Turk. J. Math. Comput. Sci.
x(x)(20yy) 1–19

DOI : 10.47000/tjmcs.XX

Design of Bézier Curves of Some Surfaces with Matlab Applications

CEYDA YILMAZ LUZUM[1], ŞENAY BAYDAŞ[1,*], BÜLENT KARAKAŞ[2]

[1]*Department of Mathematics, Faculty of Science, Van Yuzuncu Yil University, 65080, Van, Turkiye.*
[2]*Department of Mathematics, Faculty of Science, Bartin University, 74100, Bartın, Turkiye.*

Received: xx-xx-20xx • Accepted: xx-xx-20xx

Abstract. Bézier curves are special types of curves defined by control points. The large number of control points naturally affects the form of the Bézier curve. When the control points of a Bézier curve are the points of a surface in $\mathbb{R}^3$, the Bézier curve will be obtained depending on the surface as well as the control points. As the number of points on the surface increases, the obtained Bézier curve will approach the limit of positioning on the surface. The study examines this approach. In addition, the theory is exemplified using the Matlab program.

2020 AMS Classification: 65D17, 14H50, 51N05, 68U07

Keywords: Bernstein polynomials, Bézier curves, De Casteljau algorithms, Matlab.

1. Introduction

The behavior of the curve or surface around a point is considered, by mentioning the local features. The surface triangulation network, which Gauss established between 1821 and 1825 for the measurement of the kingdom of Hanover with observations of angle and length, is an important step by which differential geometry for curves and surfaces moved from theory to practice. Operating with polynomials provides great convenience in the studies carried out [7]. The history of Bernstein polynomials dates back to the Ukrainian mathematician Sergei Natanovic Bernstein (1880-1968). Bernstein's work attracted great attention in France, thanks to which he was elected as a member of the French Academy of Sciences in 1955 [14]. The definition and some properties of a Bézier curve given by the parametric equation $P(t)$, $0 \leq t \leq 1$ in terms of Bernstein polynomials were examined, and the De Casteljau algorithm was discussed.

Bézier curves and surfaces are of great importance for computer aided geometric design (CAGD). Bézier curves and surfaces were first developed independently by different mathematical approaches by French engineer Pierre Etienne Bézier (1910-1999) and French mathematician Paul de Faget De Casteljau (1930-2022) between 1958 and 1960, respectively. Pierre Etienne Bézier was a mechanical and industrial engineer; after graduation, he started to work in the mechanical method planning department of Renault in 1933. In 1960, he was transferred to the vehicle body method planning department and worked on the design of vehicle bodies ([12], [13]). The Bézier curves and surfaces theory, which developed completely with the expression of polynomial curves and surfaces in Bernstein form, was established during the period of long career of Bézier in the company and called by the name of Bézier. It was developed to define automobile body surfaces with curves that can be controlled by changing a few parameters [3].

*Corresponding Author
Email addresses: ceydayilmazz92@gmail.com (C.Y. Luzum), sbaydas@yyu.edu.tr (Ş. Baydaş), bk@bartin.edu.tr (B. Karakaş)
The author is supported by ...

De Casteljau, who joined the Citroen company in 1958, utilized the same concept with an entirely different approach, and introduced the De Casteljau algorithm, which was named after him. This algorithm, which contributed greatly to the development of the concept of Bézier curves and surfaces, was equivalent to the use of Bernstein polynomials. De Casteljau's work was kept secret by Citroen for a very long time. However, De Casteljau used these polynomials long before Bézier realized that Bézier curves and surfaces could be expressed with Bernstein polynomials ([1], [6], [9], [12], [13]).

2. Preliminaries

2.1. **Bernstein polynomials.** Binomial expansion of $(x+y)^n$ for $n \in \mathbb{N}$ and $x, y \in \mathbb{R}$ given by;

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{(n-k)} \tag{2.1}$$

This expansion has been known since Omar Khayyam, and the coefficients $\binom{n}{k}$ can be given by an algorithm known as Pascal's triangle. Bernstein gave this expansion a statistical identity if x and y are probability variables. If the probabilities of an event are x and y, then we have the following form ;

$$x + y = 1 \tag{2.2}$$

So the sum of $x + y$ can be written as the sum of $x + (1 - x)$. In this case, the probability expansion for the event is given by

$$1 = (x + (1-x))^n \tag{2.3}$$

$$= \sum_{k=0}^{n} \binom{n}{k} x^k (1-x)^{(n-k)} \tag{2.4}$$

The (2.4) expansion is called the Bernstein polynomial. Here, each

$$\binom{n}{k} x^k (1-x)^{(n-k)} \tag{2.5}$$

term accompanying the polynomial is called the components (coefficients) of the Bernstein polynomial expansion. Bernstein polynomials occupy an important position in the ring of polynomials, and being base property is one of its most important properties. They are also important in the construction of Bézier curves.

2.2. **Basic properties of Bernstein polynomials.**

Definition 2.2.1. for k= 0,1,2,. . . ,n, an n-degree Bernstein polynomial is defined by the coefficients;

$$B_{k,n}(x) = \binom{n}{k} x^k (1-x)^{n-k} \tag{2.6}$$

Where we have the following equality

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} \tag{2.7}$$

It is pretty easy to write these polynomials. Coefficients $\binom{n}{k}$ can be easily obtained from Pascal's triangle; As k increases, the exponent of the x term increases by one, and the exponent of the $(1 - x)$ term decreases by one. The zeroth, first, second, and third-order Bernstein polynomials can be calculated as follows:
The Bernstein polynomial of degree zero is defined as

$$B_{0,0}(x) = 1 \tag{2.8}$$

and the graph for $0 \le x \le 1$ can be drawn as follows (Figure 2.1) [7].

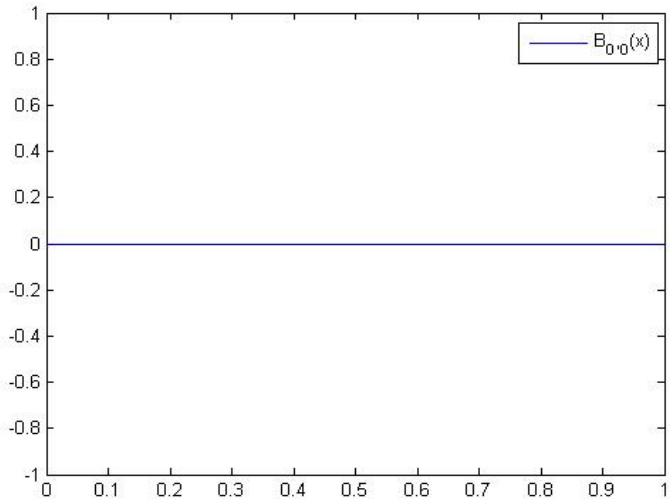

Figure 2.1. Graph of a zero-order Bernstein polynomial.

Bernstein polynomials of degree one are obtained as

$$B_{0,1}(x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} x^0(1-x)^{1-0} = (1-x) \tag{2.9}$$

$$B_{1,1}(x) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} x^1(1-x)^{1-1} = x \tag{2.10}$$

And the graph for $0 \leq x \leq 1$ can be drawn as follows (Figure 2.2)

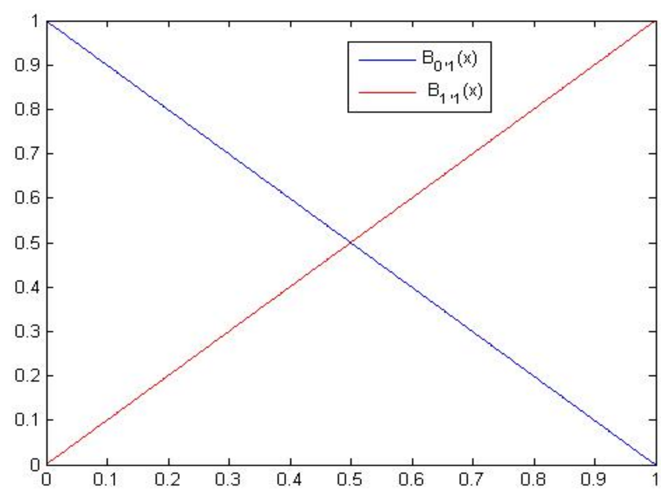

Figure 2.2. Graph of first-order Bernstein polynomials.

