

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

**\mathbb{R}^3 DEKİ YÜZEY EĞRİLERİNİN BEZIER EĞRİLERİ VE MATLAB
UYGULAMALARI**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Ceyda YILMAZ LUZUM
DANIŞMAN : Doç. Dr. Şenay BAYDAŞ

VAN-2018

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

**\mathbb{R}^3 DEKİ YÜZEY EĞRİLERİNİN BEZIER EĞRİLERİ VE MATLAB
UYGULAMALARI**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Ceyda YILMAZ LUZUM

VAN-2018

KABUL VE ONAY SAYFASI

Matematik Anabilim Dalı'nda Doç. Dr. Şenay BAYDAŞ danışmanlığında, Ceyda YILMAZ LUZUM tarafından sunulan “ \mathbb{R}^3 deki Yüzey Eğrilerinin Bezier Eğrileri ve Matlab Uygulamaları” isimli bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince 07/06/2018 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile başarılı bulunmuş ve yüksek lisans tezi olarak kabul edilmiştir.

Başkan: Prof. Dr. Bülent KARAKAŞ



Üye: Doç. Dr. Şenay BAYDAŞ



Üye: Dr. Öğretim Üyesi Muhsin İNCESU



Fen Bilimleri Enstitüsü Yönetim Kurulu'nun .../.../..... tarih ve sayılı kararı ile onaylanmıştır.

İmza

.....
Enstitü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.



Ceyda YILMAZ LUZUM

ÖZET

\mathbb{R}^3 DEKİ YÜZEY EĞRİLERİNİN BEZIER EĞRİLERİ VE MATLAB UYGULAMALARI

YILMAZ LUZUM, Ceyda
Yüksek Lisans Tezi, Matematik Anabilim Dalı
Tez Danışmanı: Doç. Dr. Şenay BAYDAŞ
Haziran 2018, 51 sayfa

\mathbb{R}^3 de yüzey eğrilerinin Bezier eğrilerini incelediğimiz tez yedi bölümden oluşmaktadır. İlk iki bölümde sırasıyla giriş ve kaynak bildirişlerine, üçüncü bölümde temel kavramlara yer verilmiştir.

Dördüncü bölümde bilgisayar programlamada ve bir çok alanda kullanımı oldukça yaygın olan, önemli uygulamalara sahip Bernstein polinomları ve bazı özelliklerinden bahsedilmiştir. Beşinci bölümde Bernstein polinomları cinsinden Bezier eğri tanımı ve özellikleri incelenip Bezier eğri ve yüzey kavramının gelişmesine katkı sağlayan De Casteljau algoritması ele alınmıştır.

Altıncı bölümde, yüzey üzerinde bir eğri ve bu eğrinin ikiden fazla noktası alınarak buna ait Bezier eğrisi hesaplanmıştır. \mathbb{R}^3 deki yüzey eğrilerinin Bezier eğrileri incelenip Matlab programı yardımıyla çizilmiş ve Matlab çıktıları eklenmiştir. Yedinci bölümde sonuç ve tartışma verilmiştir.

Anahtar kelimeler: Bernstein polinomları, Bezier eğrileri, De Casteljau algoritması, Matlab.

ABSTRACT

THE BEZIER CURVES OF SURFACE CURVES ON \mathbb{R}^3 AND MATLAB APPLICATIONS

YILMAZ LUZUM, Ceyda
M.Sc. Thesis, Department of Mathematics
Supervisor: Assos. Prof. Dr. Şenay BAYDAŞ
June 2018, 51 pages

This thesis we examined the Bezier curves and surface curves on \mathbb{R}^3 consists of seven chapters. In the first two chapters, the introduction and the literature review are involved and in the third chapter, the basic concepts are given with respect to the main purpose of the thesis study.

In the fourth chapter, Bernstein polynomials, which have important applications in many areas and more particularly commonly utilized in the scope of computer programming, are examined and some significant features of Bernstein polynomials are mentioned. In the fifth chapter, definition and characteristics of a Bezier curve in terms of Bernstein polynomials and De Casteljau algorithms that contributed to the development of Bezier curves and surface concepts are investigated, as well.

In the sixth chapter, a curve on surface, as well as multiple points on this curve, are taken in order to calculate the Bezier curve belonging to these points. The Bezier curves on the surface curves on \mathbb{R}^3 are investigated and plotted with Matlab of which outputs are also included in the study. Finally, in the last chapter, the discussion and conclusion are presented.

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



ÖN SÖZ

Bu tez çalışmasında, her türlü ilgi ve yardımlarını esirgemeyen bana her zaman destek olan değerli hocalarım Sayın Doç. Dr. Şenay BAYDAŞ ve Sayın Prof. Dr. Bülent KARAKAŞ'a ve ayrıca verdikleri desteklerden ötürü eşim Arş. Gör. Murat LUZUM ve aileme teşekkürlerimi borç bilirim.

2018

Ceyda YILMAZ LUZUM



İÇİNDEKİLER

	Sayfa
ÖZET	i
ABSTRACT	iii
ÖN SÖZ.....	v
İÇİNDEKİLER.....	vii
ŞEKİLLER LİSTESİ.....	ix
1. GİRİŞ.....	1
2. KAYNAK BİLDİRİŞLERİ	3
3. TEMEL KAVRAMLAR.....	5
3.1. \mathbb{R}^3 de eğriler	5
3.2. \mathbb{R}^3 de yüzeyler.....	6
4. BERNSTEIN POLİNOMLARI.....	9
4.1. Bernstein polinomları	9
4.2. Bernstein polinomlarının temel özellikleri	10
4.2.1. Birimin parçalanması özelliği.....	12
4.2.2. Anti-negatiflik özelliği	13
4.2.3. Simetri özelliği	13
4.2.4. Rekürsiyon özelliği.....	14
4.3. Bernstein polinomları için matris temsili	15
5. BEZIER EĞRİLERİ VE DE CASTELJAU ALGORİTMASI	17
5.1. Bezier eğrileri	17
5.1.1. Bezier eğri parçasının oluşturulması	17
5.2. De Casteljau algoritması.....	18
5.2.1. Bezier eğrisi için De Casteljau algoritması	19
5.2.2. Bezier eğrilerinin bazı özellikleri	27
5.2.2.1. Son nokta interpolasyon özelliği	27
5.2.2.2. Son nokta teğet özelliği	27

	Sayfa
6. \mathbb{R}^3 DEKİ YÜZEY EĞRİLERİNİN BEZIER EĞRİLERİ VE MATLAB UYGULAMALARI.....	29
6.1. \mathbb{R}^3 deki yüzey eğrilerinin Bezier eğrileri	29
6.2. Matlab uygulamaları.....	34
7. TARTIŞMA VE SONUÇ.....	47
KAYNAKLAR.....	49
ÖZ GEÇMİŞ.....	51



ŞEKİLLER LİSTESİ

Şekil	Sayfa
Şekil 3.1. $\alpha(t)$ eğrisi.....	5
Şekil 4.1. Sıfırıncı dereceden Bernstein polinomunun grafiği.	10
Şekil 4.2. Birinci dereceden Bernstein polinomunun grafiği	11
Şekil 4.3. İkinci dereceden Bernstein polinomunun grafiği.....	12
Şekil 4.4. Üçüncü dereceden Bernstein polinomunun grafiği	12
Şekil 5.1. $P_0^1(t)$ nin P_0 ve P_1 e göre temsili.....	17
Şekil 5.2. De Casteljaou algoritması : $b_0^3(t)$ noktası tekrarlı lineer interpolasyon. ..	18
Şekil 5.3. Bezier eğrisi için De Casteljaou algoritması.	19
Şekil 5.4. $P_0 = (3,3)$, $P_1 = (2,5)$ kontrol noktalarından geçen Bezier eğrisi.....	21
Şekil 5.5. P_0, P_1 ve P_2 kontrol noktalarının oluşturduğu kuadratik Bezier eğrisi....	22
Şekil 5.6. P_0, P_1, P_2 ve P_3 kontrol noktalarının oluşturduğu kübik Bezier eğrisi.	23
Şekil 5.7. 2. , 3. , 4. ve 5. dereceden Bernstein polinomları.....	24
Şekil 5.8. Düzlemsel kuadratik Bezier eğrisi	25
Şekil 5.9. Düzlemsel kübik Bezier eğrisi	26
Şekil 5.10. 3-D kübik Bezier eğrisi	26
Şekil 6.1. $\varphi(u, v)$ yüzeyi üzerindeki parametrik eğriler	29
Şekil 6.2. 5 kontrol noktasına ait silindir yüzeyi üzerindeki Bezier eğrisi	35
Şekil 6.3. 5 kontrol noktasına ait Bezier eğrisinin üstten görünümü.....	35
Şekil 6.4. Sırasıyla 5, 10 ve 30 kontrol noktalarına ait silindir yüzeyi üzerindeki Bezier eğrisi	36
Şekil 6.5. 10 kontrol noktası için küre yüzeyi üzerindeki Bezier eğrisi	37

Şekil	Sayfa
Şekil 6.6. 10 kontrol noktasına ait Bezier eğrisinin üstten görünümü.....	37
Şekil 6.7. 7 kontrol noktası için saddle yüzeyi üzerindeki Bezier eğrisi.....	39
Şekil 6.8. 7 kontrol noktası için torus yüzeyi üzerindeki Bezier eğrisi.....	41
Şekil 6.9. 5 kontrol noktası için mobius yüzeyi üzerindeki Bezier eğrisi	42
Şekil 6.10. 6 kontrol noktası için lemniscate yüzeyi üzerinde Bezier eğrisi	44
Şekil 6.11. 5 kontrol noktası için Bezier eğrisi.....	45



1. GİRİŞ

Matematięi oluřturan konular, yařanılan dđnemle doęrudan iliřkili olduęundan ilgi ve anlamlandırılmada zamana baęlı deęiřimler gđrđlđr. Örneęin antik Yunanda geometri, konumsal iliřkilerin incelendięi bir matematik dalı olarak algılanırken, gđnümüzde diferansiyel topoloji veya diferansiyel manifoldların incelendięi oldukça karmařık bir matematik dalına dđnüşmüřtür. Benzer durum, geometrinin bir alt dalı olan diferansiyel geometri için de geçerlidir.

Bezier eęrisi bilgisayar grafiklerinde ve ilgili alanlarda sıklıkla kullanılan parametrik bir eęridir. Bernstein polinomlarının takdiminden elli yıl sonra Fransız mühendis Pierre Bezier tarafından otomobil tasarlanırken kullanılan zamana kadar grafięe uygulanmamıřtır. Aynı dđnem dięer bir Fransız mühendis Paul De Casteljaou tarafından da Casteljaou algoritması kullanılarak geliřtirilmiřtir.

Bir Bezier eęrisi, verilen belli sayıdaki kontrol noktasına baęlı olarak Bernstein polinom yaklařımı ve De Casteljaou algoritması ile oluřturulan eęridir. Benzer olarak üç boyutlu uzayda parametre sayısını iki alarak belirlenmiř noktalara göre Bezier yüzeyleri verilir.

Bu çalıřmada, yüzey bir Bezier yüzeyi olarak verilmedięi zaman, parametre eęrilerinin ve herhangi eęrilerin sonlu sayıdaki noktaları üstünden, yüzey eęrisinin Bezier eęrisinin nasıl olacaęı, hangi řart ve kısıtlarda yüzey üstünde kalıp kalamayacaęı çalıřılmıřtır.

2. KAYNAK BİLDİRİŞLERİ

Klasik diferansiyel geometri, eğri ve yüzeylerin lokal özelliklerini ele alır. Burada lokal özellikler denildiğinde, eğri ya da yüzeyin bir nokta civarındaki davranışı anlaşılır. Gauss'un 1821-1825 yılları arasında Hannover krallığının ölçümü için açı ve uzunluk gözlemleriyle tesis ettiği yüzey nirengi ağı, eğriler ve yüzeyler için diferansiyel geometrinin teoriden pratiğe geçtiği önemli bir adımdır.

Polinomlar kolay tanımlandıkları, bilgisayar sistemlerinde hızlı hesaplandıkları için çok kullanışlı matematiksel araçlardır. Ayrıca polinomlar kolaylıkla diferensiyellenebilir ve integrallenebilir. Bu nedenle polinomlarla işlem yapmak, yapılan çalışmalarda büyük kolaylıklar sağlamaktadır (Joy, 2000).

Bernstein polinomlarının tarihi, Ukraynalı matematikçi Sergei Natanovic Bernstein'e (1880-1968) dayanır. Bernstein'in yapmış olduğu çalışmalar Fransa'da büyük ilgi görmüştür. 1955'de Fransız Bilim Akademisi üyesi olarak seçilmiştir (Steffens, 2006). Daha sonra $P(t)$ parametrik denkleminle verilen bir Bezier eğrisinin $0 \leq t \leq 1$ Bernstein polinomları cinsinden tanımı ve bazı özellikleri incelenip, De Casteljau algoritması ele alınmıştır. Bezier eğrileri ve yüzeyleri, bilgisayar destekli geometrik tasarım (CAGD) için büyük önem taşımaktadır. Bezier eğrileri ve yüzeyleri ilk olarak 1958- 1960 yılları arasında Fransız mühendis Pierre Etienne Bezier (1910-1999) ve Fransız matematikçi Paul de Faget De Casteljau (1930-) tarafından farklı matematiksel yaklaşımlar ile birbirinden bağımsız olarak geliştirilmiştir.

Pierre Etienne Bezier makine ve endüstri mühendisidir. Mezun olduktan sonra 1933 yılında Renault firmasında mekanik yöntem planlama bölümünde çalışmaya başlamıştır. 1960 yılından itibaren de araç gövde yöntem planlama bölümüne geçip araç gövdelerinin tasarımı üzerinde çalışmıştır (Rabut, 2002; Rogers, 2002). Uzun yıllar bu firmada çalışan Bezier'in ismiyle anılır hale gelen polinom eğri ve yüzeylerinin Bernstein formunda ifadesiyle tamamen gelişen Bezier eğri ve yüzeyleri teorisi bu dönem içerisinde kurulmuştur. Otomobil kaporta yüzeylerinin az sayıda parametre değiştirilerek kontrol edilebilen eğrilerle tanımlanabilmesi için geliştirilmiştir (Bezier ve Sioussiou, 1983).

1958 yılında Citroen firmasına katılan De Casteljau tamamen farklı bir yaklaşımla aynı kavramı; kendisinden sonra adını alan De Casteljau algoritmasını taktim etmiştir. Bezier eğri ve yüzeyleri kavramının gelişmesine büyük katkı sağlayan bu algoritma Bernstein polinomlarının kullanımına denktir. De Casteljau'nun çalışmaları çok uzun bir süre Citroen firması tarafından gizli tutulmuştur. Hâlbuki De Casteljau, Bezier'in Bezier eğri ve yüzeylerinin Bernstein polinomları ile ifade edilebileceğini fark etmesinden çok önce bu polinomları kullanmıştır (Farin, 2002; Laurent ve Sablonniere, 2001; Rabut, 2002; Rogers, 2002).



