Subdivision Algorithms and Convexity Analysis for Rational Bézier Triangular Patches

R. Sharma\(^{1}\) and O. P. Sha\(^{2}\)

**ABSTRACT**

Triangular surfaces are important because in areas where the geometry is not similar to rectangular domain, the rectangular surface patch will collapse into a triangular patch. In such a case, one boundary edge may collapse into a boundary vertex of the patch, giving rise to geometric dissimilarities (e.g. shape parameters, Gaussian curvature distribution, cross boundary continuities etc.) and topological inconsistency. Furthermore, since a triangular patch (i.e. defined as a closed polygon) is a basic figure in algebraic topology; hence any fairly irregular complex geometry can be efficiently modeled/designed/generated with rational triangular patches. But, still triangular surfaces over triangular domain remain relatively unexplored as compared to rectangular surfaces over rectangular domain. In the present paper, we present subdivision algorithms for rational Bézier triangular patches for arbitrary subdivision. The algorithms are generated by constructing degenerate Bézier triangles and tetrahedron and using their edges to subdivide the original curve. Using the ‘classical probability theory’, we derive the algorithms by constructing higher dimensional degenerate Bézier simplices and incorporate their triangular sub-simplices. Additionally, we analyze the algorithms for convexity and derive conditions that preserve convexity.

**AMS MSC 2000:** 65D07, 65D10, 65D17, and 68U07.

**ACM CCS:** G.1.1, G.1.2, and I.3.5.

**Keywords:** Algorithms, Convex analysis, Rational Bézier patches, Triangular arrays.

1. **INTRODUCTION**

In engineering sciences (i.e. aircraft science, mechanical science, and ship science) a surface is designed/modelled/generated with a set of discrete input points called control points (i.e. *control net* or *offset points*). The control points are either supplied by the designer or collected via any mean of sampling or collected from scanned data. In these processes it is not always

---

\(^{1}\) Assistant Professor, Department of Ocean Engineering, Indian Institute of Technology Madras, Chennai (TN) – 600 036, India.

\(^{2}\) Professor, Design Laboratory, Department of Ocean Engineering and Naval Architecture, Indian Institute of Technology, Kharagpur (WB) – 721 302, India. E-mail: ops@naval.iitkgp.ernet.in

* Author for Correspondence: rajivatri@iitm.ac.in
possible to have a set of control points which are distributed fairly and form a net/wire frame with a good distribution of aspect ratio. In design and analysis practices a derivationally continuous (i.e. $C^n$ where $n = 0, 1, \text{ and } 2, \text{ and/or a combination of } n = 0, 1, \text{ and } 2$) surface is generated through or over the given control points. The imposition of derivational continuities does not ensure that the elements in the generated surface will have a good distribution of the aspect ratio. Hence, there is no practical control over the aspect ratio of the elements generated in surface modeling/design/generation scheme. Furthermore, for surfaces of underwater or floating bodies, the flow changes drastically in the normal direction to the surface and little in the tangential longitudinal direction. This forces the discretized elements in the surface to have extreme side ratios, Bertram (2000). Again, in case of some surfaces the manufacturing related features (i.e. the intrinsic properties such as unit tangent vector, curvature vector, bi-normal vector, torsion, and higher-order transversal and tangential derivatives; Gaussian and principal mean curvatures) also show drastic change from one end to another. This demands the surface modeling/design/generation to be done over differently shaped topological domains and geometries, and therefore in some cases of surface generation process, the poor aspect ratio of individual domain cannot be avoided.

The poor aspect ratios of long, narrow, and thin elements/patches, often affect the stability, efficiency, and robustness of the solution process in any of the surface interrogation processes. It is not only difficult to compute either numerical or computational objective function within a domain or boundary of a long, narrow, and thin element/patch; but even the simple geometric and algebraic functions fail to converge for a long, narrow, and thin element/patch.

Subdivision is important because it is in this process that not only the number of elements can be increased but also the aspect ratios of the discretized elements can be improved. Furthermore, in some problems of shape interrogation like surface/surface intersection (e.g. Dokken (1985), Dokken et al. (1989) and Barnhill and Kersey (1990)), surface/plane intersection (e.g. Sharma and Sha (2007)) and solution of nonlinear polynomial equations (e.g. Reuter et al. (2008)), and in computer graphics and visualization like ray tracing of rational trimmed patches (e.g. Stürzlinger (1998)) the process of subdivision coupled with convex hull property is used to improve the computations in algorithms via splitting the triangles into subtriangles and then using these subtriangles directly in the algorithms or using only those that are of interest. Additionally, subdivision algorithms are important in CAD because they allow easy trimming of surfaces, application of extension of formulas from vertices at arbitrary points along the surface and simplification of the computation of curvatures and normals. In addition subdivision algorithms are used in geometric construction algorithms and these geometric construction algorithms are considered simple, numerically stable algorithms to evaluate the surface properties at any specified parameter value. The improvement in the aspect ratio of a triangle via the process of subdivision is shown in Figure 1-1. The triangular surfaces can be subdivided with an extended form of the ‘de Casteljau algorithm’ (de Casteljau (1963 and 1986)) or any of its variant. The triangular control point net is recursively subdivided with tri-linear interpolation for a parameter set,
and the results being the control net for the surfaces and a point on the surface. Furthermore, the geometric, topological, and differential geometry properties are computed as for the initial surface. The successive recursion to the resulting subsurfaces, with de Casteljau algorithm leads to long, narrow, and thin triangular patches with poor aspect ratio. The triangular patches with poor aspect ratio often cause numerical and computational problems (Stürzlinger (1998)), and thus the de Casteljau algorithm is of only limited purpose for arbitrary subdivision.

Farin (1979 and 1991) presented a simple geometric construction based subdivision algorithm which is an extended form of ‘de Casteljau algorithm’. He subdivides the Bézier triangle directly without the need for evaluation of the blending functions and their derivatives at arbitrary parameter values or to invert a matrix (which is computationally expensive). Using the ‘classical probability theory’ (Yates and Goodman (2004)), and exploring the intrinsic probabilistic relationships between Bernstein-Bézier polynomials and their use in triangular Bernstein-Bézier surfaces; Goldman (1983) presented general subdivision algorithms based on arbitrary subdivision that allows explicit control over the aspect ratio. Though, his algorithms use ‘3/4/5 – simplex’ of polar value numerical computations which are computationally expensive, but they allow insight into the subdivision process, control of subdivision steps, and offer a wide variation in the subdivision patterns. Böhm (1983) discussed geometrical subdivision algorithms for multivariate splines that are variations of ‘de Casteljau algorithm’ and ‘de Boor algorithm’ (de Boor (1972 and 1982)). His algorithms are based upon geometric arguments and they split the triangle at mid point of the edges or edges are divided at equal intervals. Using subdivision at pre selected ratios he minimizes the number of calls to the standard ‘de Casteljau algorithm’. Prautzsch (1984) and Prautzsch et al. (2002) also worked in similar line. Fillip (1986) explored subdivision strategies using a variation of ‘de Casteljau algorithm’ by first dividing an edge of a Bézier triangle at mid point and then generating sub divisions by connecting mid point of the edge with vertices of the triangle. It can be noted here that in the existing literature subdivision strategies have been explored mainly for a Bézier triangle and subdivision strategies for a rational triangular patch remain unexplored. It can be argued though naively that subdivision for Bézier can be computed easily via the technique of ‘blossoming’. However, in ‘blossoming’ no deeper understanding of the subdivision process is achieved and to deal with ‘blossoming’ in a projective space restricts the open choices in the process of subdivision.
Our interest is to explore arbitrary subdivision of rational triangular Bernstein-Bézier patch. In our investigation we retain the use of probabilistic arguments (Yates and Goodman (2004) and Goldman (1983)), and use of geometric construction (de Casteljau (1963), Farin (1979 and 1991), and Goldman (1983)). We investigate subdivision strategies that allow insight into the subdivision process, control of subdivision steps, and offer a wide variation in the subdivision patterns. Also, we explore the relationship between subdivision and convexity. Furthermore, in our analysis of subdivision we show that the presented algorithms are general and other known subdivision strategies of subdivisions can be derived from the presented algorithms.

The remaining of the paper is organized as follows: Section 2 presents some mathematical preliminaries that are essential for the presentation of this paper; Section 3 presents subdivision algorithms; Section 4 presents convexity analysis of subdivision; and Section 5 concludes the paper with identifying some future scope of research.

