A de Casteljau Algorithm for Bernstein type
Polynomials based on \((p, q)\)-integers in CAGD

Khalid Khan and D.K. Lobiyal
School of Computer and System Sciences, SC & SS, J.N.U., New Delhi-110067., India
khalidga1517@gmail.com; dklobiyal@gmail.com

Abstract

In this paper, a de Casteljau algorithm to compute \((p, q)\)-Bernstein Bézier curves based on \((p, q)\)-integers is introduced. We also introduce affine de Casteljau algorithm for Lupaš type \((p, q)\)-Bernstein Bézier curves. The new curves have some properties similar to \(q\)-Bézier curves. Moreover, we construct the corresponding tensor product surfaces over the rectangular domain \((u, v) \in [0,1] \times [0,1]\) depending on four parameters. We also study the de Casteljau algorithm and degree evaluation properties of the surfaces for these generalization over the rectangular domain. Furthermore, some fundamental properties for \((p, q)\)-Bernstein Bézier curves and Lupaš type \((p, q)\)-Bernstein Bézier curves are discussed.

We get \(q\)-Bézier curves and surfaces for \((u, v) \in [0,1] \times [0,1]\) when we set the parameter \(p_1=p_2=1\). In Comparison to \(q\)-Bézier curves and surfaces based on Phillips \(q\)-Bernstein polynomials, our generalizations show more flexibility in choosing the value of \(p_1, p_2\) and \(q_1, q_2\) and superiority in shape control of curves and surfaces. The shape parameters provide more convenience for the curve and surface modeling.

Keywords and phrases: de Casteljau algorithm; \((p, q)\)-integers; tensor product; \((p, q)\)-Bernstein polynomials; \(q\)-Bernstein polynomials; Lupaš type \((p, q)\)-Bernstein polynomials; Lupaš \((p, q)\)-Bézier curve; Lupaš \((p, q)\)-Bézier surface; Shape preserving; Total positivity; Degree elevation.

MSC: primary 65D17; secondary 41A10, 41A25, 41A36.

1 Introduction

In computer aided geometric design (CAGD), Bernstein polynomials and its variants are used in order to preserve the shape of the curves or surfaces. One of the most important curve in CAGD [26] is the classical Bézier curve [2] constructed with the help of Bernstein basis functions.

In 1912, S.N. Bernstein [1] introduced his famous operators \(B_n : C[0,1] \to C[0,1]\) defined for any \(n \in \mathbb{N}\) and for any function \(f \in C[0,1]\)

\[ B_n(f;x) = \sum_{k=0}^{n} \binom{n}{k} x^k (1-x)^{n-k} f\left(\frac{k}{n}\right), \quad x \in [0,1]. \]  

(1.1)

and named it Bernstein polynomials to prove the Weierstrass theorem [6]. Later it was found that Bernstein polynomials possess many remarkable properties and has various applications in areas such as approximation theory [6], numerical analysis, computer-aided geometric design, and solutions of differential equations due to its fine properties of approximation [20].
In recent years, generalization of the Bézier curve with shape parameters has received continuous attention. Several authors were concerned with the problem of changing the shape of curves and surfaces, while keeping the control polygon unchanged and thus they generalized the Bézier curves in [17, 20, 21].

The rapid development of \( q \)-calculus [25] has led to the discovery of new generalizations of Bernstein polynomials involving \( q \)-integers [7, 14, 18, 20]. The aim of these generalizations is to provide appropriate and powerful tools to application areas such as numerical analysis, computer-aided geometric design, and solutions of differential equations.

In 1987, Lupas [7] introduced the first \( q \)-analogue of Bernstein operator and investigated its approximating and shape-preserving properties.

\[
B_{n,q}(f;x) = \sum_{k=0}^{n} \binom{n}{k}_q x^k \prod_{s=0}^{n-k-1} (1 - q^s x) f\left(\frac{[k]_q}{[n]_q}\right), \quad x \in [0,1] \tag{1.2}
\]

where \( B_{n,q} : C[0,1] \to C[0,1] \) defined for any \( n \in \mathbb{N} \) and any function \( f \in C[0,1] \).

In 1996, Phillips [16] proposed another \( q \)-variant of the classical Bernstein operator, the so-called Phillips \( q \)-Bernstein operator which attracted lots of investigations. The \( q \)-variants of Bernstein polynomials provide one shape parameter for constructing free-form curves and surfaces, Phillips \( q \)-Bernstein operator was applied well in this area. G.M. Phillips in [15] also presented a De Casteljau algorithm for generalized Bernstein Polynomials.

In 2003, Oruk and Phillips [20] used the basis functions of Phillips \( q \)-Bernstein operator for construction of \( q \)-Bézier curves, which they call Phillips \( q \)-Bézier curves, and studied the properties of degree reduction and elevation.

Cetin et al in [3] proposed tensor Product \( q \)-Bernstein Polynomials. Li-Wen Hana et al in [17] introduced generalized Bézier curves and surfaces based on Lupas \( q \)-analogue of Bernstein operator. They also constructed the corresponding tensor product surfaces over the rectangular domain, and study the properties of the surfaces, as well as the degree evaluation and de Casteljau algorithms for Lupas \( q \)-analogue of Bernstein operators.