Bernstein polynomials of degree two are obtained as

$$B_{0,2}(x) = \begin{pmatrix} 2 \\ 0 \end{pmatrix} x^0(1-x)^{2-0} = (1-x)^2 \tag{2.11}$$

$$B_{1,2}(x) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} x^1(1-x)^{2-1} = 2x(1-x) \tag{2.12}$$

$$B_{2,2}(x) = \begin{pmatrix} 2 \\ 2 \end{pmatrix} x^2(1-x)^{2-2} = x^2 \tag{2.13}$$

and the graph for $0 \leq x \leq 1$ can be drawn as follows (Figure 2.3).
The Bernstein polynomial of degree three is defined as

$$B_{0,3}(x) = \begin{pmatrix} 3 \\ 0 \end{pmatrix} x^0(1-x)^{3-0} = (1-x)^3 \tag{2.14}$$

$$B_{1,3}(x) = \begin{pmatrix} 3 \\ 1 \end{pmatrix} x^1(1-x)^{3-1} = 3x(1-x)^2 \tag{2.15}$$

$$B_{2,3}(x) = \begin{pmatrix} 3 \\ 2 \end{pmatrix} x^2(1-x)^{3-2} = 3x^2(1-x) \tag{2.16}$$

$$B_{3,3}(x) = \begin{pmatrix} 3 \\ 3 \end{pmatrix} x^3(1-x)^{3-3} = x^3 \tag{2.17}$$

and the graph for $0 \leq x \leq 1$ can be drawn as follows (Figure 2.4).

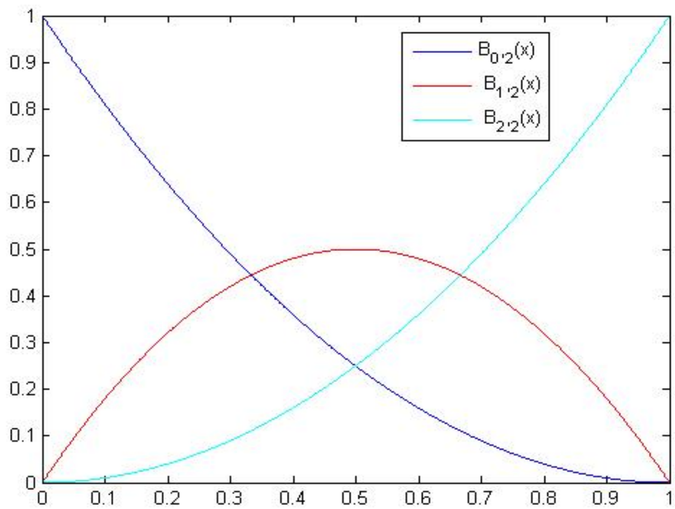

Figure 2.3. Graphs of quadratic Bernstein polynomials.

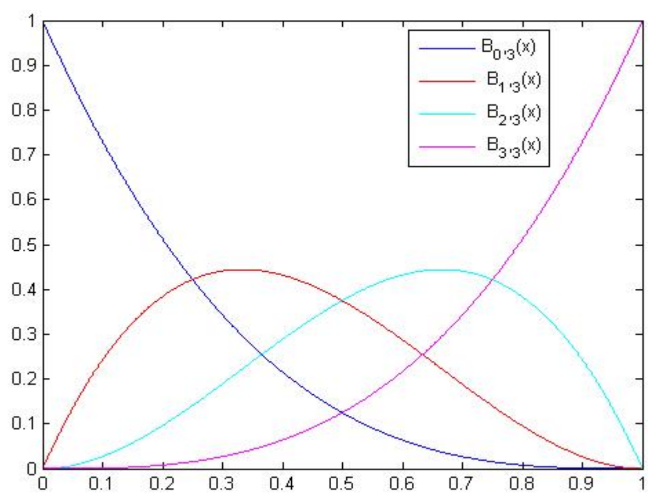

Figure 2.4. Graph of third-order Bernstein polynomials.

2.3. **Matrix representation for Bernstein polynomials.** Matrix representation for Bernstein polynomials is useful in practice. If viewed in terms of linear combinations of dot products, they can be developed directly. Let

$$B(x) = c_0 B_{0,n}(x) + c_1 B_{1,n}(x) + \cdots + c_n B_{n,n}(x) \tag{2.18}$$

be a polynomial given as a linear combination of Bernstein basis functions. It is easy to write this equation as the dot product of two vectors as follow;

$$B(x) = \left[B_{0,n}(x)\ B_{1,n}(x) \ldots\ B_{n,n}(x) \right] \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} \tag{2.19}$$

When we arrange equation (2.19) it turns into following form

$$B(x) = \left\lfloor 1\ x\ x^2 \ldots\ x^n \right\rfloor \begin{bmatrix} b_{0,0} & 0 & 0 & \cdots & 0 \\ b_{1,0} & b_{1,1} & 0 & \cdots & 0 \\ b_{2,0} & b_{2,1} & b_{2,2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ b_{n,0} & b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \tag{2.20}$$

Here $b_{i,j}$ are the coefficients of the power base used to determine the related Bernstein polynomial. Matrix representation in quadratic case (for $n = 2$)

$$B(x) = \left[1\ x\ x^2 \right] \begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \tag{2.21}$$

and matrix representation in cubic case (for $n = 3$)

$$B(x) = \left[1\ x\ x^2\ x^3 \right] \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$ [7].

3. Bézier Curves And De Casteljau Algorithm

In this section, Bézier curves and Bernstein polynomials were used in obtaining these curves and some properties of them were given.

The Bézier curve is defined by control points and a basis function to construct them. The first and last control point selected creates the beginning and end of the curve. Other points in between were used to determine the structure of the curve. In this context, these points are usually not located on the curve [2].

3.1. Creating the Bézier curve segment.

Definition 3.1.1. When given the points P_0 and P_1 in E^n $(n = 2, 3)$ Euclidean space, any point on the line segment $\overline{P_0P_1}$ is represented by $P_0^1(t)$;

$$P(t) = P_0^1(t) = (1-t)P_0 + tP_1 \quad ; t \in [0, 1]. \tag{3.1}$$

This equation forms the parameterization of a linear curve segment. Here;

$$B_0^1(t) = (1-t) \tag{3.2}$$

$$B_1^1(t) = t \tag{3.3}$$

called first-order Bernstein polynomials and control points P_0 and P_1

(Fig. 3.1) ([8], [11]).

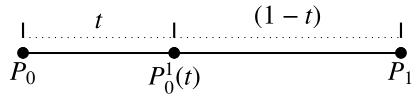

Figure 3.1. Representation of $P_0^1(t)$ with respect to P_0 and P_1.

3.2. **De Casteljau algorithm.** Although this algorithm is one of the most basic in curve space and surface design, it is surprisingly simple. The main interesting point is a good interplay between geometry and algebra. An intuitive geometric structure leads to a strong theory.

Definition 3.2.1. $b_0,\ b_1,\ldots,b_n \in \mathbb{E}^3$ and $t \in \mathbb{R}$ for

$$\boldsymbol{b}_i^r(t) = (1-t)\,\boldsymbol{b}_i^{r-1}(t) + t\boldsymbol{b}_{i+1}^{r-1}(t) \begin{cases} r = 1,\ldots,n \\ i = 0,\ldots,n-r \end{cases} \tag{3.4}$$

and $\boldsymbol{b}_i^0(t) = \boldsymbol{b}_i$. Then $\boldsymbol{b}_0^n(t)$, $\boldsymbol{b}^n$ is on the Bézier curve, t is the parameter-valued point. So $\boldsymbol{b}^n(t) = \boldsymbol{b}_0^n(t)$. The polygon P formed by $\boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is called the Bézier polygon or control polygon of the $\boldsymbol{b}^n$ (in the cubic case, there are four control points). Similarly, the $\boldsymbol{b}_i$ polygon vertices are called control points or Bézier points. The figure below shows the cubic case (Fig. 3.2).

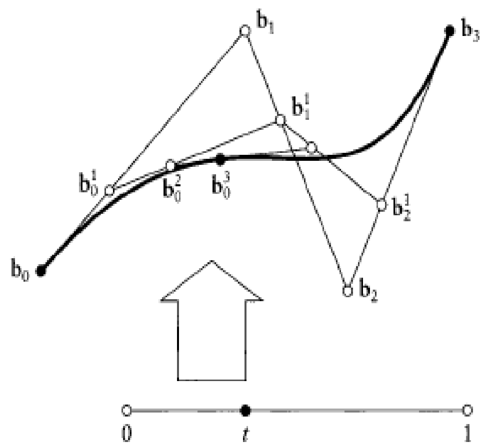

Figure 3.2. De Casteljau algorithm: $\boldsymbol{b}_0^3(t)$ point iterative linear interpolation.