2. MATHEMATICAL PRELIMINARIES

2.1 A Rational Triangular Bernstein-Bézier Patch

Following Lorentz (1986), Farin (1986 and 2001) and Hoschek et al. (1992), let us define a rational triangular Bernstein-Bézier patch as,

\[
RBBTP(u) = \frac{\sum_{d=0}^{d} w_i B^d_i (u) P_i}{\sum_{d=0}^{d} w_i B^d_i (u)}
\]  

(2.1)

\[
= \sum_{d=0}^{d} \left[ \frac{\sum_{i=0}^{d} w_i B^d_i (u) P_i}{\sum_{d=0}^{d} w_i B^d_i (u)} \right]
\]

(2.2)

where \( d \in \mathbb{N} \), \( i = (i_0, i_1, i_2) \in \mathbb{N}_0^3 \), \( d = i_0 + i_1 + i_2 \), \( u = (u_0, u_1, u_2) \in \mathbb{R}^3 \), \( u_0 + u_1 + u_2 = 1 \), \( 0 \leq u_0, u_1, u_2 \leq 1 \) are the barycentric co-ordinates of a point in a domain triangle, \( B^d_i (u) = \binom{(d!)}{(i_0!, i_1!, i_2!)} (u_0^{i_0} u_1^{i_1} u_2^{i_2}) \) are the generalized Bernstein polynomials and the \( w_i \in \mathbb{R} \) are the weights associated with the control vertices \( P_i = (x_i, y_i, z_i) \in E^3 \). If all the weights \( w_i \) are positive then it is well known via the de Casteljau algorithm (de Casteljau (1986)) that the rational triangular Bernstein-Bézier patch \( RTBBP(u) \) satisfies the convex hull property as,

\[
\min_{d=0} P_i \leq RTBBP(u) \leq \max_{d=0} P_i.
\]

(2.3)
Throughout this work, in our all formulations we assume that the weights \((w_i)\) are positive.

### 2.2 A Multinomial Distribution and Bernstein-Bézier Simplices

For a random weight dependent event \(RE\), let \(\mathbf{IO} = (IO_0, \ldots, IO_m)\) be all the possible outcomes, \(w = (w_0, \ldots, w_m)\) be the respective weights assigned to each outcome and 
\(u = (u_0, \ldots, u_m)\) be their respective probabilities of occurrence. From basic probability theory it means that \(w_k \geq 0\), \(u_k \geq 0\) and \(k = 0, \ldots, m\) and \(|u| = \sum_{k=0}^{m} u_k = 1\). Further, let \(i = (i_0, \ldots, i_m)\) be an \((m+1)\) tuple of integers which satisfies \(i_k \geq 0\) where \(k = 0, \ldots, m\) and \(|i| = \sum_{k=0}^{m} i_k = d\).

Now we define \(B_i^d(u)\): normalized probability of exactly \(i_k\) successes of \(k\) weight dependent trials. The distribution defined by \(B_i^d(u)\) is multinomial distribution and it gets reduces to a binomial distribution when \(m = 1\). Since, we are utilizing basic probability theory it is clear that the following properties hold as:

\begin{align*}
\text{a. } & B_i^d(u) \geq 0. \\
\text{b. } & \sum_{i=0}^{d} B_i^d(u) = 1, |i| = d. 
\end{align*}

The functions \(B_i^d(u)\) are explicitly represented in polynomial forms (Lorentz (1986), Yates and Goodman (2004)). Following Goldman (1983) and Farin (1986 and 1991), let \(\Delta^m = \{u = (u_0, \ldots, u_m) | u_k \geq 0, \text{ and } |u| = 1\}\) denote the standard \(m\) dimensional simplex and let \(\{P_i^* = (P_i, w_i) = (P_{i_0}, w_{i_0}), \ldots, (P_{i_m}, w_{i_m}) | i_k \geq 0, \text{ and } |i| = d\}\) be a collection of points. The rational \(m\) dimensional Bernstein-Bézier simplex of degree \(d\) is determined by the points \(P_i^*\) as \(\sum_{i=0}^{d} B_i^d(u)P_i^* |u \in \Delta^m, |i| = d\). The points \(P_i^*\) are basically the generators of the rational Bernstein-Bézier simplex, and there exists a canonical map \(B^d[P_i^*]: \text{standard rational simplex} \rightarrow \text{rational Bernstein-Bézier simplex} \) and that is defined as

\[ B^d[P_i^*](u) = \sum_{i=0}^{m} B_i^d(u)P_i^*, |i| = d. \]

It is clear that the standard rational simplex space basically serves as the parameter space for the rational Bernstein-Bézier simplex. Furthermore, it is also clear from the Equations (2.4) and (2.5) that the rational Bernstein-Bézier simplex lies in the convex hull
of its generators and is free from the choices of origin co-ordinates. For \( m = 1 \), the rational Bernstein-Bézier simplex reduces to a rational Bernstein-Bézier curve, for \( m = 2 \) the rational Bernstein-Bézier simplex reduces to a rational triangular Bernstein-Bézier surface patch and for \( m = 3 \) the rational Bernstein-Bézier simplex reduces to a rational Bernstein-Bézier triangular tetrahedral solid. A rational Bernstein-Bézier simplex (for \( m = 2 \)) is shown in Figure 1-2.

The subsets of standard rational \( m \) dimensional simplex are defined by the equations:

\[
\begin{align*}
\sum_{j=0}^{k} u_j &= 1 \\
\sum_{j=0}^{k} u_j &= 0
\end{align*}
\]

where \( u_j \) is for rational \( m-k \) dimensional subsimplex of the standard rational \( m \) dimensional simplex. A subset of a rational \( m \) dimensional Bernstein-Bézier simplex is of \( p \) dimensional subsimplex if it is the image of a \( p \) dimensional subsimplex of the standard rational \( m \) dimensional simplex under the canonical map. If the dimension is 1 then the subsimplices are rational edges, if the dimension is 2 then the subsimplices are rational subtriangular patches, if the dimension is 3 then the subsimplices are rational subtetrahedrons and in general \( m - 1 \) dimensional simplices are rational faces. Now, it is clear that rational \( m-k \) dimensional subsimplices of a rational Bernstein-Bézier simplex are rational Bernstein-Bézier simplices, and the image of the standard rational \( m-k \) dimensional subsimplex of \( \Delta^m \) is defined as \( u_{i_1} = u_{i_2} = \ldots = u_{i_k} = 0 \) and is generated by the array as \( \{ P^*_i | i_1 = i_2 = \ldots = i_k = 0 \} \). Additionally, the rational subtriangular patches of a rational Bernstein-Bézier simplex are rational triangular Bernstein-Bézier surfaces, and the image of the standard rational subtriangular patch of \( \Delta^m \) is defined as \( u_{i_1} = 0 \), where \( k \neq 1,2,3 \), and is generated by the array as \( \{ P^*_i | i_1 = 0, \text{ where } k \neq 1,2,3 \} \).

![Figure 1-2: Description of a Rational Triangular Bernstein-Bézier Patch and its Subdivision](image-url)
2.3 Probabilistic Preliminaries for Bernstein-Bernstein-Bézier basis Functions

Let \( RE_k, k = 0, ..., m \) be the \((m+1)\) tuple defined as \( RE_k = (0, ..., 1, ..., 0) \), and \( RE_k = 1 \) for \( k \)-th tuple and 0 for every other tuple.

**Lemma 1:** Following Lorentz (1986) and Yates & Goodman (2004), a general recursion formula is given by,

\[
B^{d+1}_i(u) = \sum_{k=0}^{m} u_k B^{d}_{i-RE_k}(u). \quad (2.6)
\]

**Proof:** In Equation (2.6) \( B^{d+1}_i(u) \) is the normalized probability of exactly \( i \) successes of \( IO_j \) where \( j = 0, ..., m \) in \( n+1 \) weight dependent trials. This can be further expanded into,

\[
B^{d+1}_i(u) = (\text{normalized probability of exactly } i \text{ successes of } IO_j \text{ where } j \neq k, \text{ and } i_{k-1} \text{ successes of } IO_j \text{ in } n \text{ weight dependent trials}) \times \\
(\text{probability of successes of } IO_j \text{ on } (n+1)-\text{th trial})
\]

\[
= \sum_{k=0}^{m} u_k B^{d}_{i-RE_k}(u).
\]

**Lemma 2:** Proceeding as previously, a contraction formula is given by,

\[
B_i^d \left( \sum_{k=0}^{m} u_k \right) = \sum_{i=0}^{n} B_i^{d}_{ih_0,...,ih_p} \left( u_1,...,u_n \right), \text{where } ih_1+...+ih_p = 1. \quad (2.7)
\]