Recently, Mursaleen et al [8] applied \((p,q)\)-calculus in approximation theory and introduced the first \((p,q)\)-analogue of Bernstein operators based on \((p,q)\)-integers. For similar works based on \((p,q)\)-integers, one can refer [9, 10, 11, 13].

Thus with the development of \((p,q)\)-analogue of Bernstein operators and its variants, one natural question arises, how it can be used in computer aided geometric design in order to preserve the shape of the curves or surfaces. In this way, it opens a new research direction which requires further investigations.

Before proceeding further, let us recall certain notations of \((p,q)\)-calculus.

The \((p,q)\) integers \([n]_{p,q}\) are defined by

\[
[n]_{p,q} = \frac{p^n - q^n}{p - q}, \quad n = 0, 1, 2, \cdots, \quad p > q > 0
\]

whereas \(q\)-integers are given by

\[
[n]_q = \begin{cases} 
\frac{1-q^n}{1-q}, & q \neq 1 \\
\frac{1}{n}, & q = 1
\end{cases}
\]
The formula for \((p, q)\)-binomial expansion is as follow:

\[
(ax + by)^n_{p,q} := \sum_{k=0}^{n} p^{-\frac{(n-k)(n-k-1)}{2}} q^{\frac{k(k-1)}{2}} \binom{n}{k}_{p,q} a^{n-k} b^k x^{n-k} y^k,
\]

\[
(x + y)^n_{p,q} = (x + y)(px + qy)(p^2 x + q^2 y) \cdots (p^{n-1} x + q^{n-1} y),
\]

\[
(1 - x)^n_{p,q} = (1 - x)(p - qx)(p^2 - q^2 x) \cdots (p^{n-1} - q^{n-1} x),
\]

where \((p, q)\)-binomial coefficients are defined by

\[
\binom{n}{k}_{p,q} = \frac{[n]_{p,q}!}{[k]_{p,q}![n-k]_{p,q}!},
\]

Details on \((p, q)\)-calculus can be found in [4, 8, 23].

The \((p, q)\)-Bernstein Operators introduced by Mursaleen et al is as follow:

\[
B_n_{p,q}(f; x) = \sum_{k=0}^{n} \binom{n}{k}_{p,q} x^k \prod_{s=0}^{n-k-1} (p^s - q^s x) \left( \frac{[k]_{p,q}}{[n]_{p,q}} \right), \quad x \in [0, 1] \quad (1.3)
\]

Note when \(p = 1\), \((p, q)\)-Bernstein Operators given by (1.3) turns out to be \(q\)-Bernstein Operators. Also, we have

\[
(1 - x)^n_{p,q} = \prod_{s=0}^{n-1} (p^s - q^s x) = (1 - x)(p - qx)(p^2 - q^2 x) \cdots (p^{n-1} - q^{n-1} x)
\]

\[
= \sum_{k=0}^{n} (-1)^k p^{-\frac{(n-k)(n-k-1)}{2}} q^{\frac{k(k+1)}{2}} \binom{n}{k}_{p,q} x^k
\]

Again by some simple calculations and using the property of \((p, q)\)-integers, we get \((p, q)\)-analogue of Pascal’s relation as follow:

\[
\binom{n}{k}_{p,q} = q^{n-k} \binom{n-1}{k-1}_{p,q} + p^k \binom{n-1}{k}_{p,q} \quad (1.4)
\]

\[
\binom{n}{k}_{p,q} = p^{n-k} \binom{n-1}{k-1}_{p,q} + q^k \binom{n-1}{k}_{p,q} \quad (1.5)
\]

Very recently, we have applied \((p, q)\)-calculus in computer aided geometric design and introduced first the \((p, q)\)-Bézier curves and surfaces based on \((p, q)\)-integers in [5] which is further generalization of \(q\)-Bézier curves and surfaces, for example, [17, 18, 19, 20]. We have also introduced a new analogue, i.e, Lupas type \((p, q)\)-analogue of the Bernstein operators in [5].

Now in this paper, we further extend our results on a rectangular domain and introduce \((p, q)\)-analogue of de Casteljau algorithm for the operators discussed in [5].

The outline of this paper is as follow: Section 2 introduces a de Casteljau type algorithm for \(B_{k,n}^{p,q}\). In Section 3 we define a tensor product patch based on algorithm 1 and its geometric properties as well as a degree elevation technique are investigated. Furthermore tensor product of \((p, q)\)-Bézier
surfaces on $[0,1] \times [0,1]$ for $(p,q)$-Bernstein polynomials are introduced and its properties that is inherited from the univariate case are being discussed.

Similarly section 4 again introduces a de Casteljau type algorithm which is in affine form for $k^{p,q}_{k,n}$. In Section 5 we define a tensor product patch and its geometric properties as well as a degree elevation technique is introduced. Finally, we define tensor product Lupas type $(p,q)$-Bézier surfaces on $[0,1] \times [0,1]$ for Lupas type $(p,q)$-Bernstein polynomials, and discuss its properties that are derived from the univariate case.

In next section, we present de Casteljau [3, 15] type algorithm for $(p,q)$-Bernstein polynomials.