3. TEMEL KAVRAMLAR

3.1. \mathbb{R}^3 te eğriler

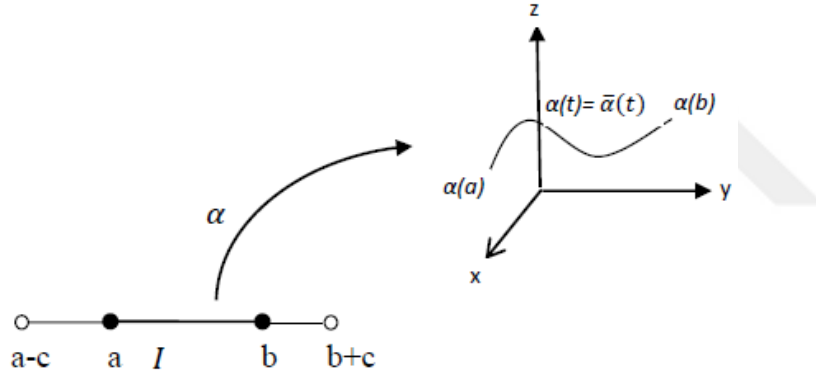
$I \subseteq \mathbb{R}$ olmak üzere, $\alpha: I \rightarrow \mathbb{R}^3$ fonksiyonuna \mathbb{R}^3 te bir eğri denir. α , C^k sınıfından ise α ya C^k sınıfından eğri veya kısaca C^k eğri olarak adlandırılır.

α nın tanımlı olduğu I aralığı $[a, b]$ şeklinde bir kapalı aralık ise α nın diferensiyellenebilirliğinden kasıt şudur:

$\exists c > 0$ reel sayısı ve $\bar{\alpha}: (a - c, b + c) \rightarrow \mathbb{R}^3$ diferensiyellenebilir fonksiyonu,

$$\bar{\alpha}(t) = \alpha(t), \forall t \in [a, b]$$

olacak şekilde var ise $\alpha(t)$ ye diferensiyellenebilir eğri denir (Şekil 3.1) (Brickell ve Clark, 1970).



Şekil 3.1. $\alpha(t)$ eğrisi.

$\alpha(a)$ ve $\alpha(b)$ ye eğrinin başlangıç ve bitiş noktaları denir. $I = [a, b]$ ise uç noktalar eğriye aittir. $t \in I$ için $\alpha(t) = (\alpha_1(t), \alpha_2(t), \alpha_3(t))$ olup, α_i ler $\alpha(t)$ eğrisinin koordinat temsili olan fonksiyonlardır. $(\alpha_1(t), \alpha_2(t), \alpha_3(t))$ ifadesi de eğrinin parametrik temsili (koordinat temsili) olarak adlandırılır. $t \in I \subseteq \mathbb{R}$ parametresi zaman parametresi gibi düşünülürse $\alpha(t)$ eğrisi hareketli bir noktanın \mathbb{R}^3 deki yörüngesi olarak ele alınabilir.

Örnek 3.1.1

$$\alpha: I \subseteq \mathbb{R} \rightarrow \mathbb{R}^3$$

$$t \rightarrow \alpha(t) = (t^2, t^4, 3 + 5t^3) \in \mathbb{R}^3$$

olup $\alpha(t)$ nin diferensiyellenebilir bir eğri olduğunu gösterelim.

Eğer I aralığı $[a, b]$ olarak alınırsa $\alpha(t)$ nin diferensiyellenebilir olmasından kastedilen şudur:

$\exists c > 0$ için

$$\bar{\alpha}: (a - c, b + c) \rightarrow \mathbb{R}^3$$

$$\bar{\alpha}(t) = \alpha(t), \quad t \in [a, b]$$

olacak şekilde var ise $\alpha(t)$ ya diferensiyellenebilir eğri denir. Bir eğri, fonksiyon, fonksiyonun tanım cümlesi ve görüntü cümlesiyle $(\alpha, I, \mathbb{R}^3)$ birlikte bir diferensiyellenebilir eğri oluşturur.

$\alpha(t)$ için,

$$\alpha'(t) = (2t, 4t^3, 15t^2)$$

olursa ifadenin türevi var ve türevi sürekli olduğundan $\alpha(t)$ bir diferensiyellenebilir eğridir.

3.2. \mathbb{R}^3 te yüzeyler

\mathbb{R}^3 de bir alt küme $G = \{(u, v) | (u, v) \in \mathbb{R}^2\} \subset \mathbb{R}^2$ olarak verilsin. $\varphi: G \rightarrow \mathbb{R}^3$ dönüşümüne veya görüntü cümlesine \mathbb{R}^3 de bir yüzey denir.

φ sürekli (diferensiyellenebilir) ise, φ ye sürekli (diferensiyellenebilir) yüzey denir (Brickell ve Clark, 1970).

Örnek 3.2.1.

$$D = \left(0, \frac{\pi}{2}\right) \times (0, 1)$$

$$\varphi: D \rightarrow \varphi(D) \subset \mathbb{R}^3$$

$$(u, v) \rightarrow \varphi(u, v) = (\cos(u), \sin(u), v)$$

fonksiyonu diferensiyellenebilir yüzey tanımlar. Örnekte ayrıca regülerlik şartının sağlandığı da gösterilecektir.

$$\varphi(u_1, v_1) = \varphi(u_2, v_2)$$

$$(\cos(u_1), \sin(u_1), v_1) = (\cos(u_2), \sin(u_2), v_2)$$

$$\cos(u_1) = \cos(u_2)$$

$$\sin(u_1) = \sin(u_2)$$

$$u_1 = u_2$$

$$v_1 = v_2$$

olduğundan φ birebirdir.

$\varphi, s > 1$ sınıfındadır. Şöyle ki;

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

$$\frac{\partial \varphi}{\partial u} = \varphi_u = (-\sin(u), \cos(u), 0)$$

olup diferensiyellenebilir. Ayrıca,

$$\frac{\partial \varphi}{\partial v} = \varphi_v = (0, 0, 1)$$

$$\varphi_{uu} = (-\cos(u), -\sin(u), 0)$$

$$\varphi_{uv} = (0, 0, 0)$$

$$\varphi_{vu} = (0, 0, 0)$$

$$\varphi_{vv} = (0, 0, 0)$$

olduğundan $\varphi, s > 1$ sınıfındadır.

J Jakobiyen matrisi verilsin.

$$J = \begin{bmatrix} -\sin u & \cos u & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

rank $J=2$ olmalıdır.

J Jakobiyen matrisinin rankını hesaplamak için alt kare matrislerinin determinantlarına bakılmalıdır.

$$\det \begin{bmatrix} -\sin u & 0 \\ 0 & 1 \end{bmatrix} = -\sin u \neq 0$$

$$\det \begin{bmatrix} \cos u & 0 \\ 0 & 1 \end{bmatrix} = \cos u \neq 0$$

$$\det \begin{bmatrix} -\sin u & \cos u \\ 0 & 0 \end{bmatrix} = 0$$

şeklindedir. En az bir alt 2×2 kare matrisinin determinantı sıfırdan farklı olduğu için rank $J=2$ dir. Sonuç olarak φ , bir regüler yüzeydir.



4. BERNSTEIN POLİNOMLARI

4.1. Bernstein polinomları

$x, y \in \mathbb{R}$; $n \in \mathbb{N}$ için $(x + y)^n$ binom açılımı,

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{(n-k)} \quad (4.1)$$

olarak verilir. Bu açılım Ömer Hayyam'dan beri bilinmektedir ve katsayılar olan $\binom{n}{k}$ lar Pascal üçgeni olarak bilinen bir algoritma ile verilebilir. Bernstein x ve y nin olasılık değişkenleri olması durumunda, bu açılıma istatistiki bir kimlik kazandırmıştır. Bir olaya ait olasılıklar x ve y ise;

$$x + y = 1 \quad (4.2)$$

şeklindedir. Böylece $x + y$ toplamı, $x + (1 - x)$ toplamı olarak yazılabilir. Bu durumda olaya ait olasılık açılımı,

$$1 = (x + (1 - x))^n \quad (4.3)$$

$$= \sum_{k=0}^n \binom{n}{k} x^k (1 - x)^{(n-k)} \quad (4.4)$$

olarak verilir. Eş. 4.4 açılımı Bernstein polinomu olarak adlandırılır. Polinoma eşlik eden her bir,

$$\binom{n}{k} x^k (1 - x)^{(n-k)} \quad (4.5)$$

terimi Bernstein polinom açılımının bileşenleri (katsayıları) olarak isimlendirilir. Bernstein polinomları polinomlar halkasında önemli bir pozisyona sahiptir ve baz olma özelliği en önemli özelliklerinden birisidir. Ayrıca Bezier eğrilerinin inşasında da önemli bir konumdadırlar.

4.2. Bernstein polinomlarının temel özellikleri

Tanım 4.2.1. n. dereceden bir Bernstein polinomu $k=0,1,2,\dots,n$ için,

$$B_{k,n}(x) = \binom{n}{k} x^k (1-x)^{n-k} \quad (4.6)$$

katsayıları ile tanımlanır. Burada,

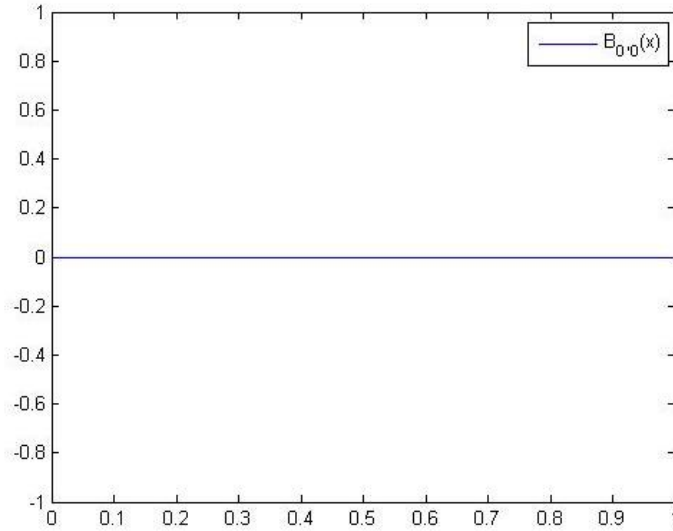
$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (4.7)$$

dir. Bu polinomları yazmak oldukça kolaydır. $\binom{n}{k}$ katsayıları Pascal üçgeninden kolayca elde edilebilir; k sayısı arttıkça x teriminin üssü bir artar ve $(1-x)$ teriminin üssü bir azalır. Sıfırıncı, birinci, ikinci ve üçüncü dereceden Bernstein polinomları aşağıdaki gibi hesaplanabilir:

Derecesi sıfır olan Bernstein polinomu,

$$B_{0,0}(x) = 1 \quad (4.8)$$

olarak tanımlanır ve $0 \leq x \leq 1$ için grafiği aşağıdaki gibi çizilebilir (Şekil 4.1) (Joy, 2000).



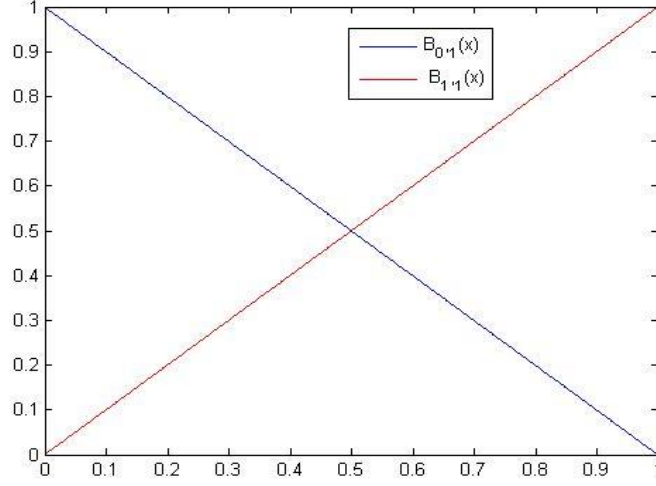
Şekil 4.1. Sıfırıncı dereceden Bernstein polinomunun grafiği.

Derecesi bir olan Bernstein polinomları

$$B_{0,1}(x) = \binom{1}{0} x^0(1-x)^{1-0} = (1-x) \quad (4.9)$$

$$B_{1,1}(x) = \binom{1}{1} x^1(1-x)^{1-1} = x \quad (4.10)$$

olarak elde edilir ve $0 \leq x \leq 1$ için grafiği aşağıdaki gibi çizilebilir (Şekil 4.2).



Şekil 4.2. Birinci dereceden Bernstein polinomlarının grafiği.

Derecesi iki olan Bernstein polinomları,

$$B_{0,2}(x) = \binom{2}{0} x^0(1-x)^{2-0} = (1-x)^2 \quad (4.11)$$

$$B_{1,2}(x) = \binom{2}{1} x^1(1-x)^{2-1} = 2x(1-x) \quad (4.12)$$

$$B_{2,2}(x) = \binom{2}{2} x^2(1-x)^{2-2} = x^2 \quad (4.13)$$

olarak elde edilir ve $0 \leq x \leq 1$ için grafiği aşağıdaki gibi çizilebilir (Şekil 4.3).

Derecesi üç olan Bernstein polinomları,

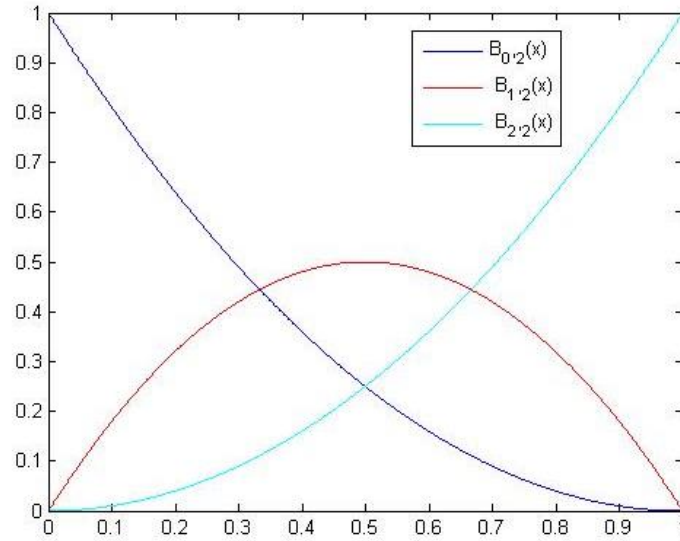
$$B_{0,3}(x) = \binom{3}{0} x^0(1-x)^{3-0} = (1-x)^3 \quad (4.14)$$

$$B_{1,3}(x) = \binom{3}{1} x^1(1-x)^{3-1} = 3x(1-x)^2 \quad (4.15)$$

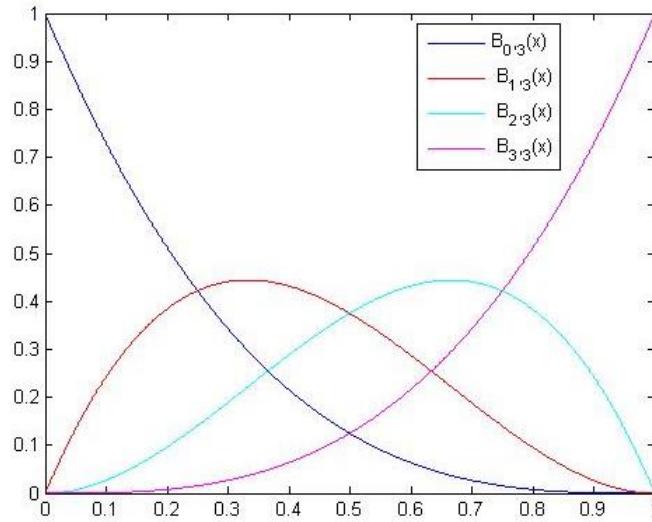
$$B_{2,3}(x) = \binom{3}{2} x^2(1-x)^{3-2} = 3x^2(1-x) \quad (4.16)$$

$$B_{3,3}(x) = \binom{3}{3} x^3(1-x)^{3-3} = x^3 \quad (4.17)$$

olarak elde edilir ve $0 \leq x \leq 1$ için grafiği aşağıdaki gibi çizilebilir (Şekil 4.4).



