18th International Conference on the Application of Computer
Science and Mathematics in Architecture and Civil Engineering
K. Gürlebeck and C. Könke (eds.)
Weimar, Germany, 07–09 July 2009

ON THE GENERATION OF HIERARCHICAL MESHES FOR MULTILEVEL FEM AND BEM SOLVERS FROM CAD DATA

Maharavo Randrianarivony*

**Institute for Numerical Simulation*
University of Bonn
Wegelerstrasse 6
Bonn 53115, Germany
E-mail: randrian@ins.uni-bonn.de

Keywords: Nested mesh, geometric processing, CAD, BEM, FEM, hierarchy.

Abstract. *As numerical techniques for solving PDE or integral equations become more sophisticated, treatments of the generation of the geometric inputs should also follow that numerical advancement. This document describes the preparation of CAD data so that they can later be applied to hierarchical BEM or FEM solvers.*

For the BEM case, the geometric data are described by surfaces which we want to decompose into several curved foursided patches. We show the treatment of untrimmed and trimmed surfaces. In particular, we provide prevention of smooth corners which are bad for diffeomorphism. Additionally, we consider the problem of characterizing whether a Coons map is a diffeomorphism from the unit square onto a planar domain delineated by four given curves. We aim primarily at having not only theoretically correct conditions but also practically efficient methods.

As for FEM geometric preparation, we need to decompose a 3D solid into a set of curved tetrahedra. First, we describe some method of decomposition without adding too many Steiner points (additional points not belonging to the initial boundary nodes of the boundary surface). Then, we provide a methodology for efficiently checking whether a tetrahedral transfinite interpolation is regular. That is done by a combination of degree reduction technique and subdivision.

Along with the method description, we report also on some interesting practical results from real CAD data.

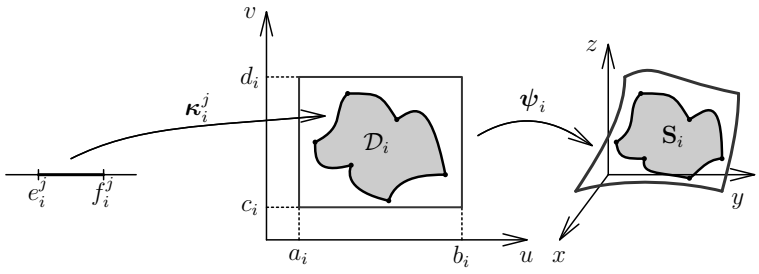


Figure 1: The boundary of $\mathcal{D}_i \subset \mathbf{R}^2$ is the image of several curves $\boldsymbol{\kappa}_i^j$. Apply $\boldsymbol{\psi}_i$ to have the trimmed surface $\mathbf{S}_i$.

1 INTRODUCTION

Traditionally, solving PDE and integral equations have been done on a very fine triangular or tetrahedral mesh. Over the last periods, the use of hierarchical methods has emerged and developed quickly. Hierarchical settings have been demonstrated to be numerically very efficient because they give rise to subdivision algorithms. Such a hierarchical setting produces in general good accuracy with low computational cost [2, 14]. Unfortunately, their applications on real geometric models cannot seem to have attained maturity if compared to traditional mesh-based approaches. We address here the problem of processing CAD data for use in hierarchical BEM and FEM. This document summarizes our earlier works [6, 10, 11, 12] about processing CAD data for numerical applications and it supplements them with new results pertaining to tetrahedral cases. The structure of this paper is as follows. It starts with a description of the CAD inputs in Section 2. Afterwards, we concentrate on the BEM treatment which consists of decomposition and regularity in Section 3. We complete the BEM case by presenting some practical results. As for FEM modeling, we consider the generation of curved tetrahedral meshes in Section 4. That will be followed by regularity verification for a curved tetrahedron in Section 5. Finally, we report on some CAD results from IGES files for FEM decomposition.

2 CAD REPRESENTATION AND PROBLEM FORMULATION

2.1 Description of the input CAD models

The initial CAD input is a solid Ω bounded by a closed surface $\Gamma \subset \mathbf{R}^3$ that is the union of M trimmed [1] parametric surfaces $\mathbf{S}_1, \cdots, \mathbf{S}_M$ defined on the domains $\mathcal{D}_1, \cdots, \mathcal{D}_M$ which are multiply connected regions in $\mathbf{R}^2$. The external and internal (when relevant) boundary curves of each domain $\mathcal{D}_i$ are supposed to be composite curves. That is, there are univariate smooth functions $\boldsymbol{\kappa}_i^j$ defined on $[e_i^j, f_i^j]$ such that $\partial\mathcal{D}_i = \bigcup_j \mathrm{Im}(\boldsymbol{\kappa}_i^j)$. We suppose further that the parametric functions defining $\mathbf{S}_i$

$$\boldsymbol{\psi}_i : \mathcal{D}_i \longrightarrow \mathbf{S}_i \tag{1}$$

are bijective, regular and smooth [11]. A graphical illustration of this formulation can be found in Fig. 1. Furthermore, we need that the bounding curves $\boldsymbol{\kappa}_i^j$ are sufficiently smooth, regular and without zero angles. More precisely, we assume that for all i, j, the tangents satisfy $\dot{\boldsymbol{\kappa}}_i^j(\tau) \neq \mathbf{0}$. Additionally, in order to forbid cusps or zero angles inside each $\mathcal{D}_i$, we suppose that if the terminating point of $\boldsymbol{\kappa}_i^{j_1}$ and the starting point of $\boldsymbol{\kappa}_i^{j_2}$ coincide, we must have

$$\lim_{t \to (f_i^{j_1})^-} \dot{\boldsymbol{\kappa}}_i^{j_1}(t) \neq -\lambda \lim_{t \to (e_i^{j_2})^+} \dot{\boldsymbol{\kappa}}_i^{j_2}(t) \qquad \forall\, \lambda > 0. \tag{2}$$

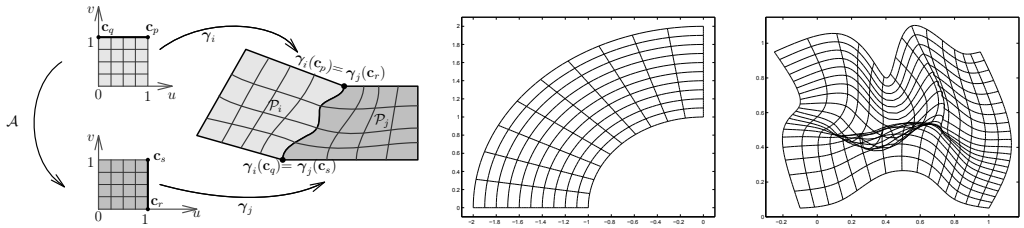


Figure 2: (a)Matching condition with affine transformation $\mathcal{A}$. (b)Regular Coons patch. (c)Undesired overspill phenomenon.

2.2 Problem setting for BEM and FEM

For preparation of BEM geometry, our objective is to tessellate the surface Γ into a collection of four-sided patches Γ_i, i.e., $\Gamma = \cup_i \Gamma_i$, where the splitting is conforming [6]. We need also some regular (differentiable and the Jacobian matrix has maximal rank) functions $\boldsymbol{\gamma}_i$ such that $\Gamma_i = \boldsymbol{\gamma}_i([0,1]^2)$. Additionally, we require global continuity meaning that for two adjacent patches Γ_i and Γ_j, there is a bijective, affine mapping $\mathcal{A} : [0,1]^2 \to [0,1]^2$ such that for all $\mathbf{x} = \boldsymbol{\gamma}_i(s)$ on a common edge of Γ_i and Γ_j it holds that $\boldsymbol{\gamma}_i(s) = (\boldsymbol{\gamma}_j \circ \mathcal{A})(s)$. That is, $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_j$ coincide pointwise at common edges up to some reorientation. Practically, global continuity means that the images by $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_j$ of u-constant and v-constant isolines match well at the interface (Fig. 2(a)). Unfortunately, we are not able to solve the problem of global continuity exactly and we could not find in the literature any method which can do that. As a consequence, we will only show how to solve that problem numerically without too much computational cost.

As for FEM geometric preparation, we intend to decompose the solid Ω into a set of curved tetrahedra T_i which again form a conforming decomposition. We want also to find functions $\boldsymbol{\gamma}_i : \Delta^3_{\text{ref}} \longrightarrow T_i$ which are regular. Additionally, we need a global continuity as follows. Suppose that the curved tetrahedra T_i and T_j share a curved triangular face t. For each $\mathbf{x} \in t$, its preimages $\boldsymbol{\gamma}_i^{-1}(\mathbf{x})$ and $\boldsymbol{\gamma}_j^{-1}(\mathbf{x})$ should have the same barycentric coordinates on their respective triangular face on the unit tetrahedron Δ^3_{ref}.

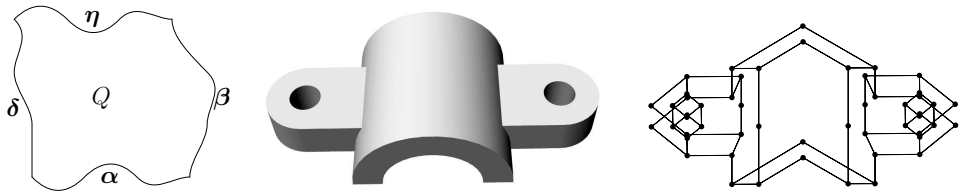
The results of the geometric operations can be graphically illustrated by Fig. 6 and Fig. 12 where the grids represent the images by $\boldsymbol{\gamma}_i$ of a uniform grid on $[0,1]^2$ or Δ^3_{ref}.