\section{$(p,q)$-Bernstein polynomials}

The $(p,q)$-Bernstein functions is as follow:

$$B^{k,n}_{p,q}(t) = \binom{n}{k}_{p,q} t^k (1-t)^{n-k}, \quad t \in [0,1] \quad (2.1)$$

\begin{align*}
(1-t)^{n-k} &= \prod_{s=0}^{n-k} (p^s - q^s t)
\end{align*}

Let us recall some recurrence relation introduced by us in [5] for $(p,q)$-Bernstein polynomials, i.e

\begin{theorem}[5]
Each $(p,q)$-Bernstein functions of degree $n$ is a linear combination of two $(p,q)$-Bernstein functions of degree $n - 1$:

$$B^{k,n}_{p,q}(t) = q^{n-k} t B^{k-1,n-1}_{p,q}(t) + (p^{n-1} - p^k q^{n-k-1}) B^{k,n-1}_{p,q}(t) \quad (2.2)$$

\end{theorem}

\begin{theorem}[5]
Each $(p,q)$-Bernstein function of degree $n$ is a linear combination of two $(p,q)$-Bernstein functions of degree $n + 1$.

$$B^{k,n}_{p,q}(t) = \left( \frac{p^k [n+1-k]_{p,q}}{[n+1]_{p,q}} \right) B^{k,n+1}_{p,q}(t) + \left( 1 - \frac{p^{k+1} [n-k]_{p,q}}{[n+1]_{p,q}} \right) B^{k+1,n+1}_{p,q}(t) \quad (2.3)$$

The de Casteljau algorithm describes how to subdivide a Bézier curve, when a Bézier curve is repeatedly subdivided, the collection of control polygons converge to the curve. Thus, the way of computing a Bézier curve is to simply subdivide it an appropriate number of times and compute the control polygons.

\subsection{de Casteljau algorithm for $(p,q)$-Bernstein polynomials:}

We now give a de Casteljau type algorithm for computing the $(p,q)$-Bernstein Bézier curves. This algorithm is based on the first recurrence relation given by Theorem (2.1) for $B^{k,n}_{p,q}(t)$, (or see [5]) and it is this algorithm that will enables us to construct tensor product $(p,q)$-Bernstein Bézier surfaces.

\begin{algorithm}
Let $P_0, P_1, ..., P_n \in \mathbb{R}^2$ or $\mathbb{R}^3$ be the given control points. Compute
\[ P_r(t, p, q) = q^n - t P_{i+1}(t, p, q) + (p^n - p^i q^{n-i-1} t) P_i(t, p, q), \]

\[
\begin{cases}
  r = 1, \ldots, n, \\
  i = 0, 1, 2, \ldots, n - r,
\end{cases}
\]

(2.4)

where

\[ P_0^0(t, p, q) = P(t) \]

The algorithm yields

\[ P_n^0(t, p, q) = \sum_{k=0}^n P_i B_{i,n}^{p,q}(t) \]

3. **Tensor product \((p, q)\)-Bézier surfaces on \([0, 1] \times [0, 1]\)**

We define a two-parameter family of tensor product surface of degree \((m, n)\) as follow:

\[
S(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{i,j} B_{i,m}^{p_1,q_1}(u) B_{j,n}^{p_2,q_2}(v), \quad (u, v) \in [0,1] \times [0,1]
\]

(3.1)

where \(P_{i,j} \in \mathbb{R}^3\) are control net points and \(B_{i,m}^{p,q}(u), B_{j,n}^{p,q}(v)\)

are \((p, q)\)-Bernstein basis polynomials with respect to the parameters \(p_1, p_2\) and \(q_1, q_2\) respectively for \(i = 0, 1, \ldots, m\) and \(j = 0, 1, \ldots, n\). The parameters \(p_1, p_2\) and \(q_1, q_2\) add extra flexibility to the basis functions which play important role in the convergence of \((p, q)\)-Bernstein polynomials. Furthermore they vary the shape of the surface.

3.1 **Properties.**

1. **Affine invariance property:** Since for some \((u, v) \in [0, 1] \times [0, 1]\)

\[
S(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_{i,m}^{p_1,q_1}(u) B_{j,n}^{p_2,q_2}(v) \neq 1,
\]

(3.2)

but for

\[
S(1, 1) = \sum_{i=0}^m \sum_{j=0}^n B_{i,m}^{p_1,q_1}(u) B_{j,n}^{p_2,q_2}(v) = 1,
\]

(3.3)

\(S(u, v)\) is not an affine combination of its control net points. Thus \(S(u, v)\) is not affinely invariant but if we put \(p_1 = p_2 = 1\) then it will become affine invariant.

2. **Convex hull property:** When \(0 < q_1 < p_1 \leq 1, 0 < q_2 < p_2 \leq 1\) the bases polynomials are nonnegative but do not forms a partition of unity at each point. Hence \(S(u, v)\) is not a convex combination of \(P_{i,j}\) and may or may not lies in the convex hull of its control net points.

3. **Boundary curves:** Boundary curves of \(S(u, v)\) are evaluated by \(S(u, 0), S(u, 1), S(0, v)\) and \(S(1, v)\). The first two curves are \((p, q)\)-Bernstein Bézier curves in \(u\) and the last two curves are \((p, q)\)-Bernstein Bézier curves in \(v\).

4. **Corner point interpolation:** The control points of the boundary curves are the boundary points of control net of \(S(u, v)\). Thus it follows from one sided endpoint interpolation property of the
Thus, the problem is reduced to expressing \( m \) where

\[
P_{e}
\]

For this purpose we first write tensor product Bézier curves that the corner control net points may not coincide with some of the four corners of the surface. Namely, \( S(0,0) \neq P_{0,0}, S(0,1) \neq P_{0,n}, S(1,0) \neq P_{m,0}, S(1,1) = P_{m,n} \).