Şekil 4.3. İkinci dereceden Bernstein polinomlarının grafikleri.



Şekil 4.4. Üçüncü dereceden Bernstein polinomlarının grafiği.

4.2.1. Birimin parçalanması özelliği:

Bernstein polinomu $B_k^n(t)$ için, binom açılımından

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{(n-k)} \quad (4.18)$$

şeklinde yazılır.

$$x + y = 1 \quad (4.19)$$

olursa

$$x = t$$

ise o halde (4.20)

$$y = 1 - t$$

dir. $n \in Z^+$ için

$$1 = 1^n = (t + (1 - t))^n = \sum_{k=0}^n \binom{n}{k} (1 - t)^{n-k} t^k \quad (4.21)$$

olur (Marsh, 2005).

4.2.2. Anti-negatiflik özelliği:

$t \in [0,1]$ için $B_k^n(t) \geq 0$ dir.

Bu özellik; $\binom{n}{k} = \frac{n!}{k!(n-k)!} > 0$, $t \geq 0$ ve $(1 - t) \geq 0$ olduğundan

$$B_k^n(t) = \binom{n}{k} (1 - t)^{n-k} t^k \geq 0 \quad (4.22)$$

olur (Marsh, 2005).

4.2.3. Simetri özelliği:

Bernstein polinomları $B_k^n(t)$ için;

$$B_{n-k}^n(t) = B_k^n(1 - t) \quad (4.23)$$

eşitliği geçerlidir.

Burada, n. dereceden bir Bernstein polinomu Eş. 4.6 eşitliği ile verilmişti. Böylece

$$B_{n-k}^n(t) = \binom{n}{n-k} (1 - t)^k t^{n-k} \quad (4.24)$$

ve

$$B_k^n(1 - t) = \binom{n}{k} (t)^{n-k} (1 - t)^k \quad (4.25)$$

şeklindedir. Bununla birlikte Eş. 4.24 ve Eş. 4.25 eşitliklerinin sağ tarafları eşit olduğundan sol tarafları da eşittir. Bu nedenle

$$B_{n-k}^n(t) = B_k^n(1 - t) \quad (4.26)$$

olur (Marsh, 2005).

4.2.4. Rekürsiyon özelliği:

n . dereceden Bernstein polinomları $B_k^n(t)$ $k = 0, \dots, n$ için $(n - 1)$. dereceden polinomlarla aşağıdaki gibi ifade edilebilir:

$$B_k^n(t) = (1 - t)B_k^{n-1}(t) + tB_{k-1}^{n-1}(t) \quad (4.27)$$

Burada; $B_{-1}^{n-1}(t) = 0$ ve $B_n^{n-1}(t) = 0$ dır. n . dereceden bir Bernstein polinomu Eş. 4.6 eşitliği ile verilmişti. Buradan;

$$B_k^{n-1}(t) = \binom{n-1}{k} (1-t)^{n-1-k} t^k \quad (4.28)$$

ve

$$B_{k-1}^{n-1}(t) = \binom{n-1}{k-1} (1-t)^{n-k} t^{k-1} \quad (4.29)$$

elde edilir. Böylece $k = 0$ için, $B_{-1}^{n-1}(t)$ yardımıyla

$$B_0^n(t) = (1-t)^n = (1-t)B_0^{n-1}(t) + tB_{-1}^{n-1}(t) \quad (4.30)$$

bulunur. Benzer şekilde $k = n$ için, $B_n^{n-1}(t) = 0$ yardımıyla

$$B_n^n(t) = t^n = (1-t)B_n^{n-1}(t) + tB_{n-1}^{n-1}(t) \quad (4.31)$$

yazılabilir. Genel halde $1 \leq k \leq n$ için Eş. 4.28 ve Eş. 4.29 eşitliklerini sırasıyla $(1-t)$ ve t ile çarpılırsa Bernstein polinomu tanımından

$$\begin{aligned} & (1-t).B_k^{n-1}(t) + t.B_{k-1}^{n-1}(t) \\ &= (1-t). \binom{n-1}{k} (1-t)^{n-1-k} t^k + t. \binom{n-1}{k-1} (1-t)^{n-k} t^{k-1} \\ &= \binom{n-1}{k} (1-t)^{n-k} t^k + \binom{n-1}{k-1} (1-t)^{n-k} t^k \\ &= (1-t)^{n-k} t^k. \left[\binom{n-1}{k} + \binom{n-1}{k-1} \right] \\ &= (1-t)^{n-k} t^k. \left[\frac{(n-1)!}{(n-1-k)!k!} + \frac{(n-1)!}{(n-k)!(k-1)!} \right] \\ &= (1-t)^{n-k} t^k. \left[\frac{(n-k)+k}{n} \right]. \frac{n!}{k!(n-1)!} \\ &= \binom{n}{k} (1-t)^{n-k} t^k \\ &= B_k^n(t) \end{aligned} \quad (4.32)$$

dir (Marsh, 2005).

4.3. Bernstein polinomları için matris temsili

Uygulamada Bernstein polinomları için matris temsili kullanışlıdır. Eğer noktasal çarpımların lineer birleşimi cinsinden bakılırsa, bunlar doğrudan geliştirilebilir.

Bernstein baz fonksiyonlarının lineer birleşimi şeklinde verilen bir polinom

$$B(x) = c_0 B_{0,n}(x) + c_1 B_{1,n}(x) + \dots + c_n B_{n,n}(x) \quad (4.33)$$

olsun. Bu eşitliği iki vektörün noktasal çarpımı olarak yazmak kolaydır.

$$B(x) = [B_{0,n}(x) \ B_{1,n}(x) \ \dots \ B_{n,n}(x)] \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (4.34)$$

bu eşitlik ise,

$$B(x) = [1 \ x \ x^2 \ \dots \ x^n] \begin{bmatrix} b_{0,0} & 0 & 0 & \dots & 0 \\ b_{1,0} & b_{1,1} & 0 & \dots & 0 \\ b_{2,0} & b_{2,1} & b_{2,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ b_{n,0} & b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad (4.35)$$

şekline dönüşebilir. Burada $b_{i,j}$ ler ilgili Bernstein polinomunu belirlemede kullanılan kuvvet bazının katsayılarıdır. Kuadratik durumda (n=2 için) matris temsili

$$B(x) = [1 \ x \ x^2] \begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \quad (4.36)$$

dir ve kübik durumda (n=3 için) matris temsili,

$$B(x) = [1 \ x \ x^2 \ x^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (4.37)$$

biçimindedir (Joy, 2000).



5. BEZIER EĞRİLERİ VE DE CASTELJAU ALGORİTMASI

5.1. Bezier eğrileri

Bu bölümde, Bezier eğrileri ve bu eğrileri elde ederken kullanılan Bernstein polinomları ve bazı özellikleri verilmiştir.

Bezier eğrisi kontrol noktaları ve onları inşa edecek bir temel fonksiyon ile tanımlanır. Seçilen ilk ve son kontrol noktası, eğrinin başı ve sonunu oluşturur. Aradaki diğer noktalar ise eğrinin yapısını belirlemek için kullanılır. Bu bağlamda bu noktalar genellikle eğrinin üzerinde yer almaz (Bezier, 1971).

5.1.1. Bezier eğri parçasının oluşturulması

Tanım 5.1.1.1. E^n ($n = 2,3$) Öklid uzayında P_0 ve P_1 noktaları verildiğinde $\overline{P_0P_1}$ doğru parçası üzerindeki herhangi bir nokta $P_0^1(t)$ ile gösterilirse;

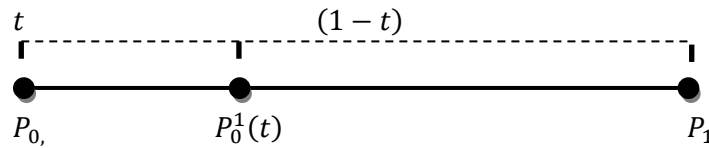
$$P(t) = P_0^1(t) = (1-t)P_0 + tP_1 \quad ; t \in [0,1] \quad (5.1)$$

eşitliği bir lineer eğri parçasının parametrisasyonunu oluşturur. Burada;

$$B_0^1(t) = (1-t) \quad (5.2)$$

$$B_1^1(t) = t \quad (5.3)$$

birinci dereceden Bernstein polinomları ile P_0 ve P_1 kontrol noktaları olarak adlandırılır (Şekil 5.1) (Kaplan ve Mann, 2006; Marsh, 2005).



Şekil 5.1. $P_0^1(t)$ nin P_0 ve P_1 e göre temsili.

5.2. De Casteljau algoritması

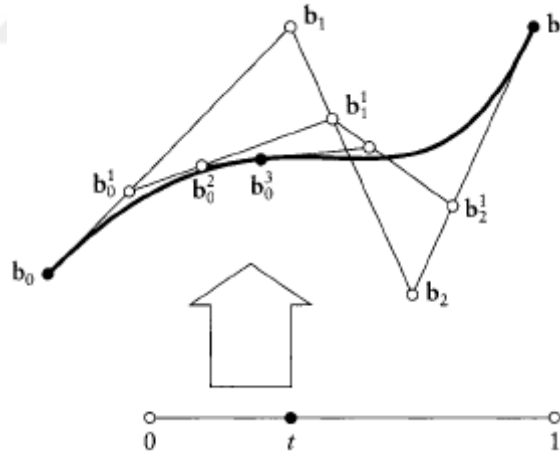
Bu algoritma eğri alanında ve yüzey tasarımı en temel olanlardan birisi olmasına rağmen sürpriz bir şekilde basittir. Ana cazibesi geometri ve cebir arasında güzel bir karşılıklı etkileşimdir. Sezgisel bir geometrik yapı, güçlü bir teoriye yol açar.

Tanım 5.2.1. $b_0, b_1, \dots, b_n \in \mathbb{E}^3$ ve $t \in \mathbb{R}$ için

$$\mathbf{b}_i^r(t) = (1-t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t) \quad \begin{cases} r = 1, \dots, n \\ i = 0, \dots, n-r \end{cases} \quad (5.4)$$

ve $\mathbf{b}_i^0(t) = \mathbf{b}_i$. O zaman $\mathbf{b}_0^n(t)$, \mathbf{b}^n Bezier eğrisi üzerinde t parametre değerli noktadır. Bu yüzden $\mathbf{b}^n(t) = \mathbf{b}_0^n(t)$ dir.

$\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ ile oluşturulmuş \mathbf{P} poligonuna \mathbf{b}^n eğrisinin Bezier poligonu veya kontrol poligonu denir (kübik durumda dört kontrol noktası vardır). Benzer olarak \mathbf{b}_i poligon köşelerine kontrol noktaları veya Bezier noktaları denir. Aşağıdaki şekil kübik durumu gösterir (Şekil 5.2).



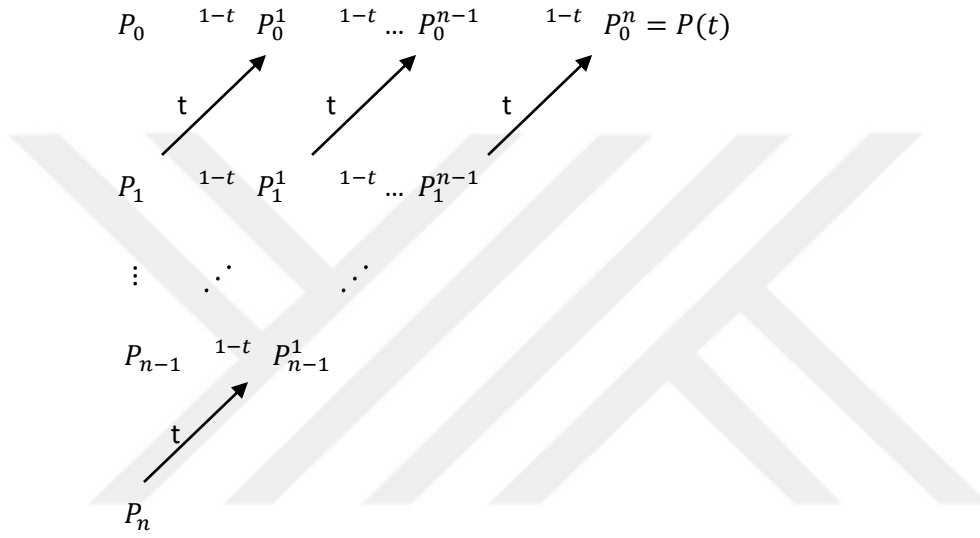
Şekil 5.2. De Casteljau algoritması : $\mathbf{b}_0^3(t)$ noktası tekrarlı lineer interpolasyon.

Bazen $\mathbf{b}^n(t) = B[b_0, b_1, \dots, b_n; t] = B[P; t]$ ile veya denk olarak daha kısa bir şekilde $\mathbf{b}^n = B[b_0, b_1, \dots, b_n] = BP$ ile de gösterebiliriz. Bu notasyon B yi, onun kontrol poligonuyla Bezier eğrisine ortak yapan lineer operatör olarak tanımlar. $B[b_0, b_1, \dots, b_n]$ eğrisinin kontrol noktasına Bernstein-Bezier yaklaşımı olduğunu

söyleyebiliriz. $b_i^r(t)$ ara katsayıları, noktaların üçgensel bir dizilimine uygun bir şekilde yazılır (Farin, 2002).

5.2.1. Bezier eğrisi için De Casteljau algoritması

Bezier eğrisi kontrol noktaları yardımıyla aşağıdaki şekilde oluşturulur (Şekil 5.3)



Şekil 5.3. Bezier eğrisi için De Casteljau algoritması.

Bu ifade Bezier eğrisi için De Casteljau algoritmasını verir. De Casteljau algoritması, bir Bezier eğrisi üzerindeki bir noktanın hesaplanması için bir metot verir. Aşağıdaki teorem De Casteljau algoritmasını ifade eder (De Casteljau, 1991).

Teorem 5.2.1.1. P_0, P_1, \dots, P_n kontrol noktalarının oluşturduğu bir Bezier eğrisi $P(t)$ olsun. Bu takdirde

$$P_i^j(t) = (1-t)P_i^{j-1}(t) + tP_{i+1}^{j-1}(t) \quad ; i = 0, \dots, n-j, j = 1, \dots, n \quad (5.5)$$

ve $P_i^0 = P_i$ olmak üzere $P(t) = P_0^n$ ($0 \leq t \leq 1$) dir (Farin, 1983; Marsh, 2005).