**Proof:** In Equation (2.7) \( B_i^{d} \left( \sum_{k=0}^{m} u_k \right) \) is the normalized probability of exactly \( i \) successes of the outcomes that are represented by \( u_1 \) or \( u_2 \) or \( u_p \) in \( n \) weight dependent trials. This can be further expanded into,

\[
B_i^d \left( \sum_{k=0}^{m} u_k \right) = \sum_{i=0}^{n} \left( \text{normalized probability of exactly } ih_k \text{ successes of the outcomes that are represented by } u_k, k = 1,...,n_p \text{ in } n \text{ weight dependent trials, where } ih_1+...+ih_{n_p} = 1 \right)
\]

\[
= B_i^d \left( \sum_{k=0}^{m} u_k \right) = \sum_{i=0}^{n} B_i^{d}_{ih_0,...,ih_p} \left( u_1,...,u_{n_p} \right), \text{where } ih_1+...+ih_{n_p} = 1.
\]

**Lemma 3:** Proceeding as previously, an expansion formula is derived as,

\[
B^{d}_{i,h_0,...,ih_{k-1}}(u) = \sum_{i=0}^{k} B^{d}_{ih_i}(QS)B^{d}_{i}(u). \quad (2.8)
\]
Proof: In Equation (2.8), let \( IP = (IP_0, ..., IP_i) \) be all the possible weight dependent outcomes of some event \( EF \) and let \( QS = (qS_0, ..., qS_j) \) be their respective probabilities of occurrence. Then, 

\[
B^d_{i_0, ..., i_d} \left( u_{i_0}, ..., u_{i_d} QS, ..., u_{i_d} \right) = \text{normalized probability of exactly } i_j \text{ successes of } IO_j, j \neq k \text{ and exactly } i_h \text{ successes of } IO_k \text{ and } IP_{i_j} n_g = 0, ..., n_p \text{ in } n \text{ weight dependent trials of } RE \text{ and } EF
\]

\[
= \sum_{i=0}^{i_k} (n \text{ normalized probability of exactly } i_j \text{ successes of } IO_j, j = 0, ..., m \text{ in } n \text{ weight dependent trials}) \times (n \text{ normalized probability of exactly } i_h \text{ successes of } IP_{i_j} n_g = 0, ..., n_p, \text{ during the } i_k \text{ success of } IO_k)
\]

\[
= \sum_{i=0}^{i_k} B^d_{i_h} (QS) B^d_{i_d} (u).
\]

3. SUBDIVISION ALGORITHMS

3.1 1-step Subdivision Algorithm

Extending the earlier works by Farin (1986 and 1991), let the generators of a rational triangular Bernstein-Bézier surface patch which is defined as in Equation (2.2) be \( \left\{ P_j^* \right\} = \left\{ P_j, w_{P_j} \right\}, i = (i_0, i_1, i_2), |i| = d \), and let \( QS \) be a point in parameter space which is defined: \( QS = ((u, v, w), w_{QS}) \in \Delta^2 \) where \( (u, v, w) \) are the barycentric co-ordinate of \( QS \) within the domain of a rational triangular patch of vertices at \( \left\{ P_j^* \right\} \) and \( w_{QS} \) is the assigned weight of \( QS \). We define a tetrahedral array of points \( \left\{ P_j^i (QS) \right\}, j = (j_0, j_1, j_2), |j| + k = d \) by recursive subdivision as,

\[
P_j^0 (QS) = \frac{w_{P_j}}{w_{QS}} P_j,
\]

and

\[
P_j^{i+1} (QS) = u P_j^{i+1} (QS) + v P_j^{i+1} (QS) + w P_j^{i+1} (QS). \tag{2.10}
\]

We shall show that the tetrahedron arrays \( \left\{ P_j^i (QS) \right\} \) in Equations (2.9-2.10) generate a degenerate rational Bernstein-Bézier tetrahedron and that the triangular faces of this degenerate rational tetrahedron subdivide the original surface patch from \( S_z(u, v) \) (surface parametrized in parameters \( u, v \)) at \( QS \).
Lemma 4: Using Equations (2.6-2.10) we derive,

\[ P^k_j (QS) = \sum_{h=0}^{n} B_{ih}^k (QS) P_{j+h} = B^k \left[ P_{j+h} \right] (QS). \] (2.11)

Proof: From the Equation (2.6), the following can be derived,

\[ P^k_{j+1} (QS) = u B_{ih-RE_0}^k (QS) + v B_{ih-RE_1}^k (QS) + w B_{ih-RE_2}^k (QS). \]

And then the principle of inductive hypothesis results into,

\[ P^k_{j+1} (QS) = u P_{j+1,RE_0}^k (QS) + v P_{j+1,RE_1}^k (QS) + w P_{j+1,RE_2}^k (QS) \]

\[ = \sum \left[ u B_{ih-RE_0}^k (QS) + v B_{ih-RE_1}^k (QS) + w B_{ih-RE_2}^k (QS) \right] P_{j+h} \]

\[ = \sum B_{ih}^{k+1} (QS) P_{j+h} = B^{k+1} \left[ P_{j+h} \right] (QS). \]

Equation (2.11) is very general. For example, for non rational triangular Bernstein-Bézier patch, combining Equations (2.8-2.10) results into,

\[ B(QS) = P_{000}^d (QS), \]

which is the geometric construction algorithm for a triangular Bernstein-Bézier patch as given by Farin (1991).

Algorithm 1: Now we present the 1-step subdivision algorithm as,

\[ B^d \left[ P^k_j (QS) \right] (u) = B^d \left[ P_i \right] \left[ (u_0, u_1, u_2) + u_3 w^* QS \right] \]

\[ = B^d \left[ P_i \right] \left[ (u_0, u_1, u_2) + u_3 w^*_QS \right]. \] (2.12)

Proof: From Equations (2.7-2.8 and 2.11), we can have,

\[ B^d \left[ P^k_j (QS) \right] (u) = \sum_{i=0}^{l} \sum_{j=0}^{d-k} B_{ij}^d (u) P^k_j (QS) = \sum_{i=0}^{l} \sum_{j=0}^{d-k} B_{i,j}^d (u) \sum_{h=0}^{n} B_{ih}^k (QS) P_{j+h} \]

\[ = \sum_{i=0}^{l} \sum_{j=0}^{d-k} \left[ \sum_{h=0}^{n} B_{ih}^k (QS) B_{i,j}^d (u) \right] P_{j+h} = \sum_{i=0}^{l} \sum_{j=0}^{d-k} \left[ \sum_{h=0}^{n} B_{i,j+h}^d (u_0, u_1, u_2, u_3) \right] P_i \]

\[ = \sum_{i=0}^{l} B_{i}^d \left[ (u_0, u_1, u_2) + u_3 w^*_QS \right] P_i = B^d \left[ P \right] \left[ (u_0, u_1, u_2) + u_3 w^* QS \right]. \]
It is clear from Equation (2.12) that the Bernstein-Bézier tetrahedron generated by the array \( \{P_{i}^{k}(QS)\} \) is degenerate. Since, it collapses onto the rational triangular Bernstein-Bézier patch generated by the original array \( \{P_{i}^{k}\} \), therefore the triangular faces of this degenerate rational Bernstein-Bézier tetrahedron subdivides the original rational triangular Bernstein-Bézier patch on the surface \( S_{i}(u,v) \).

The selection of an arbitrary step \((QS)\) from \( \{P_{i}^{k}(QS)\} \) for each of the \( d \) steps of the sequence will compute a control point of the sub-triangular patch. The computation for all possible permutations of this step sequence will give all the control points of the subtriangular surface patch. Since, the four triangular faces of the standard rational 3D simplex are defined by the equations \( u_{i} = 0, i = 0,\ldots,3 \), the four triangular faces of our degenerate rational Bernstein-Bézier simplex are generated by the four arrays \( \{P_{i}^{k}(QS)\}|_{j_{i} = 0 \text{ or } k = 0} \). Geometrically, our 1-step subdivision algorithm is shown in Figure 1-3.

![Figure 1-3: 1-step Subdivision of the Rational Triangular Bernstein-Bézier Patch](image)

**Remark 1:** In general a rational triangular Bernstein-Bézier patch is subdivided into three rational subpatches by Equation (2.12) when \( P_{0,0,0}^{d}(QS) \) is arbitrarily positioned anywhere within the rational triangular patch. However, if \( P_{0,0,0}^{d}(QS) \) is allowed to lie on any of the edges (i.e. \( P_{d,0,0}^{d}P_{0,d,0} \) or \( P_{0,d,0}P_{0,0,d} \) or \( P_{d,0,0}P_{d,0,0} \)) then a rational triangular Bernstein-Bézier patch is subdivided into two rational subpatches by Equation (2.12). In that case the number of terms in the recursive definition of \( P_{i}^{k}(QS) \) is reduced from three to two. Hence, this reduction in terms can be utilized to implement an automatic algorithm in which \( P_{0,0,0}^{d}(QS) \) is simply taken as the mid point of any of the edges (i.e. \( P_{d,0,0}P_{0,d,0} \) or \( P_{0,d,0}P_{0,0,d} \) or \( P_{0,0,d}P_{d,0,0} \)) with suitable weights. Though this has limited application in practice because it does not allow explicit control on the quality of the generated triangular
subpatches but it is computationally fast. This approach has been taken in the works of Prautzsch (1984), Prautzsch et al. (2002) and Gallier (2006) for a non rational triangular Bernstein-Bézier patch.