Sometimes with $b^n(t) = B[b_0,\ b_1,\ldots,b_n;t] = B[P;t]$ or equivalently shorter $b^n = B[b_0,\ b_1,\ldots,b_n] = BP$ We can also show it with. This notation defines B as the linear operator that associates its control polygon with the Bézier curve. It can be said that the curve$B[b_0,\ b_1,\ldots,b_n]$ is the Bernstein-Bézier approximation to the control point. $\boldsymbol{b}_i^r(t)$ intermediate coefficients were written in accordance with a triangular array of points [6].

3.3. **De Casteljau algorithm for the Bézier curve.** The Bézier curve is created with the help of control points as follows (Figure 3.3):

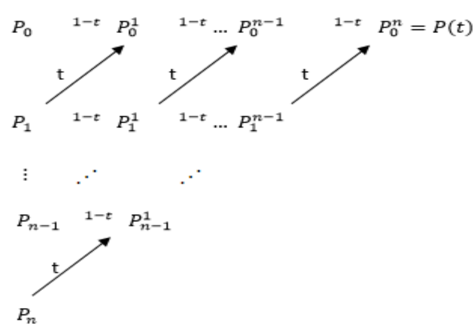

Figure 3.3. De Casteljau algorithm for the Bézier curve.

This expression provides the De Casteljau algorithm for the Bézier curve. The De Casteljau algorithm provides a method for calculating a point on a Bézier curve. The following theorem expresses the De Casteljau algorithm [4].

Theorem 3.3.1. *Let $P_0, P_1, \dots, P_n$ be a Bézier curve P(t) formed by the control points. In this case*

$$P_i^j(t) = (1-t)\,P_i^{j-1}(t) + tP_{i+1}^{j-1}(t) \quad ; i = 0,\dots,n-j, \quad j = 1,\dots,n \tag{3.5}$$

where $P_i^0 = P_i$ and $P(t) = P_0^n$ $(0 \le t \le 1)$([5], [11]).

Let's state this theorem as follows; The De Casteljau algorithm emerges from the recursion property of the Bernstein polynomial. Similarly;

$$\begin{aligned} P(t) &= \sum_{i=0}^{n} P_i B_i^n(t) = \sum_{i=0}^{n} P_i((1-t)\,B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)) \\ &= \sum_{i=0}^{n} (1-t)P_i B_i^{n-1}(t) + \sum_{i=0}^{n} tP_i B_{i-1}^{n-1}(t) \end{aligned}$$

obtained. Since $B_n^{n-1}(t) = 0$ and $B_{-1}^{n-1}(t) = 0$ we get

$$P(t) = \sum_{i=0}^{n-1} (1-t)\,P_i B_i^{n-1}(t) + \sum_{i=1}^{n} tP_i B_{i-1}^{n-1}(t) \tag{3.6}$$

In the second sum of (3.6), if the expression is rewritten by replacing i with $i+i$;

$$P(t) = \sum_{i=0}^{n-1} (1-t)P_i B_i^{n-1}(t) + \sum_{i=0}^{n-1} tP_{i+1} B_i^{n-1}(t)$$

obtained. for $i = 0,\dots,n-1$ if $P_i^1 = (1-t)\,P_i + tP_{i+1} = (1-t)\,P_i^0 + tP_{i+1}^0$ is taken;

$$P(t) = \sum_{i=0}^{n-1} P_i^1 B_i^{n-1}(t)$$

obtained. Here $P(t)$ represents an $(n-1)$ degree Bézier curve with control points $P_0^1, \dots, P_{n-1}^1$. If we proceed in a similar way we get;

$$P(t) = \sum_{i=0}^{n-2} P_i^2 B_i^{n-2}(t)$$

Here for $i = 0,\dots,n-2$ $P_i^2 = (1-t)\,P_i^1 + iP_{i+1}^1$. Similarly,

$$P(t) = \sum_{i=0}^{n-j} P_i^j B_i^{n-j}(t)$$

obtained. Here; for $i = 0,\dots,n-j$ $P_i^j = (1-t)\,P_i^{j-1} + iP_{i+1}^{j-1}$.
Specifically, the final equation for, $n = j$:

$$P(t) = \sum_{i=0}^{0} P_i^0 B_i^0(t) = P_0^0$$

obtained.

Definition 3.3.2. for $t \in [0,1]$ given the points P_0, P_1 and P_2 in $E^n (n = 2, 3)$ Euclidean space,

$$\begin{aligned} P_0^1(t) &= (1-t)\,P_0 + tP_1 \\ P_1^1(t) &= (1-t)\,P_1 + tP_2 \\ P_0^2(t) &= (1-t)\,P_0^1(t) + tP_1^1(t) \end{aligned} \tag{3.7}$$

with the help of the equations,

$$\begin{aligned} P(t) &= P_0^2(t) \\ &= (1-t)^2 P_0 + 2\,(1-t)\,tP_1 + t^2 P_2 \end{aligned}$$

$$= \sum_{i=0}^{2} P_i B_i^2(t) \tag{3.8}$$

quadratic Bézier curve is defined. Here;

$$\begin{aligned} B_0^2(t) &= (1-t)^2 \\ B_1^2(t) &= 2(1-t)t \\ B_2^2(t) &= t^2 \end{aligned} \tag{3.9}$$

with the quadratic Bernstein polynomials P_0, P_1 and P_2 are called control points (see Fig. 3.5)

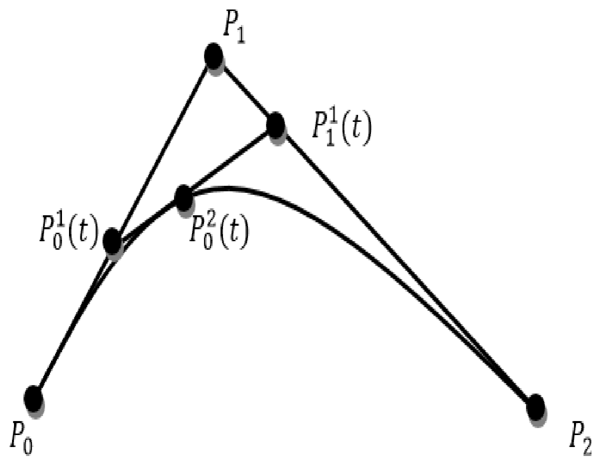

Figure 3.4. Quadratic Bézier curve formed by control points P_0, P_1 and P_2

The triangle obtained by combining the control points P_0, P_1 ve P_2 with the line segments in the prescribed order is called the control polygon. ([8], [11]).

Definition 3.3.3. Given the points P_0, P_1, P_2 and P_3 in $E^n (n = 2, 3)$ Euclidean space for $t \in [0, 1]$;

$$\begin{aligned} P_0^1(t) &= (1-t)P_0 + tP_1 \\ P_1^1(t) &= (1-t)P_1 + tP_2 \\ P_2^1(t) &= (1-t)P_2 + tP_3 \\ P_0^2(t) &= (1-t)P_0^1(t) + tP_1^1(t) \\ P_1^2(t) &= (1-t)P_1^1(t) + tP_2^1(t) \\ P_0^3(t) &= (1-t)P_0^2(t) + tP_1^2(t) \end{aligned} \tag{3.10}$$

with the help of equations given below;

$$\begin{aligned} P(t) &= P_0^3(t) \\ &= (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3 \\ &= \sum_{i=0}^{3} P_i B_i^3(t) \end{aligned} \tag{3.11}$$

cubic Bézier curve is defined. Here;

$$\begin{aligned} B_0^3(t) &= (1-t)^3 \\ B_1^3(t) &= 3(1-t)^2 t \\ B_2^3(t) &= 3(1-t)t^2 \\ B_3^3(t) &= t^3 \end{aligned} \tag{3.12}$$

third-order Bernstein polynomials and P_0, P_1, P_2 and P_3 are also called control points (see Fig. 3.6).

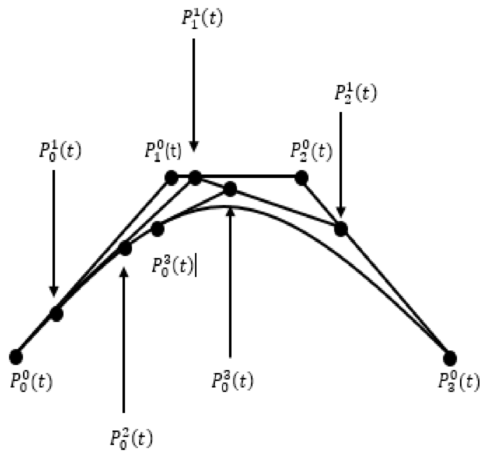

Figure 3.5. Cubic Bézier curve formed by control points P_0, P_1, P_2 and P_3.