Bu teoremi şöyle ifade edelim; Bernstein polinomunun rekürsiyon özelliğinden De Casteljau algoritması ortaya çıkar. Böylece;

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

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

elde edilir. $B_n^{n-1}(t) = 0$ ve $B_{-1}^{n-1}(t) = 0$ olduğundan dolayı;

$$P(t) = \sum_{i=0}^{n-1} (1-t) P_i B_i^{n-1}(t) + \sum_{i=1}^n t P_i B_{i-1}^{n-1}(t) \quad (5.6)$$

olur. Eş. 5.6 eşitliğinin ikinci toplamında i yerine $i+1$ yazılarak ifade yeniden düzenlenirse

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

elde edilir. $i = 0, \dots, n-1$ için $P_i^1 = (1-t)P_i + tP_{i+1} = (1-t)P_i^0 + tP_{i+1}^0$ alınır;

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

olur. Son ifadedeki $P(t)$, P_0^1, \dots, P_{n-1}^1 kontrol noktaları ile $(n-1)$. dereceden bir Bezier eğrisini ifade eder. Bu son ifadeye benzer bir tartışma uygulanırsa;

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

olur. Burada $i = 0, \dots, n-2$ için $P_i^2 = (1-t)P_i^1 + tP_{i+1}^1$ dir. Böylece

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

olur. Burada; $i = 0, \dots, n-j$ için $P_i^j = (1-t)P_i^{j-1} + tP_{i+1}^{j-1}$ dir.

Özel olarak, $n = j$ için son eşitlik;

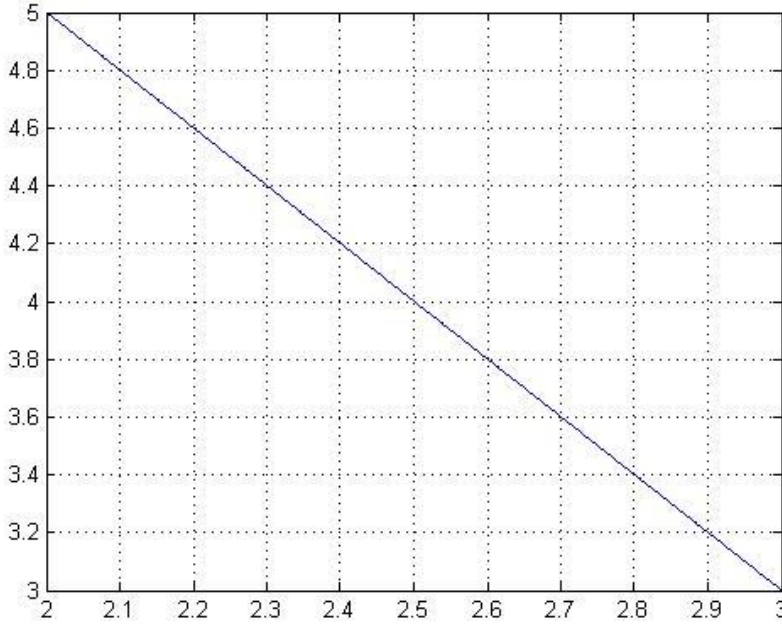
$$P(t) = \sum_{i=0}^0 P_i^0 B_i^0(t) = P_0^0$$

olur.

Örnek 5.2.1.1. E^2 de $P_0 = (3,3)$, $P_1 = (2,5)$ kontrol noktalarından geçen $\overline{P_0P_1}$ doğru parçası;

$$\begin{aligned} P(t) &= P_0^1(t) = (1-t)(3,3) + t(2,5) \quad ; t \in [0,1] \\ &= (3-t, 3+2t) \end{aligned} \quad (5.7)$$

olur (Şekil 5.4)



Şekil 5.4. $P_0 = (3,3)$, $P_1 = (2,5)$ kontrol noktalarından geçen Bezier eğrisi.

Tanım 5.2.1.1 E^n ($n = 2,3$) Öklid uzayında P_0, P_1 ve P_2 noktaları verildiğinde $t \in [0,1]$ için,

$$P_0^1(t) = (1-t)P_0 + tP_1$$

$$P_1^1(t) = (1-t)P_1 + tP_2 \quad (5.8)$$

$$P_0^2(t) = (1-t)P_0^1(t) + tP_1^1(t)$$

denklemleri yardımıyla,

$$P(t) = P_0^2(t)$$

$$= (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_2$$

$$= \sum_{i=0}^2 P_i B_i^2(t) \quad (5.9)$$

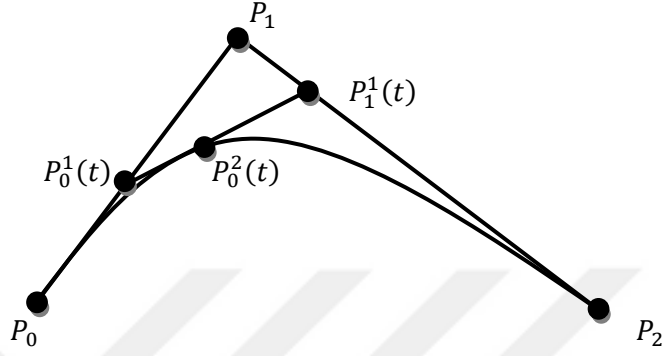
kuadratik Bezier eğrisi tanımlanır. Burada

$$B_0^2(t) = (1-t)^2$$

$$B_1^2(t) = 2(1-t)t \quad (5.10)$$

$$B_2^2(t) = t^2$$

ikinci dereceden Bernstein polinomları ile P_0 , P_1 ve P_2 kontrol noktaları olarak adlandırılır (Şekil 5.5)



Şekil 5.5. P_0 , P_1 ve P_2 kontrol noktalarının oluşturduğu kuadratik Bezier eğrisi.

P_0 , P_1 ve P_2 kontrol noktalarının öngörölmüş sırayla doğru parçaları ile birleştirilmesi ile elde edilen üçgene kontrol poligonu denir (Kaplan ve Mann, 2006; Marsh, 2005).

Tanım 5.2.1.2. E^n ($n = 2,3$) Öklid uzayında P_0, P_1, P_2 ve P_3 noktaları verildiğinde $t \in [0,1]$ için;

$$P_0^1(t) = (1-t)P_0 + tP_1$$

$$P_1^1(t) = (1-t)P_1 + tP_2$$

$$P_2^1(t) = (1-t)P_2 + tP_3$$

$$P_0^2(t) = (1-t)P_0^1(t) + tP_1^1(t) \quad (5.11)$$

$$P_1^2(t) = (1-t)P_1^1(t) + tP_2^1(t)$$

$$P_0^3(t) = (1-t)P_0^2(t) + tP_1^2(t)$$

denklemleri yardımıyla

$$P(t) = P_0^3(t)$$

$$= (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3$$

$$= \sum_{i=0}^3 P_i B_i^3(t) \quad (5.12)$$

kübik Bezier eğrisi tanımlanır. Burada;

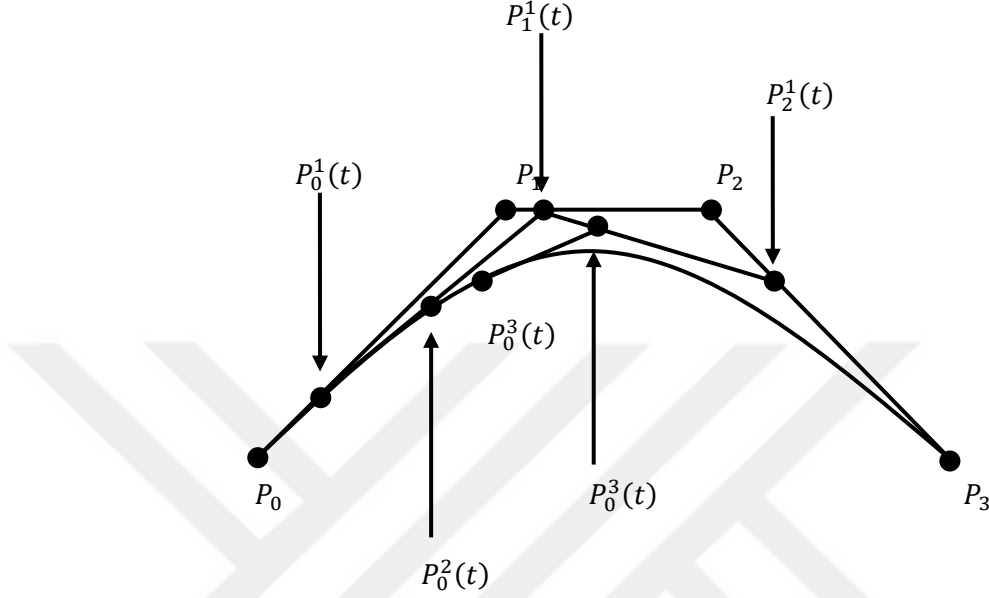
$$B_0^3(t) = (1-t)^3$$

$$B_1^3(t) = 3(1-t)^2t$$

$$B_2^3(t) = 3(1-t)t^2 \quad (5.13)$$

$$B_3^3(t) = t^3$$

üçüncü dereceden Bernstein polinomları ve P_0, P_1, P_2 ve P_3 de kontrol noktaları olarak adlandırılır (Şekil 5.6).



Şekil 5.6. P_0, P_1, P_2 ve P_3 kontrol noktalarının oluşturduğu kübik Bezier eğrisi.

P_0, P_1, P_2 ve P_3 kontrol noktalarının öngörölmüş sırayla doğru parçaları ile birleştirilmesiyle elde edilen poligona kontrol poligonu denir (Kaplan ve Mann, 2006; Marsh, 2005).

Tanım 5.2.1.3. $E^n (n = 2,3)$ Öklid uzayında $P_0, P_1, P_2, \dots, P_n$ kontrol noktaları verildiğinde her $t \in [0,1]$ için;

$$P(t) = \sum_{i=0}^n P_i B_i^n(t) \quad (5.14)$$

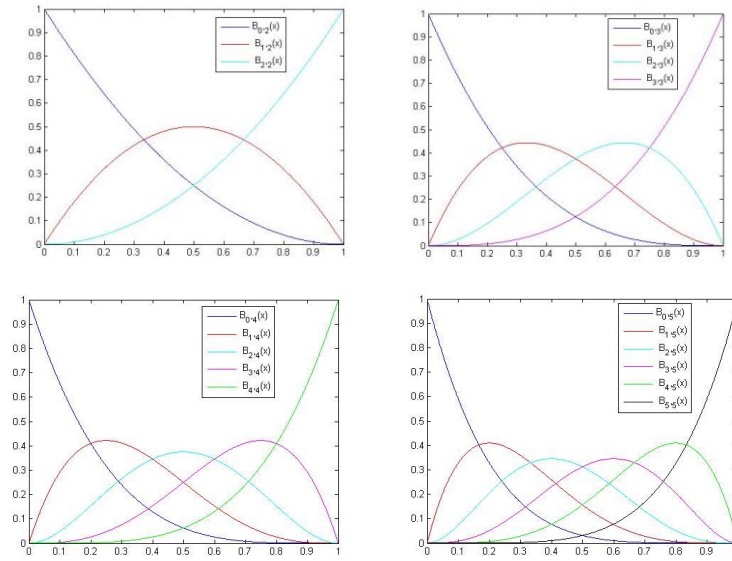
parametrik eşitliği bu kontrol noktalarının oluşturduğu Bezier eğrisi adını alır. Burada;

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i \quad ; 0 \leq i \leq n \quad (5.15)$$

n . dereceden Bernstein polinomları olarak adlandırılır.

$P_0, P_1, P_2, \dots, P_n$ kontrol noktalarının öngörölmüş sırayla oluşturulan doğru parçaları yardımıyla elde edilen poligona kontrol poligonu denir (Marsh,2005).

Şekil 5.7. de sırasıyla 2. , 3. , 4. ve 5. dereceden Bernstein polinomları verilmiştir.



Şekil 5.7. 2. , 3. , 4. ve 5. dereceden Bernstein polinomları.

Örnek 5.2.1.2. $P_0 = (0,0)$, $P_1 = (4,0)$ ve $P_2 = (4, -7)$ kontrol noktalarının oluşturduğu düzlemsel kuadratik Bezier eğrisinin parametrik denklemi Eş. 5.9 eşitliği yardımıyla

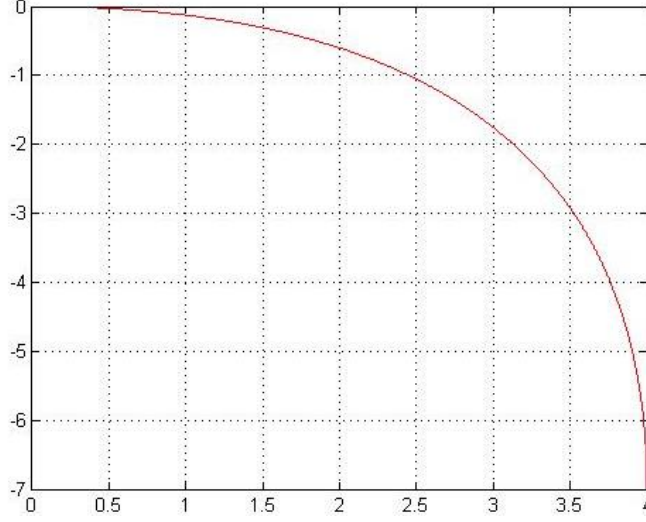
$$P(t) = \sum_{i=0}^2 P_i B_i^2(t) = (1-t)^2 P_0 + 2(1-t)t P_1 + t^2 P_2$$

$$= (1-t)^2(0,0) + 2(1-t)t(4,0) + t^2(4,-7)$$

$$= (0,0) + (4t(1-t), 0) + (4t^2, -7t^2)$$

$$= (4t, -7t^2) \quad ; t \in [0,1] \quad (5.16)$$

olur (Şekil 5.8).



Şekil 5.8. Düzlemsel kuadratik Bezier eğrisi.

Örnek 5.2.1.3. $P_0 = (0,0)$, $P_1 = (4,0)$, $P_2 = (8,-8)$ ve $P_3 = (-4,0)$ kontrol noktalarının oluşturduğu düzlemsel kübik Bezier eğrisinin parametrik denklemi Eş. 5.12 eşitliği yardımıyla

$$\begin{aligned}
 P(t) &= \sum_{i=0}^3 P_i B_i^3(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3 \\
 &= (1-t)^3(0,0) + 3(1-t)^2 t(4,0) + 3(1-t)t^2(8,-8) + t^3(-4,0) \\
 &= (0,0) + (12t(1-t)^2, 0) + (24(1-t)t^2, -24(1-t)t^2) + (-4t^3, 0) \\
 &= (12t(1-t)^2 + 24(1-t)t^2 - 4t^3, -24(1-t)t^2) \\
 &= (12t - 16t^3, 24t^3 - 24t^2) \quad ; t \in [0,1] \quad (5.17)
 \end{aligned}$$

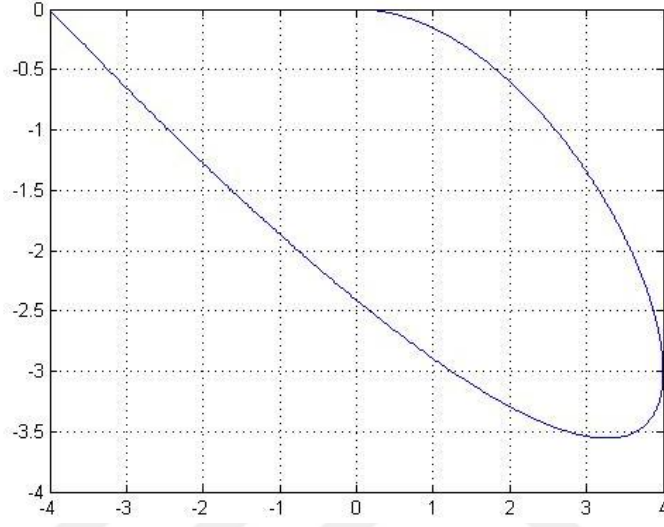
bulunur (Şekil 5.9).