3 GEOMETRIC PROCESSING FOR BEM

3.1 Decomposition into four-sided patches

A complete detail of decomposing a 3D-model is beyond the scope of this paper. Therefore, we will summarize only the main steps and point out the principal difficulties [11]. First of all, we approximate the curved boundaries of $\{\mathbf{S}_i\}$ by straight line segments separated by nodes $\{X_k\} \subset \mathbf{R}^3$ as in Fig. 3(c). In order to achieve that approximation while having conforming splitting in mind, we create planar polygonalizations of $\{\mathcal{D}_i\}_{i=1}^M$ which amount to doing the following. For each trimmed surface $\mathbf{S}_i$, we generate a polygon $P^{(i)}$ whose nodes $\mathbf{v}_k^{(i)}$ are taken from the curved boundary of the 2D domain $\mathcal{D}_i$. We have to make sure that for two adjacent different surfaces $\mathbf{S}_i$ and $\mathbf{S}_j$ sharing a curve $\mathbf{C}$, if $\boldsymbol{\psi}_i(\mathbf{v}_k^{(i)}) \in \mathbf{C}$, then there must exist a vertex $\mathbf{v}_l^{(j)} \in P^{(j)}$ such that

$$\boldsymbol{\psi}_i(\mathbf{v}_k^{(i)}) = \boldsymbol{\psi}_j(\mathbf{v}_l^{(j)}). \tag{3}$$



(a) (b) (c)

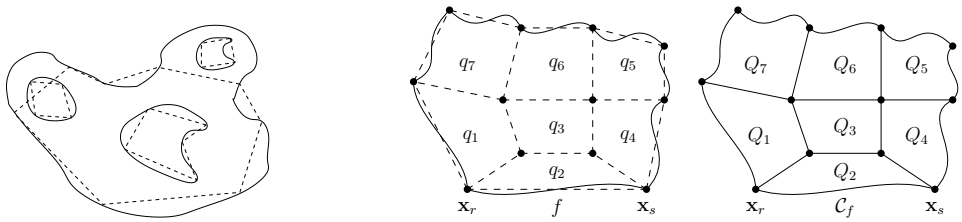
Figure 3: (a)Four-sided region. (b)Input 3D model. (c)Polygonal approximation.

Let us note that if we take too few vertices, the resulting polygon $P^{(i)}$ may have imperfections such that its edges do not form an admissible polygon as illustrated in Fig. 4(a). But if the polygonal approximation is too fine, then it results in overly many four-sided surfaces. As a consequence, one has to split the curved edges adaptively while trying to maintain relation (3) which involves some preimage computations. Let us emphasize that only polygons having an *even* number of boundary vertices can be decomposed into quadrilaterals. It is not straightforward to convert odd faces into even ones inside a closed surface with arbitrary genus. One should assemble the adjacency graph which is used in the Dijkstra algorithm to search for the shortest path joining two odd polygons in order that the number of additional nodes to be inserted are not too many. We could theoretically prove that the number of odd faces must be even for a closed model and that the odd faces can be converted to even ones pairwise.

Our main approach consists in splitting the 2D regions $\mathcal{D}_i$ into four-sided regions $Q_{k,i} \subset \mathbf{R}^2$ such that $\mathcal{D}_i = \bigcup_k Q_{k,i}$. The four-sided patches $\mathcal{P}_k$ are therefore the images by $\boldsymbol{\psi}_i$ of the 2D domains $Q_{k,i}$

$$\mathcal{P}_k = \boldsymbol{\psi}_i(Q_{k,i}). \tag{4}$$

As for the decomposition into $Q_{k,i}$, we consider the polygon $P^{(i)}$ which we decompose into a set of convex quadrilaterals $q_{k,i}$. The four-sided domains $Q_{k,i}$ are obtained from $q_{k,i}$ by replacing the straight boundary edges of $q_{k,i}$ by the corresponding curve portion of $\mathcal{D}_i$ as illustrated in Fig. 4(b) and Fig. 4(c). In the decomposition of a polygon $P^{(i)}$ into quadrilaterals $q_{k,i}$, we use only the preimages $\boldsymbol{\psi}_i^{-1}(X_k)$ of the nodes $\{X_k\}$ as boundary vertices. That is, we do not use any additional boundary nodes in the course of the quadrangulation (usual term for "quadrilateral mesh" or "process of quadrilating") process. We have developed in [11] an approach that decomposes a polygon with n boundary vertices into $\mathcal{O}(n)$ convex quadrilaterals. There are two main difficulties in that process. First, converting nonconvex quadrilaterals into convex ones requires many cases to be handled individually. Second, finding cuts connecting an internal boundary and the exterior boundary of a multiply connected polygon is complicated. The process of replacing a boundary edge with the corresponding curve can generate three serious problems. First, it is possible that the curve intersects an internal edge causing a boundary interference. The second problem is that some corners in a four-sided region $Q_{k,i}$ might be smoothened out. Third, it is possible that the corresponding Coons patch is not regular [11]. In those cases, we have to make a polygonal refinement. We have developed a method for making only a small local rectification while keeping the large part of the quadrangulation. The things about which one has to be careful is that it is not easy to detect those three problems and we have to guarantee relation (3) when we insert new nodes.



(a) (b) (c)

Figure 4: (a)Inadmissible polygonal approximation. (b),(c)Conversion of a quadrangulation $\{q_k\}$ into subregions having curved sides $\{Q_k\}$.

3.2 Transfinite interpolation and subdivision

Let us consider four sufficiently smooth curves $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta} : \mathbb{R} \longrightarrow \mathbb{R}^2$. We are interested in their restriction on $[0, 1]$ and we suppose that they fulfill the *compatibility conditions* at the corners

$$\boldsymbol{\alpha}(0) = \boldsymbol{\delta}(0), \qquad \boldsymbol{\alpha}(1) = \boldsymbol{\beta}(0), \qquad \boldsymbol{\gamma}(0) = \boldsymbol{\delta}(1), \qquad \boldsymbol{\gamma}(1) = \boldsymbol{\beta}(1). \tag{5}$$

We assume that besides those four common points in, there are no further intersection points. We are interested in generating a parametric surface $\mathbf{c}(u, v)$ defined on $\square := [0, 1]^2$ such that the image of $\partial\square$ by $\mathbf{c}$ coincides with the four curves. That is, we have for all $u, v \in [0, 1]$,

$$\mathbf{c}(u, 0) = \boldsymbol{\alpha}(u), \qquad \mathbf{c}(u, 1) = \boldsymbol{\gamma}(u) \qquad \mathbf{c}(0, v) = \boldsymbol{\delta}(v), \qquad \mathbf{c}(1, v) = \boldsymbol{\beta}(v). \tag{6}$$

This transfinite interpolation problem can be solved by a first order Coons patch which is defined in matrix form as

$$\mathbf{c}(u, v) := -\begin{bmatrix} -1 \\ F_0(u) \\ F_1(u) \end{bmatrix}^T \begin{bmatrix} \mathbf{0} & \mathbf{c}(u, 0) & \mathbf{c}(u, 1) \\ \mathbf{c}(0, v) & \mathbf{c}(0, 0) & \mathbf{c}(0, 1) \\ \mathbf{c}(1, v) & \mathbf{c}(1, 0) & \mathbf{c}(1, 1) \end{bmatrix} \begin{bmatrix} -1 \\ F_0(v) \\ F_1(v) \end{bmatrix}, \tag{7}$$

where F_0 and F_1 denote two arbitrary smooth functions satisfying [10]:

$$F_i(j) = \delta_{ij}, \quad i, j = 0, 1 \qquad \text{and} \qquad F_0(t) + F_1(t) = 1 \quad \forall\, t \in [0, 1].$$

For most cases, a Coons patch is already regular. However, when the boundary curves become too wavy, we observe overlapping isolines indicating that the mapping is not invertible as in Fig. 2(c). Our next goal is to find an efficient method which can quickly verify if a Coons patch is regular. In this document, we will treat only the case where the boundary curves are in Bézier form [3]. For general curves, refer to [6]. Thus, we suppose that the boundary curves $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}$ are expressed in terms of their respective control points $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}_i, \boldsymbol{\delta}_i$ $(i = 0, \ldots, n)$ as follows

$$\boldsymbol{\alpha}(t) = \sum_{i=0}^{n} \boldsymbol{\alpha}_i B_i^n(t), \quad \boldsymbol{\beta}(t) = \sum_{i=0}^{n} \boldsymbol{\beta}_i B_i^n(t), \quad \boldsymbol{\gamma}(t) = \sum_{i=0}^{n} \boldsymbol{\gamma}_i B_i^n(t), \quad \boldsymbol{\delta}(t) = \sum_{i=0}^{n} \boldsymbol{\delta}_i B_i^n(t).$$

The polynomial blending function F_1 is also expressed in Bézier form $F_1(t) = \sum_{i=0}^{n} \phi_i B_i^n(t) = 1 - F_0(t)$. Furthermore, we suppose that the range of F_0 and F_1 is $[0, 1]$ and we define

$$\mu := \max\{|F_1'(t)| : t \in [0, 1]\}. \tag{8}$$

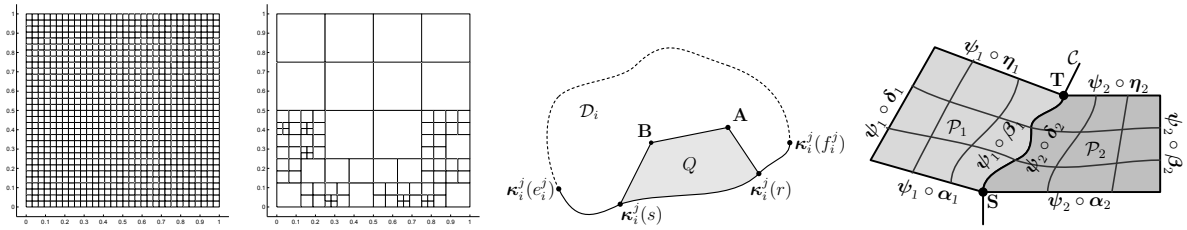


Figure 5: (a)Uniform subdivision. (b)Adaptive subdivision. (c)Restriction of $\boldsymbol{\kappa}_i^j$ to have boundary of a Coons patch inside the domain $\mathcal{D}_i$. (d)Two Coons maps match well at interface curve $\mathbf{C}$ for chord length parametrization.