Consequently \( S(u,v) \) mimics the shape of control net. Moreover putting \( p_1 = p_2 = 1 \) reduces 3.1 to a tensor product \( q \)-Bernstein Bézier patch. What follows is the de Casteljau type algorithm to compute \( S(u,v) \).

Algorithm 2. Given the control net \( P_{i,j} \in \mathbb{R}^3; i = 0, 1, ..., m, \ j = 0, 1, ..., n \). Compute

\[
P_{i,j}^{r} = \sum_{i=0}^{r} \sum_{j=0}^{r} P_{i,j} B_{i,j}^m(u) B_{i,j}^n(v)
\]

For this purpose we first write tensor product Bézier patches in the form

\[
S(u,v) = \sum_{j=0}^{n} P_{j} B_{j,n}^2(v)
\]

where

\[
P_{j} = \sum_{i=0}^{m} P_{i,j} B_{i,j}^m(u)
\]

Thus, the problem is reduced to expressing \( m \) th degree \((p,q)\)-Bernstein Bézier curve \( P_{j} \) by one of \((m+1)\) th degree. From the degree elevation procedure (see, 3.3) for \( P_{j} \) in the latter equation, we obtain for \( P_{j} \) in the latter equation, we obtain

\[
P_{j} = \sum_{i=0}^{m} P_{i,j} B_{i,j}^m(u) = \sum_{i=0}^{m+1} P_{i,j}^{(1.0)} B_{i,j}^{m+1}(u),
\]

where

\[
P_{i,j}^{(1.0)} = \left( 1 - \frac{p_1^{-1} [m+1-i]_{p_1,q_1}}{[m+1]_{p_1,q_1}} \right) P_{i-1,j} + \left( \frac{p_1^{i} [m+1-i]_{p_1,q_1}}{[m+1]_{p_1,q_1}} \right) P_{i,j}, \ \ i = 0, 1, ..., m + 1.
\]

and \([n]_{p_1,q_1}\) denotes the \((p,q)\)-integer \( [n] \) with parameter value \( p_1, q_1 \) in the place of \( p, q \). Similarly, to obtain the same surface as one of degree \((m,n+1)\) we need new control points \( P_{i,j}^{(0.1)} \) such that

\[
P_{i,j}^{(0.1)} = \left( 1 - \frac{p_2^{-1} [n+1-j]_{p_2,q_2}}{[n+1]_{p_2,q_2}} \right) P_{i-1,j} + \left( \frac{p_2^{j} [n+1-j]_{p_2,q_2}}{[n+1]_{p_2,q_2}} \right) P_{i,j}, \ \ i = 0, 1, ..., m + 1.
\]
Finally, to obtain $S(u, v)$ as a surface of degree $(m + 1, n + 1)$ evaluate the new control net points from the product

$$P_{i,j}^{1,1} = \left[ 1 - \frac{p_i^{-1} [m+1-i]_{p_1,q_1}}{(m+1)_{p_1,q_1}} \right] \left[ \frac{p_i^{-1} [m+1-i]_{p_1,q_1}}{(m+1)_{p_1,q_1}} \right] \left[ \begin{array}{c} P_{i,j}^{r-1,r-1} \\ P_{i,j+1}^{r-1,r-1} \\ P_{i+1,j}^{r-1,r-1} \end{array} \right]$$

The repeated degree elevation procedure can be computed by following the univariate case described in [5].

**Note:** We get $q$-Bézier curves and surfaces for $(u, v) \in [0, 1] \times [0, 1]$ when we set the parameter $p_1 = p_2 = 1$ as given in [3].

In the next section, we study about Lupaş type $(p, q)$-analogue of the Bernstein polynomials.

## 4 Lupaş type $(p, q)$-analogue of the Bernstein polynomials

The new analogue, that is Lupaş $(p, q)$-analogue of the Bernstein polynomials introduced by us in [3] is as follows:

$$b_{p,q}^{k,n}(t) = \begin{bmatrix} n \\ k \end{bmatrix} p_{p,q}^{(n-k)(q-k-1)} q_{p,q}^{k(k+1)} t^k (1-t)^{n-k} \prod_{j=1}^{n} \{p^{j-1}(1-t) + q^{j-1}t\}, (4.1)$$

where $b_{p,q}^{0,n}(t), b_{p,q}^{1,n}(t), \ldots, b_{p,q}^{n,n}(t)$ are the Lupaş $(p, q)$-analogue of the Bernstein functions of degree $n$ on the interval $[0, 1]$.

When $p = 1$, Lupaş $(p, q)$-Bernstein functions turns out to be Lupaş $q$-Bernstein functions as given in [17].

