with respect to $\mathcal{A}_{\boldsymbol{\delta}}$ such as $\mathbf{u} = \sum_{p=0}^{d} \lambda_{\boldsymbol{\delta},p}(\mathbf{u})\mathbf{t}_{p,\delta_p}$. In order to define the basis functions [21, 4], we initialize $\mathcal{B}^0_{\mathbf{0}}(\mathbf{u}) := 1$ where $\mathbf{0} \in \Omega^d_0$. For $k \in [\![1, n]\!]$, we define recursively

$$\mathcal{B}^k_{\boldsymbol{\delta}}(\mathbf{u}) := \sum_{p=0}^{d} \lambda_{\boldsymbol{\delta},p}(\mathbf{u})\mathcal{B}^{k-1}_{\boldsymbol{\delta}-\mathbf{e}_p}(\mathbf{u}) \qquad \text{for} \qquad \boldsymbol{\delta} \in \Omega^d_k, \tag{11}$$

where $\mathbf{e}_p \in \Omega^d_1$ is the multi-index having 1 at the p-th entry and 0 elsewhere. A B-patch is defined to be a linear combination of those basis functions $\mathcal{B}^{\mathcal{A}}_{\boldsymbol{\beta}} := \mathcal{B}^n_{\boldsymbol{\beta}}$:

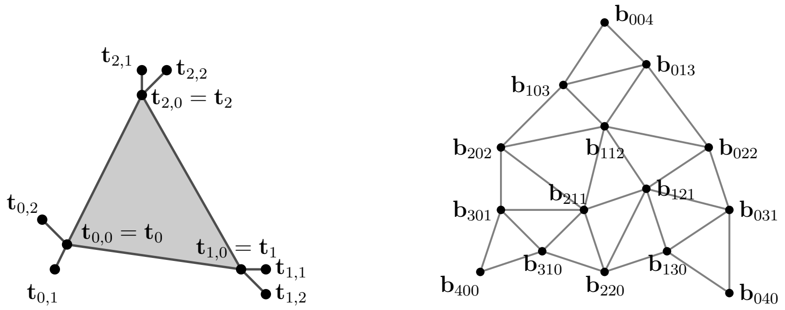
$$\mathbf{Y}(\mathbf{u}) = \sum_{\boldsymbol{\beta} \in \Omega^d_n} \mathbf{b}_{\boldsymbol{\beta}} \mathcal{B}^{\mathcal{A}}_{\boldsymbol{\beta}}(\mathbf{u}). \tag{12}$$

The set of the control points $\mathbf{b}_{\boldsymbol{\beta}}$ generates the control net of the B-patch. A graphical illustration of a control net for $n = 4$ can be seen in Fig. 4(b). In our case of tetrahedral transfinite interpolation, we suppose that the four triangular faces $\mathbf{F}_p$ are B-patches:

$$\mathbf{F}_p(s,t) = \sum_{\boldsymbol{\beta} \in \Omega^2_n} \mathbf{b}^p_{\boldsymbol{\beta}} \mathcal{B}^{\mathcal{A}}_{\boldsymbol{\beta}}(s,t) \qquad \forall\, p = 1, ..., 4. \tag{13}$$

For the knot arrangement, we will denote the first entries by $\mathbf{t}_p := \mathbf{t}_{p,0}$. Further, we make the assumption that the distance of the clouds $(\mathbf{t}_{p,i})_{i=1}^{n-1}$ about each apex $\mathbf{t}_p$ is of order quadratic to the length of the edges (see Fig. 4(a)):

$$\|\mathbf{t}_p - \mathbf{t}_{p,i}\| = \mathcal{O}\left(\max_{j \neq k} \|\mathbf{t}_j - \mathbf{t}_k\|^2\right) \qquad \forall\, p \in [\![1, d]\!], \quad \forall\, i \in [\![1, n-1]\!]. \tag{14}$$



(a) (b)

Figure 4: (a)Knot arrangement (b)Control net for $n = 4$

It is well known [21, 22] that every multivariate polynomial can be expressed in term of one B-patch. Additionally, if each cloud of points is reduced to one point such as $\mathbf{t}_p := \mathbf{t}_{p,i}$ for all $i \in [\![0, n-1]\!]$ then we retrieve the usual triangular Bézier surface with the basis functions

$$\frac{n!}{\beta_0!\beta_1!\cdots\beta_d!}(1 - u_0 - ... - u_d)^{\beta_0} u_1^{\beta_1} \cdots u_d^{\beta_d}. \tag{15}$$

On the other hand, a B-patch can be seen as a piece of triangular B-spline which have been developed by Dahmen, Micchelli and Seidel [4].
The relationship of the control points and the polynomials can be expressed with the help of the blossoming which we recall briefly now. A function b is a polar form or a blossom function [16] if it is multiaffine:

$$\begin{aligned} b(\mathbf{u}_1, \cdots, \lambda_a \mathbf{u}_i^a \quad + \quad & \lambda_b \mathbf{u}_i^b, \cdots, \mathbf{u}_n) = \lambda_a b(\mathbf{u}_1, \cdots, \mathbf{u}_i^a, \cdots, \mathbf{u}_n) + \\ & \lambda_b b(\mathbf{u}_1, \cdots, \mathbf{u}_i^b, \cdots, \mathbf{u}_n) \qquad \forall\, \lambda_a + \lambda_b = 1. \end{aligned} \tag{16}$$

and symmetric:

$$b(\mathbf{u}_1, \cdots, \mathbf{u}_i, \cdots, \mathbf{u}_j, \cdots, \mathbf{u}_n) = b(\mathbf{u}_1, \cdots, \mathbf{u}_j, \cdots, \mathbf{u}_i, \cdots, \mathbf{u}_n). \tag{17}$$

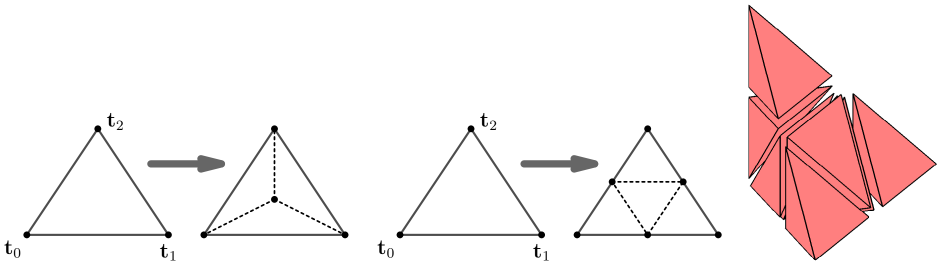
There is a close connection between blossoms and the space of d-variate polynomials of degree n:

$$\Pi_n(\mathbb{R}^d) := \left\{ f : \quad f(\mathbf{u}) = \sum_{|\boldsymbol{\gamma}| \le n} a_{\boldsymbol{\gamma}} \mathbf{u}^{\boldsymbol{\gamma}}, \quad a_{\boldsymbol{\gamma}} \in \mathbb{R} \right\}. \tag{18}$$