### 3.2 2-step Subdivision Algorithm

Again extending the earlier works by Farin (1986 and 1991) and Goldman (1983), let the generators of a rational triangular Bernstein-Bézier surface patch which is defined as in Equation (2.2) be \( \{ P_i \} = \{ P_i, w_{i,j} \}, i = (i_0, i_1, i_2), \| i \| = d \). and let \( QS_0 \) and \( QS_1 \) be two points in parameter space which are defined as \( QS_0 = \{ (u_0, v_0, w_0), w_{Q5} \} \in \Delta^2 \), \( QS_1 = \{ (u_1, v_1, w_1), w_{Q5} \} \in \Delta^2 \), where \((u_0, v_0, w_0)\) and \((u_1, v_1, w_1)\) are the barycentric coordinates of \( QS_0 \) and \( QS_1 \) respectively within the domain of a rational triangular patch of vertices at \( \{ P_i \} \), and \( w_{Q5} \) and \( w_{Q5} \) are the assigned weights of \( QS_0 \) and \( QS_1 \) respectively.

Further let us define a 4D triangular array of points \( \{ P_j^{k} (QS_0, QS_1) \} \), \( j = (j_0, j_1, j_2) \), \( k (k_0, k_1) \), \( j + k = d \) by recursive subdivision as,

\[
P_j^{0,0} (QS_0) = \frac{w_{P_i}}{w_{Q5}} P_j^i, \quad (2.13)
\]

\[
P_j^{0,0} (QS_1) = \frac{w_{P_i}}{w_{Q5}} P_j^i, \quad (2.14)
\]

\[
P_j^{k+RE_0} (QS_0, QS_1) = u_0 P_{j+RE_0}^k (QS_0, QS_1) + v_0 P_{j+RE_1}^k (QS_0, QS_1) + w_0 P_{j+RE_2}^k (QS_0, QS_1), \quad (2.15)
\]

\[
P_j^{k+RE_1} (QS_0, QS_1) = u_1 P_{j+RE_0}^k (QS_0, QS_1) + v_1 P_{j+RE_1}^k (QS_0, QS_1) + w_1 P_{j+RE_2}^k (QS_0, QS_1). \quad (2.16)
\]

By applying construction (Equation (2.6 -2.8)) we get,

\[
P_j^{k,0} (QS_0, QS_1) = P_j^{k} (QS_0), \quad (2.17)
\]

and \( P_j^{0,k} (QS_0, QS_1) = P_j^{k} (QS_1) \).

The arrays of Equations (2.17-2.18) contain all the points of the tetrahedron array \( \{ P_j^{k} (QS) \} \) defined previously in Section (3.1). Now, we shall show that the array \( \{ P_j^{k} (QS_0, QS_1) \} \) generates a degenerate 4D rational triangular Bernstein-Bézier simplex whose triangular subsimplices subdivide the original rational Bernstein-Bézier surface patch.
at $QS_0$ and $QS_1$. Geometrically, it is clear that there exist two separate options to construct the point $P_{j+RE+E}^k (QS_0, QS_1)$ as either constructing first $P_{j+RE}^k (QS_0, QS_1)$, $P_{j+RE+E}^k (QS_0, QS_1)$, $P_{j+RE}^k (QS_0, QS_1)$, and then applying the recursion formulas of Equations (2.13-2.16) to compute $P_{j+RE+E}^k (QS_0, QS_1)$; or constructing first $P_{j+RE+E}^k (QS_0, QS_1)$, $P_{j+RE}^k (QS_0, QS_1)$, and then applying the recursion formulas of Equations (2.13-2.16) to compute $P_{j+RE+E}^k (QS_0, QS_1)$. The two constructions are similar and compute the same result. The similarity is the result of a recursion that is extended from Equations (2.13-2.16) as,

$$P_j^k (QS_0, QS_1) = \sum_{ih=0}^{n_d-n_p} \sum_{g=0}^{d-n_p} B_{ig}^{*k} (QS_0) B_{ih}^{*k} (QS_1) P_{j+ig+ih}. \quad (2.19)$$

**Algorithm 2:** Now we present the 2-step subdivision algorithm as,

$$B^{d} [P_j^k (QS_0, QS_1)] (u) = B^{d} [P_j^k] \left[ (u_0, u_1, u_2) + u_4 w_{P_{j+RE+E}}^* (QS_0) + u_4 w_{P_{j+RE+E}}^* (QS_1) \right]$$

$$= B^{d} [P_j^k] \left[ (u_0, u_1, u_2) + u_4 \frac{w_{P_{j+RE}}}{w_{QS_0}} (QS_0) + u_4 \frac{w_{P_{j+RE}}}{w_{QS_1}} (QS_1) \right]. \quad (2.20)$$

**Proof:** This result comes from Equations (2.6-2.8 and 2.19) and the proof is analogous to the proof of Equation (2.12).

It is clear from Equation (2.20) that the points $P_{0,0,0}^k (QS_0, QS_1)$ generate a rational Bernstein-Bézier curve along the edges from $B (QS_0)$ to $B (QS_1)$ that lies on the rational Bernstein-Bézier surface patch. Furthermore, the $4D$ rational triangular Bernstein-Bézier simplex generated by the array $\{P_j^k (QS_0, QS_1)\}$ collapses onto the rational triangular Bernstein-Bézier patch generated by the original array $\{P_j^*\}$. Therefore the triangular faces of this $4D$ rational triangular Bernstein-Bézier simplex subdivides the original rational triangular Bernstein-Bézier patch on the surface $S_5 (u, v)$. The selection of two arbitrary step $QS_0$ and $QS_1$ from $\{P_j^k (QS_0, QS_1)\}$ for each of the $d$ steps of the sequence will compute two control points of the sub-triangular patch. The computation for all possible permutations of this step sequence will give all the control points of the subtriangular surface patch. The triangular subsimplices of the standard $4D$ rational simplex are defined by the equations $\{u_h = u_{ih} = 0, ih \neq ih \}$. It means that the standard $4D$ rational simplex has 10 triangular subsimplices. And, these different triangular subsimplices can be grouped into 5 distinct types as,
Subdivision Algorithms and Convexity Analysis for Rational Bézier Triangular Patches

\[ P_j^{0,0}(QS_0) = \frac{w_{P_j}}{w_{QS_0}} P_j. \]  
\[ (2.21) \]

\[ P_j^{0,0}(QS_1) = \frac{w_{P_j}}{w_{QS_1}} P_j. \]  
\[ (2.22) \]

\[ P_j^{k_0,0}(QS_0,JQS_1) = P_j^{k_0}(QS_0) \quad \text{for one} \quad j_p = 0. \]  
\[ (2.23) \]

\[ P_j^{k_0,0}(QS_0,JQS_1) = P_j^{k_0}(QS_1) \quad \text{for one} \quad j_p = 0. \]  
\[ (2.24) \]

\[ P_j^{k_0,k_0}(QS_0,JQS_1) = \sum_{ih=0}^{d_{ih}} \sum_{iq=0}^{d_{iq}} B_{d_{ih}}^{k_0} (QS_0) B_{d_{iq}}^{k_0} (QS_1) P_{j+iq+ih} \quad \text{for one} \quad j_p \neq 0. \]  
\[ (2.25) \]

The arrays of Equations (2.21-2.22) generate the original rational triangular surface patch. The arrays of Equation (2.23-2.24) generate the subdivision each at \( QS_0 \) and \( QS_1 \) like the 1-step subdivision of Equation (2.12), and generate six triangular subpatches. The six triangular subpatches have vertices at: subpatch 1 - whose vertices are at \( P_{d,0,0}, P_{0,0,d} \); subpatch 2 - whose vertices are at \( P_{d,0,0}, P_{0,0,d} \) and \( P_{0,0,0}, (QS_0) \); subpatch 3 - whose vertices are at \( P_{d,0,0}, P_{0,0,d} \) and \( P_{0,0,0}, (QS_1) \); subpatch 4 - whose vertices are at \( P_{d,0,0}, P_{0,0,d} \) and \( P_{0,0,0}, (QS_0) \); subpatch 5 - whose vertices are at \( P_{0,0,0}, (QS_0), (QS_1) \); and subpatch 6 - whose vertices are at \( P_{0,0,0}, (QS_0), (QS_1) \). And, the array of Equation (2.25) generates the subdivisions which form three 'wedge shaped' subpatches (i.e. subpatch 1: whose vertices are at \( P_{d,0,0}, P_{0,0,d} \) and \( P_{0,0,0}, (QS_0) \); subpatch 2: whose vertices are at \( P_{d,0,0}, (QS_0), P_{0,0,0}, (QS_1) \) and \( P_{0,0,0}, (QS_0) \); subpatch 3: whose vertices are at \( P_{d,0,0}, (QS_0), P_{0,0,0}, (QS_1) \) and \( P_{0,0,0}, (QS_0) \)). Geometrically, our 2-step subdivision algorithm is shown in Figure 1-4.