For more details on Lupaş $q$-analogue of Bernstein operators, one can refer [22] [24]

**Theorem 4.1** [3] Each Lupaş $(p, q)$-analogue of the Bernstein function of degree $n$ is a linear combination of two Lupaş $(p, q)$-analogues of the Bernstein functions of degree $n-1$.

$$b_{p,q}^{k,n}(t) = \frac{q^{n-1}t}{p^{n-1}(1-t) + q^{n-1}t} b_{p,q}^{k-1,n-1}(t) + \frac{p^{n-1}(1-t)}{p^{n-1}(1-t) + q^{n-1}t} b_{p,q}^{k,n-1}(t) \quad (4.2)$$

$$b_{p,q}^{k,n}(t) = \frac{p^{n-1}q^{i-1}t}{p^{n-1}(1-t) + q^{n-1}t} b_{p,q}^{k-1,n-1}(t) + \frac{p^{n-i-1}q^{i}(1-t)}{p^{n-1}(1-t) + q^{n-1}t} b_{p,q}^{k,n-1}(t) \quad (4.3)$$

### 4.1 Lupaş $(p, q)$-Bézier curves:

Let us define the Lupaş $(p, q)$-Bézier curves of degree $n$ using the Lupaş $(p, q)$-analogues of the Bernstein functions as follows:

$$P(t; p, q) = \sum_{i=0}^{n} P_i b_{p,q}^{k,n}(t) \quad (4.4)$$
where \( P_i \in \mathbb{R}^3 \) (\( i = 0, 1, \ldots, n \)) and \( p > q > 0 \). \( P_i \) are control points. Joining up adjacent points \( P_i, i = 0, 1, 2, \ldots, n \) to obtain a polygon which is called the control polygon of Lupas \((p, q)\)-Bézier curves.

### 4.2 Degree elevation for Lupas \((p, q)\)-Bézier curves

Lupas \((p, q)\)-Bézier curves have a degree elevation algorithm that is similar to that possessed by the classical Bézier curves. Using the technique of degree elevation, we can increase the flexibility of a given curve.

\[
P(t; p, q) = \sum_{k=0}^{n} P_k b_{p,q}^k(t) \tag{4.5}
\]

\[
P(t; p, q) = \sum_{k=0}^{n+1} P_k^* b_{p,q}^{k,n+1}(t),
\]

where

\[
P^* = \left(1 - \frac{p^k [n + 1 - k]}{[n + 1]_{p,q}}\right) P_{k-1} + \left(\frac{p^k [n + 1 - k]}{[n + 1]_{p,q}}\right) P_k \tag{4.6}
\]

The statement above can be derived from Theorem (4.1). When \( p = 1 \), formula (4.5) reduce to the degree evaluation formula of the \( q \)-Bézier curves. If we let \( P = (P_0, P_1, \ldots, P_n)^T \) denote the vector of control points of the initial Lupas \((p, q)\)-Bézier curve of degree \( n \), and \( P^{(1)} = (P^*_0, P^*_1, \ldots, P^*_{n+1}) \) the vector of control points of the degree elevated Lupas \((p, q)\)-Bézier curve of degree \( n + 1 \), then we can represent the degree elevation procedure as:

\[
P^{(1)} = T_{n+1} P,
\]

where

\[
T_{n+1} = \frac{1}{[n + 1]_{p,q}} \begin{bmatrix}
\frac{p[n + 1]_{p,q}}{[n + 1]_{p,q}} & 0 & \cdots & 0 & 0 \\
\frac{p[n + 1]_{p,q} - p[n]_{p,q}}{[n + 1]_{p,q}} & \cdots & 0 & 0 \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
0 & \cdots & \frac{p[n + 1]_{p,q} - p^{n-1}[2]_{p,q}}{[n + 1]_{p,q}} & \frac{p^{n-1}[2]_{p,q}}{[n + 1]_{p,q}} & 0 \\
0 & 0 & \cdots & \frac{p[n + 1]_{p,q} - p^n[1]_{p,q}}{[n + 1]_{p,q}} & \frac{p^n[1]_{p,q}}{[n + 1]_{p,q}}
\end{bmatrix}_{(n+2) \times (n+1)}
\]

For any \( l \in \mathbb{N} \), the vector of control points of the degree elevated Lupas \((p, q)\)-Bézier curves of degree \( n + l \) is: \( P^{(l)} = T_{n+l} T_{n+2} \cdots T_{n+1} P \). As \( l \to \infty \), the control polygon \( P^{(l)} \) converges to a Lupas \((p, q)\)-Bézier curve.

### 4.3 de Casteljau algorithm:

Lupas \((p, q)\)-Bézier curves of degree \( n \) can be written as two kinds of linear combination of two Lupas \((p, q)\)-Bézier curves of degree \( n - 1 \), and we can get the two selectable algorithms to evaluate Lupas \((p, q)\)-Bézier curves. The algorithms can be expressed as:

**Algorithm 3.**
Pr where 
\[
\begin{align*}
  & \text{P}^0_r(t; p, q) = \text{P}^0_r \equiv \text{P}_i = 0, 1, 2, \ldots, n \\
  & \text{P}^r_i(t; p, q) = \frac{p^{n-r} - q^{n-r}}{p^{n-r}(1-t) + q^{n-r}} \text{P}^r_{i+1}(t; p, q) + \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} \text{P}^r_{i-1}(t; p, q) \\
  \end{align*}
\]

(4.7)
or
\[
\begin{align*}
  & \text{P}^0_r(t; p, q) = \text{P}^0_r \equiv \text{P}_i = 0, 1, 2, \ldots, n \\
  & \text{P}^r_i(t; p, q) = \frac{p^{n-r} - q^{n-r}}{p^{n-r}(1-t) + q^{n-r}} \text{P}^r_{i+1}(t; p, q) + \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} \text{P}^r_{i-1}(t; p, q) \\
  \end{align*}
\]