The polygon obtained by combining the control points P_0, P_1, P_2 and P_3 with the line segments in the prescribed order is called the control polygon. ([8], [11]).

Definition 3.3.4. For each $t \in [0, 1]$ when given the control points

$$P_0, P_1, P_2, \ldots, P_n \text{ in Euclidean space } E^n (n = 2, 3)$$

$$P(t) = \sum_{i=0}^{n} P_i B_i^n(t) \tag{3.13}$$

the parametric equation is called the Bézier curve formed by these control points. Here;

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i \qquad ; 0 \leq i \leq n \tag{3.14}$$

are called n -degree Bernstein polynomials. The polygon obtained with the help of line segments of control points $P_0, P_1, P_2, \ldots, P_n$ formed in the prescribed order is called the control polygon [11].

In Figure 3.7 also 2nd, 3rd, 4th and 5th degree Bernstein polynomials are given, respectively.

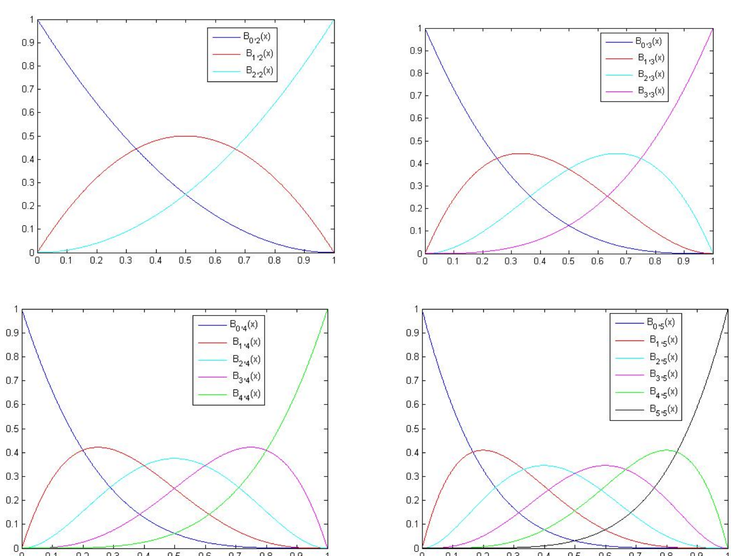

Figure 3.6. 2nd, 3rd, 4th and 5th order Bernstein polynomials.

4. Bézier Curves of Surface Curves in $\mathbb{R}^3$ And Matlab Applications

4.1. **Bézier curves of surface curves in $\mathbb{R}^3$.** Bézier curves of the control points are calculated by taking a curve on a surface, and then by taking a certain number of control points on the curve.

Let a surface in $\mathbb{R}^3$ be given by the parameterization $\varphi(u,v)$. With the constant selections of $u = u_0$ and $v = v_0$, the parameter curves of the surface are obtained. In addition, any curves on the surface are obtained with the relation $f(u,v) = 0$ (Figure 4.1).

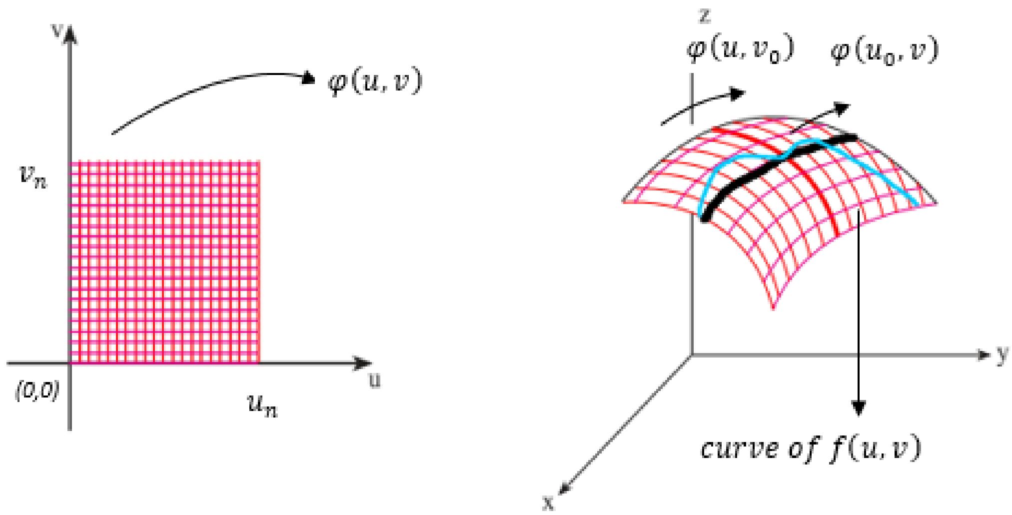

Figure 4.1. Parametric curves on the surface $\varphi(u,v)$.

Let a curve $f(u,v) = 0$ be chosen on the surface $\varphi(u,v)$. Consider the curves $u = f(v)$ from $f(u,v) = 0$. The points P_0 and P_n are obtained for $u = 0$ and $u = u_n$. The points $P_0,\ P_1,\ldots,\ P_n$ can be calculated with a partition of u; $\{u_0,\ u_1,\ldots,\ u_n\}$. Let. $B_z(u)$ or $P(u)$ is the Bézier curve that accepts the points $P_0,\ P_1,\ldots,\ P_n$ as control points. The Bézier curve with $(n+1)$ control points corresponding to the curve $f(u,v) = 0$ are coincident curves for $n \to \infty$.

This method is valid for any space curves, and there are many reasons for getting a curve on the surface, among which one is that the surface acts as a panel in CAGD. Another reason is to create a preparation to obtain a Bézier surface corresponding to a surface with the algorithm here.

Examples of Bézier curves belonging to curves on a right circular cylinder and sphere surface are provided, then algorithms and Matlab applications are demonstrated.

Example 4.1.1. Let's draw the Bézier curve on the helix curve that provides the relation $v = u$ on the vertical circular cylinder surface $\varphi(u,v) = (\cos u\ , sinu, v)$;

$$\varphi(u,v) = (\cos u\ , sinu, v)$$

$$v = u$$

$$\varphi(u) = (\cos u\ , sinu, u)$$

Here, $\varphi(u)$ are $u-$parameter curves. These curves are the helix curves on the cylinder surface $\varphi(u,v)$. A parameter curve on the vertical circular cylinder surface and five control points on this curve are taken. Then, the Bézier curve of these control points is computed. The Matlab algorithm of the Bézier curves of five or more control points is given and their graphs are drawn;

* for $u = 0$

$$\varphi(0) = (\cos(0)\ , \sin(0)\ , 0) \longrightarrow P_0$$

* for $u = \frac{\pi}{2}$

$$\varphi\left(\frac{\pi}{2}\right) = \left(\cos\left(\frac{\pi}{2}\right), \sin\left(\frac{\pi}{2}\right), \frac{\pi}{2}\right) \longrightarrow P_1$$

* for $u = \pi$

$$\varphi(\pi) = (\cos(\pi\ )\ , \sin(\pi\ )\ , \pi) \longrightarrow P_2$$

* for $u = \frac{3\pi}{2}$

$$\varphi\left(\frac{3\pi}{2}\right) = \left(\cos\left(\frac{3\pi}{2}\right), \sin\left(\frac{3\pi}{2}\right), \frac{3\pi}{2}\right) \longrightarrow P_3$$

*and for $u = 2\pi$

$$\varphi(2\pi) = (\cos(2\pi), \sin(2\pi), 2\pi) \longrightarrow P_4$$

is obtained.

For $k = 0, 1, 2, ..., n$, $n+1$ Bernstein polynomials of order n are

$$B_{k,n}(t) = \binom{n}{k} t^k (1-t)^{n-k}$$

and Bernstein polynomials of degree four are

$$B_{0,4}(t) = \binom{4}{0} t^0 (1-t)^{4-0} = (1-t)^4$$

$$B_{1,4}(t) = \binom{4}{1} t (1-t)^{4-1} = 4t(1-t)^3$$

$$B_{2,4}(t) = \binom{4}{2} t^2 (1-t)^{4-2} = 6t^2(1-t)^2$$

$$B_{3,4}(t) = \binom{4}{3} t^3 (1-t)^{4-3} = 4t^3(1-t)$$

$$B_{4,4}(t) = \binom{4}{4} t^4 (1-t)^{4-4} = t^4.$$

According to these polynomials while the t parameter changes between 0 and 1 a curve was drawn between the first point (P_0) and the last point (P_4).