For each polynomial $f \in \Pi_n(\mathbb{R}^d)$, there is a unique blossom function $\mathcal{P}(f)$ such that we have the next diagonal property:

$$\mathcal{P}(f)(\underbrace{\mathbf{u}, \cdots, \mathbf{u}}_{n}) = f(\mathbf{u}) \qquad \forall\, \mathbf{u} \in \mathbb{R}^d. \tag{19}$$

The blossom of the function $\mathbf{Y}$ can be evaluated at $(\mathbf{u}_1, \cdots, \mathbf{u}_n)$ with the help of a pyramid algorithm [21]:



(a) (b) (c)

Figure 5: (a)Seidel's subdivision (b),(c)Subdivision with similar shape

	Algorithm: Pyramid algorithm for multivariate blossom
1:	Initialize $\mathbf{b}^0_{\boldsymbol{\delta}} := \mathbf{b}_{\boldsymbol{\delta}}$ for all $\boldsymbol{\delta} \in \Omega^d_n$
2:	**for** $(l = 1, \cdots, n)$
3:	$\quad \mathbf{b}^l_{\boldsymbol{\delta}} := \sum_{k=0}^d \lambda_{\boldsymbol{\delta},k}(\mathbf{u}_l)\mathbf{b}^{l-1}_{\boldsymbol{\delta}+e_k} \quad \forall\, \boldsymbol{\delta} \in \Omega^d_{n-l}$
4:	**enddo**
5:	Define $\mathcal{P}(\mathbf{Y})(\mathbf{u}_1, \cdots, \mathbf{u}_n) := \mathbf{d}^n_{\mathbf{0}}$ where $\mathbf{0} = (0, \cdots, 0) \in \Omega^d_0$

The blossom function of the B-patch in (12) and the control points are related with the following relation

$$\mathbf{b}_{\boldsymbol{\gamma}} = \mathcal{P}(\mathbf{Y})(\underbrace{\mathbf{t}_{0,0}, \cdots, \mathbf{t}_{0,\gamma_0-1}}_{\gamma_0}, \underbrace{\mathbf{t}_{1,0}, \cdots, \mathbf{t}_{1,\gamma_1-1}}_{\gamma_1}, \cdots, \underbrace{\mathbf{t}_{d,0}, \cdots, \mathbf{t}_{d,\gamma_d-1}}_{\gamma_d}). \tag{20}$$

4 Subdivision of B-patches

In this section, we state our results about subdivision and its application to our tetrahedral transfinite interpolation. To that end, let us formulate the process of subdivision clearly. Seidel has proposed a subdivision scheme where one inserts a new point inside one triangle which is then split into three sub-triangles as seen in Fig. 5(a). Below, we propose a more general subdivision for the multidimensional case where a simplex is split into several simplices which have the same shape up to some rotations. The following subdivision technique is correct for any dimension d but we will need it mainly for $d = 3$.

Before formulating the recursive subdivision, let us formulate the expression of a B-patch inside a smaller knot arrangement where $\mathbf{t}_p = \mathbf{t}_{p,0}$. Let $\mathbf{Y}$ be a B-patch with respect to $\mathcal{A} = \{\mathbf{t}_{p,i}\}$ as in (12). Consider another knot arrangement $\bar{\mathcal{A}} = \{\bar{\mathbf{t}}_{p,i}\}$ with $\bar{\mathbf{t}}_p := \bar{\mathbf{t}}_{p,i}$ such that the simplex $conv\{\bar{\mathbf{t}}_p : p \in [\![0, d]\!]\}$ is inside the simplex $conv\{\mathbf{t}_p : p \in [\![0, d]\!]\}$. We want to express $\mathbf{Y}$ from (12) with respect to the new knot arrangement $\bar{\mathcal{A}}$ such that

$$\mathbf{Y}(\mathbf{u}) = \sum_{\boldsymbol{\beta} \in \Omega^d_n} \bar{\mathbf{b}}_{\boldsymbol{\beta}} \mathcal{B}^{\bar{\mathcal{A}}}_{\boldsymbol{\beta}}(\mathbf{u}). \tag{21}$$

In order to find the control points $\bar{\mathbf{b}}_{\boldsymbol{\beta}}$ with respect to the new knot arrangement $\bar{\mathcal{A}}$, we apply the pyramid algorithm from Section 3 to $(\mathbf{u}_1, ..., \mathbf{u}_n) = (\bar{\mathbf{t}}_{0,0}, \cdots, \bar{\mathbf{t}}_{0,\beta_0-1}, \bar{\mathbf{t}}_{1,0}, \cdots, \bar{\mathbf{t}}_{1,\beta_1-1}, \cdots, \bar{\mathbf{t}}_{d,0}, \cdots, \bar{\mathbf{t}}_{d,\beta_d-1})$ by using the blossom of the original B-patch (12). The new control points are then obtained from the connection formula (20), that is to say we have

$$\bar{\mathbf{b}}_{\boldsymbol{\beta}} = \mathcal{P}(\mathbf{Y})(\bar{\mathbf{t}}_{0,0}, \cdots, \bar{\mathbf{t}}_{0,\beta_0-1}, \bar{\mathbf{t}}_{1,0}, \cdots, \bar{\mathbf{t}}_{1,\beta_1-1}, \cdots, \bar{\mathbf{t}}_{d,0}, \cdots, \bar{\mathbf{t}}_{d,\beta_d-1}). \tag{22}$$

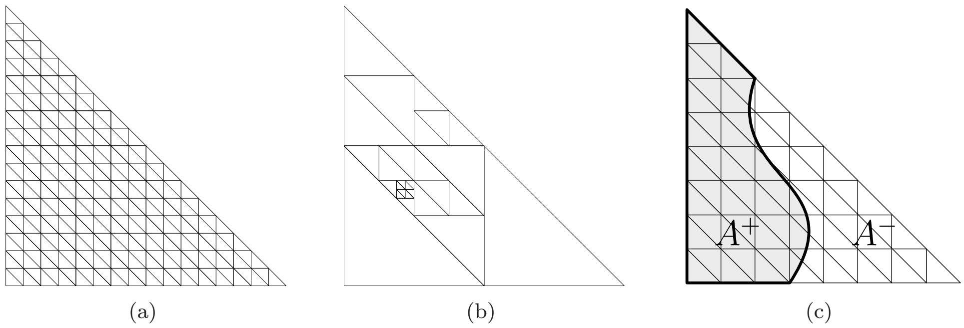


Figure 6: (a)Uniform subdivision with ω_5^2 triangles (b)Adaptive subdivision (c)Sign distribution on the subdivided simplex

In order to introduce the notion of recursive subdivisions, suppose that a B-patch has a knot arrangement where the apices $\mathbf{t}_p$ are the corners of a simplex $\Delta \subset \mathbb{R}^d$. In fact, the simplex Δ is subdivided into several subsimplices as follows. First, a new node is introduced at the middle of each edge of Δ. Then, one generates subsimplices of the same size by using only the nodes of Δ and the newly created ones. For example, in case of triangles or tetrahedra where $d = 2$ and $d = 3$, we have the subdivisions in Fig. 5(b) and Fig. 5(c) respectively. The same subdivision process can be applied to each one of the resulting subsimplices. By doing that repeatedly, let us denote by ω_N^d the number of simplices after N subdivisions as illustrated in Fig. 6(a). That is, we have simplices $\Delta^{N,k}$ for $k = 1, 2, ..., \omega_N^d$ on which we have the B-patches

$$\mathbf{Y}^{N,k}(\mathbf{u}) = \sum_{\boldsymbol{\beta} \in \Omega_n^d} \mathbf{b}_{\boldsymbol{\beta}}^{N,k} \mathcal{B}_{\boldsymbol{\beta}}^{\mathcal{A}^{N,k}}(\mathbf{u}). \tag{23}$$

