

**EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**

**(YÜKSEK LİSANS TEZİ)**

**AÇILABİLİR BEZIER YÜZEYLERİNİN  
İNCELENMESİ ve BİLGİSAYAR TASARIMI**

**Abdullah GÜNGÖZ**

Matematik Anabilim Dalı

Bilim Dalı Kodu: 403.02.01

**Sunuş Tarihi: 29.07.2008**

**Tez Danışmanı: Yrd. Doç. Dr. Bahadır TANTAY**

**Bornova – İZMİR**

**2008**

Abdullah GÜNGÖZ tarafından YÜKSEK LİSANS tezi olarak sunulan “AÇILABİLİR BEZİER YÜZEYLERİNİN İNCELENMESİ ve BİLGİSAYAR TASARIMI” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 29/07/2008 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

**Jüri Üyeleri:**

**İmza**

**Jüri Başkanı : Prof. Dr. Ali ÇALIŞKAN**

.....

**Raportör Üye: Yrd. Doç. Dr. Bahadır TANTAY**

.....

**Üye : Yrd. Doç. Dr. Aybars UĞUR**

.....

**ÖZET****AÇILABİLİR BEZİER YÜZELERİNİN İNCELENMESİ****ve BİLGİSAYAR TASARIMI**

GÜNGÖZ, Abdullah

Yüksek Lisans Tezi, Matematik Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Bahadır TANTAY

Temmuz 2008, 52 sayfa

Bu tez çalışmasının ilk bölümünde, daha sonraki bölümler için gerekli matematiksel altyapı çalışmalarına yer verilmiştir.

İkinci bölümde, Bernstein polinomları ve bunların özellikleri ayrıntılı olarak incelenmiştir. Böylece  $n$ . dereceden Bernstein polinomlarının oluşturduğu kümenin  $n$ . dereceden polinomlar uzayı için bir baz teşkil ettiği gösterilmiştir.

Üçüncü bölümde, Bézier polinom eğrilerinin tanımı yapılmış ve bazı gerekli özellikleri incelenmiştir. Sıkça sözü edilen “Bézier eğrileri için derece yükseltme” yöntemi anlatılmıştır.

Dördüncü bölümde, regle yüzeyin tanımı yapılmıştır. Ayrıca regle yüzeylerin açılabilirlik şartı araştırılıp, açılabilir regle yüzeyler tanımlanmıştır.

Beşinci bölümde, regle Bézier yüzeyleri ve açılabilirlik şartları incelenmiştir. Açılabilir Bézier yüzeyler sınıflandırılmış, her bir tür için çeşitli koşullara göre oluşan örnekler incelenerek, bilgisayar uygulaması için kolaylık sağlayacak yöntemler geliştirilmeye çalışılmıştır..

Çalışmamızın son bölümünde, bu çalışmanın uygulaması olacak şekilde üç boyutlu çizim yapan bir bilgisayar programı geliştirilmiştir.

**Anahtar sözcükler:** Bernstein polinomları, Bézier eğrileri, regle yüzeyler, açılabilir regle yüzeyler.

**ABSTRACT**

**THE INVESTIGATION and COMPUTER DESIGN OF  
DEVELOPABLE BEZIER SURFACES**

GÜNGÖZ, Abdullah

Msc. in Mathematics

Supervisor: Assistant Prof. Dr. Bahadır TANTAY

July 2008, 52 pages

In the first chapter of this thesis, the mathematical constructions needed for the other chapters are given.

In the second chapter Bernstein polynomials and their properties are studied in detail. Thus it is shown that the set of Bernstein polynomials of  $n$ th degree forms a basis for the space of  $n$ th degree polynomials.

In the third chapter Bézier polynomial curves are defined and some necessary properties are investigated. Also “Degree elevation for Bézier curves” that frequently mentioned is explained.

In the fourth chapter definition of the ruled surface is made. Also developability conditions of ruled surfaces are studied and developable surfaces are defined.

In the fifth chapter ruled Bézier surfaces and developability conditions of them are discussed. Developable Bézier surfaces are classified, via the examination of examples formed by various conditions for every class, the methods that could be helpful for computer design are developed .

In the last chapter of this thesis, a computer software that makes 3D drawings is developed as an application of this work.

**Keywords:** Bernstein polynomials, Bézier curves, ruled surfaces, developable surfaces

## TEŐEKKÜR

Bu alıőmamda deneyim, dűőünce ve destekleri ile her zaman yanımda olan danıőman hocam Yrd. Do. Dr. Bahadır TANTAY' a, bilgi ve tecrűbesini esirgemeyen deėerli hocam Prof. Dr. Ali ALIŐKAN' a, yardımları ve paylaőımları iin yűksek lisans arkadaőım Ferhat TAŐ'a yetiőmemdeki emei ve alıőmalarım sırasındaki sabrı iin anneme ayrıca uezgű bir birey olabilmemi ve bu eėitimi alabilmemi saėlayan űlkeme teőekkűrű bir bor bilirim.