In order to express the next result, we define τ as the minimum of the following expressions over $i, j = 0, \cdots, n$

$$\begin{aligned} A_{ij} &:= n^2 \det[\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i, \boldsymbol{\delta}_{j+1} - \boldsymbol{\delta}_j], \quad B_{ij} := n^2 \det[\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i, \boldsymbol{\beta}_{j+1} - \boldsymbol{\beta}_j], \\ C_{ij} &:= n^2 \det[\boldsymbol{\gamma}_{i+1} - \boldsymbol{\gamma}_i, \boldsymbol{\delta}_{j+1} - \boldsymbol{\delta}_j], \quad D_{ij} := n^2 \det[\boldsymbol{\gamma}_{i+1} - \boldsymbol{\gamma}_i, \boldsymbol{\beta}_{j+1} - \boldsymbol{\beta}_j]. \end{aligned}$$

Introduce also $G := \max\{G_1, G_2\}$ where

$$\begin{aligned} G_1 &:= \max_i\{\mu\|(\boldsymbol{\beta}_i - \boldsymbol{\delta}_i) + \phi_i(\boldsymbol{\gamma}_0 - \boldsymbol{\gamma}_n + \boldsymbol{\alpha}_n - \boldsymbol{\alpha}_0) + (\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}_n)\|\}, \\ G_2 &:= \max_i\{\mu\|(\boldsymbol{\gamma}_i - \boldsymbol{\alpha}_i) + \phi_i(\boldsymbol{\gamma}_0 - \boldsymbol{\gamma}_n + \boldsymbol{\alpha}_n - \boldsymbol{\alpha}_0) + (\boldsymbol{\alpha}_0 - \boldsymbol{\gamma}_0)\|\}. \end{aligned} \tag{9}$$

Theorem 3.1 *[6] Let M be a constant such that*

$$\begin{aligned} n\|\phi_j(\boldsymbol{\gamma}_{i+1} - \boldsymbol{\gamma}_i + \boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i+1}) + (\boldsymbol{\alpha}_{i+1} - \boldsymbol{\alpha}_i)\| &\le M, \\ n\|\phi_j(\boldsymbol{\beta}_{i+1} - \boldsymbol{\beta}_i + \boldsymbol{\delta}_i - \boldsymbol{\delta}_{i+1}) + (\boldsymbol{\delta}_{i+1} - \boldsymbol{\delta}_i)\| &\le M, \end{aligned} \tag{10}$$

for all $i = 0, \ldots, n-1$ and $j = 0, \ldots, n$. If $2MG + G^2 < \tau$ and $\tau > 0$, then $\mathbf{c}$ *is regular.*

In order to employ the technique of adaptive subdivision, let us introduce two notions. First, one can show [10] that a Bézier surface $\sum_{i,j=0}^{n} \mathbf{E}_{ij} B_i^n(u) B_j^n(v)$ has as Jacobian a Bézier function of degree $2n$ with the next control coefficients:

$$J_{pq} := \sum_{\substack{i+k=p \\ j+l=q}} C(i,j,k,l) \frac{\binom{n}{i}\binom{n}{k}}{\binom{2n}{i+k}} \frac{\binom{n}{j}\binom{n}{l}}{\binom{2n}{j+l}}, \qquad p, q = 0, \ldots, 2n, \tag{11}$$

where

$$\begin{aligned} C(i,j,k,l) := \frac{l}{n}\left[\frac{i}{n} D(i-1,j,k,l-1) + \left(1 - \frac{i}{n}\right) D(i,j,k,l-1)\right] \\ + \left(1 - \frac{l}{n}\right)\left[\frac{i}{n} D(i-1,j,k,l) + \left(1 - \frac{i}{n}\right) D(i,j,k,l)\right] \end{aligned}$$

and

$$D(i,j,k,l) := n^2 \det[\mathbf{E}_{i+1,j} - \mathbf{E}_{ij}, \mathbf{E}_{k,l+1} - \mathbf{E}_{kl}].$$

On the other hand, a Bézier surface F defined on $[a,b] \times [c,d]$ can be subdivided into four

Bézier surfaces F^A, F^B, F^C, F^D which are respectively defined on

$$I^A := [a,(a+b)/2]\times[c,(c+d)/2], \qquad I^B := [a,(a+b)/2]\times[(c+d)/2,d],$$
$$I^C := [(a+b)/2,b]\times[c,(c+d)/2], \qquad I^D := [(a+b)/2,b]\times[(c+d)/2,d],$$

by using the following recursions. Suppose the control points of F are F_{ij}, $i,j=0,\ldots,n$. We define for $i,j=0,\ldots,n, \quad k\geq 1$

$$\left\{\begin{array}{lll} F_{ij}^{[0]} := F_{ij} & \text{and} & F_{ij}^{[k]} := 0.5\big(F_{i-1,j}^{[k-1]} + F_{ij}^{[k-1]}\big) \\ P_{ij}^{[0]} := F_{ij}^{[i]} & \text{and} & P_{ij}^{[k]} := 0.5\big(P_{i,j-1}^{[k-1]} + P_{ij}^{[k-1]}\big) \\ Q_{ij}^{[0]} := F_{nj}^{[n-i]} & \text{and} & Q_{ij}^{[k]} := 0.5\big(Q_{i,j-1}^{[k-1]} + Q_{ij}^{[k-1]}\big) \end{array}\right. \tag{12}$$

The control points of F^A, F^B, F^C and F^D are respectively $A_{ij} := P_{ij}^{[j]}$, $B_{ij} := P_{in}^{[n-j]}$, $C_{ij} := Q_{ij}^{[j]}$, $D_{ij} := Q_{in}^{[n-j]}$. We have in particular

$$F(u,v) = F^r(u,v) \quad \text{for} \quad (u,v)\in I^r, \quad \text{where} \quad r = A,B,C,D.$$

We can apply the same subdivision technique to each of the resulting 4 Bézier surfaces. A recursive application of that subdivision on the unit square generates a uniform grid consisting of σ^2 little squares as illustrated in Fig. 5(a).

Theorem 3.2 *[6] Suppose that the Coons patch* $\mathbf{c}$ *defined with* $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}$ *is regular. Suppose that its Jacobian function* J *has been subdivided into* σ^2 *functions* J^{ij} *defined on*

$$I^{ij} := [(i-1)/\sigma, i/\sigma]\times[(j-1)/\sigma, j/\sigma], \qquad i,j=1,\ldots,\sigma, \tag{13}$$

and with Bézier coefficients J_{pq}^{ij}, $p,q=0,\ldots,2n$. *Then, for a sufficiently large* σ *all coefficients* J_{pq}^{ij} *have the same sign.*

Theorem 3.3 *[6] Let* $\mathbf{c}$ *be a Coons patch that is not regular. Then, for sufficiently large* σ *(see Theorem 3.2), there must exist* (i_1,j_1) *and* (i_2,j_2) *such that*

$$\left.\begin{array}{l} J_{pq}^{i_1,j_1} > 0 \\ J_{pq}^{i_2,j_2} < 0 \end{array}\right\} \quad \forall\, p,q=0,\ldots,2n. \tag{14}$$

In practice, we do not need to subdivide the Jacobian uniformly because we can perform adaptive subdivision. We start from a single Jacobian function written in Bézier form defined on the unit square. Then, we split it recursively by adaptively using the former subdivision techniques. That is, we subdivide only those Bézier functions that have Bézier coefficients J_{pq}^{ij} with different signs. The preceding two theorems serve as abortion conditions for that recursion. An instance of that adaptive subdivision process is illustrated in Fig. 5(b). For more structured description about that adaptive process refer to [6].

3.3 Global continuity

In order to obtain a mapping $\boldsymbol{\gamma}_k$ from the unit square $[0,1]^2$ to $\mathcal{P}_k \subset \mathbf{R}^3$, we need to find a mapping $\mathbf{x}_{k,i}$ from $[0,1]^2$ to the planar four-sided domain $Q_{k,i}\subset\mathbf{R}^2$ of relation (4) and we take the composition:

$$\mathcal{P}_k = \boldsymbol{\psi}_i(Q_{k,i}) = \boldsymbol{\psi}_i \circ \mathbf{x}_{k,i}([0,1]^2). \tag{15}$$

Now, we will discuss about the construction of the mapping $\mathbf{x} := \mathbf{x}_{k,i}$ from $[0,1]^2$ to each of the four-sided domains $Q := Q_{k,i} \subset \mathbf{R}^2$ by means of Coons maps. On account of the decomposition algorithm of Section 3.1, we note that the boundaries $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\eta}$, $\boldsymbol{\delta}$ of the Coons maps are either straight lines or restrictions of the 2D boundary curves $\boldsymbol{\kappa}_i^j$. Since the Coons patch (7) requires $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\eta}$, $\boldsymbol{\delta}$ to be defined on $[0,1]$, we use one of the next two representations:

$$\begin{array}{rcll} \boldsymbol{\mu}(t) & = & t\mathbf{B} + (1-t)\mathbf{A}, & \forall\, \boldsymbol{\mu} = \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\delta}, \\ \boldsymbol{\mu}(t) & = & \boldsymbol{\kappa}_i^j(tr + (1-t)s), & \forall\, \boldsymbol{\mu} = \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\delta}. \end{array} \tag{16}$$

The first representation is used when $\boldsymbol{\mu}$ is a straight internal edge joining two interior points $\mathbf{A}$ and $\mathbf{B}$ while the second one applies if $\boldsymbol{\mu}$ is a restriction of $\boldsymbol{\kappa}_i^j$ on some $[s,r] \subset [e_i^j, f_i^j]$ as seen in Fig. 5(c).

An arbitrary parametrization of $\boldsymbol{\kappa}_i^j$ does not guarantee the global continuity. Additionally, we cannot modify the base surfaces $\boldsymbol{\psi}_i$ because they are given as input in the initial CAD storage. Therefore, our objective is to replace the 2D curves $\boldsymbol{\kappa}_i^j$ by $\tilde{\boldsymbol{\kappa}}_i^j$ so that their 3D images by $\boldsymbol{\psi}_i$ agree pointwise as long as they are incident. Let us introduce the length function

$$\chi_i^j(t) := \int_{e_i^j}^{t} \left\| \frac{d(\boldsymbol{\psi}_i \circ \boldsymbol{\kappa}_i^j)}{d\theta}(\theta) \right\| d\theta. \tag{17}$$

This function is defined from $[e_i^j, f_i^j]$ to $[0, L]$ where L is the total length of the curve $\boldsymbol{\psi}_i \circ \boldsymbol{\kappa}_i^j$. On account of the properties of $\boldsymbol{\kappa}_i^j$ and $\boldsymbol{\psi}_i$ that we met in Section 2, let us observe that

$$\frac{d\chi_i^j}{d\theta}(\theta) = \left\| \frac{d(\boldsymbol{\psi}_i \circ \boldsymbol{\kappa}_i^j)}{d\theta}(\theta) \right\| \neq 0 \qquad \forall\, \theta \in [e_i^j, f_i^j]. \tag{18}$$

Hence, there is an inverse function $\phi_i^j := (\chi_i^j)^{-1}$ and our approach is to replace the function $\boldsymbol{\kappa}_i^j$ by $\tilde{\boldsymbol{\kappa}}_i^j := \boldsymbol{\kappa}_i^j \circ \phi_i^j$. In [10], we have shown that the use of the chord length [5] parametrization ϕ_i^j ensures the matching condition at the interface curve of two adjacent Coons patches. To that end, let us consider two adjacent four-sided patches $\mathcal{P}_{k,i}$ and $\mathcal{P}_{p,j}$. We will treat only the non-obvious case where those two patches belong to two nonidentical trimmed surface $\mathbf{S}_i$ and $\mathbf{S}_j$. In order to facilitate the presentation, we may suppose that they are $\mathbf{S}_1$, $\mathbf{S}_2$, $\mathcal{P}_1$, $\mathcal{P}_2$ and we omit the superscripts. In the sequel, we denote the corresponding length functions by χ_1 and χ_2, the two Coons maps from relation (15) by $\mathbf{x}_1$ and $\mathbf{x}_2$ such as $\mathcal{P}_1 = \boldsymbol{\psi}_1 \circ \mathbf{x}_1([0,1]^2)$ and $\mathcal{P}_2 = \boldsymbol{\psi}_2 \circ \mathbf{x}_2([0,1]^2)$ which share only one curved edge because of the conforming condition. Further, we denote by $[e_1, f_1]$ and $[e_2, f_2]$ the intervals of definition of $\boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1$ and $\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2$ which have coinciding images upon which $\mathcal{P}_1$ and $\mathcal{P}_2$ are incident:

$$(\boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1)([e_1, f_1]) = (\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2)([e_2, f_2]) =: \mathbf{C}. \tag{19}$$