Örnek 5.1.2.4. $P_0 = (0,0,0)$, $P_1 = (0,0,1)$, $P_2 = (0,1,0)$ ve $P_3 = (1,0,0)$ kontrol noktalarının oluşturduğu 3D kübik Bezier eğrisinin parametrik denklemi Eş. 5.12 eşitliği yardımıyla

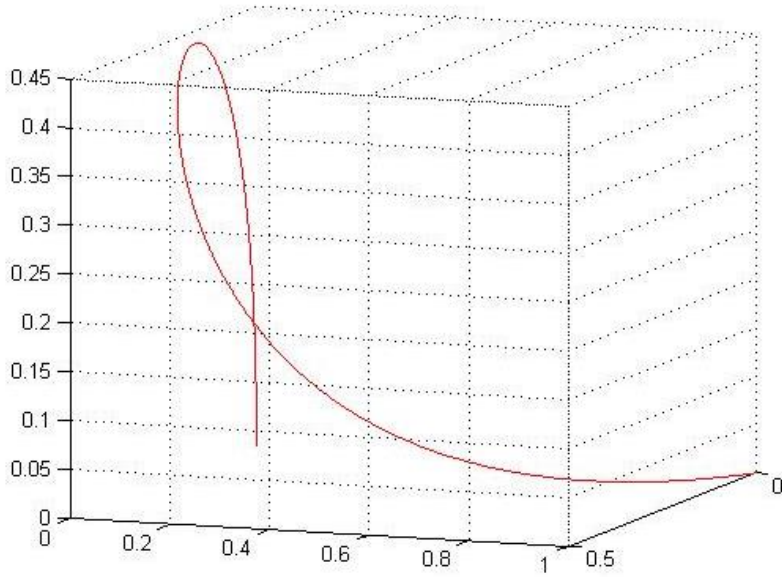
$$\begin{aligned}
 P(t) &= (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3 \\
 &= (1-t)^3(0,0,0) + 3(1-t)^2 t(0,0,1) + 3(1-t)t^2(0,1,0) + t^3(1,0,0)
 \end{aligned}$$

$$\begin{aligned}
&= (0,0,0) + (0,0,3t(1-t)^2) + (0,3t^2(1-t),0) + (t^3, 0,0) \\
&= (t^3, 3t^2 - 3t^3, 3t - 3t^3,) \quad ; t \in [0,1] \quad (5.18)
\end{aligned}$$

elde edilir (Şekil 5.10).



Şekil 5.9. Düzlemsel kübik Bezier eğrisi.



Şekil 5.10. 3-D kübik Bezier eğrisi.

5.2.2. Bezier eğrilerinin bazı özellikleri

5.2.2.1. Son nokta interpolasyon özelliği

Eş. 5.14 eşitliğiyle verilen $P(t)$ Bezier eğrisi $t=0$ için

$$P(0) = P_0$$

ve $t=1$ için

$$P(1) = P_n$$

olmaktadır (Marsh, 2005).

5.2.2.2. Son nokta teğet özelliği

Eş. 5.14 eşitliği ile verilen $P(t)$ Bezier eğrisi için

$$P'(0) = n(P_1 - P_0) \text{ ve } P'(1) = n(P_n - P_{n-1}) \text{ dir (Marsh, 2005).}$$

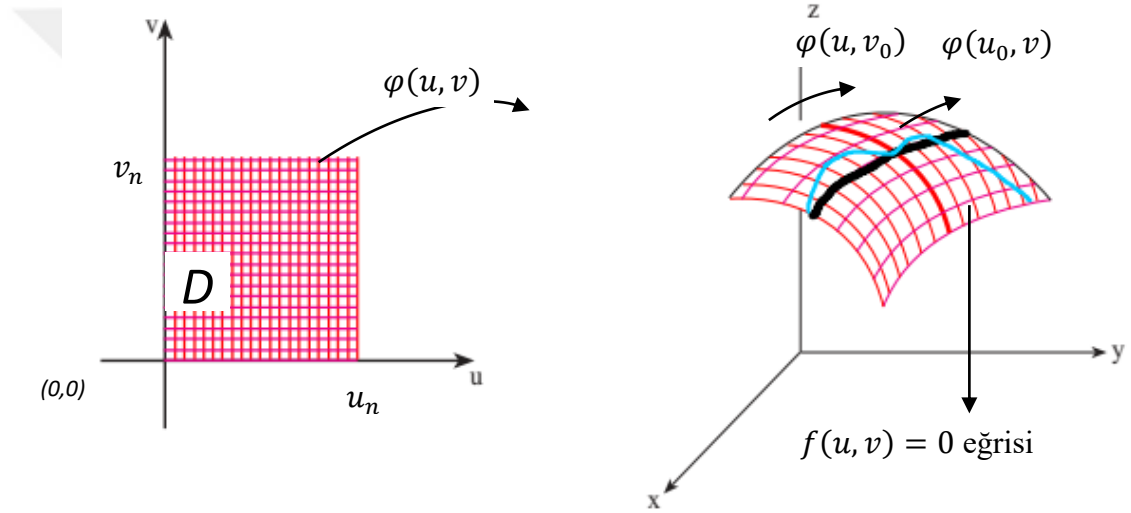


6. \mathbb{R}^3 DEKİ YÜZEY EĞRİLERİNİN BEZIER EĞRİLERİ VE MATLAB UYGULAMALARI

6.1. \mathbb{R}^3 deki yüzey eğrilerinin Bezier eğrileri

Bir yüzey üzerinde bir eğri ve eğri üzerinde de belirli sayıda kontrol noktası alınarak bu kontrol noktalarına ait Bezier eğrisi hesaplanmıştır.

\mathbb{R}^3 de bir yüzey $\varphi(u, v)$ parametrizasyonu ile verilsin. $u = u_0$ ve $v = v_0$ sabit yüzey üzerinde herhangi eğriler elde edilir (Şekil 6.1).



Şekil 6.1. $\varphi(u, v)$ yüzeyi üzerindeki parametrik eğriler.

$\varphi(u, v)$ yüzeyi üstünde bir $f(u, v) = 0$ eğrisi seçilsin. $f(u, v) = 0$ dan $u = f(v)$ eğrilerini ele alalım. $u = 0$ ve $u = u_n$ için P_0 ve P_n noktaları elde edilir. u nun bir parçalanışı $\{u_0, u_1, \dots, u_n\}$ olmak üzere P_0, P_1, \dots, P_n noktaları hesaplanabilir. P_0, P_1, \dots, P_n noktalarını kontrol noktası kabul eden Bezier eğrisi $B_z(u)$ ya da $P(u)$ ile gösterilsin. $f(u, v) = 0$ eğrisine eşlik eden $(n+1)$ kontrol noktalı Bezier eğrisi $n \rightarrow \infty$ için çakışık eğrilerdir.

Bu yöntem herhangi uzay eğrileri için de geçerlidir. Yüzey üstünde bir eğri almanın bir çok gereği vardır. Bunlardan biri yüzeyin CAGD de pano görevini üstlenmesidir. Bir diğer sebep buradaki algoritmayla bir yüzeye eşlik eden Bezier yüzeyi elde etmek için bir hazırlık oluşturmaktır.

Şimdi dik dairesel silindir, küre, saddle, torus, mobius, lemniscete yüzeyleri üstündeki eğrilere ait Bezier eğrisi örnekleri ve herhangi bir eğri üstündeki Bezier eğrisi örneği, algoritmaları ve Matlab uygulamaları verilmiştir.

Örnek 6.1.1. $\varphi(u, v) = (\cos u, \sin u, v)$ dik dairesel silindir yüzeyi üzerindeki $v = u$ bağıntısını sağlayan helis eğrisi üzerindeki Bezier eğrisini çizelim;

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

$$v = u$$

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

u -parametre eğrileridir. Bu eğriler $\varphi(u, v)$ silindir yüzeyi üzerindeki helis eğrileridir.

Dik dairesel silindir yüzeyi üzerinde bir parametre eğrisi ve bu eğri üzerinde 5 kontrol noktası alınarak bu kontrol noktalarına ait Bezier eğrisi hesaplanmıştır. 5 ve daha fazla kontrol noktasına ait Bezier eğrilerinin Matlab algoritması verilip grafikleri çizilmiştir;

* $u = 0$ için

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

* $u = \frac{\pi}{2}$ için

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

* $u = \pi$ için

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

* $u = \frac{3\pi}{2}$ için

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

* $u = 2\pi$ için

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

bulunur. n . dereceden $n + 1$ tane Bernstein polinomu $k = 0, 1, 2, \dots, n$ için

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

olup derecesi 4 olan Bernstein polinomları

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

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

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

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

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

olur. Buna göre t parametresi 0 ile 1 arasında değişirken ilk nokta (P_0) ve son nokta (P_4) arasında bir eğri çizilmektedir. Beş noktanın bir doğru üzerinde olması hali hariç (P_1, P_2, P_3) noktaları bu eğri üzerinde yer almayacaktır. Gerçekte 4. Dereceden bir Bezier eğrisinin üzerindeki noktaların koordinatları bu eğriyi tanımlamak için kullanılan beş kontrol noktanın koordinatlarının ağırlıklı bir ortalamasıdır. Kontrol nokta sayısını artırdıkça Bezier eğrisinin bu eğriye gittikçe yaklaştığını görmekteyiz.

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

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

$$P(t) = \sum_{k=0}^4 P_k \binom{4}{k} t^k (1-t)^{4-k}$$

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

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

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

olup silindir yüzeyi üzerinde bir helis eğrisi oluşturulmuş ve bu helis eğrisi üzerinde 5 kontrol noktasına ait beraber Bezier eğrisini hesaplanmıştır.

Örnek 6.1.2. $\varphi(u, v) = (\cos u \sin v, \sin u \sin v, \cos v)$ orijin merkezli 1 yarıçaplı küre yüzeyi üzerinde $v = \frac{\pi}{4}$ sabit eğrilerini bulalım.

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

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

u -parametre eğrileridir. Bu eğriler küre yüzeyi üzerindeki çember eğrileridir.

Küre yüzeyi üzerinde çember eğrisi ve bu eğri üzerinde 4 kontrol noktası alınarak, bu kontrol noktalarına ait Bezier eğrisi hesaplanmıştır. 10 kontrol noktasına ait Bezier eğrisinin Matlab algoritması verilip grafiği çizilmiştir. Şöyle ki;

$u = 0$ için,

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

$u = \frac{\pi}{2}$ için,

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

$u = \pi$ için,

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

$u = \frac{3\pi}{2}$ için,

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

olur.

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

$$P(t) = \sum_{k=0}^3 P_k \binom{3}{k} t^k (1-t)^{3-k}$$

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

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

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

Örnek 6.1.3. $\varphi(u, v) = (\sqrt{u} \cos v, \sqrt{u} \sin v, u \cos v \sin v)$ Saddle yüzeyi üzerindeki $u = 0.5v$ bağıntısını sağlayan u -parametre eğrisi üzerindeki Bezier eğrisini çizelim;

$$\varphi(v) = (\sqrt{0.5v} \cos v, \sqrt{0.5v} \sin v, 0.5v \cos v \sin v)$$

Saddle yüzeyi üzerinde bir parametre eğrisi ve bu eğri üzerinde 7 kontrol noktası olarak bu kontrol noktalarına ait Bezier eğrisi hesaplanmıştır.

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

Örnek 6.1.4. $\varphi(u, v) = ((c + a \cos v) \cos u, (c + a \cos v) \sin u, a \sin v)$ Torus yüzeyi üzerindeki

$u = 0.28v$ bağıntısını sağlayan parametre eğrisi üzerindeki Bezier eğrisini çizelim;

$$u = 0.28v;$$

$$a = 1;$$

$$c = 3;$$

$$\varphi(u, v) = ((3 + \cos u) \cos u, (3 + \cos u) \sin u, \sin u)$$

Torus yüzeyi üzerinde v - parametre eğrisi ve bu eğri üzerinde 7 kontrol noktası olarak, bu kontrol noktalarına ait Bezier eğrisi hesaplanmıştır. Bezier eğrisinin Matlab algoritması verilip grafikleri çizilmiştir.

Örnek 6.1.5. $\varphi(u, v) = (\cos u + v \cos(\frac{u}{2}) \cos u, \sin u + v \cos(\frac{u}{2}) \sin u, v \sin(\frac{u}{2}))$

Mobius yüzeyi üzerindeki $u = 5v$ bağıntısını sağlayan parametre eğrisi üzerindeki Bezier eğrisini çizelim;

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

Mobius yüzeyi üzerinde v - parametre eğrisi ve bu eğri üzerinde 5 kontrol noktası alınarak, bu kontrol noktalarına ait Bezier eğrisi hesaplanmıştır. Bezier eğrisinin Matlab algoritması verilip grafikleri çizilmiştir.

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

Örnek 6.1.6.

$$X = \cos(v) \sqrt{|\sin(2u)|} \cos u$$

$$Y = \cos(v) \sqrt{|\sin(2u)|} \sin u$$

$$Z = X^2 - Y^2 + 2XY \tan^2(v)$$

olmak üzere;

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

Lemnscete yüzeyi üzerindeki $u = v$ bağıntısını sağlayan v -parametre eğrisi üzerindeki Bezier eğrisini çizelim;

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

Lemnscete yüzeyi üzerinde bir parametre eğrisi ve bu eğri üzerinde 6 kontrol noktası olarak bu kontrol noktalarına ait Bezier eğrisi hesaplanmıştır.

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

Örnek 6.1.7. $\varphi(u, v) = (u^3, 2u - 5, u^2 + 2u)$ eğrisi üzerinde 5 kontrol noktası alınarak bu kontrol noktalarına ait Bezier eğrisi hesaplanmıştır.