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET.....</b>	<b>V</b>
<b>ABSTRACT.....</b>	<b>VII</b>
<b>TEŞEKKÜR.....</b>	<b>IX</b>
<b>ŞEKİLLER DİZİNİ.....</b>	<b>XIV</b>
<b>SİMGELER VE KISALTMALAR.....</b>	<b>XVI</b>
<b>I. GİRİŞ.....</b>	<b>1</b>
I.1 Parabol.....	2
I.2 De Casteljou Algoritması .....	3
I.3 1.mertebeden Sonlu Fark.....	5
I.4 n.mertebeden Sonlu Fark.....	5
I.5 1.mertebeden İleri Fark.....	5
I.6 n.mertebeden İleri Fark.....	5
<b>II. BERNSTEIN POLİNOMLARI.....</b>	<b>6</b>
II.1 Bernstein Polinomları.....	6
II.1.1 1.Dereceden Bernstein Polinomları.....	6
II.1.2 2.Dereceden Bernstein Polinomları.....	7
II.1.3 3.Dereceden Bernstein Polinomları.....	7

## İÇİNDEKİLER(devam)

	<u>Sayfa</u>
II.2 Bernstein Polinomlarının Bazı özellikleri.....	8
II.2.1 n. Dereceden Bir Bernstein Polinomunun (n-1). Dereceden Bernstein Polinomları Kullanılarak Yazılması.....	8
II.2.2 n.Dereceden Bir Bernstein Polinomunun, Kuvvet Bazının Lineer Birleşimi Olarak İfade Edilmesi.....	8
II.2.3 Kuvvet Bazının, n. Dereceden Bernstein Polinomlarının Lineer Birleşimi Olarak İfade Edilmesi.....	9
II.2.4 n. Dereceden Bernstein Polinomlarının Lineer Bağımsızlığı.....	10
II.2.5 n. Derecede Polinom Uzayı İçin Baz Oluşturma Özelliği.....	11
<b>III. BEZIER POLİNOM EĞRİSİ.....</b>	<b>12</b>
III.1 Bézier Polinom Eğrisi.....	12
III.2 Bézier Eğrileri ile Tasarım.....	14
III.3 Bézier Eğrileri İçin Derece Yükseltme.....	14
<b>IV. REGLE YÜZEYLER.....</b>	<b>17</b>
IV.1 Regle Yüzey.....	17
IV.2 Açılabilir Yüzey.....	18
<b>V. REGLE BEZIER YÜZEYLERİ ve AÇILABİLİRLİK ŞARTLARI.....</b>	<b>20</b>
V.1 Silindir Olma Durumu.....	21
V.2 Koni veya Bir Uzay Eğrisinin Teğetleri Tarafından Oluşturulan Teğet yüzeyi Olma Durumu.....	23

**İÇİNDEKİLER(devam)**

	<b><u>Sayfa</u></b>
V.2.1 Koni Olma Durumu.....	23
V.2.2 Bir Uzay Eğrisinin Teğetleri Tarafından Oluşturulan Teğet Yüzeyi Olma Durumu.....	27
<b>VI.BİLGİSAYAR UYGULAMASI.....</b>	<b>33</b>
<b>VII.SONUÇ.....</b>	<b>50</b>
<b>YARARLANILAN KAYNAKLAR.....</b>	<b>51</b>
<b>ÖZGEÇMİŞ.....</b>	<b>52</b>



## ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 1.1 Tekrarlı lineer interpolasyon ile parabollerin oluşturulması.....	2
Şekil 1.2. Eğri üzerindeki üç noktadaki teğetlerin kesişim noktaları.....	3
Şekil 1.3. de Casteljau algoritması.....	4
Şekil 2.1. $0 \leq t \leq 1$ için 1. dereceden Bernstein polinomları.....	6
Şekil 2.2. $0 \leq t \leq 1$ için 2. dereceden Bernstein polinomları.....	7
Şekil 2.3. $0 \leq t \leq 1$ için 3. dereceden Bernstein polinomları.....	7
Şekil 3.1. Bézier eğri örnekleri.....	13
Şekil 3.2. Bazı Bézier eğri örnekleri.....	14
Şekil 3.3. 4. dereceden 5. dereceye yükseltilen Bézier eğrisi.....	16
Şekil 3.4. Çeşitli derecelere yükseltilmiş Bézier eğrisi.....	17
Şekil 4.1. Açılabilirlik şartlarını sağlayan vektörlerin durumu.....	18
Şekil 5.1. Bézier silindirleri.....	23
Şekil 5.2. Bézier konileri.....	25
Şekil 5.3. Bézier konileri.....	27
Şekil 5.4. Bézier teğet yüzeyi.....	29