That is, we reparametrize the 2D curves $\boldsymbol{\kappa}_i$ from which the boundary of the planar Coons maps are obtained in order that the 3D curves $\boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1$ and $\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2$ agree pointwise. In fact, by using the chain rule and $\lambda := (\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2)^{-1} \circ \boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1$, we obtain

$$\begin{array}{rcl} \chi_1(t) & = & \displaystyle\int_{e_1}^{t} \left\| \frac{d(\boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1)}{d\theta}(\theta) \right\| d\theta = \int_{e_1}^{t} \left\| \frac{d(\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2)}{d\theta}[\lambda(\theta)] \right\| \lambda'(\theta) d\theta. \\ & = & \displaystyle\int_{\lambda(e_1)}^{\lambda(t)} \left\| \frac{d(\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2)}{d\sigma}(\sigma) \right\| d\sigma = \int_{e_2}^{\lambda(t)} \left\| \frac{d(\boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2)}{d\sigma}(\sigma) \right\| d\sigma = \chi_2(\lambda(t)). \end{array}$$

Models	Nb. trimmed surfaces	Nb. patches	Total runtimes
Fig. 6(a)	16	38	1.919 sec.
Fig. 6(b)	26	90	6.588 sec.
Fig. 6(c)	243	727	44.232 sec.

Table 1: Number of initial surfaces, number of patches, run times for decomposition and evaluation

Hence, we obtain $\chi_2^{-1} \circ \chi_1 = \lambda$. Therefore, if we denote the total length by $L := \chi_1(f_1) = \chi_2(f_2)$, we have

$$\boldsymbol{\psi}_2 \circ \tilde{\boldsymbol{\kappa}}_2(t) = \boldsymbol{\psi}_2 \circ \boldsymbol{\kappa}_2 \circ \phi_2(t) = \boldsymbol{\psi}_1 \circ \boldsymbol{\kappa}_1 \circ \phi_1(t) = \boldsymbol{\psi}_1 \circ \tilde{\boldsymbol{\kappa}}_1(t) \qquad \forall\, t \in [0, L]. \tag{20}$$

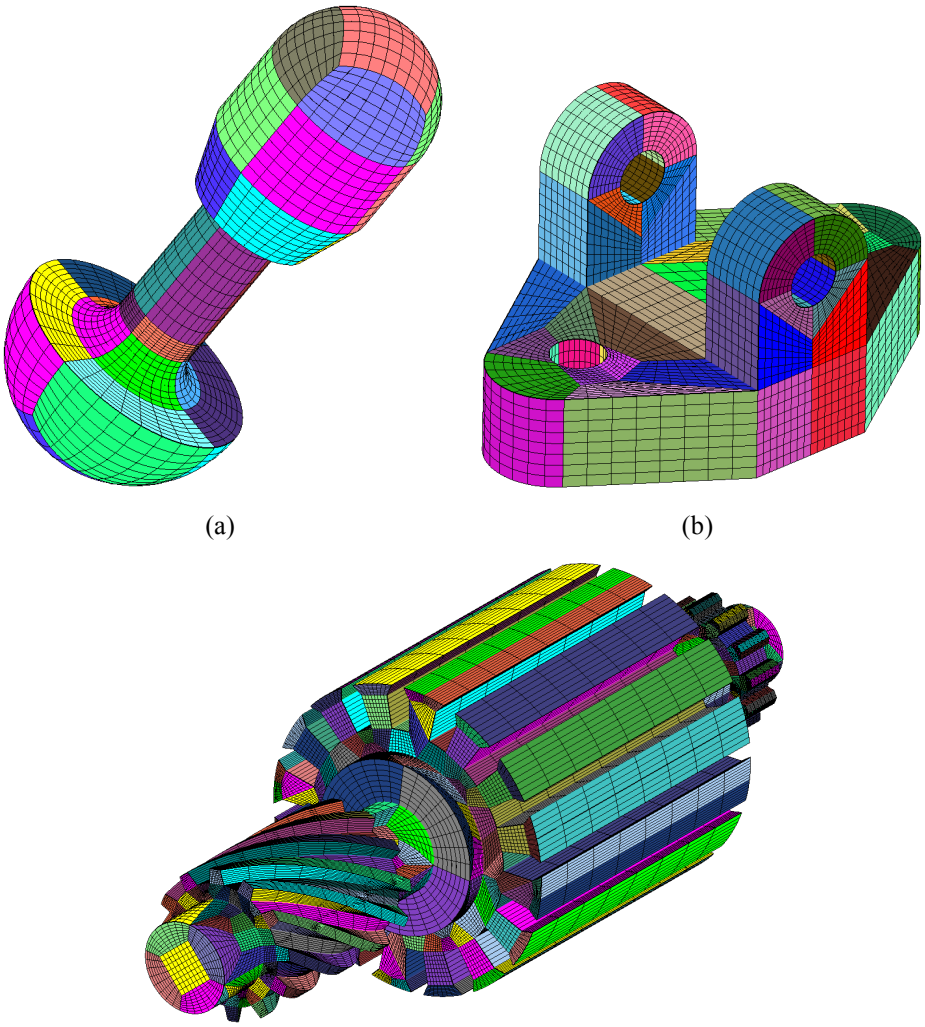
The former sections imply that we have to evaluate chord lengths many times and accurately. In [10], we proposed a length computation algorithm with exponential speed.
We have applied the former methods to real CAD objects. The initial CAD models corresponding to the mechanical parts which are found in Fig.6 have been designed with a CAD system. In Table 1, we gather the number of entities corresponding to the mechanical parts. We see there also the required time to generate the decomposition together with the evaluation to obtain the gridpoints at level $J = 6$.

4 CURVED DECOMPOSITION FOR FEM

In this section, we would like to describe the process of generating a tetrahedral mesh containing tetrahedra which might have curved edges and non-planar faces of a domain Ω. Such a generation will be performed in several steps. As a starting step, we create a conforming curved triangular mesh $\mathcal{M}$ of the boundary of Ω. That can be obtained for examples by using a method similar to that in Section 3.1 but instead of quadrangulations, one uses triangulations. Another alternative is to create a curved quadrilateral boundary mesh as in the BEM case and then one inserts a diagonal edge inside each quadrilateral to obtain two triangular patches. According to our experience, that second method does not give quality results. Generating a curved triangular mesh directly from the CAD models gives better results. Note that we require also a global continuity for the curved triangular boundary mesh $\mathcal{M}$. That is obtained by using the method from Section 3.3.

4.1 Straight tetrahedral mesh

Consider the triangular mesh $\overline{\mathcal{M}}$ that is obtained from the curved mesh $\mathcal{M}$ in which one replaces every curved edge e of $\mathcal{M}$ by a straight edge joining the two endpoints of e. The next step now consists in generating a tetrahedral mesh bounded by $\overline{\mathcal{M}}$. Note that in two dimensions, it is possible to decompose a *nonconvex* polygon directly into triangles. That can be theoretically proved by using the 2-ear theorem [11]. That is, one can always chop off a triangle from the polygon recursively until the polygon becomes empty. Unfortunately, that nice fact does not generally hold in higher dimension. To treat a nonconvex polyhedron $\overline{\mathcal{M}}$ in 3D, one needs a *constrained* tetrahedralization as follows. First, one tetrahedralizes the set $\mathbf{V}$ of vertices $\mathbf{x}_i$ of the polyhedron $\overline{\mathcal{M}}$. Thus, one obtains a tetrahedralization of the convex hull of $\mathbf{V}$. At this point, some faces of the polyhedron $\overline{\mathcal{M}}$ might be crossed by some tetrahedra. That is, there might be some tetrahedra which are partly inside and partly outside the polyhedron $\overline{\mathcal{M}}$. Therefore, the second step consists in modifying the tetrahedralization so that one recovers the faces of



(a) (b)

(c)

Figure 6: CAD decomposition for BEM.

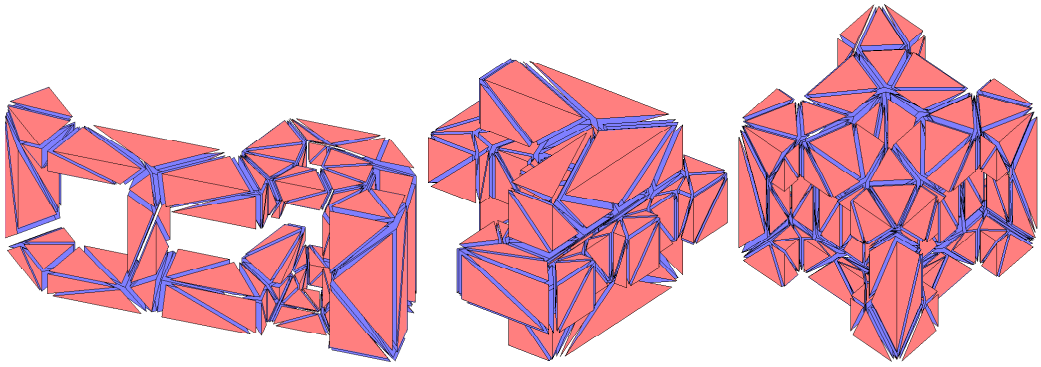
the polyhedron $\overline{\mathcal{M}}$. At this step every tetrahedron is either inside or outside the polyhedron. Finally, the tetrahedra outside $\overline{\mathcal{M}}$ are discarded. An illustration of those steps are depicted in Fig. 8. Now, we will describe each of those steps briefly.