Thus, on the N-th subdivision, we obtain ω_N^d knot arrangements $\mathcal{A}^{N,k}$ for $k = [\![1, w_N^d]\!]$ having knots $\mathbf{t}_q^{N,k}$. We suppose that the relation (14) about the cloud still persists in each subdivision step. That is, we have at each apex $\mathbf{t}_r^{N,k}$:

$$\|\mathbf{t}_r^{N,k} - \mathbf{t}_{r,i}^{N,k}\| = \mathcal{O}\left(\max_{p \neq q} \|\mathbf{t}_p^{N,k} - \mathbf{t}_q^{N,k}\|^2\right) \qquad \forall\, i \in [\![1, n-1]\!]. \tag{24}$$

Note that the length $\|\mathbf{t}_q^{N,k} - \mathbf{t}_p^{N,k}\|$ of an edge on the current subdivision is the half of the parent edge. For the k-th subsimplex ($k \in [\![1, \omega_N^d]\!]$) of the N-th subdivision, we will denote the control point by $b_{\boldsymbol{\beta}}^{N,k}$. Additionally, the maximal length of the initial edges will be denoted by

$$h := \max_{p \neq q} \|\mathbf{t}_p^0 - \mathbf{t}_q^0\|. \tag{25}$$

In our case of tetrahedral transfinite interpolation, we suppose that the apices $\mathbf{t}_i$ of the first knot arrangement are composed of the corners of the unit reference tetrahedron Δ_{ref}^3, that is $\Delta := \Delta_{\text{ref}}^3$.

A tetrahedral mapping is regular if the determinant is of constant sign. Throughout this paper, we suppose it is positive. Additionally, although the determinant is positive but very small, we consider that as irregularity. As a consequence, the regularity definition in (7) can be replaced by

$$\mathcal{J}(\mathbf{u}) = \det[\mathbf{X}(u, v, w)] \geq \delta \qquad \forall\, \mathbf{u} = (u, v, w) \in \Delta_{\text{ref}}^3, \tag{26}$$

for some prescribed constant $\delta > 0$.

Theorem 4.1 *Suppose that we have regularity and that $\mathcal{J}(\mathbf{u}) \geq \delta > 0$. Then, for sufficiently many subdivisions, for all $k \in [\![1, \omega_N^d]\!]$ and $\boldsymbol{\beta} \in \Omega_N^d$ the B-patch coefficients $b_{\boldsymbol{\beta}}^{N,k}$ on each subsimplex verify:*

$$b_{\boldsymbol{\beta}}^{N,k} = \mathcal{P}(\mathcal{J})(T) + \mathcal{O}\left(\max_{p\neq q} \|\mathbf{t}_p^{N,k} - \mathbf{t}_q^{N,k}\|^2\right). \tag{27}$$

Thus, the expected number of subdivisions to ensure the positivity of those coefficients $b_{\boldsymbol{\beta}}^{N,k}$ is

$$N \quad \sim \quad \left\lceil \log_2\left(\frac{\sqrt{\delta}}{h}\right)\right\rceil \tag{28}$$

where $\lceil x \rceil$ denotes the smallest integer which is larger than x.

PROOF. Consider the k-th subsimplex on the N-th subdivision. For each multi-index $\boldsymbol{\gamma} = (\gamma_0, \cdots, \gamma_d) \in \Omega_n^d$, we define $\boldsymbol{\theta}_{\boldsymbol{\gamma}} := \sum_{p=0}^{d} (\gamma_p/n)\mathbf{t}_p^{N,k}$. Consider the following ordered knot sequence

$$T := (\underbrace{\mathbf{t}_{0,0}^{N,k}, \cdots, \mathbf{t}_{0,\gamma_0-1}^{N,k}}_{\gamma_0}, \cdots, \underbrace{\mathbf{t}_{d,0}^{N,k}, \cdots, \mathbf{t}_{d,\gamma_d-1}^{N,k}}_{\gamma_d}) \tag{29}$$

and that of θ as

$$\tilde{T} = (\underbrace{\theta_{\boldsymbol{\beta}}, \cdots, \theta_{\boldsymbol{\beta}}}_{n}) \qquad \bar{T} := (\underbrace{\mathbf{t}_{0,0}^{N,k}, \cdots, \mathbf{t}_{0,0}^{N,k}}_{\gamma_0}, \cdots, \underbrace{\mathbf{t}_{d,0}^{N,k}, \cdots, \mathbf{t}_{d,0}^{N,k}}_{\gamma_d}). \tag{30}$$

We apply a multi-variate Taylor expansion of second order to the blossom $\mathcal{P}(\mathcal{J})$.

$$\mathcal{P}(\mathcal{J})(\tilde{T}) = \mathcal{P}(\mathcal{J})(T) + \sum_{|\boldsymbol{\gamma}|=1} \partial_{\boldsymbol{\gamma}}\mathcal{P}(\mathcal{J})(T)(T-\tilde{T})^{\boldsymbol{\gamma}} + \mathcal{O}\left(\max_{p\neq q} \|\mathbf{t}_p^{N,k} - \mathbf{t}_q^{N,k}\|^2\right). \tag{31}$$

Since the blossom is symmetric, the first partial derivatives are the same such that for all $|\boldsymbol{\gamma}| = 1$ we have $\partial_{\boldsymbol{\gamma}}\mathcal{P}(\mathcal{J})(T) = K$. Moreover, we have for $|\boldsymbol{\gamma}| = 1$:

$$(T - \tilde{T})^{\boldsymbol{\gamma}} = t_{p,q}^r - \theta_{\boldsymbol{\gamma}}^r. \tag{32}$$

By using relation (24), we obtain for $\boldsymbol{\gamma}$ having unity length for entry corresponding to p, r:

$$(T - \tilde{T})^{\boldsymbol{\gamma}} = (T - \bar{T})^{\boldsymbol{\gamma}} + \mathcal{O}\left(\max_{p\neq q} \|\mathbf{t}_p^{N,k} - \mathbf{t}_q^{N,k}\|^2\right). \tag{33}$$

As a consequence, we have

$$\mathcal{P}(\mathcal{J})(\tilde{T}) = \mathcal{P}(\mathcal{J})(T) + K\sum_{q=1}^{d} \gamma_q(t_q^{N,k} - \theta_{\boldsymbol{\gamma}}^{N,k}) + \mathcal{O}\left(\max_{p\neq q} \|\mathbf{t}_p^{N,k} - \mathbf{t}_q^{N,k}\|^2\right). \tag{34}$$