6.2. Matlab uygulamaları

Örnek 6.2.1. Bir silindir yüzeyi üzerindeki helis eğrisi üzerinde beş kontrol noktasına ait Beizer eğrisinin Matlab algoritması ve grafiği aşağıdaki gibidir (Şekil 6.2, 6.3).

```
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)
%5 nokta için Bezier eğrisinin grafiği
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];

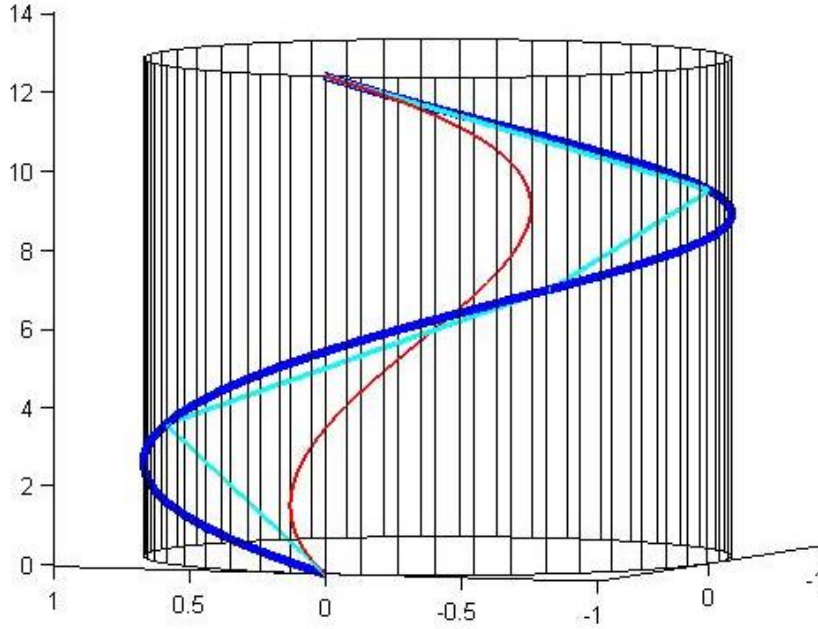
px=(1-t).^4*x(1)+4*t.*(1-t).^3*x(2)+6*t.^2.*(1-t).^2*x(3)+4*t.^3.*(1-t)*x(4)+t.^4*x(5)
py=(1-t).^4*y(1)+4*t.*(1-t).^3*y(2)+6*t.^2.*(1-t).^2*y(3)+4*t.^3.*(1-t)*y(4)+t.^4*y(5)
pz=(1-t).^4*z(1)+4*t.*(1-t).^3*z(2)+6*t.^2.*(1-t).^2*z(3)+4*t.^3.*(1-t)*z(4)+t.^4*z(5)
```



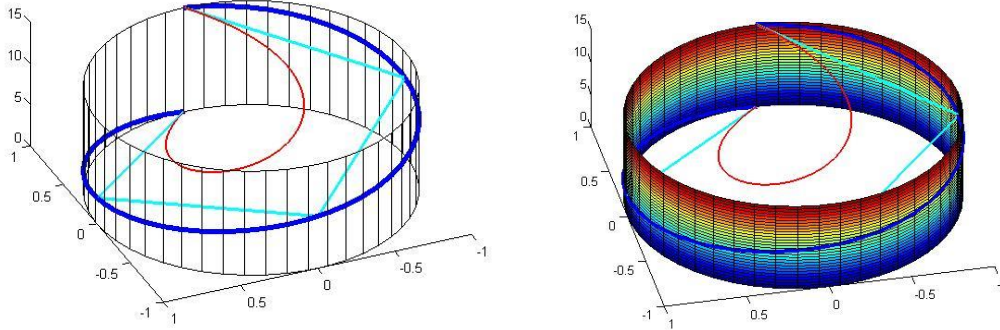
```

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

```

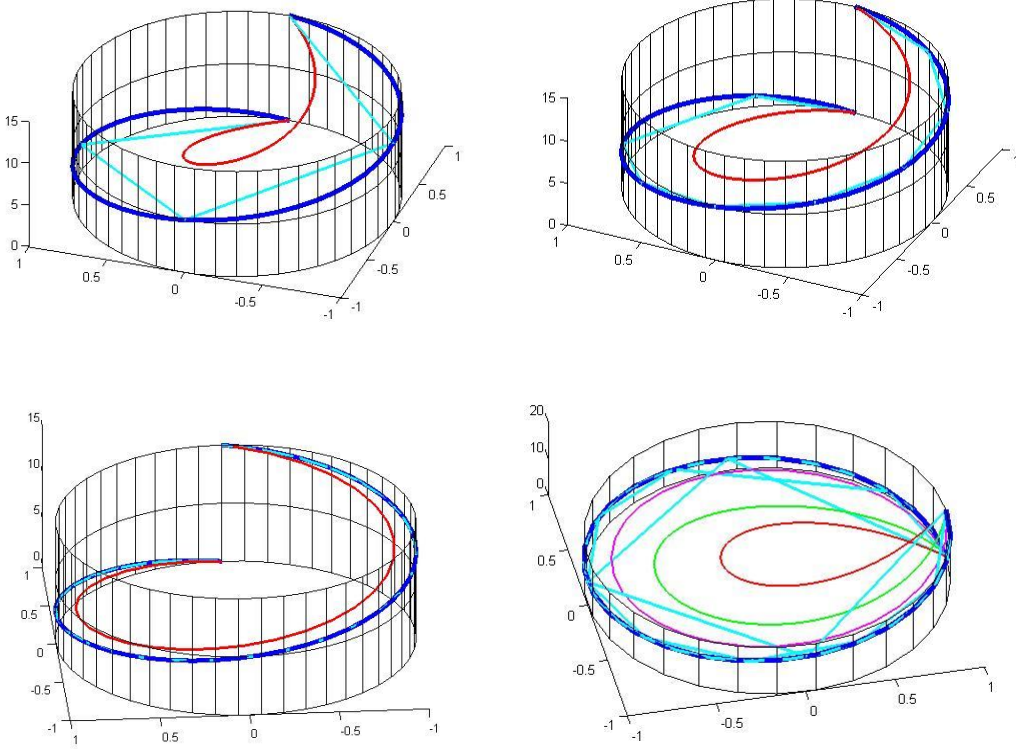


Şekil 6.2. 5 kontrol noktasına ait silindir yüzeyi üzerindeki Bezier eğrisi.



Şekil 6.3. 5 kontrol noktasına ait Bezier eğrisinin üstten görünümü.

Silindir yüzeyi üzerindeki helis eğrisi üzerinde kontrol nokta sayısını artırdıktan sonra oluşan Bezier eğrisinin Matlab grafikleri aşağıdaki gibidir (Şekil 6.4).



Şekil 6.4. Sırasıyla 5, 10 ve 30 kontrol noktalarına ait silindir yüzeyi üzerindeki Bezier eğrisi.

Örnek 6.2.2. Küre yüzeyi üzerindeki, çember eğrisi üzerinde kontrol nokta sayısını arttırdıktan sonra oluşan Bezier eğrisinin Matlab algoritması ve grafiği aşağıdaki gibidir (Şekil 6.5, 6.6).

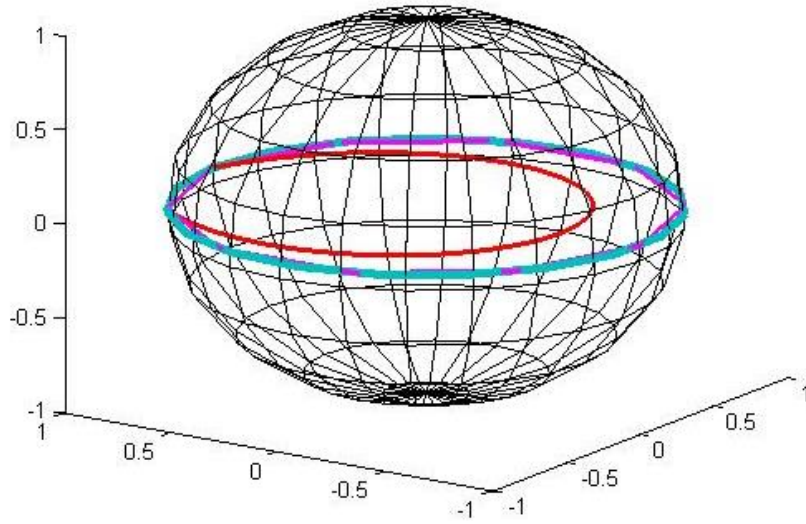
```
%Küre yüzeyi üzerindeki çember eğrisi için Bezier eğrisi
u=linspace(0,2*pi,20); v=linspace(0,2*pi,25);
[U,V]=meshgrid(u,v);
X=sin(U).*cos(V)
Y=sin(U).*sin(V)
Z=cos(U)
a=zeros(size(X))
plot3(sin(u),cos(u),a,'LineWidth',3), hold on
surf(X,Y,Z), colormap white
alpha(0.0001)

t=0:1/100:1;
x=[sin(0) sin(pi/5) sin(2*pi/5) sin(3*pi/5) sin(4*pi/5) sin(5*pi/5)
sin(6*pi/5) sin(7*pi/5) sin(8*pi/5) sin(9*pi/5) sin(10*pi/5)];
y=[cos(0) cos(pi/5) cos(2*pi/5) cos(3*pi/5) cos(4*pi/5) cos(5*pi/5)
cos(6*pi/5) cos(7*pi/5) cos(8*pi/5) cos(9*pi/5) cos(10*pi/5)];
z=[0 0 0 0 0 0 0 0 0 0 0];
```

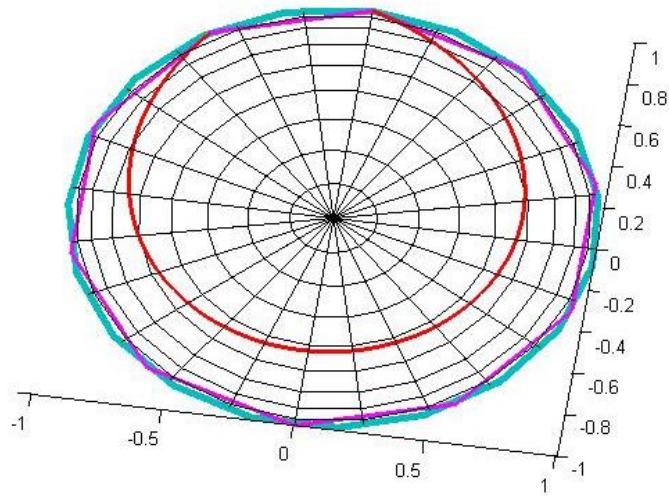
```

px=(1-t).^9*x(1)+9*t.*(1-t).^8*x(2)+36*t.^2.*(1-
t).^7*x(3)+84*t.^3.*(1-t).^6*x(4)+126*t.^4.*(1-
t).^5*x(5)+126*t.^5.*(1-t).^4*x(6)+84*t.^6.*(1-
t).^3*x(7)+36*t.^7.*(1-t).^2*x(8)+9*t.^8.*(1-t)*x(9)+t.^9*x(10)
py=(1-t).^9*y(1)+9*t.*(1-t).^8*y(2)+36*t.^2.*(1-
t).^7*y(3)+84*t.^3.*(1-t).^6*y(4)+126*t.^4.*(1-
t).^5*y(5)+126*t.^5.*(1-t).^4*y(6)+84*t.^6.*(1-
t).^3*y(7)+36*t.^7.*(1-t).^2*y(8)+9*t.^8.*(1-t)*y(9)+t.^9*y(10)
pz=(1-t).^9*z(1)+9*t.*(1-t).^8*z(2)+36*t.^2.*(1-
t).^7*z(3)+84*t.^3.*(1-t).^6*z(4)+126*t.^4.*(1-
t).^5*z(5)+126*t.^5.*(1-t).^4*z(6)+84*t.^6.*(1-
t).^3*z(7)+36*t.^7.*(1-t).^2*z(8)+9*t.^8.*(1-t)*z(9)+t.^9*z(10)
plot3(x,y,z,'m','LineWidth',2);
hold on
plot3(px,py,pz,'r','LineWidth',2);

```



Şekil 6.5. 10 kontrol noktası için küre yüzeyi üzerindeki Bezier eğrisi.



Şekil 6.6. 10 kontrol noktasına ait Bezier eğrisinin üstten görünümü.

Örnek 6.2.3. Saddle yüzeyi üzerindeki, v - parametre eğrisi üzerinde 7 kontrol nokta sayısı ile oluşturulan Beizer eğrisinin Matlab algoritması ve grafiği aşağıdaki gibidir (Şekil 6.7).

```

u=linspace(0,2*pi,20);
v=linspace(0,2*pi,20);
[u,v]=meshgrid(u,v)

x=sqrt(u).*cos(v);
y=sqrt(u).*sin(v);
z=u.*cos(v).*sin(v);
surf(x,y,z)
hold on
alpha(0.001)
u=0.5.*v;
x=sqrt(u).*cos(v);
y=sqrt(u).*sin(v);
z=u.*cos(v).*sin(v);
p1=plot3(x,y,z,'b','LineWidth',2)

t=0:0.0025:1;

x=[sqrt(0.5.*0).*cos(0) sqrt(0.5.*pi/3).*cos(pi/3)
sqrt(0.5.*2.*pi/3).*cos(2.*pi/3) sqrt(0.5.*pi).*cos(pi)
sqrt(0.5.*4.*pi/3).*cos(4.*pi/3) sqrt(0.5.*5.*pi/3).*cos(5.*pi/3)
sqrt(0.5.*2.*pi).*cos(2.*pi)];
y=[sqrt(0.5.*0).*sin(0) sqrt(0.5.*pi/3).*sin(pi/3)
sqrt(0.5.*2.*pi/3).*sin(2.*pi/3) sqrt(0.5.*pi).*sin(pi)
sqrt(0.5.*4.*pi/3).*sin(4.*pi/3) sqrt(0.5.*5.*pi/3).*sin(5.*pi/3)
sqrt(0.5.*2.*pi).*sin(2.*pi)];
z=[0.5.*0.*cos(0).*sin(0) 0.5.*pi/3.*cos(pi/3).*sin(pi/3)
0.5.*2.*pi/3.*cos(2.*pi/3).*sin(2.*pi/3) 0.5.*pi.*cos(pi).*sin(pi)
0.5.*4.*pi/3.*cos(4.*pi/3).*sin(4.*pi/3)
0.5.*5.*pi/3.*cos(5.*pi/3).*sin(5.*pi/3)
0.5.*2.*pi.*cos(2.*pi).*sin(2.*pi)];

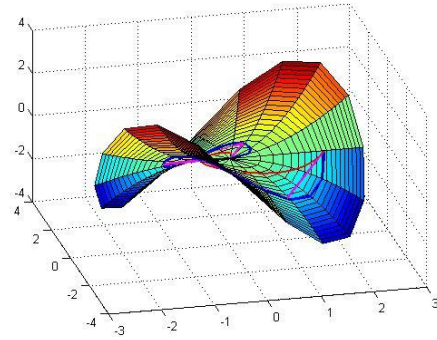
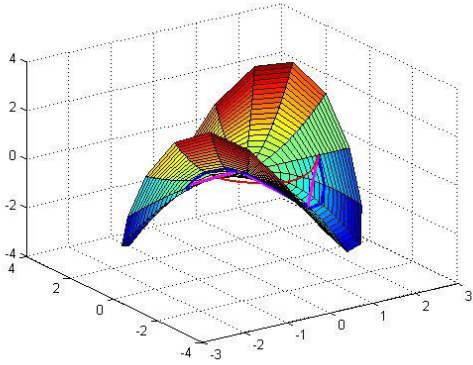
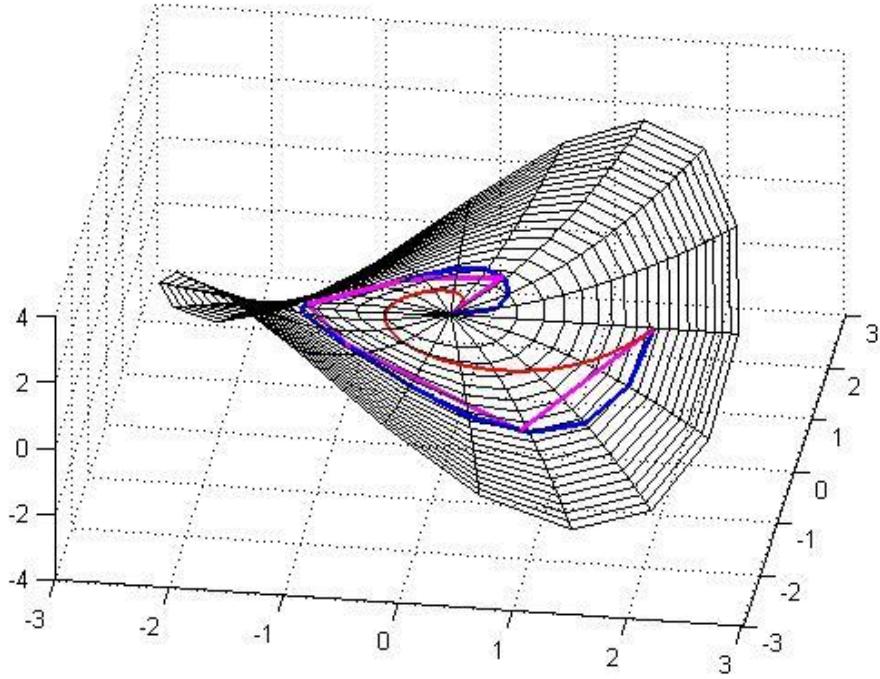
px=(1-t).^6.*x(1)+6.*t.*(1-t).^5*x(2)+15.*t.^2.*(1-
t).^4*x(3)+20.*t.^3.*(1-t).^3.*x(4)+15.*t.^4.*(1-
t).^2*x(5)+6.*t.^5.*(1-t).*x(6)+t.^6.*x(7)
py=(1-t).^6.*y(1)+6.*t.*(1-t).^5*y(2)+15.*t.^2.*(1-
t).^4*y(3)+20.*t.^3.*(1-t).^3.*y(4)+15.*t.^4.*(1-
t).^2*y(5)+6.*t.^5.*(1-t).*y(6)+t.^6.*y(7)
pz=(1-t).^6.*z(1)+6.*t.*(1-t).^5*z(2)+15.*t.^2.*(1-
t).^4*z(3)+20.*t.^3.*(1-t).^3.*z(4)+15.*t.^4.*(1-
t).^2*z(5)+6.*t.^5.*(1-t).*z(6)+t.^6.*z(7)

p2=plot3(x,y,z,'m','LineWidth',2);

hold on

p3=plot3(px,py,pz,'r','LineWidth',1.5);