The (P_1, P_2, P_3) points would not lie on this curve unless the five points are on a straight line. In fact, the coordinates of the points on a 4th-order Bézier curve are a weighted average of the coordinates of the five control points used to describe that curve. As we increase the number of control points, we see that the Bézier curve gets closer and closer to this curve.

$$P(t) = \sum_{k=0}^{n} P_k \binom{n}{k} t^k (1-t)^{n-k}$$

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

$$P(t) = \sum_{k=0}^{4} P_k \binom{4}{k} t^k (1-t)^{4-k} = P_0 \binom{4}{0} t^0 (1-t)^4 + P_1 \binom{4}{1} t(1-t)^3$$
$$+P_2 \binom{4}{2} t^2 (1-t)^2 + P_3 \binom{4}{3} t^3 (1-t) + P_4 \binom{4}{4} t^4 (1-t)^0$$

$$P(t) = (1-t)^4 P_0 + 4t(1-t)^3 P_1 + 6t^2(1-t)^2 P_2 + 4t^3(1-t) P_3 + t^4 P_4$$

$$\begin{aligned} P(t) = (1-t)^4 (\cos(0), \sin(0), 0) &+ 4t(1-t)^3 \left(\cos\left(\frac{\pi}{2}\right), \sin\left(\frac{\pi}{2}\right), \frac{\pi}{2}\right) \\ &+6t^2(1-t)^2 (\cos(\pi), \sin(\pi), \pi) \\ &+4t^3(1-t)\left(\cos\left(\frac{3\pi}{2}\right), \sin\left(\frac{3\pi}{2}\right), \frac{3\pi}{2}\right) \\ &+t^4 (\cos(2\pi), \sin(2\pi), 2\pi) \end{aligned}$$

Here helix curve was generated on the cylinder surface and the Bézier curve was computed with the five control points on this helix curve.

Example 4.1.2. Let's find constant curves $v=\frac{\pi}{4}$ on the unit sphere centered at origin $\varphi(u,v)=(\cos u\ sinv, sinu\ sinv, cosv)$.

$$\varphi(u)=\left(\cos u\ sin\frac{\pi}{4}, sinu\ sin\frac{\pi}{4}, cos\frac{\pi}{4}\right)$$
$$=\left(\frac{\sqrt{2}}{2}\cos u\ ,\ \frac{\sqrt{2}}{2}\sin u,\ \frac{\sqrt{2}}{2}\right)$$

Here, $\varphi(u)$ are $u-$parameter curves. These curves are the circle curves on the sphere surface. By taking the circle curve on the sphere surface and four control points on this curve, the Bézier curve of these control points is computed. The Matlab algorithm of the Bézier curve of the ten control points is given and its graph is drawn. That is;
for $u=0$,

$$\varphi(0)=\left(\frac{\sqrt{2}}{2}\cos 0\ ,\ \frac{\sqrt{2}}{2}\sin 0,\ \frac{\sqrt{2}}{2}\right)\longrightarrow P_0$$

for $u=\frac{\pi}{2}$,

$$\varphi\left(\frac{\pi}{2}\right)=\left(\frac{\sqrt{2}}{2}\cos\frac{\pi}{2}\ ,\ \frac{\sqrt{2}}{2}\sin\frac{\pi}{2},\ \frac{\sqrt{2}}{2}\right)\longrightarrow P_1$$

for $u=\pi$,

$$\varphi(\pi)=\left(\frac{\sqrt{2}}{2}\cos\pi\ ,\ \frac{\sqrt{2}}{2}\sin\pi,\ \frac{\sqrt{2}}{2}\right)\longrightarrow P_2$$

and for $u=\frac{3\pi}{2}$,

$$\varphi\left(\frac{3\pi}{2}\right)=\left(\frac{\sqrt{2}}{2}\cos\frac{3\pi}{2}\ ,\ \frac{\sqrt{2}}{2}\sin\frac{3\pi}{2},\ \frac{\sqrt{2}}{2}\right)\longrightarrow P_3$$

is obtained.

$$P(t)=\sum_{k=0}^{n}P_k\binom{n}{k}t^k(1-t)^{n-k}$$

$$P(t)=\sum_{k=0}^{3}P_k\binom{3}{k}t^k(1-t)^{3-k}=P_0\binom{3}{0}t^0(1-t)^3+P_1\binom{3}{1}t(1-t)^2+P_2\binom{3}{2}t^2(1-t)+P_3\binom{3}{3}t^3(1-t)^0$$

$$P(t)=(1-t)^3P_0+3t(1-t)^2P_1+3t^2(1-t)P_2+t^3P_3$$

$$P(t)=(1-t)^3\left(\frac{\sqrt{2}}{2}\cos 0\ ,\ \frac{\sqrt{2}}{2}\sin 0,\ \frac{\sqrt{2}}{2}\right)+3t(1-t)^2\left(\frac{\sqrt{2}}{2}\cos\frac{\pi}{2}\ ,\ \frac{\sqrt{2}}{2}\sin\frac{\pi}{2},\ \frac{\sqrt{2}}{2}\right)$$
$$+3t^2(1-t)\left(\frac{\sqrt{2}}{2}\cos\pi\ ,\ \frac{\sqrt{2}}{2}\sin\pi,\ \frac{\sqrt{2}}{2}\right)+t^3\left(\frac{\sqrt{2}}{2}\cos\frac{3\pi}{2}\ ,\ \frac{\sqrt{2}}{2}\sin\frac{3\pi}{2},\ \frac{\sqrt{2}}{2}\right).$$

Example 4.1.3. Let's draw the Bézier curve of Saddle surface
$\varphi(u,v)=(\sqrt{u}\cos v\ ,\ \sqrt{u}\ sinv, u\ cosv\ sinv)$ on u-parameter curve with $u=0.5v$;

$$\varphi(v)=\left(\sqrt{0.5v}\cos v\ ,\ \sqrt{0.5v}\ \ sinv, 0.5v\ cosv\ sinv\right)$$

By taking a parameter curve on the Saddle surface and seven control points on this curve, the Bézier curve of these control points is computed.

$$P(t)=(1-t)^6P_0+6t(1-t)^5P_1+15t^2(1-t)^4P_2+20t^3(1-t)^3P_3+15t^4(1-t)^2P_4+6t^5(1-t)P_5+t^6P_6$$

Example 4.1.4. Let's draw the Bézier curve of the torus surface
$\varphi(u,v)=((c+a\cos v)\,cosu, (\mathrm{c}+a\ cosv)\ sinu, a\ sinv)$ on parameter curve with $u=0.28v$;

$$u=0.28v;$$
$$a=1;$$
$$c=3;$$
$$\varphi(u,v)=((3+\cos u)\,cosu,\ (3+cosu)\ sinu, sinu)$$

By taking the $u-$parameter curve on the torus surface and seven control points on this curve, the Bézier curve of these control points is computed. The Matlab algorithm of the Bézier curve is given and its graphs are drawn.

Example 4.1.5. $\varphi(u,v) = (cosu + v\cos\left(\frac{u}{2}\right) cosu, sinu + v\cos\left(\frac{u}{2}\right) sinu, v\sin\left(\frac{u}{2}\right))$ Let's draw the Bézier curve of Möbius surface on parameter curve with $u = 5v$;

$$\varphi(v) = (\cos(5v) + v\cos\left(\frac{5v}{2}\right)\cos(5v), \sin(5v) + v\cos\left(\frac{5v}{2}\right)\sin(5v), v\sin\left(\frac{5v}{2}\right))$$

By taking the $u-$parameter curve on the Möbius surface and five control points on this curve, the Bézier curve of these control points is computed. The Matlab algorithm of the Bézier curve is given and its graphs are drawn.

$$P(t) = (1-t)^4 P_0 + 4t(1-t)^3 P_1 + 6t^2(1-t)^2 P_2 + 4t^3(1-t)P_3 + t^4 P_4$$

Example 4.1.6. With the following equation
X=$\cos(v)\ \sqrt{|\sin(2u)|}cosu$
Y=$\cos(v)\sqrt{|\sin(2u)|}sinu$
Z $=X^2 - Y^2 + 2XYtan^2(v)$
the Lemniscate surface is given by

$$\varphi(u,v) = (X, Y, Z).$$

Let's draw the Bézier curve of the Lemniscate surface on the v-parameter curve with

$$u = v;$$

$$\varphi(v) = (\cos(v)\ \sqrt{|\sin(2v)|}cosv, \cos(v)\sqrt{|\sin(2v)|}sinu, X^2 - Y^2 + 2XYtan^2(v))$$

By taking the parameter curve on the Lemniscate surface and six control points on this curve, the Bézier curve of these control points is computed.

$$P(t) = (1-t)^5 P_0 + 5t(1-t)^4 P_1 + 10t^2(1-t)^3 P_2 + 10t^3(1-t)^2 P_3 + 5t^4(1-t)P_4 + +t^5 P_5$$

Example 4.1.7. By taking five control points on the curve given by the parametric equation $\varphi(u,v) = (u^3, 2u-5, u^2 + 2u)$, the Bézier curve of these control points is computed.