(4.8)

Then
\[
\text{P}(t; p, q) = \sum_{i=0}^{n-1} \text{P}^i_1(t; p, q) = \ldots = \sum \text{P}^i_1(t; p, q) b^n_{p, q}^{n-r}(t) = \ldots = \text{P}^0_0(t; p, q)
\]

(4.9)

It is clear that the results can be obtained from Theorem (4.1). When \( p = 1 \), formula (4.7) and (4.8) recover the de Casteljau algorithms of classical \( q \)-Bézier curves. Let \( P^0 = (P_0, P_1, \ldots, P_n)^T \), \( P^r = (P_0^r, P_1^r, \ldots, P_n^r)^T \), then de Casteljau algorithm can be expressed as:

Algorithm 4.
\[
\text{P}^r(t; p, q) = M_r(t; p, q) \ldots M_2(t; p, q) M_1(t; p, q) P^0
\]

(4.10)

where \( M_r(t; p, q) \) is a \( (n-r+1) \times (n-r+2) \) matrix and

\[
M_r(t; p, q) = \begin{bmatrix}
  \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} & \frac{p^{n-r}}{p^{n-r}(1-t) + q^{n-r}} & \cdots & 0 & 0 \\
  \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} & \frac{p^{n-r}}{p^{n-r}(1-t) + q^{n-r}} & \cdots & 0 & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  0 & \cdots & \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} & \frac{p^{n-r}}{p^{n-r}(1-t) + q^{n-r}} & 0 \\
  0 & 0 & \cdots & \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} & \frac{p^{n-r}}{p^{n-r}(1-t) + q^{n-r}} \\
\end{bmatrix}
\]

or

\[
M_r(t; p, q) = \begin{bmatrix}
  \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} & \frac{p^{n-r}}{p^{n-r}(1-t) + q^{n-r}} & \cdots & 0 & 0 \\
  \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} & \frac{p^{n-r}}{p^{n-r}(1-t) + q^{n-r}} & \cdots & 0 & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  0 & \cdots & \frac{p^{n-r}}{p^{n-r}(1-t) + q^{n-r}} & \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} & 0 \\
  0 & 0 & \cdots & \frac{p^{n-r}}{p^{n-r}(1-t) + q^{n-r}} & \frac{p^{n-r}(1-t)}{p^{n-r}(1-t) + q^{n-r}} \\
\end{bmatrix}
\]

5 Tensor product Lupaş \((p, q)\)-Bézier surfaces on \([0, 1] \times [0, 1]\)

We define a two-parameter family \( P(u, v) \) of tensor product surfaces of degree \( m \times n \) as follow:
Let \( m \) take obtaining the same surface as a surface of degree \(( p, q)\)-Bézier parameter \( p_1, q_1 > 0, p_2 > q_2 > 0, \)

and \( b_{p_1, q_1}^i(u), b_{p_2, q_2}^j(v) \) are Lupa\( (p, q)\)-analogue of Bernstein functions respectively with the parameter \( p_1, q_1 \) and \( p_2, q_2 \). We call the parameter surface tensor product Lupa\( (p, q)\)-Bézier surface with degree \( m \times n \). We refer to the \( P_{i,j} \) as the control points. By joining up adjacent points in the same row or column to obtain a net which is called the control net of tensor product Lupa\( (p, q)\)-Bézier surface.

5.1 Properties

1. Geometric invariance and affine invariance property: Since

\[
P(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} P_{i,j} b_{p_1, q_1}^i(u) b_{p_2, q_2}^j(v), \quad (u, v) \in [0, 1] \times [0, 1],
\]

where \( P_{i,j} \in \mathbb{R}^3 \) \((i = 0, 1, ..., m, j = 0, 1, ..., n)\) and two real numbers \( p_1 > q_1 > 0, p_2 > q_2 > 0, \)

\[
b_{p_1, q_1}^i(u), b_{p_2, q_2}^j(v)
\]

is a tensor product Lupa\( (p, q)\)-Bézier surface. Namely, \( P(u, v) \) is an affine combination of its control points.

2. Convex hull property: \( P(u, v) \) is a convex combination of \( P_{i,j} \) and lies in the convex hull of its control net.

3. Isoparametric curves property: The isoparametric curves \( v = v^* \) and \( u = u^* \) of a tensor product Lupa\( (p, q)\)-Bézier surface are respectively the Lupa\( (p, q)\)-Bézier curves of degree \( m \) and degree \( n \), namely,

\[
P(u, v^*) = \sum_{i=0}^{m} \left( \sum_{j=0}^{n} P_{i,j} b_{p_2, q_2}^j(v^*) \right) b_{p_1, q_1}^i(u), \quad u \in [0, 1];
\]

\[
P(u^*, v) = \sum_{j=0}^{n} \left( \sum_{i=0}^{m} P_{i,j} b_{p_1, q_1}^i(u^*) \right) b_{p_2, q_2}^j(v), \quad v \in [0, 1]
\]

The boundary curves of \( P(u, v) \) are evaluated by \( P(u, 0), P(u, 1), P(0, v) \) and \( P(1, v) \).