```



Şekil 6.7. 7 kontrol noktası için saddle yüzeyi üzerindeki Bezier eğrisi.

Örnek 6.2.4. Torus yüzeyi üzerindeki, v -parametre eğrisi üzerinde 7 kontrol nokta sayısı ile oluşturulan Beizer eğrisinin Matlab algoritması ve grafiği aşağıdaki gibidir (Şekil 6.8).

```
a=1.;
c=3.;
u=linspace(-pi,pi,20);
v=linspace(0.,2.*pi,20);
[u,v]=meshgrid(u,v);
x=(c + a.*cos(v)).*cos(u);
y=(c + a.*cos(v)).*sin(u);
z=a.*sin(v);
surf(x,y,z);
```

```

axis equal;
%set(hsurf,'FaceColor','interp','FaceAlpha',0.5,'EdgeAlpha',0.25);
hold on

u=0.28.*v;

plot3((c + a.*cos(v)).*cos(u),(c +
a.*cos(v)).*sin(u),a.*sin(v),'LineWidth',3)
alpha(0.0001)

t=0:0.0025:1;

x=[(c + a.*cos(0)).*cos(0.28.*0) (c +
a.*cos(2.*pi/7)).*cos(0.28.*2.*pi/7) (c +
a.*cos(4.*pi/7)).*cos(0.28.*4.*pi/7) (c +
a.*cos(6.*pi/7)).*cos(0.28.*6.*pi/7) (c +
a.*cos(8.*pi/7)).*cos(0.28.*8.*pi/7) (c +
a.*cos(10.*pi/7)).*cos(0.28.*10.*pi/7) (c +
a.*cos(2.*pi)).*cos(0.28.*2.*pi)];
y=[(c + a.*cos(0)).*sin(0.28.*0) (c +
a.*cos(2.*pi/7)).*sin(0.28.*2.*pi/7) (c +
a.*cos(4.*pi/7)).*sin(0.28.*4.*pi/7) (c +
a.*cos(6.*pi/7)).*sin(0.28.*6.*pi/7) (c +
a.*cos(8.*pi/7)).*sin(0.28.*8.*pi/7) (c +
a.*cos(10.*pi/7)).*sin(0.28.*10.*pi/7) (c +
a.*cos(2.*pi)).*sin(0.28.*2.*pi)];
z=[a.*sin(0) a.*sin(2.*pi/7) a.*sin(4.*pi/7) a.*sin(6.*pi/7)
a.*sin(8.*pi/7) a.*sin(10.*pi/7) a.*sin(2.*pi)];

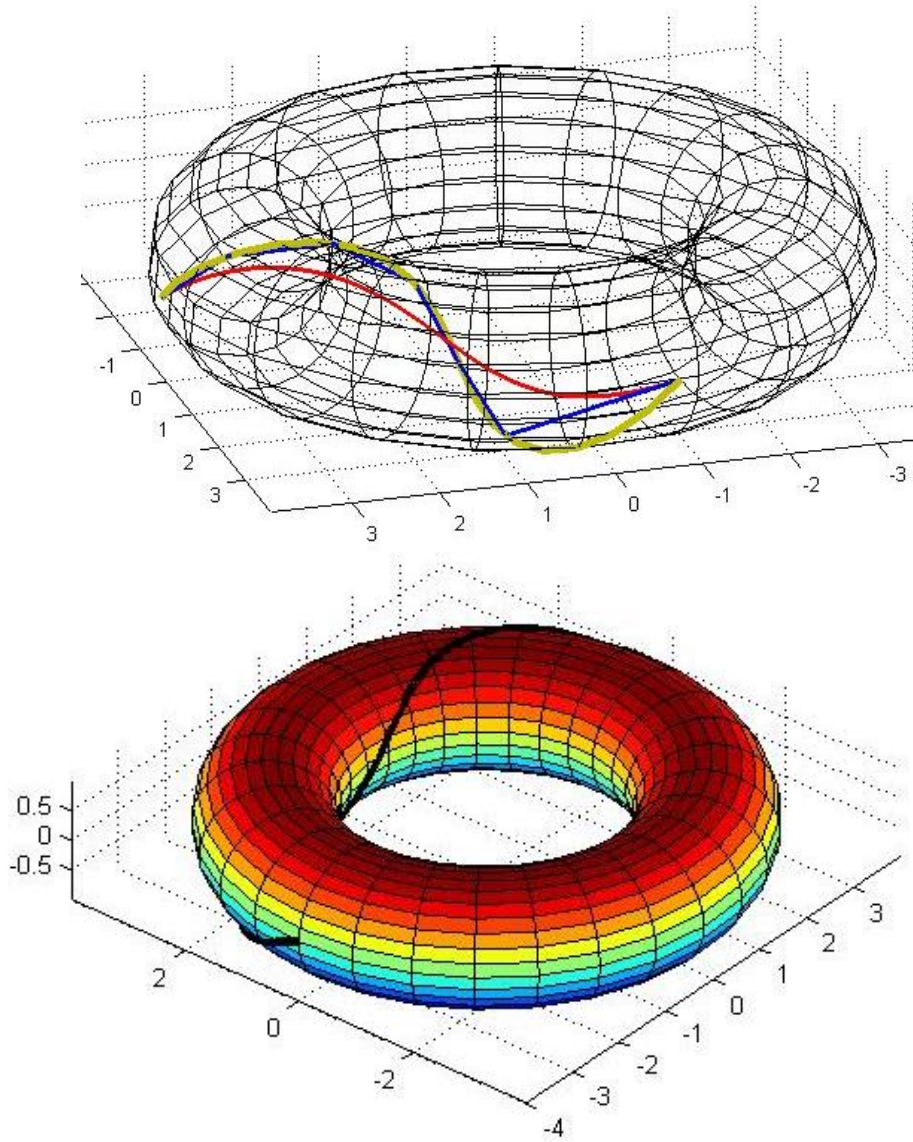
px=(1-t).^6.*x(1)+6.*t.*(1-t).^5.*x(2)+15.*t.^2.*(1-
t).^4.*x(3)+20.*t.^3.*(1-t).^3.*x(4)+15.*t.^4.*(1-
t).^2.*x(5)+6.*t.^5.*(1-t).*x(6)+t.^6.*x(7)
py=(1-t).^6.*y(1)+6.*t.*(1-t).^5.*y(2)+15.*t.^2.*(1-
t).^4.*y(3)+20.*t.^3.*(1-t).^3.*y(4)+15.*t.^4.*(1-
t).^2.*y(5)+6.*t.^5.*(1-t).*y(6)+t.^6.*y(7)
pz=(1-t).^6.*z(1)+6.*t.*(1-t).^5.*z(2)+15.*t.^2.*(1-
t).^4.*z(3)+20.*t.^3.*(1-t).^3.*z(4)+15.*t.^4.*(1-
t).^2.*z(5)+6.*t.^5.*(1-t).*z(6)+t.^6.*z(7)

plot3(x,y,z,'b','LineWidth',2)

hold on

plot3(px,py,pz,'r','LineWidth',2)

```



Şekil 6.8. 7 kontrol noktası için torus yüzeyi üzerindeki Bezier eğrisi.

Örnek 6.2.5. Mobius yüzeyi üzerindeki, v - parametre eğrisi üzerinde 5 kontrol nokta sayısı ile oluşturulan Beizer eğrisinin Matlab algoritması ve grafiği aşağıdaki gibidir (Şekil 6.9).

```
[u,v] = meshgrid(linspace(0,2*pi,50),linspace(-.4,.4,20));
surf(cos(u)+v.*cos(u/2).*cos(u),sin(u)+v.*cos(u/2).*sin(u),v.*sin(u/2))
axis equal
hold on
u=5*v;
plot3(cos(u)+v.*cos(u/2).*cos(u),sin(u)+v.*cos(u/2).*sin(u),v.*sin(u/2),'LineWidth',3), hold on
```

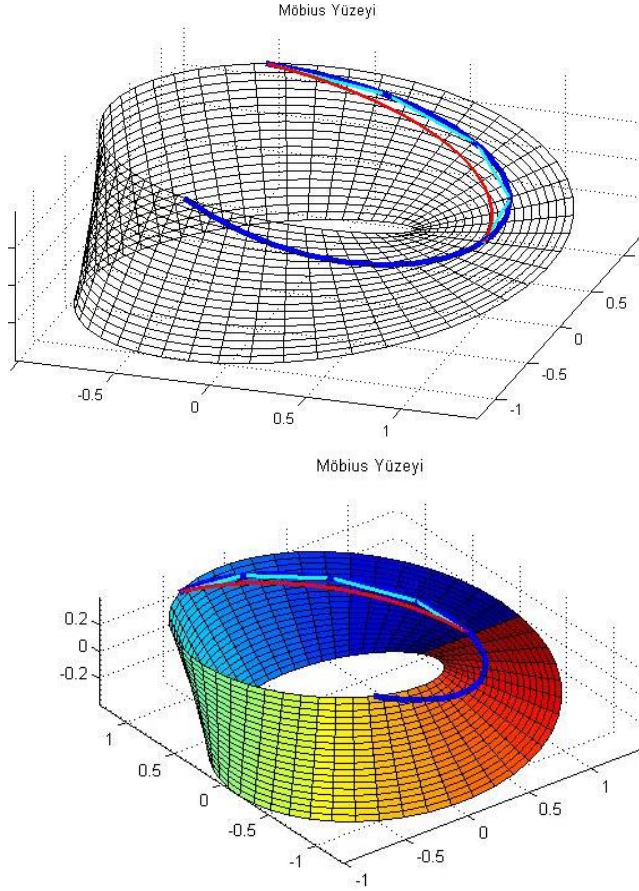
```

title('Möbius Yüzeyi')
alpha(0.001)

t=0:0.0025:1;
x=[cos(5.*0)+0.*cos(5.*0/2).*cos(5.*0)
cos(5.*(0.1))+0.1.*cos(5.*(0.1)/2).*cos(5.*(0.1))
cos(5.*(0.2))+0.2.*cos(5.*(0.2)/2).*cos(5.*(0.2))
cos(5.*(0.3))+0.3.*cos(5.*(0.3)/2).*cos(5.*(0.3))
cos(5.*(0.4))+0.4.*cos(5.*(0.4)/2).*cos(5.*(0.4))];
y=[sin(5.*0)+0.*cos(5.*0/2).*sin(5.*0)
sin(5.*(0.1))+0.1.*cos(5.*(0.1)/2).*sin(5.*(0.1))
sin(5.*(0.2))+0.2.*cos(5.*(0.2)/2).*sin(5.*(0.2))
sin(5.*(0.3))+0.3.*cos(5.*(0.3)/2).*sin(5.*(0.3))
sin(5.*(0.4))+0.4.*cos(5.*(0.4)/2).*sin(5.*(0.4))];
z=[0.*sin(5.*0/2) (0.1).*sin(5.*(0.1)/2) (0.2).*sin(5.*(0.2)/2)
(0.3).*sin(5.*(0.3)/2) (0.4).*sin(5.*(0.4)/2)];

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);
hold on plot3(px,py,pz,'r','LineWidth',2);

```



Şekil 6.9. 5 kontrol noktası için Möbius yüzeyi üzerindeki Bezier eğrisi.