**ŞEKİLLER DİZİNİ (Devam)**

<b><u>Sekil</u></b>	<b><u>Sayfa</u></b>
Şekil 5.5. Bézier teğet yüzeyi.....	30
Şekil 5.6. Bézier teğet yüzeyi.....	31
Şekil 5.7. Bézier teğet yüzeyleri.....	33
Şekil 6.1. Kullanıcı arayüzü ve kontrollerin isimleri.....	34

## SİMGELER VE KISALTMALAR

<u>Simgeler</u>	<u>Açıklama</u>
$b_i, p_i, q_i$	kontrol noktaları
$c_i$	katsayılar
Oran( $a_t, c, b_t$ )	$c$ noktasının $a_t$ ve $b_t$ noktalarına göre oranı
$b_r^i$	de Casteljau algoritmasındaki ara nokta
$\Delta$	fark operatörü
$\Delta^r$	tekrarlı ileri farklar operatörü
$B_i^n$	Bernstein polinomu
$y_u$	$y$ nin $u$ ya göre türevi
$y_v$	$y$ nin $v$ ya göre türevi
$\alpha'$	$\alpha$ nın türevi
$\ v\ $	$v$ vektörünün uzunluğu
$u \parallel v$	$u$ ve $v$ vektörleri paralel

## I. GİRİŞ

Bir doğrunun, bir uzay eğrisi boyunca (dayanarak eğrisi) yaptığı sürekli hareket sonucu oluşan yüzeye regle yüzey denir. Bu doğruya ana doğru veya doğuran doğru denir. Bir doğuran boyunca her noktada aynı teğet düzlemine sahip olan regle yüzeye açılabilir (regle) yüzey denir. Açılabilir yüzey ayrıca bir düzlem ailesinin zarfı olarak da tarif edilebilir.

Bir yüzeyle izometrik olarak eşlenebilen sadece bir açılabilir yüzey vardır. Bu nedenle imalatlarında kullanılan üretim süreçlerinin basitliğinden dolayı açılabilir yüzeyler ilginç ve kullanışlıdır. Düz bir yüzeyden hiçbir esnetme veya büzdürme yapmadan sadece eğilip, bükülerek ve yuvarlanarak elde edilirler. Bütün bu nedenlerden dolayı, esnetilmesi uygun olmayan maddelerden yapılan gemi gövdesi, boru, ayakkabı ve otomobil parçalarının yapımında geniş bir kullanım alanı bulunmaktadır. Bu suretle açılabilir yüzeyler CAD/CAM sistemlerinde de sıkça kullanılmaktadır. Bu yüzeylerin endüstride kullanımını anlatan birçok makale mevcuttur.

Bézier eğrileri ise kontrol edilebilir olması nedeniyle tasarımda çok büyük kolaylık sağlar. Öyleki şuan kullanılan ürünlerin hemen hemen hepsinin tasarımında Bézier eğrileri kullanılabilir. Bézier eğrileri kullanılarak oluşturulan regle yüzeyler ise etkileşimli olmasından dolayı tasarımda, üretimdeki kolaylığından dolayı imalatta, geniş bir kullanım alanı bulurlar.

## I.1. Paraboller

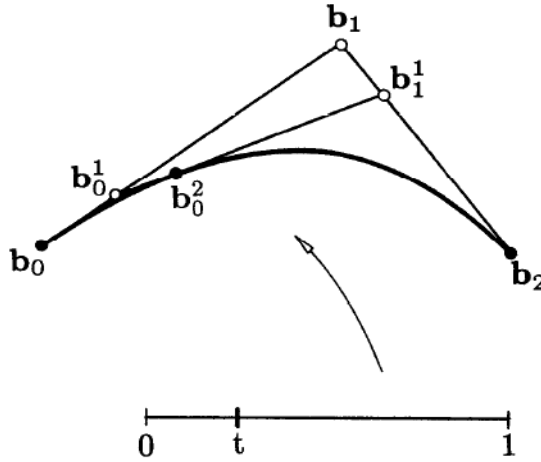
En basit eğriler olan paraboller eğri tasarımımda sıkça kullandığımız yapılardır İkinci derecen polinom eğrilerine parabol denir.

$\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2 \in E^3$  ve  $t \in \mathbb{R}$  olsun.

$t \in I = [0,1]$  için  $\mathbf{b}^2(t) \mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$  üçgeninin içindedir.