In order to generate a tetrahedral mesh of the convex hull of $\mathbf{V}$, one utilizes Delaunay tetrahedralization [17] or any other methods. One of the most stable ways of generating a quality tetrahedral mesh is using the parabolic lifting map

$$\Pi(x, y, z) := (x, y, z, x^2 + y^2 + z^2). \tag{21}$$

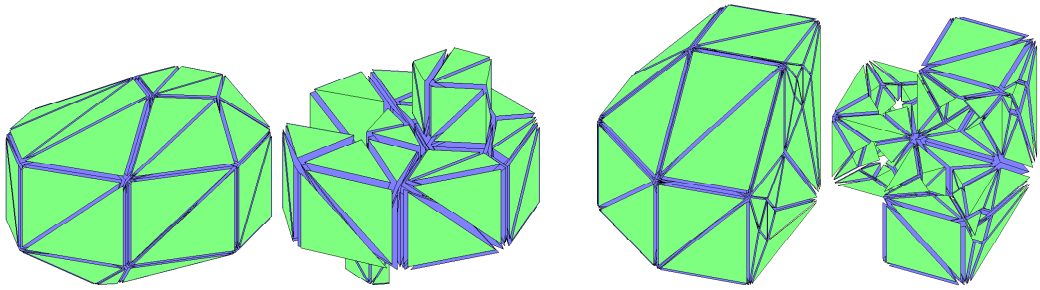
This has a good property [17] that the image by Π of a sphere $S \subset \mathbb{R}^3$ centered at A and with radius r is contained in a hyperplane $\mathcal{H}$ is $\mathbb{R}^4$. In fact, $\Pi(S)$ is the intersection of the paraboloid corresponding to Π and the hyperplane $\mathcal{H}$. Thus, a point $Q \in \mathbb{R}^3$ is not inside the sphere S as long as $\Pi(Q)$ is above (points to positive axis of fourth coordinate) that hyperplane. Computing the tetrahedralization of $\mathbf{V}$ amounts to computing the boundary of the convex hull C of $\widetilde{\mathbf{x}}_i = \Pi(\mathbf{x}_i)$ and projecting the lower part of C onto $\mathbb{R}^3$. A software such as QHULL for computing convex hulls in $\mathbb{R}^4$ is helpful in performing such a task.

Let us now see the recovery of the initial boundary to obtain a tetrahedral mesh of nonconvex polyhedra. At this position, we suppose that we have a tetrahedralization $\mathcal{T}$ of the convex hull



(a) (b) (c)

Figure 7: Coarse tetrahedralization of polyhedra where the triangular faces are planar.



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

Figure 8: (a) is an unconstrained tetrahedral mesh for (b). (c) is an unconstrained tetrahedral for (d).

of the triangular boundary $\overline{\mathcal{M}}$. Some tetrahedra of $\mathcal{T}$ might not respect the boundary $\overline{\mathcal{M}}$. That is, they are partly outside and partly inside the domain Ω. That facial recovery is performed in two steps. First, we need to recover the edges of $\overline{\mathcal{M}}$. Afterwards, we recover the triangles of $\overline{\mathcal{M}}$.

Suppose that we have a triangular edge $[N_1, N_2]$ of $\overline{\mathcal{M}}$ which does not coincide with any of the tetrahedral edges of $\mathcal{T}$. Consider a tetrahedron $\tau = [a, b, c, d]$ which is traversed by the edge $[N_1, N_2]$. Let us denote by n_1, n_2 the part of $[N_1, N_2]$ which overlaps with τ. There are several cases to consider in order that we have a tetrahedralization such that $[n_1, n_2]$ becomes an edge of the tetrahedralization. Those cases are defined according to whether n_i is located on a node, on an edge or on a face of the tetrahedron τ as illustrated in Fig. 9. The case where both n_1 and n_2 are on nodes of τ does not occur because $[N_1, N_2]$ is not an edge of the tetrahedralization. Some cases where the line segment $[N_1, N_2]$ touches an edges of the closure of τ but $[N_1, N_2]$ does not traverse the interior of τ are also treated similarly. For those five situations, the tetrahedron τ is decomposed into several subtetrahedra as follows

Case a: $[n_1, n_2, c, d]$, $[n_1, n_2, b, c]$, $[n_1, n_2, a, b]$.
Case b: $[a, n_1, b, c]$, $[n_1, b, c, n_2]$, $[n_1, c, d, n_2]$, $[n_1, b, d, n_2]$.
Case c: $[n_2, n_1, d, c]$, $[n_2, n_1, a, c]$, $[n_2, n_1, a, d]$, $[a, c, b, n_2]$, $[a, c, b, n_2]$, $[a, d, b, n_2]$.
Case d: $[n_1, n_2, c, d]$, $[n_1, b, c, n_2]$.
Case e: $[a, b, n_2, d]$, $[a, b, c, n_2]$, $[a, c, n_2, d]$.

Suppose now that the above process has been applied to every tetrahedron τ of $\mathcal{T}$ which is

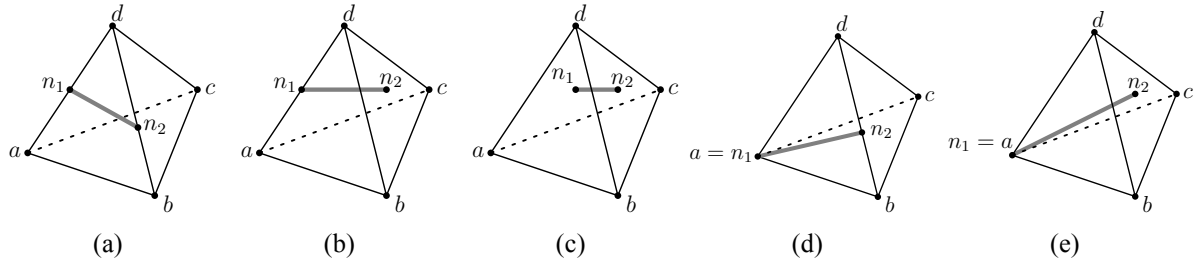


Figure 9: Some cases where a line segment $[n_1, n_2]$ traverses a tetrahedron $[a, b, c, d]$.

traversed by a certain edge of a boundary triangle. Thus, all nodes and edges of every triangle of the boundary $\overline{\mathcal{M}}$ are among the nodes and edges of the tetrahedralization of $\mathcal{T}$. Hence, the property of the resulting tetrahedral mesh from the previous step is that if a triangle t from $\overline{\mathcal{M}}$ and a tetrahedron τ from $\mathcal{T}$ intersect, then they intersect completely. That is, the intersection $t \cap \tau$ is the same as the intersection between τ and the plane $\mathcal{P}$ passing through t. In such a situation, there are only two cases which may occur. In the first case, one node of τ lies on one side $\mathcal{P}^+$ of $\mathcal{P}$ and three other nodes are on the other side $\mathcal{P}^-$. The second case occurs when two nodes of τ are on one side $\mathcal{P}^+$ while the other two are on the other side $\mathcal{P}^-$.

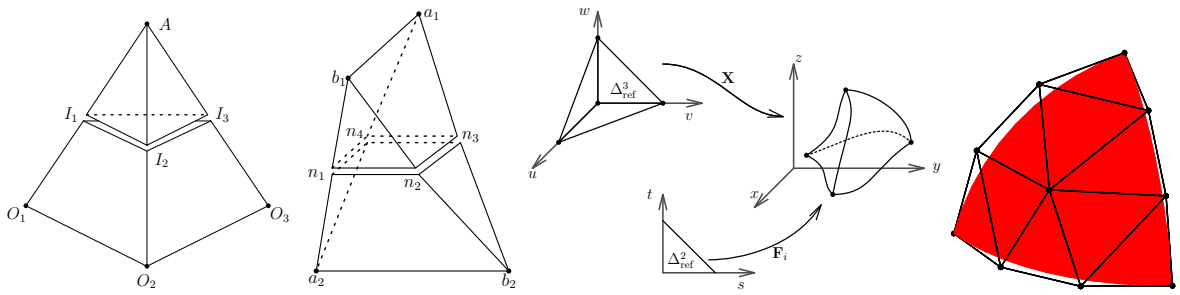
For the former case, let $[O_1, O_2, O_3]$ be the three nodes in $\mathcal{P}^-$ and let $[I_1, I_2, I_3]$ be the intersection of τ and the plane $\mathcal{P}$. Such an indexation is illustrated in Fig. 10(a). In order that all tetrahedra are either completely in $\mathcal{P}^+$ or completely in $\mathcal{P}^-$, one needs to refine τ as follows. In $\mathcal{P}^-$, the subtetrahedra are $[O_1, I_1, I_2, I_3]$, $[O_1, O_2, I_2, I_3]$ and $[O_1, O_2, O_3, I_3]$. The only new tetrahedron on the other side is of course $[I_1, I_2, I_3, A]$.

For the second case, let a_1 and b_1 be two nodes of τ in $\mathcal{P}^+$. The four intersections of the edges of τ with the plane $\mathcal{P}$ are designated by n_1, n_2, n_3, n_4. Suppose also that n_i are given in the ordering shown as in Fig. 10(b). From those nodes, the obtained local tetrahedra are $[a_1, b_1, n_3, n_4]$, $[n_1, n_3, n_4, b_1]$ and $[n_1, n_2, n_3, b_1]$. The same construction can be repeated to the nodes a_2 and b_2 on the other side $\mathcal{P}^-$.

At this point, we have a tetrahedral mesh where a tetrahedron is either completely inside or completely outside the mesh $\overline{\mathcal{M}}$. To generate the nonconvex tetrahedralization, one needs to discard the tetrahedra which are outside the mesh $\overline{\mathcal{M}}$. One generates a connectivity graph $\mathcal{G}$ whose nodes are represented by the tetrahedra. An edge between two nodes n_1 and n_2 of the graph $\mathcal{G}$ is inserted if the corresponding two tetrahedra share a triangular face. One then selects the connected component of $\mathcal{G}$ for which the union of the corresponding tetrahedra are covered by a triangular surface which coincides with the triangular boundary mesh $\overline{\mathcal{M}}$. The above process was only performed to recover the triangular boundary mesh $\overline{\mathcal{M}}$. After that recovery process, one can improve the tetrahedral mesh $\mathcal{T}$ by using shifting and flipping operations [4, 8].

4.2 Curved tetrahedral mesh

The purpose of this section is the deduction of the curved tetrahedralization $\widetilde{\mathcal{T}}$ from the curved triangulation $\mathcal{M}$ and the straight tetrahedralization $\mathcal{T}$. Thus, the process consists in converting the straight tetrahedral mesh into a curvilinear one. The first step in that convertion consists of course in replacing the boundary triangles of $\mathcal{T}$ by the triangular faces of $\mathcal{M}$. Thus, the boundary edges of $\widetilde{\mathcal{T}}$ are defined to correspond to the boundary edges of $\mathcal{M}$ while the internal edges of $\widetilde{\mathcal{T}}$ are exactly the same as those of $\mathcal{T}$. Afterwards, we need to generate the internal triangular faces of $\widetilde{\mathcal{T}}$. Note that it is possible that some triangular faces could be nonplanar



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

Figure 10: (a)Three nodes of the tetrahedron are on one side and one on the other of a cutting plane. (b)Two nodes on each side of the cutting plane. (c)Tetrahedral transfinite interpolation. (d)Triangular Bézier function for $n = 3$.

although they are completely inside the domain Ω. That occurs when some of their edges are boundary edges which are curved. To that end, one uses transfinite interpolations to combine three edges of $\widetilde{\mathcal{T}}$. For instance, one can use the Coons map of three spatial curves.