Additionally, we have

$$\begin{aligned}\sum_{q=1}^{d} \gamma_q(t_q^{N,k} - \theta_{\boldsymbol{\gamma}}^{N,k}) &= \sum_{q=1}^{d} \gamma_q \left[(1 - \frac{\gamma_q}{n})t_q^{N,k} - \sum_{p\neq q} \frac{\gamma_p}{n} t_p^{N,k}\right] \\ &= \left[\sum_{q=1}^{d} \gamma_q \sum_{p\neq q} \frac{\gamma_p}{n} t_q^{N,k}\right] - \left[\sum_{p=1}^{d}\sum_{p\neq q} \frac{\gamma_p}{n}\gamma_p t_q^{N,k}\right] = 0.\end{aligned}$$

Thus, the sum with $|\boldsymbol{\gamma}| = 1$ of (31) vanishes and we obtain

$$\mathcal{P}(\mathcal{J})(\tilde{T}) = \mathcal{P}(\mathcal{J})(T) + \mathcal{O}\left(\max_{p\neq q} \|\mathbf{t}_p^{N,k} - \mathbf{t}_q^{N,k}\|^2\right) \quad (35)$$

$$= \mathcal{P}(\mathcal{J})(T) + \mathcal{O}\left(\frac{1}{2^{2N}} \max_{p\neq q} \|\mathbf{t}_p^{0,0} - \mathbf{t}_q^{0,0}\|^2\right). \quad (36)$$

We use the relation between the control points and the blossom in order to have

$$b_{\boldsymbol{\beta}}^{N,k} \geq \delta + \mathcal{O}\left(\frac{1}{2^N} h\right)^2. \quad (37)$$

The coefficients are therefore positive for sufficiently many subdivisions. More, the expected number of subdivision N verifies

$$2^{-N} \sim \frac{\sqrt{\delta}}{h}. \quad (38)$$

■

Theorem 4.2 *Let* $\mathbf{X}$ *be a tetrahedral transfinite interpolation that is not regular. Then, in the situation of Theorem 4.1 and for sufficiently large* N*, there must exist subsimplices* Δ^{N,k_1} *and* Δ^{N,k_2} *such that*

$$\begin{cases} b_\beta^{N,k_1} > 0 & \forall\, \beta \in \Omega_n^d \\ b_\beta^{N,k_2} < 0 & \forall\, \beta \in \Omega_n^d. \end{cases} \quad (39)$$

PROOF. Since we do not have regularity, $\mathcal{J}(u, v, w)$ is not of constant sign. Consider the following two disjoint sets (see Fig. 6(c)):

$$A^+ := \{\mathbf{u} \in \Delta^3_{\text{ref}} : \quad \mathcal{J}(\mathbf{u}) > 0\}, \quad (40)$$

$$A^- := \{\mathbf{u} \in \Delta^3_{\text{ref}} : \quad \mathcal{J}(\mathbf{u}) < 0\}. \quad (41)$$

For sufficiently many subdivisions, there exist k_1 and k_2 such that the two small simplices Δ^{N,k_1} and Δ^{N,k_2} belong respectively to A^+ and A^-. That is, $Y^{N,k_1}(\mathbf{u}) > 0$ and $Y^{N,k_2}(\mathbf{u}) < 0$. In order to obtain the claim about the control points b_β^{N,k_1} and b_β^{N,k_2}, we should use the blossom functions as in the proof of Theorem 4.1. In other words, by using the same argument as in (35), we obtain (39) for sufficiently large N.

■

The former theory proposes that we subdivide everywhere. In practice, that is not necessary because if $\mathbf{X}$ is regular on a larger simplex, then it is also regular for all subsimplices. That is, we need only to subdivide the simplices which are critical as illustrated in Fig. 5. The conditions in (39) will be used as an abortion criterion in the regularity algorithm which is summarized below. Note that we perform adaptive subdivision, i.e. only those functions are subdivided that have coefficients with different signs. In the following pseudo code, we start with $K = \Delta^3_{\text{ref}}$. The routine `SignTest`$(K)$ is a function that returns $+1$ (resp. -1) if all B-patch coefficients of the Jacobian defined on the simplex K are positive (resp. negative) and say 2 else. If `Check()` returns '+1' or '-1', then $\mathbf{X}$ is regular. If `Check()` returns '0', $\mathbf{X}$ is not regular.

```
Algorithm: Check(Simplex K)
int sign=SignTest(K)
if (sign==-1) or (sign==+1) then
      return sign;
else
      subdivide K into K_1,···, K_{ω_n}
      for(i=1,···,ω_n) int sign[i]=Check(K_i); enddo
      if (sign[1]==sign[2]==···==sign[ω_n]==-1) then
            return -1;
      endif
      if (sign[1]==sign[2]==···==sign[ω_n]==+1) then
            return +1;
      endif
      return 0;
endif
```

5 Transfinite interpolation and blossoms

We have introduced the tetrahedral transfinite interpolation as a combination of the triangular faces which are B-patches in relation (6). But we did not show yet how to describe the resulting map $\mathbf{X}$ as a single B-patch which is used in the above subdivision algorithm. The purpose of this section is to show some way of achieving that by using blossoming operations. Before showing the exact computations, let us introduce some notion about products of blossoms. For $I \subset [\![1, n]\!]$ such that $|I| = m$, let $\mathbf{p}_I$ be the projection from $(\mathbb{R}^d)^n$ to $(\mathbb{R}^d)^m$ such that for $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_n) \in (\mathbb{R}^d)^n$, $\mathbf{u}_i$ appears in $\mathbf{p}_I(\mathbf{U})$ if i belongs to I. The following theorem has been proved in [23] by using homogeneous polynomials. Below, we provide an alternative direct proof which is simpler in the opinion of the author.

Lemma 5.1 *Let $\mathbf{a}_0, ..., \mathbf{a}_d$ be affinely independent in $\mathbb{R}^d$. For any $\mathbf{x} \in \mathbb{R}^d$, define $\alpha_1(\mathbf{x})$,..., $\alpha_d(\mathbf{x})$ the components of $\overrightarrow{\mathbf{a}_0\mathbf{x}}$ in $\{\overrightarrow{\mathbf{a}_0\mathbf{a}_1}, ..., \overrightarrow{\mathbf{a}_0\mathbf{a}_d}\}$.*

$$\overrightarrow{\mathbf{a}_0\mathbf{a}_1} = \sum_{i=1}^{d} \alpha_i(\mathbf{x})\overrightarrow{\mathbf{a}_0\mathbf{a}_i} \tag{42}$$

Define $\alpha_0(\mathbf{x}) := 1 - \sum_{i=1}^{d} \alpha_i(\mathbf{x})$. Then, every blossom function b verifies

$$b(\mathbf{x}_1, ..., \mathbf{x}_{N-1}, \mathbf{x}_N) = \sum_{i=0}^{d} \alpha_i(\mathbf{x}_N) b(\mathbf{x}_1, ..., \mathbf{x}_{N-1}, \mathbf{a}_i). \tag{43}$$

This lemma is useful for reducing the number of variables in the following proof using induction.