$Oran(\mathbf{b}_0, \mathbf{b}_0^1, \mathbf{b}_1) = Oran(\mathbf{b}_1, \mathbf{b}_1^1, \mathbf{b}_2) = Oran(\mathbf{b}_0^1, \mathbf{b}_0^2, \mathbf{b}_1^1) = t/(1-t)$

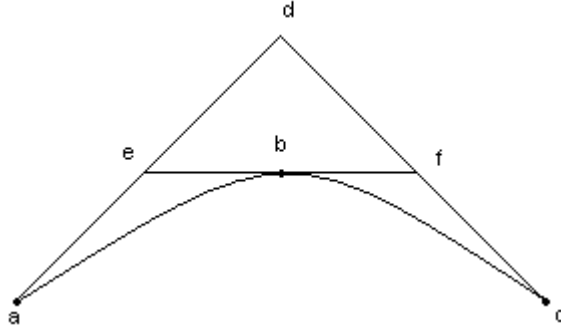
dir.



Şekil 1.1. Tekrarlı lineer interpolasyon ile parabollerin oluşturulması.

Böylece yapımız afin özelliğini sağlar çünkü parçalı lineer interpolasyon afin özelliğidir.  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  paraboldeki üç farklı nokta olsun.  $\mathbf{b}$  deki teğet  $\mathbf{a}$  ve  $\mathbf{c}$  teğetlerle ,sırasıyla  $\mathbf{e}$  ve  $\mathbf{f}$  noktalarında kesişsin.  $\mathbf{a}$  daki ve  $\mathbf{c}$  deki teğetler ise  $\mathbf{d}$  de kesişsin, (Bkz. Şekil1.2.)

O halde  $Oran(\mathbf{a}, \mathbf{e}, \mathbf{d}) = Oran(\mathbf{e}, \mathbf{b}, \mathbf{f}) = Oran(\mathbf{d}, \mathbf{f}, \mathbf{c})$  olur. Bu üç teğet teoremi parabolün bir özelliğini tanımlar (Farin, 1990).



Şekil 1.2. Eğri üzerindeki üç noktadaki teğetlerin kesişim noktaları.

## I.2. De Casteljau Algoritması

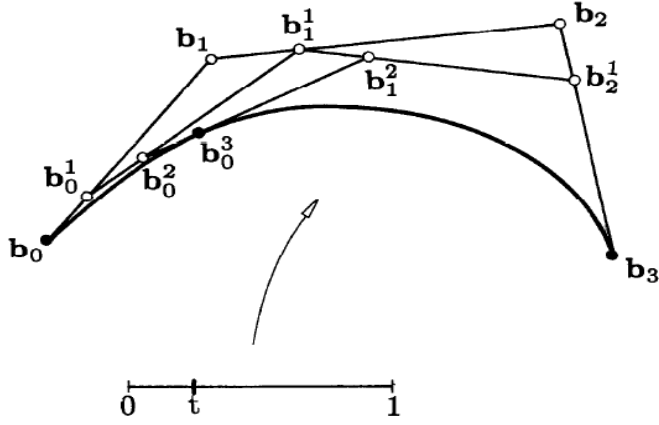
Paraboller düzlemsel eğrilerdir. Bir çok uygulama, gerçek uzay eğrileri gerektirdiğinden, bir parabolün daha önceki yapısı keyfi  $n$  dereceden bir polinom eğri üretimine genelleştirilebilir.

Verilenler:  $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n \in E^3$  ve  $t \in \mathbb{R}$ ,

$$\mathbf{b}_i^r(t) = (1-t)\mathbf{b}_i^{r-1}(t) + t\mathbf{b}_{i+1}^{r-1}(t) \quad \begin{cases} r = 1, \dots, n \\ i = 0, \dots, n-r \end{cases} \quad (1.1)$$

$$\mathbf{b}_i^0(t) = \mathbf{b}_i$$

O halde  $\mathbf{b}_0^n(t)$ ,  $\mathbf{b}^n$  Bézier eğrisinde  $t$  değerli noktadır.  $P$  çokgeni  $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$  den elde edilir ve buna  $\mathbf{b}^n$  in Bézier çokgeni veya kontrol çokgeni denir (Farin, 1990). Benzer olarak  $b_i$  ler kontrol noktalarıdır. Kısaca  $\mathbf{b}^n(t) = B[\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n; t] = B[P; t]$  veya daha kısa olarak  $B[\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n] = BP$  diyebiliriz. Bu notasyonla  $B$  ye bir lineer operatör diyebiliriz.  $B[\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n]$  eğrisi, kontrol çokgenine Bernstein-Bézier yaklaşımıdır.



Şekil 1.3. de Casteljau algoritması.