Now that all curved edges and faces are set, we generate then the curvilinear tetrahedral mesh $\widetilde{\mathcal{T}}$ in which we replace every straight tetrahedron τ of $\mathcal{T}$ by $\tilde{\tau}$ that is obtained from τ by taking the transfinite interpolation (see Section 5.1) of the four corresponding curved triangular faces. Due to that construction the global continuity introduced in Section 2.2 follows because every two adjacent tetrahedra have common triangular faces which possess the same parametrization.

At this stage, there might of course be some interference problems. Some curvilinear triangular faces of one tetrahedron in $\widetilde{\mathcal{T}}$ might have an internal intersection. Some edges which are certainly sharp from the tetrahedral mesh $\mathcal{T}$ might be smoothened out in $\widetilde{\mathcal{T}}$ after the convertion process. These interference problems can be detected by using the process which we describe in section 5. Generally speaking, the existence of such interferences inccurs that the approximation of the curved triangular mesh $\mathcal{M}$ by the straight triangular mesh $\overline{\mathcal{M}}$ in the vicinity of the interference is too coarse. That is, they usually occur next to the boundary. Below, we propose an approach for circumventing that conflicting interference.

We keep a graph which stores the adjacency information of the whole tetrahedralization $\widetilde{\mathcal{T}}$. Suppose that τ is a curvilinear tetrahedron of $\widetilde{\mathcal{T}}$ such that the transfinite interpolation with respect to τ is not regular. The remedy is processed in two attempts. First, we search for neighboring tetrahedra to τ by using the adjacency graph. We try then to apply some flipping operations [4] to them. Afterwards, we retry to check regularity of the flipped tetrahedra. If that process solves the problem then we stop. Otherwise, we group all the tetrahedra adjacent to τ and insert a new node on each curved boundary edge between every two of them. The next step consists in the local retetrahedralization of the grouping and in replacing the old local tetrahedra from the grouping by the new ones. By repeating the above steps the curved edge are getting more and more straight and after sufficient application of that process, all consisting tetrahedra must become regular.

5 REGULARITY FOR TETRAHEDRAL INTERPOLATION

Let us first introduce some notations. Multi-indices will be denoted by bold Greek letters $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_m)$ such as $\alpha_i \in \mathbb{N}_0$ in which we have $|\boldsymbol{\alpha}| := \sum_{p=1}^{m} \alpha_p$. We introduce the following

definitions:

$$\mathbf{\Omega}_n^{m-1} := \{\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_m) \in \mathbb{N}_0^m \,:\, |\boldsymbol{\alpha}| = n\}. \tag{22}$$

$$\mathbf{u}^{\boldsymbol{\alpha}} = (u_1, ..., u_m)^{(\alpha_1,...,\alpha_m)} := u_1^{\alpha_1} \cdots u_m^{\alpha_m}, \quad \text{and} \quad \partial_{\boldsymbol{\alpha}} f := \frac{\partial^{\alpha_1}}{\partial u_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_m}}{\partial u_m^{\alpha_m}} f. \tag{23}$$

5.1 Tetrahedral transfinite interpolation

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

$$\Delta^2_{\text{ref}} := \{\boldsymbol{\sigma} = (s,t) \in \mathbb{R}^2 : \quad s \geq 0, \quad t \geq 0, \quad s + t \leq 1\}, \tag{24}$$

$$\Delta^3_{\text{ref}} := \{\mathbf{u} = (u,v,w) \in \mathbb{R}^3 : \, u \geq 0, \, v \geq 0, \, w \geq 0, \, u + v + w \leq 1\}. \tag{25}$$

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

$$\begin{array}{ll} \mathbf{X}(u,v,0) = \mathbf{F}_1(v,u) & \mathbf{X}(u,0,w) = \mathbf{F}_2(u,w) \\ \mathbf{X}(0,v,w) = \mathbf{F}_4(w,v) & \mathbf{X}(u,v,1-u-v) = \mathbf{F}_3(v,1-u-v). \end{array} \tag{26}$$

In order that these conditions can be fulfilled, it is necessary to assume *compatibility conditions* which are generalization of 5 for tetrahedra. They consists of 6 conditions at the edges of Δ^3_{ref} and 4 conditions at the corners of Δ^3_{ref}. We consider also blending functions p_i which are polynomials defined on Δ^3_{ref}. Each one of them takes zero value at one side of Δ^3_{ref} and they sum to unity. The complete form of tetrahedral transfinite interpolation is

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

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

$$\mathcal{J}(u,v,w) = \det[\mathbf{X}(u,v,w)] \neq 0 \qquad \forall\, (u,v,w) \in \Delta^3_{\text{ref}}. \tag{27}$$

5.2 Blossom and multivariate Bézier representation

Since we suppose that the triangular faces $\mathbf{F}_i$ are represented in Bézier form [15], we will introduce the definition and certain properties of Bézier in this section. Consider a d-simplex ($d = 2, 3$) of definition Δ with apices $\mathbf{t}_0$,...,$\mathbf{t}_d$. By using the barycentric coordinates $\lambda_i(\mathbf{u})$ of each point $\mathbf{u} \in \Delta$, we define

$$\mathcal{B}^{\Delta,n}_{(\beta_0,...,\beta_d)}(\mathbf{u}) := \frac{n!}{\beta_0! \cdots \beta_d!} \lambda_0(\mathbf{u})^{\beta_0} \cdots \lambda_d(\mathbf{u})^{\beta_d}. \tag{28}$$

We have the identity $\mathcal{B}^{\Delta,n}_{(\beta_0,\ldots,\beta_d)}(\mathbf{u}) = \lambda_0(\mathbf{u})\mathcal{B}^{\Delta,n-1}_{(\beta_0-1,\ldots,\beta_d)}(\mathbf{u}) + \cdots + \lambda_d(\mathbf{u})\mathcal{B}^{\Delta,n-1}_{(\beta_0,\ldots,\beta_d-1)}(\mathbf{u})$. For the particular case where the simplex of definition Δ is the unit simplex, the above definition reduces to

$$\frac{n!}{\beta_0!\beta_1!\cdots\beta_d!}(1-u_1-\cdots-u_d)^{\beta_0}u_1^{\beta_1}\cdots u_d^{\beta_d}. \tag{29}$$

A Bézier function defined for $\mathbf{u} \in \Delta$ is given by

$$\mathbf{Y}(\mathbf{u}) = \sum_{|(\beta_0,\ldots,\beta_d)|=n} \mathbf{b}_{(\beta_0,\ldots,\beta_d)}\mathcal{B}^{\Delta,n}_{(\beta_0,\ldots,\beta_d)}(\mathbf{u}) \tag{30}$$

where the points $\mathbf{b}_{\boldsymbol{\beta}} = \mathbf{b}_{(\beta_0,\ldots,\beta_d)} \in \mathbb{R}^3$ are the control points. The corelation between the control points and the polynomials can be expressed with the help of the blossoming which we recall briefly now. A function $\mathcal{P}$ is a polar form or a blossom function [15] if it is *multiaffine*: for all $\lambda_a \geq 0$ and $\lambda_b \geq 0$ such that $\lambda_a + \lambda_b = 1$, we have $\mathcal{P}(\mathbf{u}_1, \cdots, \lambda_a\mathbf{u}_i^a + \lambda_b\mathbf{u}_i^b, \cdots, \mathbf{u}_n) = \lambda_a\mathcal{P}(\mathbf{u}_1, \cdots, \mathbf{u}_i^a, \cdots, \mathbf{u}_n) + \lambda_b\mathcal{P}(\mathbf{u}_1, \cdots, \mathbf{u}_i^b, \cdots, \mathbf{u}_n)$ and *symmetric*: for any permutation π of $\{1, ..., n\}$, we have $\mathcal{P}(\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n) = \mathcal{P}(\mathbf{u}_{\pi(1)}, \mathbf{u}_{\pi(2)}, \cdots, \mathbf{u}_{\pi(n)})$. For each multivariate polynomial f of degree n, there is a unique blossom function $\mathcal{P}(f)$ such that we have the next diagonal property:

$$\mathcal{P}(f)(\underbrace{\mathbf{u}, \cdots, \mathbf{u}}_{n}) = f(\mathbf{u}) \qquad \forall\, \mathbf{u} \in \mathbb{R}^d. \tag{31}$$

The blossom $\mathcal{P}(\mathbf{Y})$ of the simplex Bézier function $\mathbf{Y}$ can be evaluated at $(\mathbf{u}_1, \cdots, \mathbf{u}_n)$ with the help of the next pyramid algorithm in which e_k is the multi-index of $\boldsymbol{\Omega}^d_n$ having unity at the k-th entry and zeros at all other entries.

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

The blossom function of the simplex Bézier in (30) and the control points are related with the following relation

$$\mathbf{b}_{\boldsymbol{\beta}} = \mathcal{P}(\mathbf{Y})(\underbrace{\mathbf{t}_0, \cdots, \mathbf{t}_0}_{\beta_0}, \underbrace{\mathbf{t}_1, \cdots, \mathbf{t}_1}_{\beta_1}, \cdots, \underbrace{\mathbf{t}_d, \cdots, \mathbf{t}_d}_{\beta_d}). \tag{32}$$

Above, we use tetrahedral transfinite interpolation as a combination of the triangular faces which are Bézier functions. In [16], one can find a method of representing the blossom of a product in terms of the blossom of each factor. By using that together with the pyramid algorithm, one can represent the transfinite interpolation on the tetrahedron Δ^3_{ref} as a tetrahedral Bézier (30).