Proposition 5.2 *Let $f_1 \in \Pi_{n_1}(\mathbb{R}^d)$ and $f_2 \in \Pi_{n_2}(\mathbb{R}^d)$ with $n_i \geq 1$. Then, the product $f_1 f_2 \in \Pi_{n_1+n_2}(\mathbb{R}^d)$ and its blossom is given by*

$$\mathcal{P}(f_1 f_2)(\mathbf{U}) = \frac{n_1! n_2!}{(n_1+n_2)!} \sum_{(I_1,I_2)\in S(n_1,n_2)} \mathcal{P}(f_1)(\mathbf{p}_{I_1}(\mathbf{U}))\, \mathcal{P}(f_2)(\mathbf{p}_{I_2}(\mathbf{U})) \tag{44}$$

where $\mathbf{U} := (\mathbf{u}_1, ..., \mathbf{u}_{n_1+n_2})$ with $\mathbf{u}_i \in \mathbb{R}^d$ for all $i \in [\![1, n_1+n_2]\!]$ and

$$S(n_1, n_2) := \{(I_1, I_2) \in [\![1, n_1+n_2]\!]^2 : I_1 \cup I_2 = [\![1, n_1+n_2]\!],\, |I_i| = n_i\}. \tag{45}$$

PROOF. Our proof is done by using induction on $N := n_1 + n_2$ and by noting that $S(n_1, n_2)$ can be partitioned into $S_1(n_1, n_2) \cup S_2(n_1, n_2)$ where

$$\begin{aligned} S_1(n_1,n_2) &:= \{(I_1,I_2) : I_1 = I_1^* \cup \{N\} \quad \text{where} \quad (I_1^*, I_2) \in S(n_1-1, n_2)\}\,, \\ S_2(n_1,n_2) &:= \{(I_1,I_2) : I_2 = I_2^* \cup \{N\} \quad \text{where} \quad (I_1, I_2^*) \in S(n_1, n_2-1)\}\,. \end{aligned}$$

Consider the simplest case $N = 2$ i.e. $n_1 = n_2 = 1$. Since they are linear polynomials, we have $\mathcal{P}(f_i)(\mathbf{u}) = f_i(\mathbf{u})$ for $i = 1$ or $i = 2$. Note that the following function is multiaffine and symmetric

$$F(\mathbf{u}_1, \mathbf{u}_2) := \frac{1}{2}[f_1(\mathbf{u_1})f_2(\mathbf{u_2}) + f_1(\mathbf{u_2})f_2(\mathbf{u_1})], \tag{46}$$

Additionally, we have $F(\mathbf{u}, \mathbf{u}) = f_1(\mathbf{u})f_2(\mathbf{u})$. Thus, F is the blossom of $f_1 f_2$. Since, F verifies (44), the claim holds for $N = 2$. Suppose that it is correct for $N - 1$ and let us prove it for N. Introduce

$$R := \sum_{(I_1,I_2)\in S(n_1,n_2)} \mathcal{L}(I_1, I_2), \quad \text{where} \tag{47}$$

$$\mathcal{L}(I_1, I_2) := \mathcal{P}(f_1)(\mathbf{p}_{I_1}(\mathbf{U}))\, \mathcal{P}(f_2)(\mathbf{p}_{I_2}(\mathbf{U})). \tag{48}$$

Let us show that

$$R = \frac{(n_1 + n_2)!}{n_1! n_2!} \mathcal{P}(f_1 f_2). \tag{49}$$

We have

$$R = R_1 + R_2 \qquad \text{where} \qquad R_i := \sum_{(I_1,I_2)\in S_i(n_1,n_2)} \mathcal{L}(I_1, I_2). \tag{50}$$

Because of the former lemma, we have for $I_1 \cap I_2 = \emptyset$

$$\begin{aligned} R_1 &= \sum_{(I_1,I_2)\in S_1} \mathcal{P}(f_1)(\mathbf{p}_{I_1}(\mathbf{u}_1, ..., \mathbf{u}_N))\mathcal{P}(f_2)(\mathbf{p}_{I_2}(\mathbf{u}_1, ..., \mathbf{u}_N)) \\ &= \sum_{(I_1^*,I_2)\in S(n_1-1,n_2)} \sum_{i=0}^{d} \alpha_i(\mathbf{u}_N)\mathcal{P}(f_1)(\mathbf{p}_{I_1^*}(\mathbf{u}_1, ..., \mathbf{u}_{N-1}), \mathbf{a}_i)\mathcal{P}(f_2)(\mathbf{p}_{I_2}(\mathbf{u}_1, ..., \mathbf{u}_{N-1})). \end{aligned}$$

The last factor is obtained beause $\mathbf{p}_{I_2}(\mathbf{u}_1, ..., \mathbf{u}_N) = \mathbf{p}_{I_2}(\mathbf{u}_1, ..., \mathbf{u}_{N-1})$ because I_2 does contain $\mathbf{u}_N$. Since the blossom and the directional derivative is related by

$$\mathcal{P}(D_{\mathbf{a}} f_1)(\mathbf{v}_1, ..., \mathbf{v}_{n_1-1}) = \mathcal{P}(f_1)(\mathbf{v}_1, ..., \mathbf{v}_{n_1-1}, \mathbf{a}), \tag{51}$$

we obtain from the hypothesis of induction

$$\begin{aligned} R_1 &= \sum_{(I_1^*,I_2)\in S(n_1-1,n_2)} \sum_{i=0}^{d} \alpha_i(\mathbf{u}_N)\mathcal{P}(D_{\mathbf{a}_i} f_1)(\mathbf{p}_{I_1^*}(\mathbf{u}_1, ..., \mathbf{u}_{N-1}))\mathcal{P}(f_2)(\mathbf{p}_{I_2}(\mathbf{u}_1, ..., \mathbf{u}_{N-1})) \\ &= \frac{(n_1 + n_2 - 1)!}{(n_1 - 1)! n_2!} \sum_{i=0}^{d} \alpha_i(\mathbf{u}_N)\mathcal{P}((D_{\mathbf{a}_i} f_1) f_2). \end{aligned}$$

We can do the same computation for R_2. Define $\delta_i := n_1\alpha_i/(n_1 + n_2)$ and $\delta_{d+i} := n_2\alpha_i/(n_1 + n_2)$ for $i = 0, ..., d$. Since blossoms are additive, we obtain

$$
\begin{aligned}
R &= \frac{(n_1+n_2)!}{n_1!n_2!}\sum_{i=0}^{d}\delta_i(u_N)\mathcal{P}[(D_{\mathbf{a}_i}f_1)f_2+(D_{\mathbf{a}_i}f_2)f_1] \\
&= \frac{(n_1+n_2)!}{n_1!n_2!}\sum_{i=0}^{d}\delta_i(u_N)\mathcal{P}[D_{\mathbf{a}_i}(f_1f_2)] \\
&= \sum_{i=0}^{d}\frac{(n_1+n_2)!}{n_1!n_2!}\delta_i(\mathbf{u}_N)\mathcal{P}(f_1f_2)(\mathbf{u}_1,...,\mathbf{u}_{N-1},\mathbf{a_i}) \\
&= \frac{(n_1+n_2)!}{n_1!n_2!}\mathcal{P}(f_1f_2)(\mathbf{u}_1,...,\mathbf{u}_N).
\end{aligned}
$$

■

Since we have assumed in Section 2 that the blending functions p_i are polynomials, each term in the function $\mathbf{X}$ from relation (6) can be expressed in B-patch by using the former Theorem and the pyramid algorithm. Additionally, since the determinant function is multiaffine, the function of interest $\mathcal{J}$ from relation (26) can be written in B-patch form by using the same idea.