**Remark 2:** In general the subpatches generated by Equation (2.20) overlap when \( P_{0,0,0}, (QS_0) \) and \( P_{0,0,0}, (QS_1) \) are arbitrarily positioned anywhere within the rational triangular patch. However, if both \( P_{0,0,0}, (QS_0) \) and \( P_{0,0,0}, (QS_1) \) are allowed to lie on any of the edges (i.e. \( P_{d,0,0}, P_{0,0,d} \) or \( P_{0,0,d}, P_{d,0,0} \) or \( P_{0,0,d}, P_{d,0,0} \)) then a rational triangular Bernstein-Bézier patch is subdivided into three subpatches by Equation (2.20). In that case the number of terms in the recursive definition of \( P_j^{k_0}(QS_0,JQS_1) \) is reduced from six to four. Hence, this reduction in terms can be utilized to implement an automatic algorithm in which \( P_{0,0,0}, (QS_0) \) and \( P_{0,0,0}, (QS_1) \) are simply taken as the two equidistant points (dividing at a ratio of 1 : 2 and 2 : 1) at any of the edges (i.e. \( P_{d,0,0}, P_{0,0,d} \) or \( P_{0,0,d}, P_{0,0,d} \) or \( P_{0,0,d}, P_{d,0,0} \)) with suitable weights. Though this has limited application in practice because it does not allow explicit control on the quality of the generated triangular subpatches but it is computationally fast. This approach
has been taken in the works of Prautzsch (1984), Prautzsch et al. (2002) and Gallier (2006) for a non rational triangular Bernstein-Bézier patch.

Figure 1-4: 2-step Subdivision of the Rational Triangular Bernstein-Bézier Patch

3.3 3-step Subdivision Algorithm

Again extending the earlier works by Farin (1986 and 1991) and Goldman (1983), let the generators of a rational triangular Bernstein-Bézier surface patch which is defined as in Equation (2.2) be \( \{ P_i \} = \{ P_i^r, w_i \}, i = (i_0, i_1, i_2), |i| = d \), and let \( QS_0, QS_1, QS_2 \) be three points in parameter space which are defined as

\[
QS_0 = (u_0, v_0, w_0), w_{QS_0} \in \Delta^2,
QS_1 = (u_1, v_1, w_1), w_{QS_1} \in \Delta^2,
QS_2 = (u_2, v_2, w_2), w_{QS_2} \in \Delta^2
\]

where \( (u_0, v_0, w_0), (u_1, v_1, w_1), (u_2, v_2, w_2) \) are the barycentric co-ordinates of \( QS_0, QS_1 \) and \( QS_2 \) respectively within the domain of a rational triangular patch of vertices at \( \{ P_i \} \), and \( w_{QS_0}, w_{QS_1} \) and \( w_{QS_2} \) are the assigned weights of \( QS_0, QS_1 \) and \( QS_2 \) respectively. Further, let us define a 5D triangular array of points \( \{ P^r_j (QS_0, QS_1, QS_2) \} \), \( j = (j_0, j_1, j_2) \), \( k = (k_0, k_1) \), \( |j| + k = d \) by recursive subdivision as,

\[
P^r_{j_0,j_1,j_2} (QS_0) = \frac{w_{QS_0}}{w_{QS_0}} P_{j_0}, \tag{2.26}
\]

\[
P^r_{j_0,j_1,j_2} (QS_1) = \frac{w_{QS_1}}{w_{QS_1}} P_{j_1}, \tag{2.27}
\]

\[
P^r_{j_0,j_1,j_2} (QS_2) = \frac{w_{QS_2}}{w_{QS_2}} P_{j_2}, \tag{2.28}
\]

\[
P^k_{j_0,j_1,j_2} (QS_0, QS_1, QS_2) = u_0 P^k_{j_0+k_0} (QS_0, QS_1, QS_2) + v_0 P^k_{j_1+k_1} (QS_0, QS_1, QS_2) + w_0 P^k_{j_2+k_2} (QS_0, QS_1, QS_2), \tag{2.29}
\]
Subdivision Algorithms and Convexity Analysis for Rational Bézier Triangular Patches

\[ P_j^{k+RE_0} (QS_0, QS_1, QS_2) = u_1 P_j^{k} (QS_0, QS_1, QS_2) + v_1 P_j^{k} (QS_0, QS_1, QS_2) + w_1 P_j^{k+RE_0} (QS_0, QS_1, QS_2), \]  

and

\[ P_j^{k+RE_2} (QS_0, QS_1, QS_2) = u_2 P_j^{k+RE_2} (QS_0, QS_1, QS_2) + v_2 P_j^{k+RE_2} (QS_0, QS_1, QS_2) + w_2 P_j^{k+RE_2} (QS_0, QS_1, QS_2). \]  

By applying construction (Equation (2.30-2.31)) we get,

\[ P_j^{k_0,k_1,0} (QS_0, QS_1, QS_2) = P_j^{k_0,k_1} (QS_0, QS_1), \]  

\[ P_j^{k_0,0,k_2} (QS_0, QS_1, QS_2) = P_j^{k_0,k_2} (QS_0, QS_2), \]  

and

\[ P_j^{0,k_2,0} (QS_0, QS_1, QS_2) = P_j^{k_2,0} (QS_1, QS_2). \]  

The arrays of Equations (2.32-2.34) contain all the 4D triangular array of points defined previously in Equations (2.21-2.25). Now, we shall show that the array \( \{ P_j^{k} (QS_0, QS_1, QS_2) \} \) generates a degenerate 5D Bernstein-Bézier simplex whose triangular subsimplices subdivide the original Bernstein-Bézier surface patch at \( QS_0, QS_1 \) and \( QS_2 \). Geometrically, it is clear that there exists many options (four) to construct the point \( P_j^{k+RE_0+RE_1+RE_2} (QS_0, QS_1, QS_2) \). However, all the constructions are similar and compute the same result. The similarity is the result of a recursion that is extended from Equations (2.26-2.31) as,

\[ P_j^{k} (QS_0, QS_1, QS_2) = \sum_{l=0}^{n} \sum_{m=0}^{n} \sum_{g=0}^{d-n-m} B_{l}^{*k_0} (QS_0) B_{l}^{*k_1} (QS_1) B_{l}^{*k_2} (QS_2) P_{j+l+g+lh}. \]  

Algorithm 3: Now we present the 3-step subdivision algorithm as,

\[ B^*d [P_j^{k} (QS_0, QS_1, QS_2)](u) \]

\[ = B^*d [P_j] \left[ (u_0, u_1, u_2) + u_1 w^*_{QS_0} QS_0 + u_4 w^*_{QS_1} QS_1 + u_5 w^*_{QS_2} QS_2 \right] \]

\[ = B^*d [P_j] \left[ (u_0, u_1, u_2) + u_4 w^*_{QS_0} QS_0 + u_4 w^*_{QS_1} QS_1 + u_5 w^*_{QS_2} QS_2 \right]. \]  

Proof: This result comes from Equations (2.6-2.8 and 2.26-2.35) and the proof is analogous to the proofs of Equations (2.12 and 2.20).

As previously in Sections (3.1 and 3.2), it is clear from Equation (2.36) that the 5D rational Bernstein-Bézier simplex generated by the array \( \{ P_j^{k} (QS_0, QS_1, QS_2) \} \) collapses...
onto the rational triangular Bernstein-Bézier patch generated by the original array \( \{ P_i^j \} \).

Therefore the triangular faces of this 5D rational triangular Bernstein-Bézier simplex subdivides the original rational triangular Bernstein-Bézier patch on the surface \( S_s(u,v) \).

The selection of three arbitrary step \( QS_0, QS_1 \) and \( QS_2 \) from \( \{ P_i^j (QS_0, QS_1, QS_2) \} \) for each of the \( d \) steps of the sequence will compute three control points of the subtriangular patch. The computation for all possible permutations of this step sequence will give all the control points of the subtriangular surface patch. The triangular subsimplices of the standard 5D simplex are defined by the equations \( u_{ih} = u_{ih} = u_{ih} = 0, ih_1 \neq ih_2 \neq ih_3 \). It means that the standard 5D simplex has 20 triangular subsimplices. And, these different triangular subsimplices can be grouped into ten distinct types as,

i) \( P_j^{0,0,0} (QS_0) = \frac{w_{j,0}}{w_{QS}} P_j \). 