5.3 Subdivision of tetrahedral Bézier

Now, we want to carry over the subdivision techniques from the BEM case to tetrahedra. Let us first formulate the expression of a Bézier function inside a smaller tetrahedron. Let $\mathbf{Y}$ be a

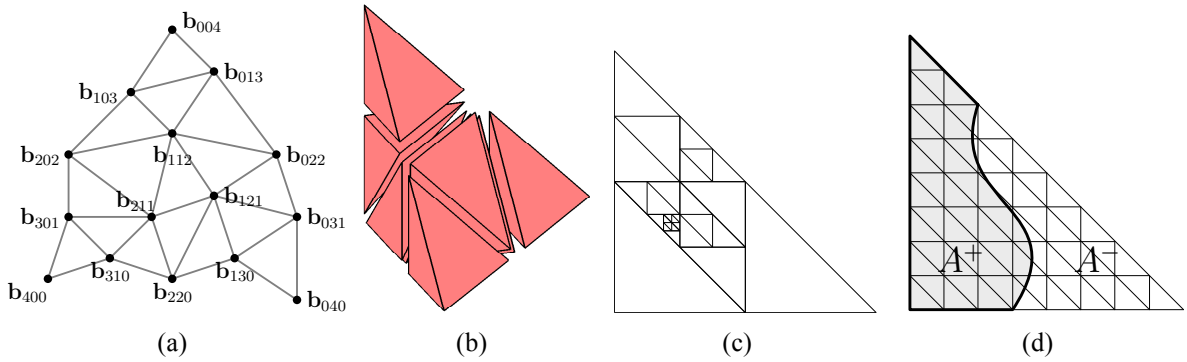


Figure 11: (a)Control net with its control points. (b)Uniform subdivision. (c)Adaptive subdivision. (d)Sign distribution on the subdivided triangle.

Bézier function with respect to $\Delta = \mathrm{HULL}\{\mathbf{t}_0, ..., \mathbf{t}_3\}$ as in (30). Consider another tetrahedron $\widetilde{\Delta} = \mathrm{HULL}\{\widetilde{\mathbf{t}}_0, ..., \widetilde{\mathbf{t}}_3\}$ such that $\widetilde{\Delta} \subset \Delta$. We want to express $\mathbf{Y}$ with respect to $\widetilde{\Delta}$ such that

$$\mathbf{Y}(\mathbf{u}) = \sum_{\boldsymbol{\beta} \in \boldsymbol{\Omega}_n^3} \widetilde{\mathbf{b}}_{\boldsymbol{\beta}} \mathcal{B}_{\boldsymbol{\beta}}^{\widetilde{\Delta}}(\mathbf{u}) = \sum_{\boldsymbol{\beta} \in \boldsymbol{\Omega}_n^3} \mathbf{b}_{\boldsymbol{\beta}} \mathcal{B}_{\boldsymbol{\beta}}^{\Delta}(\mathbf{u}) \qquad \forall\, \mathbf{u} \in \widetilde{\Delta} \subset \Delta. \tag{33}$$

For each multi-index $\boldsymbol{\beta} = (\beta_0, ..., \beta_3) \in \boldsymbol{\Omega}_n^3$, in order to find the control points $\widetilde{\mathbf{b}}_{\boldsymbol{\beta}}$ with respect to the new tetrahedron of definition $\widetilde{\Delta}$, we apply the pyramid algorithm from Section 5.2 to

$$(\mathbf{w}_1, ..., \mathbf{w}_n) := (\underbrace{\widetilde{\mathbf{t}}_0, \cdots, \mathbf{t}_0}_{\beta_0}, \underbrace{\widetilde{\mathbf{t}}_1, \cdots, \widetilde{\mathbf{t}}_1}_{\beta_1}, \cdots, \underbrace{\widetilde{\mathbf{t}}_3, \cdots, \widetilde{\mathbf{t}}_3}_{\beta_3}) \tag{34}$$

by using the blossom of the original Bézier function (30). The new control points $\widetilde{\mathbf{b}}_{\boldsymbol{\beta}}$ are then obtained from the connection formula (32). That is to say, we have $\widetilde{\mathbf{b}}_{\boldsymbol{\beta}} = \mathcal{P}(\mathbf{Y})(\mathbf{w}_1, ..., \mathbf{w}_n)$. In order to introduce the notion of recursive subdivisions, suppose that a Bézier function has a tetrahedron of definition $\Delta \subset \mathbb{R}^d$ where the apices are $\mathbf{t}_p$. The subdivision scheme consists in decomposing Δ into several subtetrahedra as follows. First, a new node is introduced at the middle of each edge of Δ. Then, one decomposes the parent tetrahedron into several subtetrahedra as illustrated in Fig. 11(b). The same subdivision process can be applied to each one of the resulting subtetrahedra. By doing that repeatedly, let us denote by ω_N the number of tetrahedra after N subdivisions. That is, we have tetrahedra $\Delta^{N,k}$ where the apices are denoted by $\mathbf{t}_p^{N,k}$ for $k = 1, 2, ..., \omega_N$. Additionally, we will denote the control point by $b_{\boldsymbol{\beta}}^{N,k}$. That is, we have on each tetrahedron $\Delta^{N,k}$ the following representation

$$\mathbf{Y}^{N,k}(\mathbf{u}) = \sum_{\boldsymbol{\beta} \in \boldsymbol{\Omega}_n^d} \mathbf{b}_{\boldsymbol{\beta}}^{N,k} \mathcal{B}_{\boldsymbol{\beta}}^{\Delta^{N,k}}(\mathbf{u}). \tag{35}$$

The maximal length of the initial edges will be denoted by $h := \max_{p \neq q} \|\mathbf{t}_p - \mathbf{t}_q\|$. In our case of tetrahedral transfinite interpolation, we suppose that the apices $\mathbf{t}_i$ of the first tetrahedron of definition are composed of the corners of the unit reference tetrahedron Δ_{ref}^3, that is $\Delta := \Delta_{\mathrm{ref}}^3$. 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 (27) can be replaced by

$$\mathcal{J}(\mathbf{u}) = \det[\mathbf{X}(u, v, w)] \geq \delta \qquad \forall\, \mathbf{u} = (u, v, w) \in \Delta_{\mathrm{ref}}^3, \tag{36}$$

for some prescribed constant $\delta > 0$. By using the above techniques, we will suppose that $\mathcal{J}$ is given as tetrahedral Bézier with scalar control points $b_{\boldsymbol{\beta}}^{N,k}$.

Theorem 5.1 *Suppose that we have regularity and that $\mathcal{J}(\mathbf{u}) \geq \delta > 0$. Then, for sufficiently many subdivisions, for all $k \in \{1, ..., \omega_N\}$ and $\boldsymbol{\beta} \in \boldsymbol{\Omega}_N^d$ the Bézier control points $b_{\boldsymbol{\beta}}^{N,k}$ on each subtetrahedron verify:*

$$b_{\boldsymbol{\beta}}^{N,k} = \mathcal{P}(\mathcal{J})(T) + \mathcal{O}\Big(\max_{i \neq j} \|\mathbf{t}_i^{N,k} - \mathbf{t}_j^{N,k}\|^2\Big). \tag{37}$$

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

$$N \quad \sim \quad \left\lceil \log_2 \left(\frac{\sqrt{\delta}}{h} \right) \right\rceil \tag{38}$$

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

PROOF. Consider the k-th subtetrahedron $\Delta^{N,k}$ on the N-th subdivision. For each multi-index $\boldsymbol{\gamma} = (\gamma_0, \cdots, \gamma_d) \in \boldsymbol{\Omega}_n^d$, we define $\mathbf{q}_{\boldsymbol{\gamma}} := \sum_{j=0}^d (\gamma_j/n) \mathbf{t}_j^{N,k}$. Consider the following two sequences of the same length

$$\mathbf{T} := (\underbrace{\mathbf{t}_0^{N,k}, \cdots, \mathbf{t}_0^{N,k}}_{\gamma_0}, \cdots, \underbrace{\mathbf{t}_d^{N,k}, \cdots, \mathbf{t}_d^{N,k}}_{\gamma_d}) \tag{39}$$

$$\widetilde{\mathbf{T}} := (\underbrace{\mathbf{q}_{\boldsymbol{\gamma}}, \cdots, \mathbf{q}_{\boldsymbol{\gamma}}}_{n}) \tag{40}$$

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

$$\mathcal{P}(\mathcal{J})(\widetilde{\mathbf{T}}) = \mathcal{P}(\mathcal{J})(\mathbf{T}) + \sum_{|\boldsymbol{\alpha}|=1} (\mathbf{T} - \widetilde{\mathbf{T}})^{\boldsymbol{\alpha}} \partial_{\boldsymbol{\alpha}} \mathcal{P}(\mathcal{J})(\mathbf{T}) + \mathcal{O}\Big(\max_{i \neq j} \|\mathbf{t}_j^{N,k} - \mathbf{t}_j^{N,k}\|^2\Big). \tag{41}$$

Since the blossom is symmetric, the first partial derivatives are the same such that for all $|\boldsymbol{\alpha}| = 1$ we have $\partial_{\boldsymbol{\alpha}} \mathcal{P}(\mathcal{J})(\mathbf{T}) = K$. Moreover, for $|\boldsymbol{\alpha}| = 1$ we have $\boldsymbol{\alpha} = (0, ..., 0, 1, 0, ..., 0)$ where the unity is at some i-th entry. Thus, by using the notation from (23), we have $(\mathbf{T} - \widetilde{\mathbf{T}})^{\boldsymbol{\alpha}} = \boldsymbol{t}_i^{N,k} - \mathbf{q}_{\boldsymbol{\gamma}}$. As a consequence, the above Taylor expansion becomes

$$\mathcal{P}(\mathcal{J})(\widetilde{\mathbf{T}}) = \mathcal{P}(\mathcal{J})(\mathbf{T}) + K \sum_{i=0}^{3} \gamma_i (\mathbf{t}_i^{N,k} - \mathbf{q}_{\boldsymbol{\gamma}}^{N,k}) + \mathcal{O}\Big(\max_{p \neq q} \|\mathbf{t}_p^{N,k} - \mathbf{t}_q^{N,k}\|^2\Big). \tag{42}$$

On the other hand, we have

$$\begin{aligned} \sum_{i=0}^{3} \gamma_i (\mathbf{t}_i^{N,k} - \mathbf{q}_{\boldsymbol{\gamma}}^{N,k}) &= \sum_{i=0}^{3} \gamma_i \Big[(1 - \frac{\gamma_i}{n}) \mathbf{t}_i^{N,k} - \sum_{j \neq i} \frac{\gamma_j}{n} \mathbf{t}_j^{N,k} \Big] \\ &= \Big[\sum_{i=0}^{3} \gamma_i \sum_{j \neq i} \frac{\gamma_j}{n} \mathbf{t}_i^{N,k} \Big] - \Big[\sum_{j=0}^{3} \sum_{j \neq i} \frac{\gamma_j}{n} \gamma_j \mathbf{t}_i^{N,k} \Big] = 0. \end{aligned}$$

Thus, the sum with respect to $|\boldsymbol{\alpha}| = 1$ in (41) vanishes and we obtain

$$\mathcal{P}(\mathcal{J})(\widetilde{\mathbf{T}}) = \mathcal{P}(\mathcal{J})(\mathbf{T}) + \mathcal{O}\Big(\max_{i \neq j} \|\mathbf{t}_i^{N,k} - \mathbf{t}_j^{N,k}\|^2\Big) = \mathcal{P}(\mathcal{J})(\mathbf{T}) + \mathcal{O}\Big(\frac{1}{2^{2N}} \max_{i \neq j} \|\mathbf{t}_i - \mathbf{t}_j\|^2\Big).$$

We use the relation (32) between the control points and the blossom in order to deduce

$$b_{\boldsymbol{\gamma}}^{N,k} \geq \delta + \mathcal{O}\Big(\frac{1}{2^N}h\Big)^2. \tag{43}$$

The coefficients $b_{\boldsymbol{\gamma}}$ are therefore positive for sufficiently many subdivisions. Moreover, the expected number of subdivision N verifies $2^{-N} \quad \sim \quad \sqrt{\delta}/h$.