6 Improving the efficiency

We focus in this section on improving the efficiency of the former algorithm. When the degree n is large and we have many control points, the former method might become computationally expensive. As a consequence, we want to show here a method of reducing the degree n while still achieving regularity check. In our next description, we will need the Jacobi polynomials [24] that are given by the Rodrigues relation:

$$P_n^{(\alpha,\beta)}(t) = \frac{1}{2^n n!}(t-1)^{-\alpha}(t+1)^{-\beta}\left(\frac{d}{dt}\right)^n[(t-1)^{n+\alpha}(t+1)^{n+\beta}] \qquad (52)$$

which can be explicitly given by

$$P_n^{(\alpha,\beta)}(t) = \frac{\Gamma(n+\alpha+1)}{\Gamma(\alpha+1)n!}F\begin{bmatrix} -n, n+\alpha+\beta+1 & ; \frac{1-t}{2} \\ \alpha+1 & \end{bmatrix} \qquad (53)$$

where $F(t) := {}_2F_1(t)$ is the hypergeometric [1] function. The Jacobi polynomials $P_n^{(\alpha,\beta)}$ can be evaluated recursively by the three-term relation

$$S_n P_{n+1}^{(\alpha,\beta)}(t) = T_n(t)P_n^{(\alpha,\beta)}(t) + U_n P_{n-1}^{(\alpha,\beta)}(t) \qquad (54)$$

where $S_n := 2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)$ and $U_n(t) := 2(n+1)(n+\beta)(2n+\alpha+\beta+2)$ while $T_n(t) := (2n+\alpha+\beta+1)(\alpha^2-\beta^2)+(2n+\alpha+\beta)_3 t$ in which the Pochhammer symbol [1] is used. Since the Jacobi polynomials are defined on $[-1,+1]$ but we need results in $[0,1]$, we introduce the modified Jacobi polynomials

$$J_n^{(\alpha,\beta)}(t) := P_n^{(\alpha,\beta)}(2t-1) \qquad \forall\, t\in[0,1]. \qquad (55)$$

In our next discussion, we will need only Jacobi polynomials for the ultraspherical case $\alpha=\beta$ where we denote $P_n^{(\alpha)} := P_n^{(\alpha,\alpha)}$ and $J_n^{(\alpha)} := J_n^{(\alpha,\alpha)}$. Additionally, we use only the case where $\alpha > (1+\sqrt{2})/4$ and $\alpha\in\mathbb{N}$. The following result is valid for any dimension d but we will only use it later for $d=3$. We need a norm for the polynomials on the unit simplex $\Delta^d := \{\mathbf{x}=(x_1,\cdots,x_d):\quad x_i\geq 0,\quad 1-|\mathbf{x}|\geq 0\}$ where $|\mathbf{x}|$ denotes the l^1 norm of $\mathbf{x}$. Consider a function f which is a multivariate polynomial of degree n:

$$f(x_1,\cdots,x_d)=\sum_{|\boldsymbol{\gamma}|\le n} b_{\boldsymbol{\gamma}} x_1^{\gamma_1}\cdots x_d^{\gamma_d} \qquad \mathbf{x}=(x_1,\cdots,x_d)\in\Delta_d. \tag{56}$$

Let us introduce for the polynomial f the quantity

$$\|f\|_{\Delta^d} := \max_{\mathbf{x}\in\Delta^d}\left[\prod_{i=1}^{d}(1-x_i)^k x_i^k\right]|f(\mathbf{x})|. \tag{57}$$

By using continuity argument, we can show that $\|\cdot\|_{\Delta^d}$ defines a norm on the polynomials of Δ^d. In order to reduce the computational cost, we would like to find $\tilde{f}$ which has the same shape as f but which has a lower degree $m<n$. Without loss of generality we suppose $m=n-1$. The definition of $\tilde{f}$ will be done by using the Jacobi polynomial and the error $\|f-\tilde{f}\|_{\Delta^d}$ will be analyzed. Consider the case $k\in\mathbb{N}$ such that $k\ge\lceil\frac{\alpha}{2}+\frac{1}{4}\rceil$.

Theorem 6.1 *Consider a multivariate polynomial $f\in\Pi_n(\mathbb{R}^d)$ with bounded n-th derivatives such that for all $|\boldsymbol{\gamma}|=n$, we have:*

$$\frac{1}{\boldsymbol{\gamma}!}|\partial_{\boldsymbol{\gamma}}f(\mathbf{x})|=\frac{1}{\gamma_1!\cdots\gamma_d!}\left|\frac{\partial^{\gamma_1}}{\partial x_1^{\gamma_1}}\cdots\frac{\partial^{\gamma_d}}{\partial x_d^{\gamma_d}}f(x_1,\cdots,x_d)\right|\le C \qquad \forall\mathbf{x}\in\Delta_d. \tag{58}$$

There is a polynomial $\tilde{f}\in\Pi_{n-1}(\mathbb{R}^d)$ of the following form

$$\tilde{f}(\mathbf{x}):=\sum_{|\boldsymbol{\gamma}|\le n-1}c_{\boldsymbol{\gamma}}\prod_{j=1}^{d}x_j^{\gamma_j}-\sum_{|\boldsymbol{\gamma}|=n}c_{\boldsymbol{\gamma}}\sum_{q=1}^{d}R_{\gamma_q}(x_q)\prod_{j=1}^{q-1}x_j^{\gamma_j}\prod_{j=q+1}^{d}(x_j^{\gamma_j}-R_{\gamma_j}(x_j)) \tag{59}$$

such that the error is given by

$$\|f-\tilde{f}\|_{\Delta^d}\le K\frac{1}{2^{2n}}\sum_{|\boldsymbol{\gamma}|=n}\prod_{i=1}^{d}2^{-2k+\alpha+0.5}\frac{(1+\alpha/\max\{1,\gamma_i\})}{\binom{2\gamma_i+2\alpha}{\gamma_i}}, \tag{60}$$

where the constant K depends only on α.