4. Corner point interpolation property: The corner control net coincide with the four corners of the surface. Namely, \( P(0, 0) = P_{0,0}, P(0, 1) = P_{0,n}, P(1, 0) = P_{m,0}, P(1, 1) = P_{m,n} \).

5. Reducibility: When \( p_1 = p_2 = 1 \), formula (5.1) reduces to a tensor product \( q \)-Bézier patch.

5.2 Degree elevation and de Casteljau algorithm

Let \( P(u, v) \) be a tensor product Lupa\( (p, q)\)-Bézier surface of degree \( m \times n \). As an example, let us take obtaining the same surface as a surface of degree \((m+1) \times (n+1)\). Hence we need to find new control points \( P_{i,j} \) such that

\[
P(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} P_{i,j} b_{p_1, q_1}^i(u) b_{p_2, q_2}^j(v) = \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} P_{i,j} b_{p_1, q_1}^i(u) b_{p_2, q_2}^j(v)
\]

\[
(5.3)
\]

Let \( \alpha_i = 1 - \frac{p_i^{-1} [m+1-i]_{p_1, q_1}}{[m+1]_{p_1, q_1}}, \quad \beta_j = 1 - \frac{p_j^{-1} [n+1-j]_{p_2, q_2}}{[n+1]_{p_2, q_2}}.
\]
Then
\[
P_{i,j}^{1,1} = \alpha_i \beta_j P_{i-1,j-1} + \alpha_i (1 - \beta_j) P_{i-1,j} + (1 - \alpha_i) (1 - \beta_j) P_{i,j}
\]
which can be written in matrix form as
\[
\begin{pmatrix}
1 - \frac{p_{i-1}^{m-1} [m+1-i]_{p_1,q_1}}{[m+1]_{p_1,q_1}} & \frac{p_{i-1}^{m-1} [m+1-i]_{p_1,q_1}}{[m+1]_{p_1,q_1}}
\end{pmatrix}
\begin{pmatrix}
P_{i-1,j-1} \\
P_{i,j}
\end{pmatrix}
\begin{pmatrix}
P_{i-1,j} \\
P_{i,j}
\end{pmatrix}
\]
which can be written as
\[
P_{i,j} = \alpha_i \beta_j P_{i-1,j-1} + \alpha_i (1 - \beta_j) P_{i-1,j} + (1 - \alpha_i) (1 - \beta_j) P_{i,j}
\]

The de Casteljau algorithms are also easily extended to evaluate points on a Lupas \((p, q)\)-Bézier surface. Given the control net \(P_{i,j} \in \mathbb{R}^3; i = 0, 1, ..., m; j = 0, 1, ..., n\).

\[
\begin{align*}
P_{i,j}^{0,0}(u,v) & \equiv P_{i,j} \quad i = 0, 1, 2, ..., m; \quad j = 0, 1, 2, ..., n. \\
P_{i,j}^{r}(u,v) & = \begin{pmatrix}
\frac{p_1^{m-r} (1-u)}{p_1^{m-r} (1-u) + q_1^{m-r}} & \frac{q_1^{m-r}}{p_1^{m-r} (1-u) + q_1^{m-r}}
\end{pmatrix}
\begin{pmatrix}
P_{i,j}^{r-1,i-1,j-1} \\
P_{i,j}^{r-1,i+1,j-1}
\end{pmatrix}
\begin{pmatrix}
P_{i,j}^{r-1,i-1,j} \\
P_{i,j}^{r-1,i+1,j}
\end{pmatrix}
\end{align*}
\]

or

\[
\begin{align*}
P_{i,j}^{0,0}(u,v) & \equiv P_{i,j}^{0,0} \equiv P_{i,j} \quad i = 0, 1, 2, ..., m; \quad j = 0, 1, 2, ..., n. \\
P_{i,j}^{r}(u,v) & = \begin{pmatrix}
\frac{p_1^{m-r} q_1^{r-1} (1-u)}{p_1^{m-r} (1-u) + q_1^{m-r}} & \frac{q_1^{m-r}}{p_1^{m-r} (1-u) + q_1^{m-r}}
\end{pmatrix}
\begin{pmatrix}
P_{i,j}^{r-1,i-1,j-1} \\
P_{i,j}^{r-1,i+1,j-1}
\end{pmatrix}
\begin{pmatrix}
P_{i,j}^{r-1,i-1,j} \\
P_{i,j}^{r-1,i+1,j}
\end{pmatrix}
\end{align*}
\]

When \(m = n\), one can directly use the algorithms above to get a point on the surface. When \(m = n\), to get a point on the surface after \(k\) applications of formula \((5.5)\) or \((5.6)\), we perform formula \((4.10)\) for the intermediate point \(P_{i,j}^{k,k}\).

**Note:** We get Lupas \(q\)-Bézier curves and surfaces for \((u,v) \in [0,1] \times [0,1]\) when we set the parameter \(p_1 = p_2 = 1\) as proved in [17].

**Remark:** One can give \((p, q)\)-analogue of Bezier curves and surfaces based on \((p, q)\)-analogue of divided difference analogous to \(q\)-analogue of Bezier curves and surfaces [3, 15] and study de Casteljau algorithm and degree evaluation properties. But for such construction, we require \((p, q)\)-analogue of divided difference which is not yet developed. So the study of \((p, q)\)-analogue of Bezier curves and surfaces based on \((p, q)\)-analogue of divided difference can be suggested as a future problem.