4.2. **Matlab applications.**

Matlab application 1. The obtained Matlab algorithm and graph of the Beizer curve of five control points on the helix curve on a cylinder surface are as follows (see Figure 4.2, 4.3);

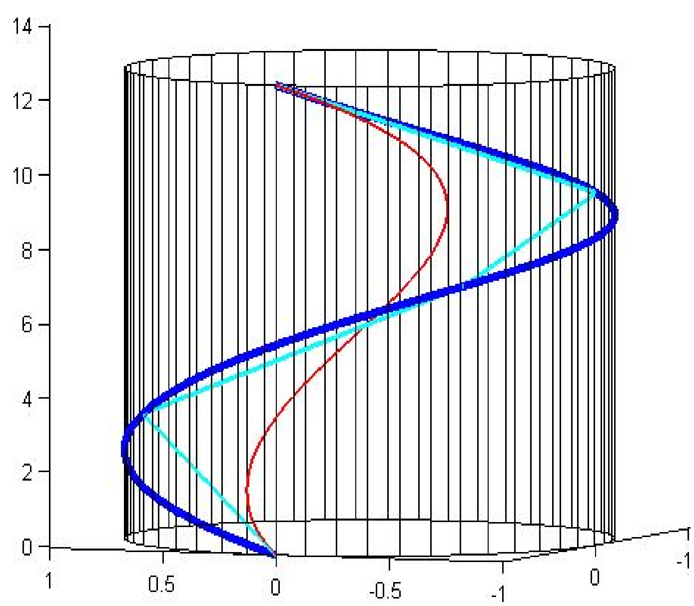

Figure 4.2. Bézier curve on cylinder surface with 5 control points.

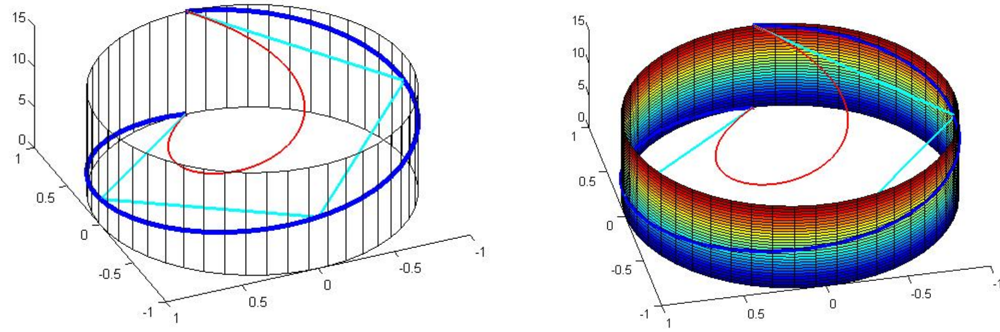

Figure 4.3. Top view of the Bézier curve with 5 control points.

The Matlab graphics of the Beizer curve formed after increasing the number of control points on the helix curve on the cylinder surface are as follows (Figure 4.4).

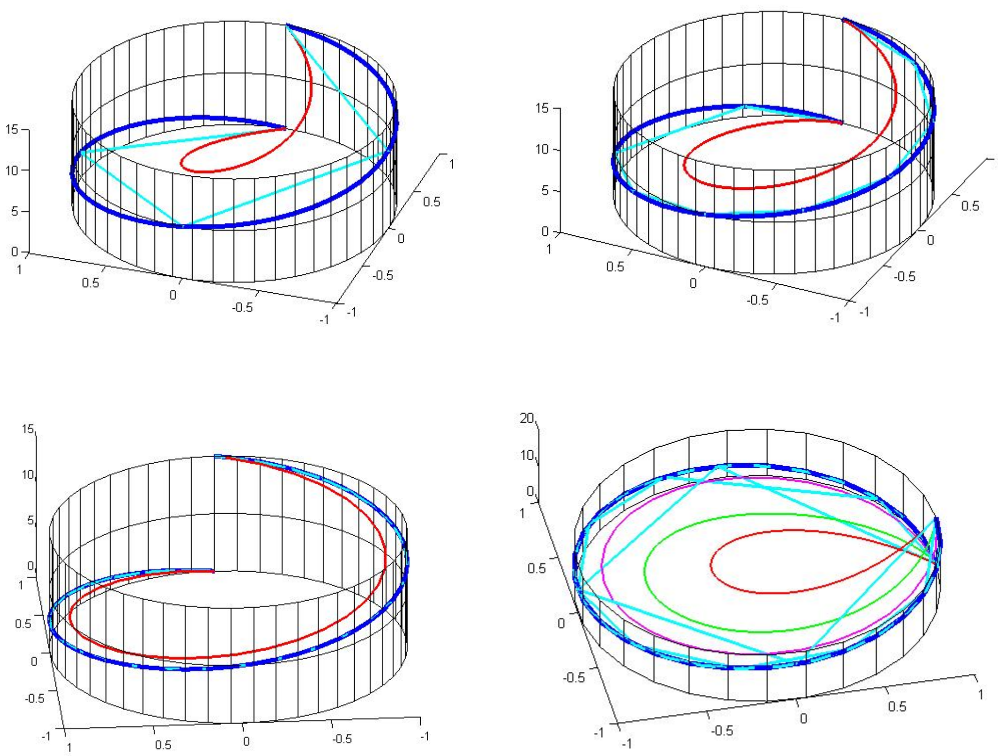

Figure 4.4. Bézier curve on the cylinder surface with 5, 10 and 30 control points, respectively.

m.file of this application is following:

```
Matlab application 1:
u = linspace(0,2*pi,50);
v = linspace(0,4*pi,3);
[U,V] = meshgrid(u,v);
r = 2;
plot3(cos(u), sin(u), r*u, 'LineWidth', 3);
```

```
hold on;
surf(cos(U), sin(U), V);
colormap white;
alpha(0.0001)

Bezier curve with 5 control points
t = 0:0.0025:1;
x = [cos(0) cos(pi/2) cos(pi) cos(3*pi/2) cos(2*pi)];
y = [sin(0) sin(pi/2) sin(pi) sin(3*pi/2) sin(2*pi)];
z = [r*0 r*pi/2 r*pi r*3*pi/2 r*2*pi];

px = (1-t).^4*x(1) + 4*t.*(1-t).^3*x(2)
+ 6*t.^2.*(1-t).^2*x(3) + 4*t.^3.*(1-t)*x(4) + t.^4*x(5);

py = (1-t).^4*y(1) + 4*t.*(1-t).^3*y(2) + 6*t.^2.*(1-t).^2*y(3)
+ 4*t.^3.*(1-t)*y(4) + t.^4*y(5);

pz = (1-t).^4*z(1) + 4*t.*(1-t).^3*z(2) + 6*t.^2.*(1-t).^2*z(3)
+ 4*t.^3.*(1-t)*z(4) + t.^4*z(5);

plot3(x, y, z, 'c', 'LineWidth', 2);
plot3(px, py, pz, 'r', 'LineWidth', 2);
```

Matlab application 2. The Matlab algorithm and graph of the Beizer curve formed after increasing the number of control points on the circle curve on the sphere surface are as follows (see Figure 4.5, 4.6).