PROOF. Since the leading coefficient of the Jacobi polynomial $P_q^{(\alpha)}$ is $l_q:=\prod_{j=1}^{q}(q+j+2\alpha)/2j$, the polynomial $\tilde{J}_q^{(\alpha)}:=J_q^{(\alpha)}/(2^q l_q)$ which is a scaled form of (55) is monic. As a consequence, by the Taylor expansion of f at $\mathbf{0}=(0,\cdots,0)$, the following multivariate polynomial is of total degree $(n-1)$:

$$\tilde{f}(\mathbf{x})=\sum_{|\boldsymbol{\gamma}|<n}\frac{1}{\boldsymbol{\gamma}!}\partial_{\boldsymbol{\gamma}}f(\mathbf{0})\mathbf{x}^{\boldsymbol{\gamma}}+\sum_{|\boldsymbol{\gamma}|=n}\frac{1}{\boldsymbol{\gamma}!}\partial_{\boldsymbol{\gamma}}f(\mathbf{0})\prod_{i=1}^{d}(x_i^{\gamma_i}-\tilde{J}_{\gamma_i}^{(\alpha)}(x_i)). \tag{61}$$

Because of the monicity of the scaled Jacobi $\tilde{J}_q^{(\alpha)}$, we can introduce $R_q(t):=t^q-\tilde{J}_q^{(\alpha)}(t)$. Let us denote $c_{\boldsymbol{\gamma}}:=\frac{1}{\boldsymbol{\gamma}!}\partial_{\boldsymbol{\gamma}}f(0,...,0)$. The former equation yields

$$\tilde{f}(\mathbf{x})=\sum_{|\boldsymbol{\gamma}|\le n}c_{\boldsymbol{\gamma}}\prod_{j=1}^{d}x_j^{\gamma_j}-\sum_{|\boldsymbol{\gamma}|=n}c_{\boldsymbol{\gamma}}\prod_{j=1}^{d}(x_j^{\gamma_j}-R_{\gamma_k}(x_j)). \tag{62}$$

After developing the last summation, we obtain (59). It is known [13, 11] that for fixed α, we have $t^{\alpha/2+1/4}(1-t)^{\alpha/2+1/4}P_m^{(\alpha)}(t)\le C(1+\alpha/\max\{1,m\})$. Therefore, we have for $t\in[0,1]$

$$\begin{aligned}(1-t)^k t^k|P_{\gamma_i}^{(\alpha)}(t)| &\le \max_{t\in[0,1]} (1-t)^{k-\alpha/2-1/4}t^{k-\alpha/2-1/4}\\ &\qquad \max_{t\in[0,1]}(1-t)^{\alpha/2+1/4}t^{\alpha/2+1/4}|P_{\gamma_i}^{(\alpha)}(t)|\\ &\le K2^{-2k+\alpha+1/2}\ (1+\alpha/\max\{1,\gamma_i\}).\end{aligned}$$

Since the n-th derivatives are bounded, we deduce from the former inequality

$$\begin{aligned}
\|f-\tilde{f}\|_{\Delta^d} &\le K\sum_{|\boldsymbol{\gamma}|=n}\max_{\mathbf{x}\in\Delta^d}\prod_{i=1}^{d}(1-x_i)^k x_i^k|\tilde{J}_{\gamma_i}^{(\alpha)}(x_i)| \\
&\le K\sum_{|\boldsymbol{\gamma}|=n}\prod_{i=1}^{d}\max_{x_i\in[0,1]}\frac{(1-x_i)^k x_i^k|J_{\gamma_i}^{(\alpha)}(x_i)|}{2^{2\gamma_i}\Pi_{j=1}^{\gamma_i}(\gamma_i+j+2\alpha)/4j} \\
&\le K\sum_{|\boldsymbol{\gamma}|=n}\frac{1}{2^{2(\sum_{i=1}^{d}\gamma_i)}}\prod_{i=1}^{d}\frac{2^{-2k+\alpha+1/2}(1+\alpha/\max\{1,\gamma_i\})}{\Pi_{j=1}^{\gamma_i}(\gamma_i+j+2\alpha)/4j} \\
&\le K\frac{1}{2^{2n}}\sum_{|\boldsymbol{\gamma}|=n}\prod_{i=1}^{d}\frac{2^{-2k+\alpha+1/2}(1+\alpha/\max\{1,\gamma_i\})}{\Pi_{j=1}^{\gamma_i}(\gamma_i+j+2\alpha)/4j}.
\end{aligned}$$

■

Note that in the previous analysis, the multivariate function f from relation (56) is given in monomial basis but we need results in B-patch structure as discussed in the former sections. That holds also for the degree reduced polynomial $\tilde{f}$. In order to obtain B-patch representation, one can represent the Jacobi polynomials in terms of Bézier as discussed in [15]:

$$P_n^{(\alpha)}(t)=\sum_{i=0}^{n}(-1)^{n-i}\frac{\binom{n+\alpha}{i}\binom{n+\alpha}{n-i}}{\binom{n}{i}}B_i^n(t). \tag{63}$$

Then, one needs only to transform that in the appropriate knot arrangement as we described in (21). Afterwards, the function $\tilde{f}$ can be represented in B-patch of degree $(n-1)$ by using the pyramid algorithm of Section 3. The B-patch coefficient of $\tilde{f}$ can be deduced from the blossom and the pyramid algorithm by applying formula (20).

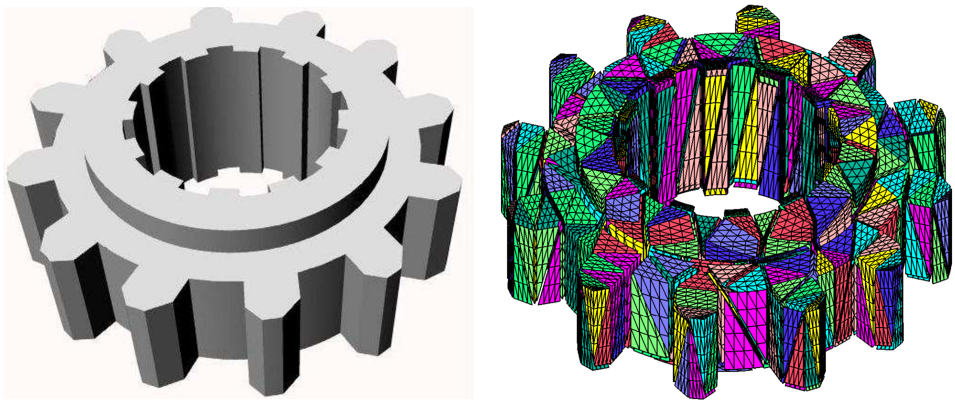
Degree n	$\varepsilon_n(1,2)$	$\varepsilon_n(2,2)$	$\varepsilon_n(3,2)$
4	8.418543e-002	1.300437e-001	2.424409e-001
7	2.852678e-003	4.010734e-003	6.843747e-003
10	7.717651e-005	1.021698e-004	1.632607e-004
13	1.849422e-006	2.347735e-006	3.565172e-006
19	7.394503e-010	8.735727e-010	1.200853e-009
22	1.144845e-011	1.336140e-011	1.802953e-011
31	3.673017e-017	4.402754e-017	6.183465e-017
40	1.170561e-022	1.481952e-022	2.241786e-022

Table 1: Errors $\varepsilon_n(\alpha,k)$ for degree reductions

Remark 6.2 *Note that a very similar result can be obtained for L^∞ norm instead of the norm $\|\cdot\|_{\Delta^d}$ but the proof is more involved. The estimation on the right hand side of equation (60) can be precomputed:*

$$\varepsilon_n:=\frac{1}{2^{2n}}\sum_{|\boldsymbol{\gamma}|=n}\prod_{i=1}^{d}2^{-2k+\alpha+0.5}\frac{(1+\alpha/\max\{1,\gamma_i\})}{\binom{2\gamma_i+2\alpha}{\gamma_i}}. \tag{64}$$