$b_i^r(t)$  aradeğer katsayıları noktaların bir üçgensel dizisi veya de Casteljau şeması olarak yazılabilir.

$$\begin{array}{l}
 b_0 \\
 b_1 \ b_0^1 \\
 b_2 \ b_1^1 \ b_0^2 \\
 b_3 \ b_2^1 \ b_1^2 \ b_0^3
 \end{array}$$

Örneğin, bir düzlemsel kübik polinomda  $t = 1/2$  için,

$$\begin{array}{l}
 \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
 \begin{bmatrix} 8 \\ 2 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3/2 \end{bmatrix} \\
 \begin{bmatrix} 4 \\ 0 \end{bmatrix} \begin{bmatrix} 6 \\ 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3/2 \end{bmatrix} \begin{bmatrix} 7/2 \\ 3/2 \end{bmatrix}
 \end{array}$$

dir (Farin, 1990).

### I.3. 1.Mertebeden Sonlu Fark

$P(t)$  eğrisi için,

$$\Delta P(t) = P(t + \delta) - P(t) \quad (1.2)$$

ifadesine **1. Mertebeden sonlu fark**, denir.

### I.4. n.Mertebeden Sonlu Fark

$$\Delta^n P(t) = \Delta^{n-1} P(t + \delta) - \Delta^{n-1} P(t) \quad (1.3)$$

ifadesine, **n. mertebeden sonlu fark**, denir.

### I.5. 1.Mertebeden İleri Fark

$x_0, x_1, x_2, \dots$  değerlerine karşılık gelen değerler sırasıyla  $y_0, y_1, y_2, \dots$  olmak üzere,

$$\begin{aligned} \Delta y_0 &= y_1 - y_0 \\ \Delta y_1 &= y_2 - y_1 \\ &\vdots \\ \Delta y_{r+1} &= y_r - \Delta y_r \end{aligned} \quad (1.4)$$

ifadesine, **1. mertebeden İleri Fark**, denir.

### I.6. n.Mertebeden İleri Fark

$x_0, x_1, x_2, \dots$  değerlerine karşılık gelen değerler sırasıyla  $y_0, y_1, y_2, \dots$  olmak üzere,

$$\begin{aligned} \Delta^2 y_r &= \Delta y_{r+1} - \Delta y_r \\ \Delta^3 y_r &= \Delta^2 y_{r+1} - \Delta^2 y_r \\ &\vdots \\ \Delta^n y_r &= \Delta^{n-1} y_{r+1} - \Delta^{n-1} y_r \end{aligned} \quad (1.5)$$

ifadesine, **n. mertebeden ileri fark**, denir.



## II. BERNSTEIN POLİNOMLARI

### II.1. Bernstein Polinomları

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & 0 \leq i \leq n \\ 0 & \text{aksi halde} \end{cases}$$

olmak üzere

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad (2.1)$$

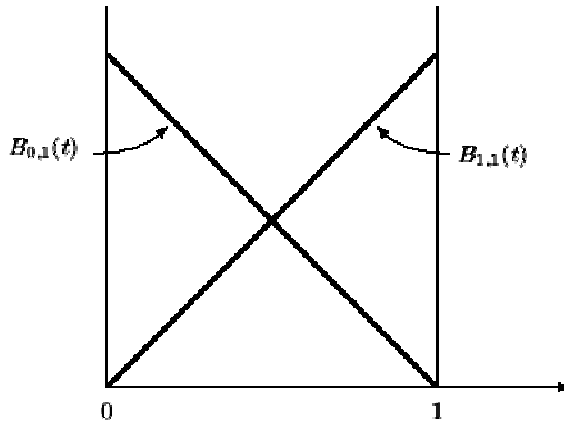
şeklinde tanımlanan polinomlara n. dereceden **Bernstein polinomu**, denir, (Biran, 1970; Joy, 2000).

#### II.1.1. 1.dereceden Bernstein polinomları

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

$$B_1^1(t) = t$$

şeklinde (Biran, 1970).



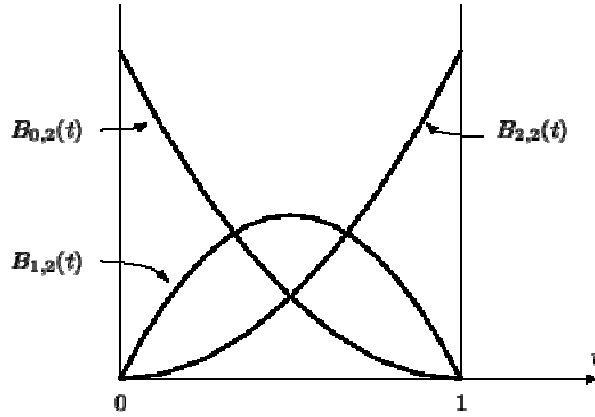
Şekil 2.1.  $0 \leq t \leq 1$  için 1. dereceden Bernstein polinomları.