(2.37)

ii) \( P_j^{0,0,0} (QS_1) = \frac{w_{j,0}}{w_{QS}} P_j \).

(2.38)

iii) \( P_j^{0,0,0} (QS_2) = \frac{w_{j,0}}{w_{QS}} P_j \).

(2.39)

iv) \( P_j^{0,0,0} (QS_0, QS_1, QS_2) = P_j^{0,0} (QS_0) \) for one \( j_p = 0 \).

(2.40)

v) \( P_j^{0,0,0} (QS_0, QS_1, QS_2) = P_j^{0,0} (QS_1) \) for one \( j_p = 0 \).

(2.41)

vi) \( P_j^{0,0,0} (QS_0, QS_1, QS_2) = P_j^{0,0} (QS_2) \) for one \( j_p = 0 \).

(2.42)

vii) \( P_j^{0,0,0} (QS_0, QS_1, QS_2) = P_j^{0,0} (QS_0, QS_1) \) for one \( j_p \neq 0 \).

(2.43)

viii) \( P_j^{0,0,0} (QS_0, QS_1, QS_2) = P_j^{0,0} (QS_0, QS_2) \) for one \( j_p \neq 0 \).

(2.44)

ix) \( P_j^{0,0,0} (QS_0, QS_1, QS_2) = P_j^{0,0} (QS_1, QS_2) \) for one \( j_p \neq 0 \).

(2.45)

x) \( P_j^{0,0,0} (QS_0, QS_1, QS_2) = \sum_{i=0}^{a_n} \sum_{j=0}^{b_n} \sum_{k=0}^{c_n} B_{ij}^h (QS_0) B_{ik}^h (QS_1) B_{jh}^h (QS_2) P_{j+i+j, j+k} \) for one \( j_p \neq 0 \).

(2.46)

The arrays of Equations (2.37-2.39) generate the original rational triangular surface patch. The arrays of Equations (2.40-2.42) generate the subdivision each at \( QS_0, QS_1 \) and \( QS_2 \) like the 1-step subdivision of Equation (2.12), and generate nine triangular subpatches. The vertices of these nine triangular subpatches can be computed as were in Sections (3.1
and 3.2). The arrays of Equations (2.43-2.45) generate the subdivisions at $QS_0 - QS_1$, $QS_1 - QS_2$ and $QS_2 - QS_0$, like the 2-step subdivision of Equation (2.20), and generate nine ‘wedge shaped’ subpatches. Again, the vertices of these nine triangular subpatches can be computed as were in Section (3.2). And, the array of Equation (2.46) generates the subdivision which form the ‘triangle shaped’ (i.e. whose vertices are at $P_{0,0,0}^d (QS_0)$, $P_{0,0,0}^d (QS_1)$ and $P_{0,0,0}^d (QS_2)$) subpatch. Geometrically, our 3-step subdivision algorithm is shown in Figure 1-5.

**Remark 3:** In general the subpatches generated by Equation (2.36) overlap when $P_{0,0,0}^d (QS_0)$, $P_{0,0,0}^d (QS_1)$ and $P_{0,0,0}^d (QS_2)$ are arbitrarily positioned anywhere within the triangular patch. However, if $P_{0,0,0}^d (QS_0)$ is selected at a pre-determined ratio at the edge $P_{0,d,0}^d P_{0,0,0}^d$, $P_{0,0,0}^d (QS_1)$ is selected at a pre-determined ratio at the edge $P_{0,0,0}^d P_{d,0,0}^d$, and $P_{0,0,0}^d (QS_2)$ is selected at a pre-determined ratio at the edge $P_{0,0,0}^d P_{d,d,0}^d$ with suitable weights, then a triangular Bernstein-Bézier patch is subdivided into four subpatches by Equation (2.36). This subdivision into four subpatches is shown in Figure 1-6.

![Figure 1-5: 3-step Subdivision of the Rational Triangular Bernstein-Bézier Patch](image1)

![Figure 1-6: The Subdivision of a Rational Triangular Bernstein-Bézier Patch into Four Subpatches by 3-step Subdivision](image2)
The pre-determination of the ratios is to improve upon the aspect ratios of the original triangular patch and subtriangular patches. In this case the number of terms in the recursive definition of \( P_j^i(QS_0, QS_1, QS_2) \) is reduced from nine to six. We utilize this approach to improve upon the aspect ratio of the subtriangular patches via subdivision of the original rational triangular Bernstein-Bézier patch. This was illustratively shown in Figure 1-1.

**Remark 4:** Equations (2.12, 2.20 and 2.36) are general and they can be derived from a variation of ‘de Casteljau algorithm’ too. For a non rational triangular Bernstein-Bézier patch and in theoretical accordance with Farin (1983 and 1986) and Kahmann (1982), let \( dC_j \) be the de Casteljau step that corresponds to \( QS_j \) and transforms the net \( [P_j] \) into the subnet \( [P_j^i(QS)] \). It means that \( dC_j^m = dC_j^0, dC_j^1, \ldots, dC_j^{m-1} \) construct the point \( P(QS_j) \) from the given net \( [P_j] \). Now, \( P_j^i(QS) \) can be obtained by applying \( dC_j^0, dC_j^1, dC_j^2 \) to the given net \( [P_j] \) \( u \) times, \( v \) times and \( w \) times respectively, meaning,

\[
[P_j^i(QS)] = (u dC_j^0 + v dC_j^1 + w dC_j^2)[P_j].
\]  

In Equation (2.47) the order of \( dC_j^0, dC_j^1 \) and \( dC_j^2 \) is not important and with any order the result will be the same. This approach has been taken in the works of Kahmann (1982), Boehm and Farin (1983), Farin (1983 and 1986) & Chang and Davis (1984). The evolutions of 1-step, 2-step and 3-step subdivision algorithms are shown in Figures 1-7 to 1-9 for a non rational triangular Bernstein-Bézier patch.

### 3.4 Tolerance in a Subdivision Algorithm

It is to be noted here, that if a surface consists of poor aspect ratios of long, narrow, and thin elements/patches; even after implementing a subdivision algorithm at least one triangle of poor aspect ratio will always remain, e.g. the remaining triangle at the edges of the smallest angle in Figure 1-1. Our idea is to introduce a user specified tolerance (i.e., \( \Delta_A = \text{triangle area tolerance} \)) in the subdivision process for each patch and hence subdivision continues till the area of each triangle with poor aspect ratio is less or equal than \( \Delta_A \). Following classical differential geometry (Do Carmo (1976) and Ko et al. (2003)), let a triangular patch on the surface be as shown in Figure 1-10.

Then,

\[
S_s(u_0, v_0 + \partial v) - S_s(u_0, v_0) = \frac{\partial S_s}{\partial v} \partial v,
\]

\[
S_s(u_0 + \partial u, v_0) - S_s(u_0, v_0) = \frac{\partial S_s}{\partial u} \partial u,
\]

|\begin{align}
S_s(u_0, v_0 + \partial v) - S_s(u_0, v_0) &= \frac{\partial S_s}{\partial v} \partial v, \\
S_s(u_0 + \partial u, v_0) - S_s(u_0, v_0) &= \frac{\partial S_s}{\partial u} \partial u,
\end{align}|
Subdivision Algorithms and Convexity Analysis for Rational Bézier Triangular Patches

\[ \partial A = \frac{1}{2} \left| S_u \partial_u \times S_v \partial_v \right| = \frac{1}{2} \left| S_u \times S_v \right| \partial_u \partial_v, \quad \text{and} \]

\[ A_{\Delta T} = \int_{\Delta T} \partial A = \frac{1}{2} \int_{\Delta T} \left| S_u \times S_v \right| \partial_u \partial_v \]

(2.50)

where \( \int_{\Delta T} \) is the integration defined within the local domains of individual triangular patch (i.e., \( \Delta T_i \)), and \( S_u \) and \( S_v \) are the partial derivatives of the surface \( S(u,v) \).