Örnek 6.2.6. Bir implicit minimal lemniscate yüzeyi üzerindeki, v - parametre eğrisi üzerinde 6 kontrol noktası ile oluşturulan Beizer eğrisinin Matlab algoritması ve grafiği aşağıdaki gibidir (Şekil 6.10).

```

u=linspace(0,pi,25); v=linspace(0,pi,25);
[U,V]=meshgrid(u,v);
X=cos(V).*sqrt(abs(sin(2.*U))).*cos(U);
Y=cos(V).*sqrt(abs(sin(2.*U))).*sin(U);
Z= X.^2-Y.^2+2.*X.*Y.*(tan(V)).^2;
u=v;
plot3(cos(v).*sqrt(abs(sin(2.*u))).*cos(u),cos(v).*sqrt(abs(sin(2.*
u))).*sin(u),(cos(v).*sqrt(abs(sin(2.*u))).*cos(u)).^2-
(cos(v).*sqrt(abs(sin(2.*u))).*sin(u)).^2+2.*(cos(v).*sqrt(abs(sin(
2.*u))).*cos(u)).*(cos(v).*sqrt(abs(sin(2.*u))).*sin(u)).*(tan(v)).
^2,'LineWidth',2), hold on
surf(X,Y,Z)
%alpha(0.0001)
t=0:0.0025:1;

x=[cos(0).*sqrt(abs(sin(2.*0))).*cos(0)
cos(2.*pi/5).*sqrt(abs(sin(2.*2.*pi/5))).*cos(2.*pi/5)
cos(4.*pi/5).*sqrt(abs(sin(2.*4.*pi/5))).*cos(4.*pi/5)
cos(6.*pi/5).*sqrt(abs(sin(2.*6.*pi/5))).*cos(6.*pi/5)
cos(8.*pi/5).*sqrt(abs(sin(2.*8.*pi/5))).*cos(8.*pi/5)
cos(10.*pi/5).*sqrt(abs(sin(2.*10.*pi/5))).*cos(10.*pi/5)];
y=[cos(0).*sqrt(abs(sin(2.*0))).*sin(0)
cos(2.*pi/5).*sqrt(abs(sin(2.*2.*pi/5))).*sin(2.*pi/5)
cos(4.*pi/5).*sqrt(abs(sin(2.*4.*pi/5))).*sin(4.*pi/5)
cos(6.*pi/5).*sqrt(abs(sin(2.*6.*pi/5))).*sin(6.*pi/5)
cos(8.*pi/5).*sqrt(abs(sin(2.*8.*pi/5))).*sin(8.*pi/5)
cos(10.*pi/5).*sqrt(abs(sin(2.*10.*pi/5))).*sin(10.*pi/5)];
z=[(cos(0).*sqrt(abs(sin(2.*0))).*cos(0)).^2-
(cos(0).*sqrt(abs(sin(2.*0))).*sin(0)).^2+2.*(cos(0).*sqrt(abs(sin(
2.*0))).*cos(0)).*(cos(0).*sqrt(abs(sin(2.*0))).*sin(0)).*(tan(0)).
^2 (cos(2.*pi/5).*sqrt(abs(sin(2.*2.*pi/5))).*cos(2.*pi/5)).^2-
(cos(2.*pi/5).*sqrt(abs(sin(2.*2.*pi/5))).*sin(2.*pi/5)).^2+2.*(cos
(2.*pi/5)).* ...

sqrt(abs(sin(2.*2.*pi/5))).*cos(2.*pi/5)).*(cos(2.*pi/5).*sqrt(abs(
sin(2.*2.*pi/5))).*sin(2.*pi/5)).*(tan(2.*pi/5)).^2
(cos(4.*pi/5).*sqrt(abs(sin(2.*4.*pi/5))).*cos(4.*pi/5)).^2-
(cos(4.*pi/5).*sqrt(abs(sin(2.*4.*pi/5))).*sin(4.*pi/5)).^2+2.*(cos
(4.*pi/5).*sqrt(abs(sin(2.*4.*pi/5))).*cos(4.*pi/5)).*(cos(4.*pi/5)
.* ...
sqrt(abs(sin(2.*4.*pi/5))).*sin(4.*pi/5)).*(tan(4.*pi/5)).^2
(cos(6.*pi/5).*sqrt(abs(sin(2.*6.*pi/5))).*cos(6.*pi/5)).^2-
(cos(6.*pi/5).*sqrt(abs(sin(2.*6.*pi/5))).*sin(6.*pi/5)).^2+2.*(cos
(6.*pi/5).*sqrt(abs(sin(2.*6.*pi/5))).*cos(6.*pi/5)).*(cos(6.*pi/5)
.*sqrt(abs(sin(2.*6.*pi/5))).*sin(6.*pi/5)).* ...
(tan(6.*pi/5)).^2
(cos(8.*pi/5).*sqrt(abs(sin(2.*8.*pi/5))).*cos(8.*pi/5)).^2-
(cos(8.*pi/5).*sqrt(abs(sin(2.*8.*pi/5))).*sin(8.*pi/5)).^2+2.*(cos

```

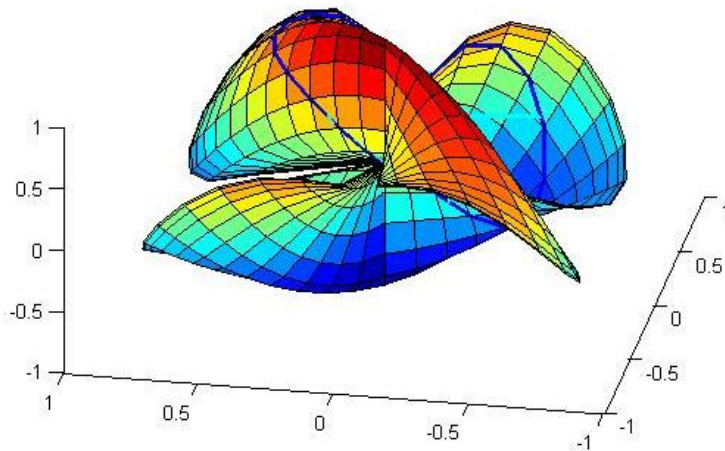
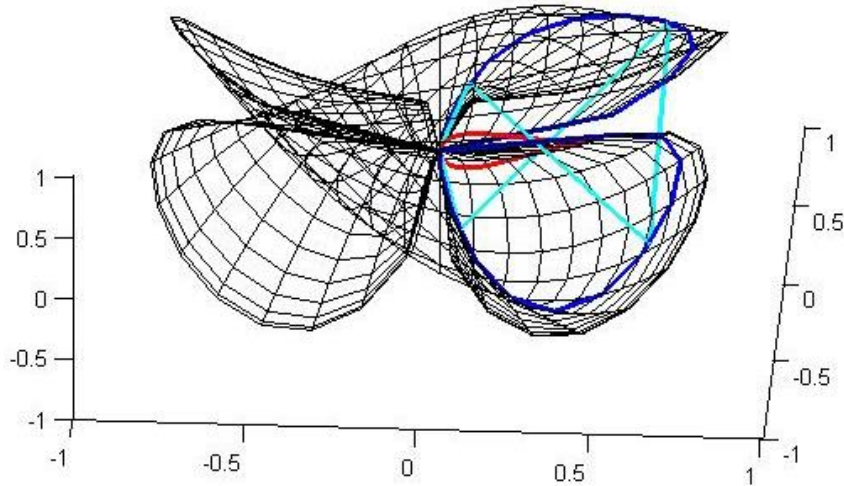
```
(8.*pi/5).*sqrt(abs(sin(2.*8.*pi/5))).*cos(8.*pi/5)).*(cos(8.*pi/5)
.*sqrt(abs(sin(2.*8.*pi/5))).*sin(8.*pi/5)).*(tan(8.*pi/5)).^2
(cos(2.*pi).*sqrt(abs(sin(2.*2.*pi))).* ...
cos(2.*pi)).^2-
(cos(2.*pi).*sqrt(abs(sin(2.*2.*pi))).*sin(2.*pi)).^2+2.*(cos(2.*pi)
).*sqrt(abs(sin(2.*2.*pi))).*cos(2.*pi)).*(cos(2.*pi).*sqrt(abs(sin
(2.*2.*pi))).*sin(2.*pi)).*(tan(2.*pi)).^2];
```

```
px=(1-t).^5.*x(1)+5.*t.*(1-t).^4.*x(2)+10.*t.^2.*(1-
t).^3.*x(3)+10.*t.^3.*(1-t).^2.*x(4)+5.*t.^4.*(1-
t).*x(5)+t.^5.*x(6)
```

```
py=(1-t).^5.*y(1)+5.*t.*(1-t).^4.*y(2)+10.*t.^2.*(1-
t).^3.*y(3)+10.*t.^3.*(1-t).^2.*y(4)+5.*t.^4.*(1-
t).*y(5)+t.^5.*y(6)
```

```
pz=(1-t).^5.*z(1)+5.*t.*(1-t).^4.*z(2)+10.*t.^2.*(1-
t).^3.*z(3)+10.*t.^3.*(1-t).^2.*z(4)+5.*t.^4.*(1-
t).*z(5)+t.^5.*z(6)
```

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



Şekil 6.10. 6 kontrol noktası için lemniscate yüzeyi üzerinde Bezier eğrisi.

Örnek 6.2.7. Bir eğri üzerinde 5 kontrol noktası olarak, bu kontrol noktalarına ait Beizer eğrisinin Matlab algoritması ve grafiği aşağıdaki gibidir (Şekil 6.11).

```

u=0:0.1:10;

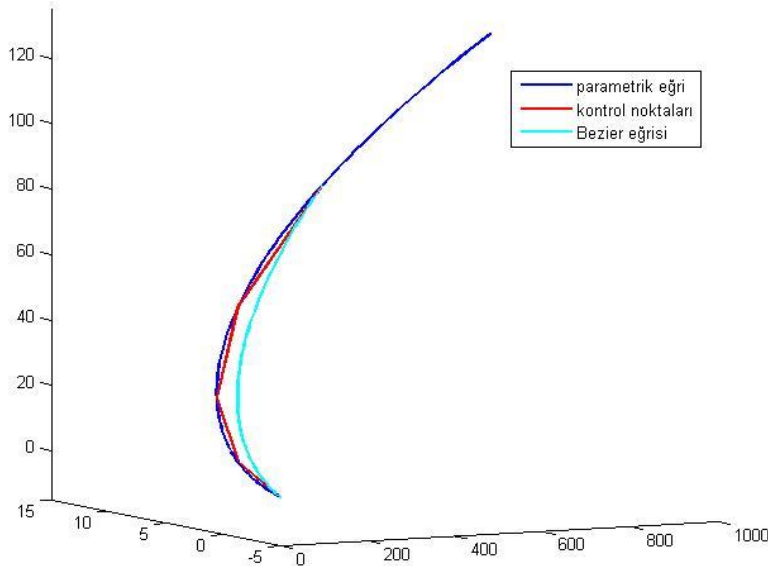
x=u.^3;
y=2.*u-5;
z=u.^2+2.*u;

plot3(x,y,z,'LineWidth',2)
hold on
%alpha(0.001)
t=0:0.01:1
x=[0.^3 2.^3 4.^3 6.^3 8.^3];
y=[2.*0-5 2.*2-5 2.*4-5 2.*6-5 2.*8-5];
z=[0.^2+2.*0 2.^2+2.*2 4.^2+2.*4 6.^2+2.*6 8.^2+2.*8];

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,'r','LineWidth',1.5)
hold on
plot3(px,py,pz,'c','LineWidth',2)
legend('parametrik eğri','kontrol noktaları','Bezier eğrisi')

```



Şekil 6.11. 5 kontrol noktası için Bezier eğrisi.



7. TARTIŞMA VE SONUÇ

Bezier eđrileri, sonlu sayıda kontrol noktası, Bernstein polinomları ve De Casteljau algoritması kullanılarak oluşturulan özel tipten eđrilerdir. Kontrol noktalarının ilki ve sonuncusu eđri üzerindedir, diđerlerinin eđri üstünde olması gerekmez.

Uzayda veya düzlemde alınan bir eđriye eşlik eden Bezier eđrisi inşası algoritması şöyledir: Eđrinin ilk noktası ve son noktası ile birlikte, eđri üzerinde sonlu sayıda nokta alınır. Bu noktalardan geçen Bezier eđrisi bilinen yöntemle elde edilir. Alınan bu noktalar, eđrinin alındığı aralığın bir parçalanmasına karşı gelen noktalardır. Aralığın parçalanma sıklığı daraltıldıkça, oluşturulan Bezier eđrisi, temel eđriye limit olarak yaklaşır. Özel olarak seçilen eđrilerin, bir yüzey üstündeki parametre veya herhangi eđriler olması bir seçimdir. Matlab uygulamalarında, yüzey ve yüzey üstündeki eđrilerin özel seçimi, teörinin görsel desteklenmesi amaçlıdır.



KAYNAKÇA

- Bezier, P., 1971. Example of an existing system in the motor industry: the Unisurf system. *Proceedings of the Royal Society of London, Series A*. 207-218
- Bezier, P., Sioussiou, S., 1983. Semi-automatic system for defining free-form curves and surfaces, *Computer Aided Design*, **15**: 65-72.
- Brickell, F., Clark, R. S., 1970. *Differentiable Manifolds: an Introduction*. Van Nostrand Reinhold., London, 289.
- De Casteljau, P., 1991. POLynomials, POLar forms, and interPOLation. In: Lyche, T., Schumaker, L.L. (Eds.), *Mathematical Methods in Computer Aided, Geometric Design II*. Academic Press, Boston. 57–68
- Farin, G., 1983. Algorithms for rational Bezier curves, *Computer Aided Design*, **15**: 73-77.
- Farin, G., 2002. *Curves and Surfaces for CAGD a Practical Guide*. Fifth edition. Academic Press, U.S.A., 521.
- Joy, K. I., 2000. On-line Geometric Modeling Notes Bernstein Polynomials. [http:// en.Wikipedia.org/wiki/Bernstein_polynomial](http://en.Wikipedia.org/wiki/Bernstein_polynomial). University of California, Erişim Tarihi: 10/05/2017.
- Kaplan, C. S., Mann, S., 2006. *Introduction to Computer Graphics*. Lecture notes CS 488/688 of University of Waterloo, 245.
- Laurent, P. J., Sablonniere, P., 2001. Pierre Beziér: An engineer and a mathematician. *Computer Aided Geometric Design*, **18**: 609-617.
- Lorentz, G. G., 1986. *Bernstein Polynomials*. Chelsea Publishing Company, New York, U.S.A., 133.
- Marsh, D., 2005. *Applied Geometry for Computer Graphics and CAD*. Springer-Verlag, London, 350.
- Rabut, C., 2002. On Pierre Beziér's life and motivations. *Computer Aided-Design*, **34**: 493-510.
- Rogers, D. F., 2002. Pierre Etienne Beziér (1910-1999), In Memoriam. *Computer Aided-Design*, **34**: 489-491.
- Steffens, K. G., 2006. *The History of Approximation Theory*. Birkhauser, Boston, U.S.A., 217.



ÖZ GEÇMİŞ

1992 yılında Van'ın Erciş ilçesinde doğdu. İlk, orta ve lise eğitimini Erciş'te tamamladı. 2011 yılında Van Yüzüncü Yıl Üniversitesi, Fen Fakültesi, Matematik Bölümüne başladı ve 2015'de mezun oldu. Aynı yıl Van Yüzüncü Yıl Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Anabilim Dalı'nda yüksek lisans eğitimine başladı.



T.C
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORJİNALLİK RAPORU

Tarih: 28/06/2018

Tez Başlığı / Konusu:

\mathbb{R}^3 deki YüzeY Eğrilerinin Bezier Eğrileri ve Matlab Uygulamaları

Yukarıda başlığı/konusu belirlenen tez çalışmamın Kapak sayfası, Giriş, Ana bölümler ve Sonuç bölümlerinden oluşan toplam 51+14 sayfalık kısmına ilişkin, 28/06/2018 tarihinde şahsım/tez danışmanım tarafından Turnitin intihal tespit programından aşağıda belirtilen filtreleme uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 13 (Onüç) tür.

Uygulanan filtreler aşağıda verilmiştir:

- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimedenden daha az örtüşme içeren metin kısımları hariç (Limit inatch size to 7 words)

Van Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.


28.06.2018

Adı Soyadı: Ceyda Yılmaz Luzum

Öğrenci No: 159102007

Anabilim Dalı: Matematik


Programı: Matematik

Statüsü: Y. Lisans

Doktora

DANIŞMAN ONAYI
UYGUNDUR

ENSTİTÜ ONAYI
UYGUNDUR


Doç. Dr. Şenay BAYDAŞ

(Unvan, Ad Soyad, İmza)