### 6 Future work

Currently, we are constructing generalizations of classical rational Bézier curves and surfaces based on Lupas type \((p, q)\)-analogue of Bernstein operators. In the near future, we also hope to construct a new generalization of B-spline based on these operators and develop blossoming and subdivision procedures for the series of Lupas curves and surfaces.
References

[1] S. N. Bernstein, Constructive proof of Weierstrass approximation theorem, *Comm. Kharkov Math. Soc.* (1912)

[2] P.E. Bézier, Numerical Control-Mathematics and applications, *John Wiley and Sons, London*, 1972.

[3] Cetin Disibuyuk and Halil Oruc, Tensor Product $q$-Bernstein Polynomials, *BIT Numerical Mathematics, Springer* 48 (2008) 689-700.

[4] R. Jagannathan, K. Srinivasa Rao, Two-parameter quantum algebras, twin-basic numbers, and associated generalized hypergeometric series, *Proceedings of the International Conference on Number Theory and Mathematical Physics*, 20-21 December 2005.

[5] Khalid Khan and D.K. Lobiyal, Bézier curves based on Lupas $(p, q)$-analogue of Bernstein polynomials in CAGD, (submitted), *Computer aided Geometric Design*, elsevier.

[6] P. P. Korovkin, Linear operators and approximation theory, *Hindustan Publishing Corporation, Delhi*, 1960.

[7] A. Lupas, A $q$-analogue of the Bernstein operator, *Seminar on Numerical and Statistical Calculus, University of Cluj-Napoca*, 9(1987) 85–92.

[8] M. Mursaleen, K. J. Ansari, A. Khan, On $(p, q)$-analogue of Bernstein Operators, *Applied Mathematics and Computation*, 266 (2015) 874-882.

[9] M. Mursaleen and Md. Nasiruzzaman, On $(p, q)$-analogue of Bernstein-Schurer operators, *arXiv:1504.05876v1 [math.CA]*.

[10] M. Mursaleen, K. J. Ansari and Asif Khan., Some Approximation Results by $(p, q)$-analogue of Bernstein-Stancu Operators, *Applied Mathematics and Computation*, 264,(2015), 392-402.

[11] M. Mursaleen, K. J. Ansari and Asif Khan, Approximation by a $(p, q)$-analogue of Bernstein-Kantorovich Operators, *arXiv: [1504.05887][math.CA]*.

[12] M. Mursaleen, Faisal Khan and Asif Khan, Approximation by $(p, q)$-Lorentz polynomials on a compact disk, *arXiv:1504.05093v1 [math.CA]*.

[13] M. Mursaleen, Md. Nasiruzzaman, Asif Khan and Khursheed J. Ansari, Some approximation results on Bleimann-Butzer-Hahn operators defined by $(p, q)$-integers, *arXiv: 1505.00392v1 [math.CA]*.

[14] M. Mursaleen, Asif Khan, Generalized q-Bernstein-Schurer Operators and Some Approximation Theorems, *Journal of Function Spaces and Applications* Volume 2013, Article ID 719834, 7 pages [http://dx.doi.org/10.1155/2013/719834](http://dx.doi.org/10.1155/2013/719834)

[15] G.M. Phillips, A De Casteljau Algorithm For Generalized Bernstein Polynomials, *BIT* 36 (1) (1996), 232-236.

[16] G.M. Phillips, Bernstein polynomials based on the $q$-integers, *The heritage of P.L.Chebyshev, Ann. Numer. Math.*, 4 (1997) 511–518.

[17] Li-Wen Hana, Ying Chua, Zhi-Yu Qiu, Generalized Bézier curves and surfaces based on Lupas $q$-analogue of Bernstein operator, *Journal of Computational and Applied Mathematics* 261 (2014) 352-363.

[18] George M. Phillips, Interpolation and Approximation by Polynomials, *Springer*

[19] G. M. Phillips, A generalization of the Bernstein polynomials based on the $q$-integers *ANZIAMJ* 42(2000), 79-86.
[20] Halil Oruk, George M. Phillips, $q$-Bernstein polynomials and Bézier curves, *Journal of Computational and Applied Mathematics* 151 (2003) 1-12.

[21] Abedallah Rababah, Stephen Manna, Iterative process for G2-multi degree reduction of Bzier curves, *Applied Mathematics and Computation* 217 (2011) 81268133.

[22] N. I. Mahmudov and P. Sabancgil, Some approximation properties of Lupas $q$-analogue of Bernstein operators, *arXiv:1012.4245v1 [math.FA]* 20 Dec 2010.

[23] Mahouton Norbert Hounkonnou, Joseph Désiré Bukweli Kymba, $R(p, q)$-calculus: differentiation and integration, *SUT Journal of Mathematics*, Vol. 49, No. 2 (2013), 145-167.

[24] Sofiya Ostrovska, On the Lupas $q$-analogue of the bernstein operator, *Rocky mountain journal of mathematics* Volume 36, Number 5, 2006.

[25] K. Victor, C. Pokman, Quantum Calculus, *Springer-Verlag* (2002), New York Berlin Heidelberg.

[26] Thomas W. Sederberg, Computer Aided Geometric Design Course Notes, Department of Computer Science *Brigham Young University*, October 9, 2014.