4. CONVEXITY ANALYSIS OF SUBDIVISION

Continuing from Sections (2.1 and 3), let the normalized shift operators (normalized with respect to the weights \( w_i \)) \( ES, ES_2 \) and \( ES_3 \) with respect to the rational triangular patch \( T \) be defined as,

\[ ES_{\Delta T} P_{0_1,1_2} = P_{0_1+1_2,1_2}, \quad ES_2 P_{0_1,1_2} = P_{0_1,1_2+1_2}, \quad \text{and} \quad ES_3 P_{0_1,1_2} = P_{0_1,1_2+1_2}, \quad (2.52) \]

then the Bernstein-Bézier polynomial is represented symbolically (Gregory and Zhou (1991) and Ablamowicz et al. (1996)) as,

\[ B_i^d(u) = (u_0 ES_i + u_1 ES_2 + u_2 ES_3)^d P_{0,0,0}. \]

(2.53)

Now, we consider \((d+1)(d+2)/2\) points of \( T \) with barycentric co-ordinates

\[
\left( \frac{i_0}{d}, \frac{i_1}{d}, \frac{i_2}{d} \right)
\]

and which are positioned as,

\[
u_{0_1,1_2} = \left( \frac{i_0}{d} \right) V_{i_0} + \left( \frac{i_1}{d} \right) V_{i_1} + \left( \frac{i_2}{d} \right) V_{i_2}, i_0 + i_1 + i_2 = d.
\]

(2.54)

By connecting a rational triangular patch with vertices \( P_{0_1+1_2,1_2}, P_{0_1,1_2+1_2} \) and \( P_{0_1,1_2+1_2} \), we obtain a rational triangular patch denoted by \( TI_{0_1,1_2} \) for \( i_0 + i_1 + i_2 = d - 1 \). In similar fashion by connecting a rational triangular patch with vertices \( P_{0_1-1_2,1_2}, P_{0_1,1_2-1_2} \) and \( P_{0_1,1_2-1_2} \), we obtain a rational triangular patch denoted by \( TJ_{0_1,1_2} \) for \( i_0 + i_1 + i_2 = d + 1 \). A piecewise linear function \( \hat{P} \) is defined on \( T \) such that it satisfies,

\[
\hat{P} \left( \frac{i_0}{d}, \frac{i_1}{d}, \frac{i_2}{d} \right) = P_{i}, |i| = d,
\]

(2.55)

and is linear on each rational triangular patch \( TI_{0_1,1_2} \) or \( TJ_{0_1,1_2} \). It is called the Bernstein-Bézier net \( \hat{P} \) of \( P \).
Figure 1-7: The Evolution of 1-step Subdivision Algorithm

Figure 1-8: The Evolution of 2-step Subdivision Algorithm

Figure 1-9: The Evolution of 3-step Subdivision Algorithm
Subdivision Algorithms and Convexity Analysis for Rational Bézier Triangular Patches

Figure 1-10: Area of a Triangular Surface Patch

Now, with the representation as mentioned above and using Equations (2.52-2.55) we state the following result.

**Lemma 5:** If the Bernstein-Bézier net \( \hat{\mathbf{P}} \) is convex with respect to \( T \), then Bernstein-Bézier polynomial is also convex.

**Lemma 6:** The Bernstein-Bézier net \( \hat{\mathbf{P}} \) is convex with respect to \( T \) if and only if,

\[
\left( ES_{i_j} - ES_{i_k} \right) \left( ES_{i_j} - ES_{i_k} \right) P_{i_j,i_k} \geq 0, i_0 + i_1 + i_2 = d - 2, \tag{2.56}
\]

for any permutative combination of \( \{i_0, i_1, i_2\} \) consisting of \( \{1,2,3\} \).

Let \( V_{i_j}^a, V_{i_k}^a \) and \( V_{i_i}^a \) be the vertices of another rational triangular patch \( T^a \) in the same plane as \( T \) and let,

\[
u^a = u_0^a V_{i_0}^a + u_1^a V_{i_1}^a + u_2^a V_{i_2}^a, \quad u_0^a + u_1^a + u_2^a = 1, \tag{2.57}
\]

define the barycentric \( \left(u_0^a, u_1^a, u_2^a\right) \) co-ordinates of a point with respect to \( T^a \). Suppose that \( V_{i_i}^a \) has barycentric co-ordinates \( \left(u_0^a, u_1^a, u_2^a\right) \) with respect to \( T \) as,

\[
V_{i_i}^a = u_0^a V_{i_0}^a + u_1^a V_{i_1}^a + u_2^a V_{i_2}^a, \quad u_0^a + u_1^a + u_2^a = 1, \tag{2.58}
\]

for \( i = 1,2,3 \). Then we can conclude the followings,

\[
u_0 = u_0^a u_0 + u_1^a u_1 + u_2^a u_2, \tag{2.59}
\]

\[
u_1 = u_0^a u_1 + u_1^a u_1 + u_2^a u_2, \tag{2.60}
\]

\[
u_2 = u_0^a u_2 + u_1^a u_2 + u_2^a u_2. \tag{2.61}
\]

Now, a Bernstein-Bézier polynomial on \( T^a \) is defined symbolically (Gregory and Zhou (1991) and Ablamowicz et al. (1996)) as,

\[
B_{i}^{n,d}(u) = \left(u_0^a ES_1^a + u_1^a ES_2^a + u_2^a ES_3^a \right) P_{i_0,i_1,i_2}^{a}, \tag{2.62}
\]
where $ES_i$ define the normalized shift operators on $T^*$. Then, using Equations (2.52-2.55, and 2.62) we can have the following.

**Theorem 1:** Let,

$$P_{i_0, i_1, i_2}^* = \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_0} \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_1} \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_2} P_{0,0,0},$$

(2.63)

for $i_0 + i_1 + i_2 = d$. Then $B_i^* u^d (u)$ is the Bernstein-Bézier representation of $B_i^* u^d (u)$ with respect to $T^*$.

**Proof:** By equating the coefficients in Equations (2.59-2.61) and (2.63) we can obtain,

$$B_i^* u^d (u) = \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_0} \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_1} \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_2} P_{0,0,0},$$

if and only if

$$ES_i^{i_0} ES_2^{i_1} ES_3^{i_2} P_{0,0,0}^* = \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_0} \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_1} \left( u_{i_0} ES_i + u_{i_1} ES_2 + u_{i_2} ES_3 \right)^{i_2} P_{0,0,0}.$$

The set $P_{d}^*$ determines a new Bernstein-Bézier net $\hat{P}^*$ which is a piecewise linear function on $T^*$. The $T^*$ is a rational subtriangular patch of $T$ if $V_{i_0}^*, V_{i_1}^*$, and $V_{i_2}^*$, the vertices of $T^*$, are all inside or at the boundary of $T$. Then $u_{i_0}, u_{i_1}, u_{i_2} \geq 0, u_{i_0} + u_{i_1} + u_{i_2} = 1$, and $i = 1, 2, 3$.

Let $T^*$ be a non degenerate rational subtriangular patch of the original rational triangular patch $T$. The $T^*$ is parallel to $T$ if each of the edges of $T^*$ is parallel to one of those of $T$. This is geometrically shown in Figure 1-11.

The $T^*$ is convexity preserving if for all Bernstein-Bézier nets that are convex with respect to $T$, the restricted Bernstein-Bézier nets on $T^*$ are also convex with respect to $T$. By using Equation (2.58) we can conclude the following.
Lemma 7: $T^*$ is parallel to $T$.

Lemma 8: Parallellity of the edges - For a non zero scalar $sc$ and a permutative combination of $\{i_1, i_2, i_3\}$ consisting of $\{1,2,3\}$ the following holds,

$$V_{i_1}^* - V_{i_2}^* = sc\left(V_{i_1} - V_{i_2}\right), \quad V_{i_0}^* - V_{i_3}^* = sc\left(V_{i_0} - V_{i_3}\right) \quad \text{and} \quad V_{i_3}^* - V_{i_0}^* = sc\left(V_{i_3} - V_{i_0}\right).$$

(2.64)

Lemma 9: The sets $\left(u_{i_1}, u_{i_2}, u_{i_3}\right), \left(u_{i_1}, u_{i_2}, u_{i_3}\right)$ and $\left(u_{i_1}, u_{i_2}, u_{i_3}\right)$ are related. For example $u_{i_2} = u_{i_2}, u_{i_1} = u_{i_1}$ and $u_{i_3} = u_{i_3}$.

Now, we can state the following.

Theorem 2: Let $T^*$ be a non degenerate rational subtriangular patch of the original rational triangular patch $T$. Then $T^*$ is Bernstein-Bézier net convex if and only if it is parallel to $T$.

Proof: Let us assume that $T^*$ is parallel to $T$ and let $\hat{P}$ be any Bernstein-Bézier net that is convex with respect to $T$.