### II.1.2. 2. dereceden Bernstein polinomları.

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

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

$$B_2^2(t) = t^2 \text{ şeklindedir (Biran, 1970).}$$

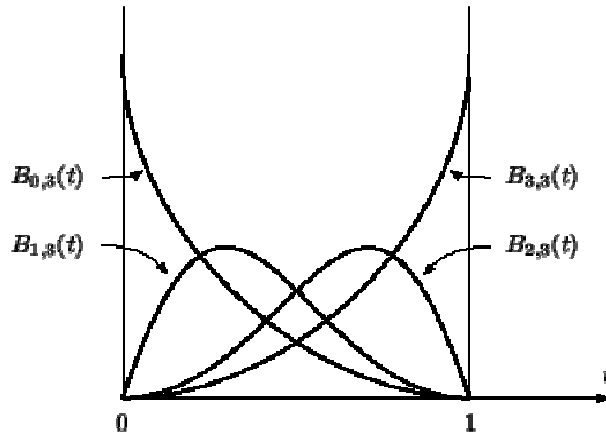


Şekil 2.2.  $0 \leq t \leq 1$  için 2. dereceden Bernstein polinomları.

### II.1.3. 3. dereceden Bernstein polinomları.

$$B_0^3(t) = (1-t)^3, \quad B_1^3(t) = 3t(1-t)^2$$

$$B_2^3(t) = 3t^2(1-t), \quad B_3^3(t) = t^3 \text{ şeklindedir (Biran, 1970).}$$



Şekil 2.3.  $0 \leq t \leq 1$  için 3. dereceden Bernstein polinomları.

## II.2. Bernstein Polinomlarının Özellikleri

### II.2.1. n. Dereceden Bir Bernstein Polinomunun (n-1). Dereceden Bernstein Polinomları Kullanılarak Yazılması

n. dereceden Bernstein polinomları (n-1). dereceden iki Bernstein polinomunun karılması ile oluşturulabilir. Yani, n. dereceden k. Bernstein polinomu;

$$B_k^n(t) = (1-t)B_k^{n-1}(t) + tB_{k-1}^{n-1}(t) \quad (2.2)$$

şeklinde ifade edilebilir.

Bunun doğruluğunu

$$\begin{aligned} (1-t)B_k^{n-1}(t) + tB_{k-1}^{n-1}(t) &= (1-t) \binom{n-1}{k} t^k (1-t)^{n-1-k} + t \binom{n-1}{k-1} t^{k-1} (1-t)^{n-1-(k-1)} \\ &= \binom{n-1}{k} t^k (1-t)^{n-k} + \binom{n-1}{k-1} t^k (1-t)^{n-k} \\ &= \left[ \binom{n-1}{k} + \binom{n-1}{k-1} \right] t^k (1-t)^{n-k} \\ &= \binom{n}{k} t^k (1-t)^{n-k} \\ &= B_k^n(t) \end{aligned}$$

şeklinde gösterebiliriz (Joy, 1990).

### II.2.2. n.Dereceden Bir Bernstein Polinomunun, Kuvvet Bazının Lineer Birleşimi Olarak İfade Edilmesi

n. dereceden herhangi bir Bernstein polinomunu, n. dereceden veya daha düşük derecelerden polinomların uzayı için bir baz olan  $\{1, t, t^2, t^3, \dots, t^n\}$  kuvvet bazının lineer birleşimi olarak ifade edilebilir.

Bu özellik,

$$\begin{aligned}
B_k^n(t) &= \binom{n}{k} t^k (1-t)^{n-k} \\
&= \binom{n}{k} t^k \sum_{i=0}^{n-k} (-1)^i \binom{n-k}{i} t^i \\
&= \sum_{i=0}^{n-k} (-1)^i \binom{n}{k} \binom{n-k}{i} t^{i+k} \\
&= \sum_{i=k}^n (-1)^{i-k} \binom{n}{k} \binom{n-k}{i} t^i \\
&= \sum_{i=k}^n (-1)^{i-k} \binom{n}{i} \binom{i}{k} t^i
\end{aligned} \tag{2.3}$$

şeklinde ispatlanabilir (Joy, 1990).

### II.2.3 Kuvvet Bazının, n. Dereceden Bernstein Polinomlarının Lineer Birleşimi Olarak İfade Edilmesi

Her bir kuvvet bazı Bernstein polinomlarının lineer birleşimi olarak ifade edilebilir (Joy, 1990). Bunun doğruluğu

$$\begin{aligned}
t^k = t(t^{k-1}) &= t \sum_{i=k-1}^n \frac{\binom{i}{k-1}}{\binom{n}{k-1}} B_i^{n-1}(t) \\
&= \sum_{i=k}^n \frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} t B_{i-1}^{n-1}(t) \\
&= \sum_{i=k-1}^n \frac{\binom{i}{k-1}}{\binom{n}{k-1}} \frac{i}{n} B_i^n(t) \\
&= \sum_{i=k-1}^{n-1} \frac{\binom{i}{k}}{\binom{n}{k}} B_i^n(t)
\end{aligned} \tag{2.4}$$

şeklinde ispatlanır (Joy, 1990).