Some computer outputs about the values of ε_n for various values of n can be found in Table 1. The assumption in (58) is not restrictive in practice because the partial



(a) (b)

Figure 7: Splitting of a mechanical part

derivatives are divided by $\boldsymbol{\gamma}!$ which is supposed to be very large for $|\boldsymbol{\gamma}| = n$. Although we have only presented degree reduction from n to $n-1$, the results can be generalized to more general cases. If we use several degree reductions from n to $m < n$, then we replace ε_n in relation (60) to

$$\mu(n,m) := \varepsilon_n + \varepsilon_{n-1} + ... + \varepsilon_m. \tag{65}$$

Now that we have gained enough insight for degree reduction, we want to describe its usefulness for our former technique. In order to improve the efficiency of the method which we described in Section 4, we use multiple degree reductions in the following way. After each subdivision step, we check if by reducing the degree from n to some $m < n$, the error introduced for each resulting patch verifies

$$\mu(n,m) < \lambda\delta \tag{66}$$

where δ is some user specified parameter in $]0,1[$. If so, then we apply a process of degree reduction. The former relations in (35) become therefore

$$\mathcal{P}(f)(\tilde{T}) = \mathcal{P}(f)(T) - \mu(n,m) + \mathcal{O}\left(\frac{1}{2^{2n}} \max_{p \neq q} \|\mathbf{t}_p^{0,0} - \mathbf{t}_q^{0,0}\|^2\right) \tag{67}$$

Hence, the former property about subdivision still remains valid by using the following inequality in place of (37)

$$b_{\boldsymbol{\beta}}^{n,k} \geq (1-\lambda)\delta + \mathcal{O}\left(\frac{1}{2^n} h\right)^2. \tag{68}$$

7 Conclusion and Discussion

We have mainly presented a theoretical method for efficiently checking whether a tetrahedral transfinite interpolation is regular. The use of subdivision is helpful to devise an adaptive method. In order to reduce the computational cost of the method, we use degree reduction. That is done by utilizing the scaled Jacobi polynomials. There are mainly two types of geometric data in CAGD. There are those which are composed of algebraic primitives such as planes, cylinders, spheres. On the other hand, there are free-form surfaces which do not have any structural features. Free-form surfaces are often seen in computer medicine including vertebral cuts, bones, cardiac parts and skulls. Rhino3D is CAD system which is very well suited for such medical models. Those organic models are extremely free-form and it is impossible to describe them in term of simple algebraic parts. Other forms of geometric data are molecular data which are mainly composed of spherical parts. Yet other types of geometric data are CAD data which are mixture of algebraic parts and free-form ones. For instance, we can usually see that in car coachworks where most part are free form. The method that is presented here is good for most types of geometries because we do not put too much restrictions on the type of the boundary faces. It is only partially included in our hierarchical mesh generator illustrated in Fig. 7. Therefore, we plan to completely include our theoretical results in our implementations in the near future.

Acknowledgment

The author is very grateful to Prof. Brunnett from the Technical University of Chemnitz who has introduced him to the topic of Geometric Modeling and CAD. He thanks also Prof. Schneider from the University of Berlin for supporting his research at the Institute of Computer Science in the University of Kiel. Last but not least, the author wants to acknowledge the help of Prof. Harbrecht from the University of Bonn who allowed him to pursue his current Post-Doctoral research.

References

[1] T. Chihara, An introduction to orthogonal polynomials, Gordon and Breach, 1978.

[2] A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet methods for elliptic operator equations-convergence rates, Math. Comp. **70** (2000) 27–75.

[3] CFD General Notation System (Standard Interface Data Structures).

[4] W. Dahmen, C. Michelli, H. Seidel, Blossoming begets B-spline bases built better by B-patches, Math. Comput. **59**, No. 199 (1992) 97–115 .

[5] W. Dahmen, A. Kunoth, Adaptive wavelet methods for linear–quadratic elliptic control problems: convergence rates, SIAM J. Contr. Optim. **43**, No. 5 (2005) 1640–1675.

[6] W. Dahmen, C. Micchelli, Continuous refinement equations and subdivision, Adv. Comput. Math. **1**, No. 1 (1993) 1–37.

[7] H. Edelsbrunner, N. Shah, Triangulating topological spaces, Int. J. Comput. Geom. Appl. **7**, No. 4 (1997) 365–378.

[8] H. Harbrecht, M. Randrianarivony, From Computer Aided Design till wavelet BEM, Berichtsreihe des mathematischen Seminars, Preprint **07-18**, Uni Kiel, October 2007 (submitted to Comput. Vis. Sci.).

[9] H. Harbrecht, R. Schneider, Wavelet Galerkin schemes for boundary integral equation-implementation and quadrature, SIAM J. Sci. Comput. **27**, No. 4 (2006) 1347–1370.

[10] H. Kim, Y. Ahn, Good degree reduction of Bézier curves using Jacobi polynomials, Pergamon **40** (2000) 1205–1215.

[11] I. Krasikov, An upper bound on Jacobi polynomials, J. Approximation Theory **149**, No. 2 (2007) 116–130.

[12] C. Micchelli, T. Sauer, Y. Xu, A construction of refineable sets of interpolating wavelets, Result Math. **34**, No. 3-4 (1998) 359–372.

[13] P. Nevai, T. Erdélyi, A. Magnus, Generalized Jacobi weights, Christoffel functions, and Jacobi polynomials, SIAM J. Math. Anal. **25**, No. 2 (1994) 602–614.

[14] S. Owen, M. Staten, S. Canann, and S. Saigal, Q-Morph: An indirect approach to advancing front quad meshing, Int. J. Numer. Methods Eng. **44**, No. 9 (1999) 1317–1340.

[15] A. Rababah, Jacobi-Bernstein basis transformation, Comput. Methods Appl. Math. **4**, No. 2 (2004) 206–214.

[16] L. Ramshaw, Blossoms are polar forms, Comput. Aided Geom. Des. **6**, No. 4 (1989) 323–358.

[17] M. Randrianarivony, Geometric processing of CAD data and meshes as input of integral equation solvers. *PhD thesis*, Technische Universität Chemnitz, 2006.

[18] M. Randrianarivony, G. Brunnett, Molecular surface decomposition using geometric techniques, In: Proc Conf. Bildverarbeitung für die Medizine, Berlin, 197–201, 2008.

[19] P. Sampl, Medial axis construction in three dimensions and its application to mesh generation, J. Eng. Comput. (Lond.) **17**, No. 3 (2001) 234–248.

[20] R. Schneider, Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur Lösung grosser vollbesetzter Gleichungssysteme, Teubner, Stuttgart, 1998.

[21] H. Seidel, Polar forms for geometrically continuous spline curves of arbitrary degree, ACM Trans. Graph. **12**, No. 1 (1993) 1–34.

[22] H. Seidel, Symmetric recursive algorithms for surfaces: B-patch and the de Boor algorithm for polynomials over triangles, Contruct. Approx. **7**, No. 2 (1991) 257–279.

[23] K. Strøm, Products of B-patches, Numer. Algorithms **4**, No. 4 (1993) 323–337.

[24] G. Watson, Approximation theory and numerical methods, John Wiley & Sons, Chichester, 1980.