■

The above method for tetrahedra gives rise to an adaptive subdivision technique which is similar to the one in Section 3.2 and [6] (see also Figs.11(c) and 11(d)).

5.4 Efficiency improvment and practical results

Let us now see the improvment of 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 [9, 7]. Since they are defined on $[-1,+1]$ but we need results in $[0,1]$, we introduce the modified Jacobi polynomials $J_n^{(\alpha,\beta)}(t) := P_n^{(\alpha,\beta)}(2t-1)$ for $t \in [0,1]$. In fact, 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 $\alpha \in \mathbb{N}$. We need a norm for the polynomials on the unit tetrahedron. Consider a function f which is a multivariate polynomial of degree n:

$$f(x_1,x_2,x_3) = \sum_{|\boldsymbol{\gamma}|\leq n} b_{\boldsymbol{\gamma}} x_1^{\gamma_1} x_2^{\gamma_2} x_3^{\gamma_3} \qquad\qquad \mathbf{x} = (x_1,x_2,x_3) \in \Delta^3_{\text{ref}}. \tag{44}$$

Let us introduce for the polynomial f the quantity

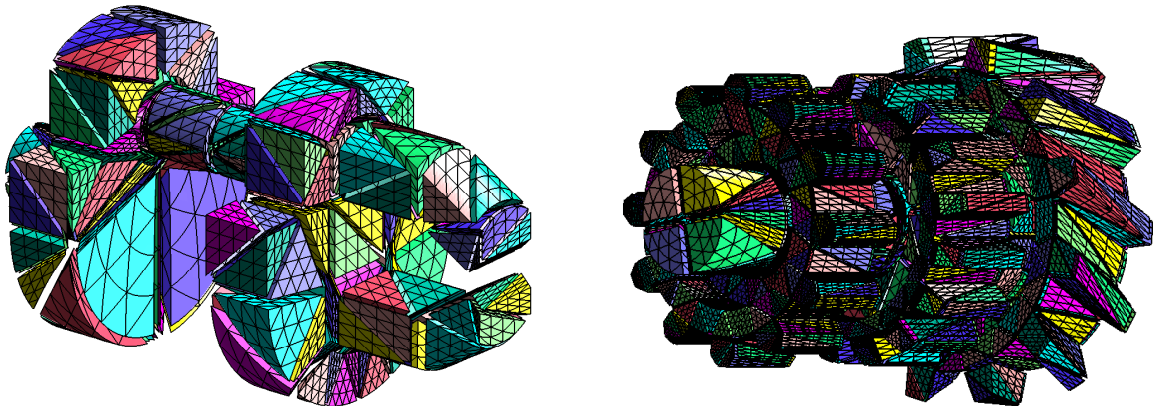
$$\|f\|_{\Delta^3_{\text{ref}}} := \max_{\mathbf{x}\in\Delta^3} \left[\prod_{i=1}^{3}(1-x_i)^k x_i^k\right] |f(\mathbf{x})|. \tag{45}$$

By using continuity argument, we can show that $\|\cdot\|_{\Delta^3_{\text{ref}}}$ defines a norm on the polynomials of Δ^3_{ref}. In order to reduce the computational cost, we would like to find $\widetilde{f}$ 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 $\widetilde{f}$ will be done by using the Jacobi polynomial and the error $\|f-\widetilde{f}\|_{\Delta^3_{\text{ref}}}$ will be analyzed. Consider the case $k \in \mathbb{N}$ such that $k \geq \left\lceil \frac{\alpha}{2} + \frac{1}{4} \right\rceil$. Consider a multivariate polynomial $f \in \Pi_n(\mathbb{R}^3)$ with bounded n-th derivatives such that for all $|\boldsymbol{\gamma}| = n$, we have:

$$\frac{1}{\boldsymbol{\gamma}!}\big|\partial_{\boldsymbol{\gamma}} f(\mathbf{x})\big| = \frac{1}{\gamma_1!\cdots\gamma_3!}\left|\frac{\partial^{\gamma_1}}{\partial x_1^{\gamma_1}}\cdots\frac{\partial^{\gamma_3}}{\partial x_3^{\gamma_3}} f(x_1,x_2,x_3)\right| \leq C \qquad \forall\, \mathbf{x} \in \Delta^3_{\text{ref}}. \tag{46}$$

It is shown in [13] that there is a polynomial $\widetilde{f} \in \Pi_{n-1}(\mathbb{R}^3)$ of the following form

$$\widetilde{f}(\mathbf{x}) := \sum_{|\boldsymbol{\gamma}|\leq n-1} c_{\boldsymbol{\gamma}} \prod_{j=1}^{3} x_j^{\gamma_j} - \sum_{|\boldsymbol{\gamma}|=n} c_{\boldsymbol{\gamma}} \sum_{q=1}^{3} R_{\gamma_q}(x_q) \prod_{j=1}^{q-1} x_j^{\gamma_j} \prod_{j=q+1}^{3} (x_j^{\gamma_j} - R_{\gamma_j}(x_j)) \tag{47}$$



(a) (b)

Figure 12: CAD decomposition for FEM

such that the error is given by

$$\|f-\widetilde{f}\|_{\Delta^3_{\text{ref}}} \leq K\frac{1}{2^{2n}}\sum_{|\boldsymbol{\gamma}|=n}\prod_{i=1}^{3}2^{-2k+\alpha+0.5}\frac{(1+\alpha/\max\{1,\gamma_i\})}{\binom{2\gamma_i+2\alpha}{\gamma_i}}, \tag{48}$$

where the constant K depends only on α. It is beyond this article to detail the proof [13] but mainly we use the next idea. 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 $\widetilde{J}_q^{(\alpha)} := J_q^{(\alpha)}/(2^q l_q)$ is monic. The result is obtained from Taylor expansion of f at $\mathbf{0} = (0,\cdots,0)$. Note that in the previous analysis, the multivariate function f from relation (44) is given in monomial basis but we need results in Bézier structure as discussed in the former sections. That holds also for the degree reduced polynomial $\widetilde{f}$. In order to obtain Bézier representation, one can represent the Jacobi polynomials in terms of Bézier as discussed in [9]:

$$P_n^{(\alpha)}(t) = \sum_{i=0}^{n}(-1)^{n-i}\frac{\binom{n+\alpha}{i}\binom{n+\alpha}{n-i}}{\binom{n}{i}}B_i^n(t). \tag{49}$$

Then, one needs only to transform that in the appropriate simplex of definition as we described in (33). Afterwards, the function $\widetilde{f}$ can be represented in Bézier of degree $(n-1)$ by using the pyramid algorithm of Section 5.2. The Bézier control points of $\widetilde{f}$ can be deduced from the blossom and the pyramid algorithm by applying formula (32). In [13], we have presented some numerical results about the estimates in (48).

Finally, let us now present some practical results of the above curved tetrahedral mesh generation. We have used the two models which are found in Fig.12 where the input CAD data are obtained from IGES formats. The CAD models have respectively 50 and 201 surfaces (trimmed and untrimmed). The resulting hierarchical meshes have respectively 374 and 1451 coarse curved tetrahedra. Our software is not yet optimal but for now the first model requires less than 2 minutes and the second one less than 3 minutes for decomposition with points evaluation at level 3. The computation has been performed on a machine with processor Intel Core 2.16GHz running Windows Vista.

REFERENCES

[1] G. Brunnett, Geometric Design with Trimmed Surfaces. *Computing Supplementum*, **10**, 101–115, 1995.

[2] W. Dahmen and A. Kunoth, Adaptive Wavelet Methods for Linear–Quadratic Elliptic Control Problems: Convergence Rates. *SIAM J. Contr. Optim.*, **43**, No. 5, 1640–1675, 2005.

[3] C. de Boor, *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.

[4] H. Edelsbrunner and N. Shah, Incremental topological flipping works for regular triangulations. *Algorithmica*, **15**, No. 3, 223–241, 1996.

[5] M. Floater, Arc Length Estimation and the Convergence of Polynomial Curve Interpolation. *BIT*, **45**, No. 4, 679–694, 2005.

[6] H. Harbrecht and M. Randrianarivony, From Computer Aided Design to Wavelet BEM. *To appear in Journal Computing and Visualization in Science (article in press)*, 2009.

[7] H. Kim and Y. Ahn, Good Degree Reduction of Bézier Curves Using Jacobi Polynomials. *Pergamon*, **40**, 1205–1215, 2000.

[8] H. Ledoux, C. Gold and G. Baciu, Flipping to Robustly Delete a Vertex in a Delaunay Tetrahedralization. *ICCSA*, **1**, 737–747, 2005.

[9] A. Rababah, Jacobi-Bernstein Basis Transformation. *Comput. Methods Appl. Math.* **4**, No. 2, 206–214, 2004.

[10] M. Randrianarivony, On Global Continuity of Coons Mappings in Patching CAD Surfaces. *To appear in Computer-Aided Design (article in press)*, 2009.

[11] M. Randrianarivony, Geometric Processing of CAD Data and Meshes as Input of Integral Equation Solvers, Ph.D. thesis, Technische Universität Chemnitz, 2006.

[12] M. Randrianarivony and G. Brunnett, Molecular Surface Decomposition using Geometric Techniques, *Conf. Bildverarbeitung für die Medizine*, Berlin, pp. 197–201, 2008.

[13] M. Randrianarivony, Tetrahedral Transfinite Interpolation with B-patch faces: construction and regularity, *INS Preprint* 0803, University of Bonn, 2008.

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

[15] H.–P. Seidel, Polar Forms for Geometrically Continuous Spline Curves of Arbitrary Degree, *ACM Trans. Graph.* **12**, No. 1, 1–34, 1993.

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

[17] H. Zimmer, *Voronoi and Delaunay Techniques*, Lecture Notes, Computer Sciences VIII, RWTH Aachen, 2005.