### II.2.4. n. Dereceden Bernstein Polinomlarının Lineer Bağımsızlığı

n. dereceden Bernstein polinomları ile oluşan küme lineer bağımsızdır.

Yani, her t değeri için

$$0 = c_0 B_0^n(t) + c_1 B_1^n(t) + \dots + c_n B_n^n(t)$$

eşitliği  $c_i$  ler hepsi birlikte sıfır olduğu takdirde sağlanıyor ise Bernstein polinomları lineer bağımsızdır.

$$\begin{aligned} 0 &= c_0 B_0^n(t) + c_1 B_1^n(t) + \dots + c_n B_n^n(t) \\ &= c_0 \sum_{i=0}^n (-1)^i \binom{n}{i} \binom{i}{0} t^i + c_1 \sum_{i=1}^n (-1)^{i-1} \binom{n}{i} \binom{i}{1} t^i + \dots + c_n \sum_{i=n}^n (-1)^{i-n} \binom{n}{i} \binom{i}{n} t^i \\ &= c_0 + \left[ \sum_{i=0}^1 c_i \binom{n}{1} \binom{1}{1} \right] t^i + \dots + \left[ \sum_{i=0}^1 c_i \binom{n}{n} \binom{n}{n} \right] t^n \end{aligned}$$

kuvvet bazı lineer bağımsız bir küme olduğundan;

$$\begin{aligned} c_0 &= 0 \\ \sum_{i=0}^1 c_i \binom{n}{1} \binom{1}{1} &= 0 \\ &\vdots \\ \sum_{i=0}^n c_i \binom{n}{n} \binom{n}{n} &= 0 \end{aligned}$$

$c_0 = 0$  olduğundan bunu ilk denklemde yerine yazarsak  $c_1 = 0$ , bunları ikinci denklemde yerine koyarsak  $c_2 = 0$  ve bu şekilde devam edersek  $c_0 = c_1 = \dots = c_n = 0$  olur. Böylece Bernstein polinomlarının oluşturduğu bazın lineer bağımsız olduğu gösterilmiş olur (Joy, 1990).

### **II.2.5. n. Dereceden Polinomların Uzayı İçin Baz Oluşturma Özelliği**

Daha önce sıraladığımız özelliklere göre n. dereceden polinomların uzayını geren kuvvet bazının her elemanı, n. dereceden Bernstein polinomlarının oluşturduğu kümedeki elemanların lineer birleşimi olarak yazılabildiği ve n. dereceden Bernstein polinomlarının oluşturduğu küme lineer bağımsız olduğu için, Bernstein polinomlarının bu kümesi n. dereceden polinomların uzayını gerer ve bu uzayın bir bazı olur (Joy, 1990).

### III. BEZIER POLİNOM EĞRİSİ

#### III.1. Bézier Polinom Eğrisi

$r_i$  ler, kontrol noktaları olmak üzere,

$$r(t) = \sum_{i=0}^n \frac{n!}{(n-i)!i!} \cdot t^i (1-t)^{n-i} r_i \quad (3.1)$$

biçiminde tanımlanan polinom eğrisine **Bézier eğrisi**, adı verilir (Tantay, 1998). Burada,

$$r(0) = r_0, \quad r(1) = r_n$$

$$\dot{r}(0) = n(r_1 - r_0), \quad \dot{r}(1) = n(r_n - r_{n-1}) \quad (3.2)$$

olduğu gösterilebilir (Tantay, 1998).

Öyleyse genel polinom eğrisi  $r_0$  ve  $r_n$  noktalarından geçer, burada teğetler  $r_0 r_1$  ve  $r_{n-1} r_n$  vektörleri yönündedirler, (Tantay, 1998). Eşitlik (3.1) ifadesinde  $n=3$  alınırsa, geometrik tasarımda en çok kullanılan Bézier Kübik eğrisi

$$r = r(t) = (1-t)^3 r_0 + 3t(1-t)^2 r_1 + 3t^2(1-t) r_2 + t^3 r_3 \quad (3.3)$$

$0 \leq t \leq 1$  biçiminde elde edilir (Tantay, 1998).