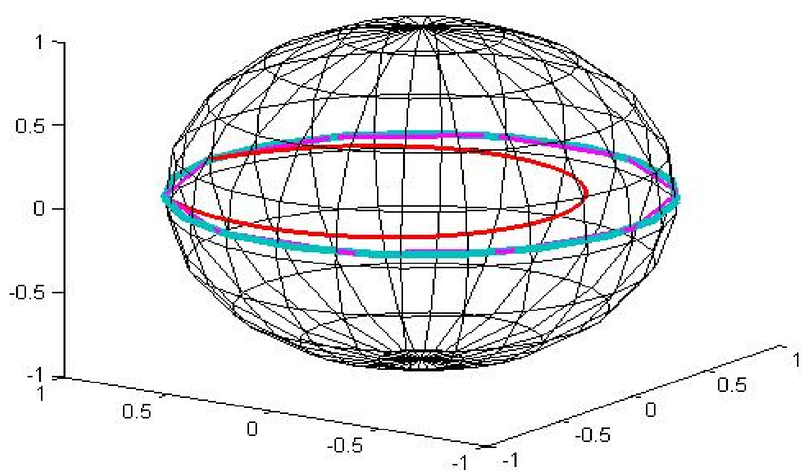

Figure 4.5. Bézier curve on sphere surface with 10 control points.

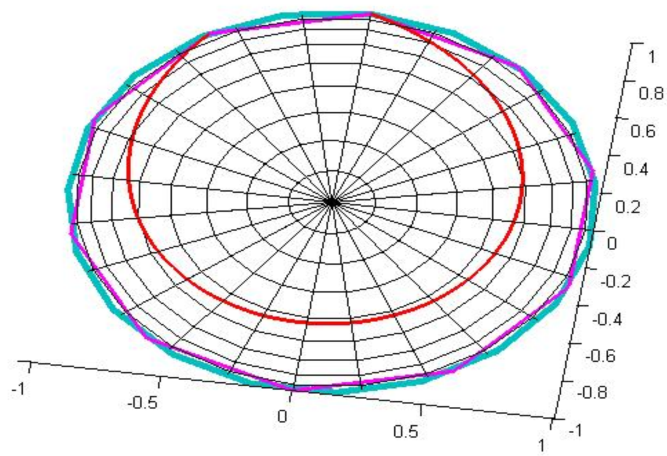

Figure 4.6. Top view of the Bézier curve with 10 control points.

Matlab application 3. The Matlab algorithm and graph of the Beizer curve generated with seven control points on the v-parameter curve on the saddle surface are as follows (Figure 4.7).

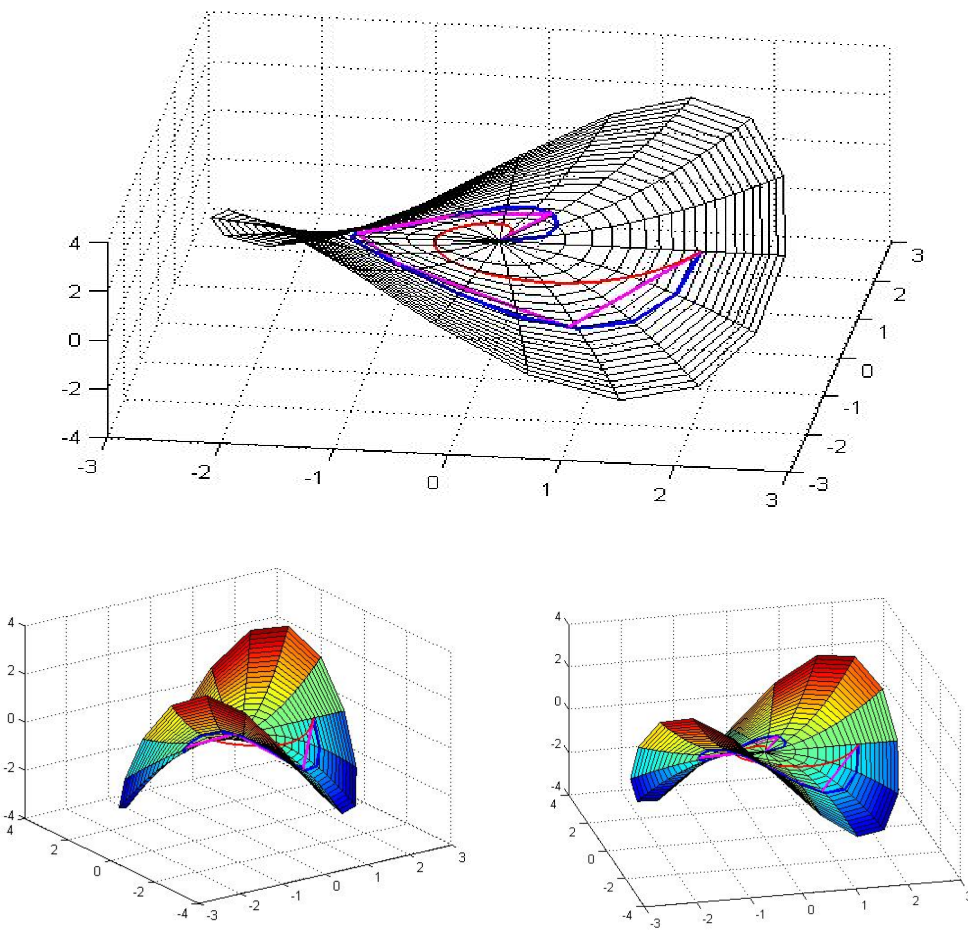

Figure 4.7. Bézier curve on saddle surface with 7 control points.

Matlab application 4. The Matlab algorithm and graph of the Bézier curve generated with seven control points on the v-parameter curve on the torus surface are as follows (Figure 4.8).

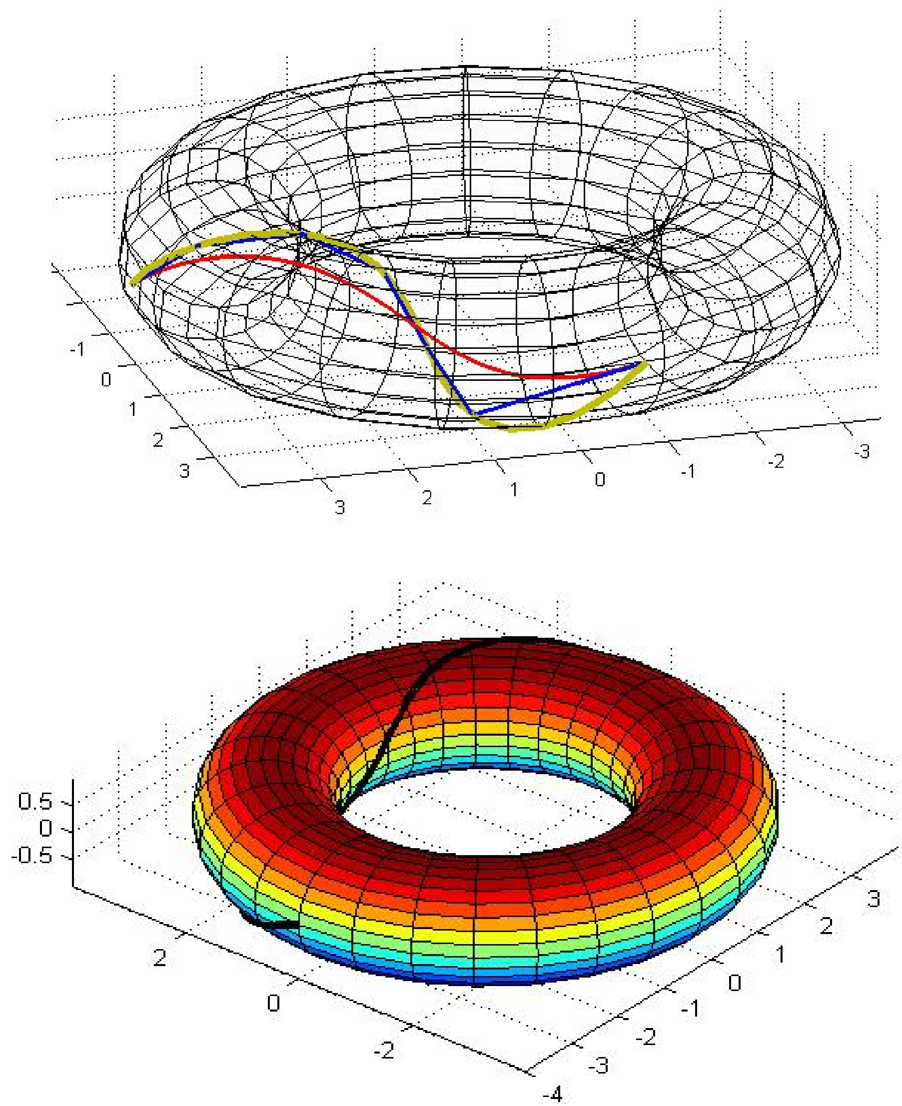

Figure 4.8. Bézier curve on torus surface with 7 control points.