Then, using Equation (2.58) and Lemma (9) it can be concluded that there exist a non zero scalar $sc$ and a permutative combination of $\{i_1, i_2, i_3\}$ consisting of $\{1,2,3\}$ such that,

$$ES_{i_1}^* - ES_{i_2}^* = sc\left(ES_{i_1} - ES_{i_2}\right), \quad ES_{i_0}^* - ES_{i_3}^* = sc\left(ES_{i_0} - ES_{i_3}\right),$$

and

$$ES_{i_2}^* - ES_{i_3}^* = sc\left(ES_{i_2} - ES_{i_3}\right).$$

(2.65)

Thus for $i_0 + i_1 + i_2 = d - 2$ we have,

$$\left(ES_{i_1}^* + ES_{i_2}^*\right)P_{i_0, i_1, i_2}^* = ES_{i_0}^* ES_{i_1}^* ES_{i_2}^* \left(ES_{i_1}^* - ES_{i_2}^*\right) \left(ES_{i_2}^* - ES_{i_3}^*\right)P_{0,0,0}^d$$
Similarly, we have,
\[
\left( E_{S_i} - ES_{i}^\# \right) \left( ES_{i}^\# - ES_{i}^{\#} \right) P_{\downarrow,i,\downarrow,i}^{\#} \geq 0, \quad \text{and} \quad \left( ES_{i}^\# - ES_{i}^{\#} \right) \left( ES_{i}^\# - ES_{i}^{\#} \right) P_{\downarrow,i,\downarrow,i}^{\#} \geq 0.
\]
Hence, we conclude that \( \hat{P} \) is convex with respect to \( T^\# \).

Now, we assume that \( T^\# \) is not parallel to \( T \). Then, from Lemma (9) it means that at least one of the sets \( \{ u_{0,i}, u_{0,j}, u_{0,k} \}, \{ u_{1,i}, u_{1,j}, u_{1,k} \} \) and \( \{ u_{2,i}, u_{2,j}, u_{2,k} \} \) has all the distinct numbers. We can assume without loss of generality that \( u_{0,i} > u_{0,j} > u_{0,k} \geq 0 \) and then we have,
\[
\left( u_{0,i} - u_{0,j} \right) \left( u_{0,j} - u_{0,k} \right) < 0.
\]
Considering, the situation where,
\[
P_{\downarrow,i,\downarrow,i} = \delta_{i,d-1} + i_0 + i_1 + i_2 = d
\]
and which is \( P_{0,0,0} = 1 \) and \( P_{\downarrow,i,\downarrow,i} = 0 \) if \( i \leq d - 1 \). It is clear that Bernstein-Bézier net \( \hat{P} \) defined in Equation (2.55) is convex with respect to \( T \). Now, using Equation (2.63) we can obtain,
\[
P_{\downarrow,i,\downarrow,i}^{\#} = u_{0,i}^i u_{0,j}^i u_{0,k}^i, i_0 + i_1 + i_2 = d.
\]
Therefore, for \( i_0 + i_1 + i_2 = d - 2 \) we can have,
\[
\left( ES_{i}^\# - ES_{i}^{\#} \right) \left( ES_{i}^{\#} - ES_{i}^{\#} \right) P_{\downarrow,i,\downarrow,i}^{\#} = u_{0,i}^i u_{0,j}^j u_{0,k}^k \left( u_{0,i} - u_{0,j} \right) \left( u_{0,j} - u_{0,k} \right),
\]
and for \( d \geq 2 \) we can have,
\[
\left( ES_{i}^\# - ES_{i}^{\#} \right) \left( ES_{i}^{\#} - ES_{i}^{\#} \right) P_{0,2,0}^{\#} = u_{0,i}^{d-2} \left( u_{0,i} - u_{0,j} \right) \left( u_{0,j} - u_{0,k} \right) < 0.
\]

Hence, the restricted Bernstein-Bézier net \( \hat{P} \) is not convex with respect to \( T^\# \).

**Remark 5:** It is clear that all the subdivisions do not preserve ‘convexity’. Infact by application of Theorem (2) it can be concluded that only very few subdivisions can preserve convexity. The 3-step subdivision algorithm of Equation (2.36) which can split the rational triangular patch into four rational subtriangular patches via subdivision at each edge of the triangle can preserve ‘convexity’ under certain conditions on ratios.
Remark 6: Theorem (2) has many interesting geometric implications. It leads to some problems that still remain unsolved. For example, is it possible to divide a rational triangular patch into 2/3 rational subtriangular patches while maintaining ‘convexity’? The answer seems no but a complete proof is unavailable.

Remark 7: The process of subdivision is important not only in CAGD but also in computational geometry. A planar triangle is a convex set and any subdivision of a planar triangle lead to convex subtriangles. Still, the division of a planar triangle into four subtriangles (similar to Equation (2.36)) is challenging from a different aspect. For example let us consider two different set of points in the plane, then what is the complexity of finding the minimum number of parallel lines to separate the plane into monochromatic regions (strips)?

The problem is similar to splitting the domain into triangles and then further subdividing each triangle into four subtriangles. This 1-to-4 refinement of a planar triangle is shown in Figure 1-12.

![Figure 1-12: 1-to-4 Refinement of a Planar Triangle](image)

Computationally, the goal is to minimize the worst-case complexity of an algorithm in terms of the total number of points \(N\). A simple \(O(N^3 \log N)\)-time algorithm considers each of the \(\binom{N}{2}\) slopes of interest (parallel to the line through two input points) and covetously constructs the optimal separating set of lines with that slope in \(O(N \log N)\) by sorting the points in the perpendicular direction. This idea can be improved to an \(O(N^2 \log N)\)-time algorithm by transforming the problem into the dual according to the standard mapping \((a, b) \leftrightarrow ax + b = 1\). The statement of the dual problem is: Given a set of non-vertical two distinct set of lines in the plane, find a set of points on a vertical line that stab the wedge enclosed by every pair of two set of lines. This dual problem can be solved in \(O(N^2 \log N)\) time by constructing the arrangement of lines and running a line sweep over the arrangement.

Now, the open problems are: is there an algorithm with running time \(O(N^2 \log N)\)?, what if the minimum number \(K\) of separating lines (the size of the output) is known to be small?, and is there an \(O(NK \log N)\)-time algorithm? For \(K \leq 2\) an \(O(N \log N)\)-time algorithm is presented in Hurtado et al. (2001). However, in general settings the problem
remains open. Also, for larger $K$ the problem remains unexplored. These problems are considered some of the most important open problems (Seara (2004)) in computational geometry.

5. CONCLUSIONS

We have investigated subdivision strategies for rational triangular Bernstein-Bézier patches by constructing generators for degenerate Bernstein-Bézier simplices of $m$ dimensions (i.e. $m = 3/4/5$). In similar line we can further explore subdivisions of Bernstein-Bézier tetrahedral solids by utilizing $m - 1$ dimensional triangular subsimplices of the $m$ dimensional simplex. We can further explore the link between the subdivision strategies of the present work and ‘normalized de Casteljau’ steps (normalized with respect to weights) to derive a variant of Casteljau algorithm (de Casteljau (1963 and 1986)) for rational triangular Bernstein-Bézier patches for arbitrary subdivision. Any point on an edge of a rational triangular Bernstein-Bézier patch can be computed as an affine combination of its normalized Bernstein-Bézier coefficients. Incorporation of affine computations in the presented subdivision algorithms will allow more insight and might allow compact proofs of the subdivision schemes. Also, we can extend the algorithms presented here to develop adaptive subdivision techniques for a network of tensor product Bernstein-Bézier patches and Bézier curves. For example, a tensor product surface can be subdivided into two triangular surfaces. Adaptive subdivision of a set of tensor product surfaces is important in the compatibility analysis of surface patches.

Our presented algorithms are strongly related to geometric construction which is inherently $C^1$ continuous. Hence, we can construct a $C^1$ continuous surface over a scattered data via subdivision. This is an important problem of computational geometry and we need to explore this. Furthermore, suppose we embed a surface inside a tetrahedron or into a collection of tetrahedrons, and then we shall be able to distort it via the optimization of weights of the rational Bernstein-Bézier tetrahedrons. This process may open new choices in the interactive shape modification of surfaces. Using the presented algorithms we can explore this.

Additionally, we need to investigate the applicability of the presented subdivision algorithms in the problems of shape interrogation like surface/surface intersection, surface/plane intersection and solution of nonlinear polynomial equations, and in computer graphics and visualization like ray tracing of rational trimmed patches. Some of these applications are explored in the present work.

We have investigated the convexity analysis of subdivision of rational triangular Bernstein-Bézier patches. Now, we need to explore if any of the problems mentioned in Remarks (6-7) can be addressed?

ACKNOWLEDGEMENTS

The first author was supported by the internal research grants of AICT, SNU, Republic of
Korea during 2008-2009. Partially, this research was supported by the Defense Research and Development Organization (DRDO), MoD, GOI, India; via the grant-in-aid scheme number: ERIP/ER/0400237/M/01. The authors thank Prof. (Retd.) R. P. Gokarn for his assistance and comments.

REFERENCES