Matlab application 5. The Matlab algorithm and graph of the Bézier curve generated with five control points on the v-parameter curve on the Mobius surface are as follows (Figure 4.9).

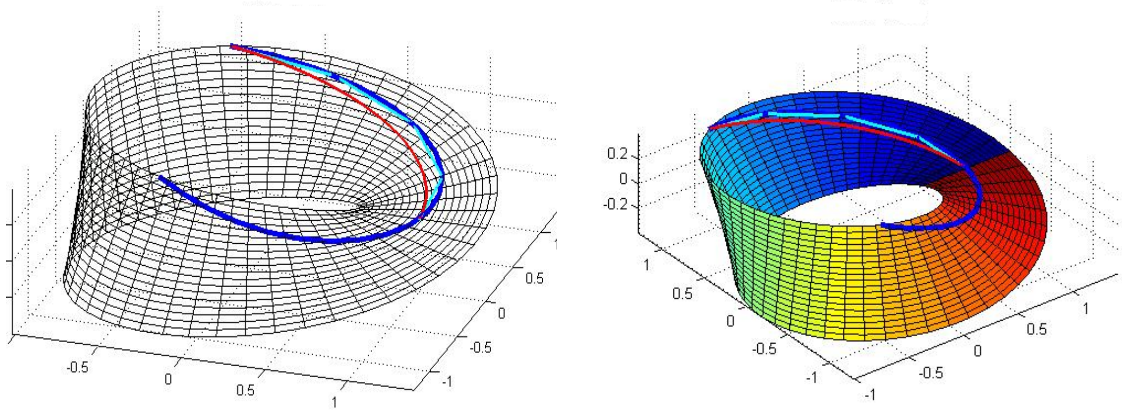

Figure 4.9. Bézier curve on Mobius surface with 5 control points.

Matlab application 6. The Matlab algorithm and graph of the Bézier curve on an implicit minimal lemniscate surface, generated with six control points on the v-parameter curve, is as follows (Figure 4.10).

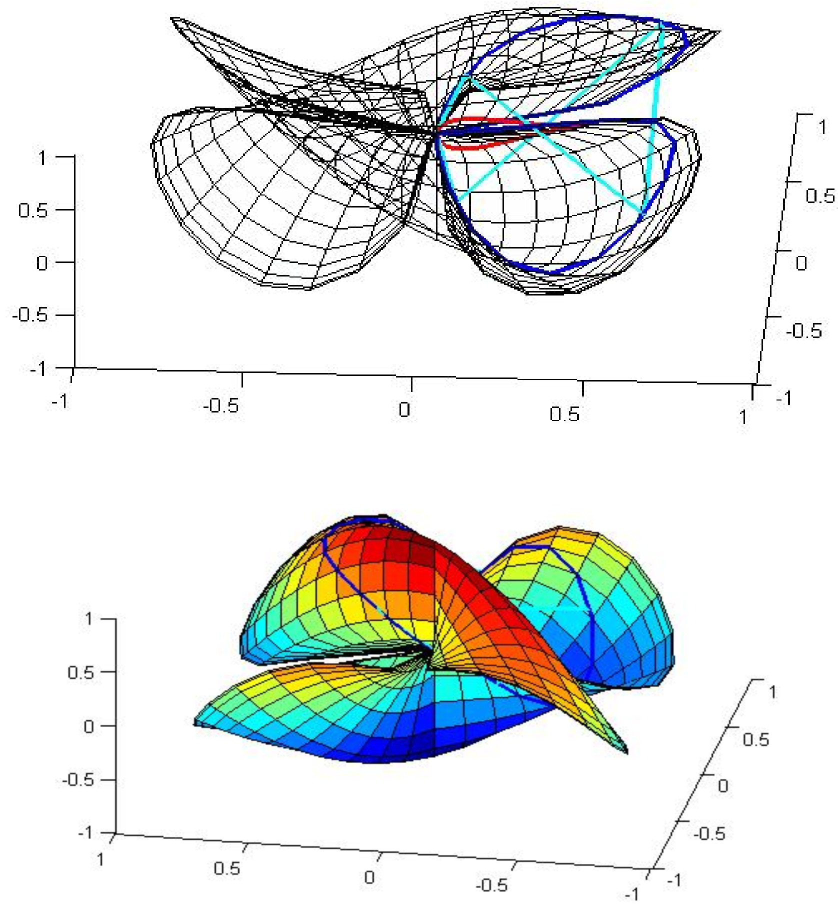

Figure 4.10. Bézier curve on a lemnescate surface with 6 control points.

Matlab application 7. By taking five control points on a curve, the Matlab algorithm and graph of the Beizer curve of these control points are as follows (Figure 4.11).

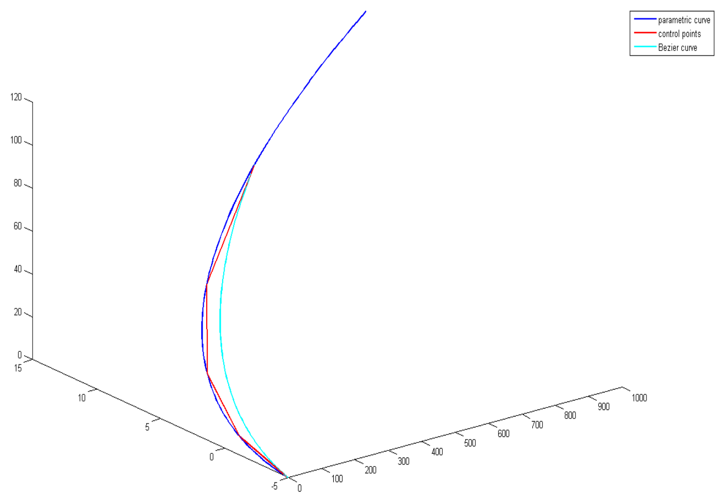

Figure 4.11. Bézier curve with 5 control points.

5. Discussion and Conclusion

Bézier curves are an important curve group determined by Bernstein polynomials and Casteljau algorithm with control points. The selection of control points is determined depending on the subject being studied. Parameter curves of a given surface or any curve on the surface can be taken as a subset from which control points are selected. Thus, Bézier curve bundle is obtained depending on the surface. If the surface is closed or convex, Bézier curve bundle is located inside the surface or inside the convex region. The findings obtained were visualized and controlled with the Matlab program.

Conflicts of Interest

The author/authors declare that there are no conflicts of interest regarding the publication of this article.

Authors Contribution Statement

All authors have read and aggreed the published version of the manuscript.

References

[1] Baydaş, Ş., Karakaş, B., Defining a curve as a Bézier curve, *J.Taibah Univ. Sci.* **13** (2019), 522-528.
[2] Bézier, P., Example of an existing system in the motor industry: the Unisurf system, *Proc. R. Soc. Lond. A.* **321** (1971), 207-218.
[3] Bézier, P., Sioussiou, S., Semi-automatic system for defining free-form curves and surfaces, *Comput. Aided Des.* **15**, (1983) 65-72.
[4] De Casteljau, P., POLynomials, POLar forms, and interPOLation. In: Lyche, T., Schumaker, L.L. (Eds.), *Mathematical Methods in Computer Aided Geometric Design II Academic Press*, 1992, 57-68.
[5] Farin, G., Algorithms for rational Bézier curves, *Comput. Aided Des.* **15** (1983), 73-77.
[6] Farin, G., Curves and Surfaces for CAGD a Practical Guide, Fifth edition, *Morgan Kaufmann*, U.S.A., 2002.
[7] Joy, K. I., On-line Geometric Modeling Notes, Bernstein Polynomials, *CAGDNotes*, 2000.
[8] Kaplan, C. S., Mann, S., *Introduction to Computer Graphics*. Lecture notes CS 488/688 of University of Waterloo, 2006.
[9] Laurent, P. J., Sablonniere, P., Pierre Beziér: An engineer and a mathematician, *Comput. Aided Des. 18* (2001) 609-617.
[10] Lorentz, G. G., Bernstein Polynomials. *Chelsea Publishing Company*, New York, U.S.A., 1986.
[11] Marsh, D., Applied Geometry for Computer Graphics and CAD.,*Springer-Verlag*, London, 2005.
[12] Rabut, C., On Pierre Beziér's life and motivations, *Comput. Aided Des.* **34** (2002), 493-510.
[13] Rogers, D. F., Pierre Etienne Beziér (1910-1999), In Memoriam, *Comput. Aided Des.* **34** 2002, 489-491.
[14] Steffens, K. G.,The History of Approximation Theory,*Birkhauser*, Boston, U.S.A., 2006.