Buradan,

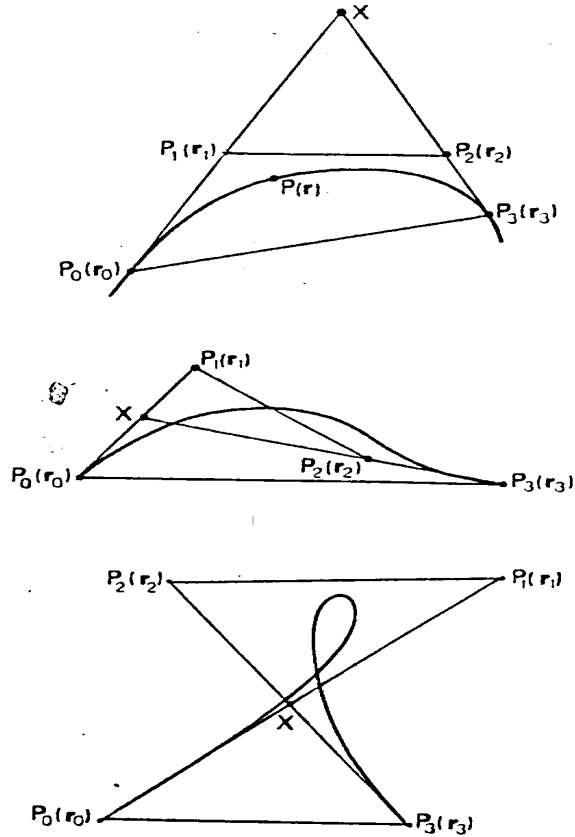
$$r(0) = r_0 ;$$

$$r(1) = r_3 ;$$

$$\dot{r}(0) = 3(r_1 - r_0) ; \quad (3.4)$$

$$\dot{r}(1) = 3(r_3 - r_2) ;$$

eşitlikleri bulunur. Böylece Bézier formu ile gösterilen eğri  $r_0$  ve  $r_3$  noktalarından geçer;  $r_0$  da,  $r_0$  dan  $r_1$  e doğru teğet yönüne sahiptir ve  $r_3$  de,  $r_2$  den  $r_3$  e doğru teğet yönüne sahiptir (Tantay, 1998).



Şekil 3.1. Bézier eğri örnekleri

$P_0P_1$ ,  $P_1P_2$  ve  $P_2P_3$  doğruları, eğrinin **karakteristik çokgeni** denilen bir şekil oluştururlar. Bu şekil genellikle bir kapalı şekil değildir. O zaman, bir eğri tasarımı için eğrinin geçmesini istediğimiz  $P_0$  ve  $P_3$  noktalarını seçer ve  $P_1$  ve  $P_2$  yi,  $P_0$  ve  $P_3$  deki istenen teğetler üzerine yerleştiririz. Daha sonra eğriye daha büyük tamlık vermek için  $P_0P_1$  ve



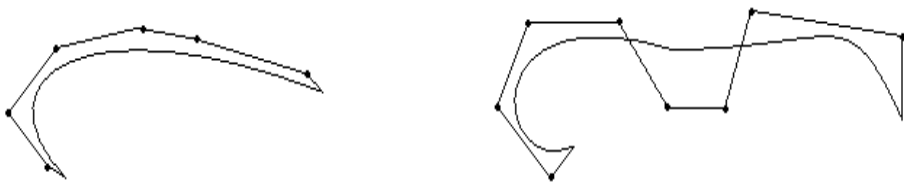
$P_2P_3$  ün uzunluklarını ayarlarız. Bézier kübik eğrisinin matris formu  $r = TMR$  dir, veya detaylı bir şekilde,

$$r(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & 6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad (3.5)$$

dir, (Tantay, 1998),  $P_0P_1$  ve  $P_2P_3$  ün  $P_0P_3$  ü geçmemesi gerektiğini söyler. Bu kurala uyan eğrilerdeki herhangi bir dönme  $0 \leq t \leq 1$  değerinin dışında olacaktır. O halde, Bézier eğrisi tamamen  $P_0P_1P_2P_3$  tetrahedronunun içinde uzanacaktır. Eğrinin karakteristik çoğeniyle olan ilişkisine **dışbükey tekne özelliği** denir ve bu bir çokgen cinsinden tanımlanan herhangi bir eğri için çok kullanışlı bir özelliktir (Tantay, 1998).

### III.2. Bézier Eğrileri ile Tasarım

Burada da kontrol çokgeninin noktalarını interaktif olarak oynatarak, eğriyi tekrar yapılandırabiliriz (Farin, 1990).



Şekil 3.2. Bazı Bézier eğri örnekleri.

### III.3. Bézier Eğrileri İçin Derece Yükseltme

İki veya daha fazla Bézier eğrisi içeren çoğu uygulama, içerdiği tüm eğrilerin aynı dereceye sahip olmalarını gerektirir. Ayrıca yüksek

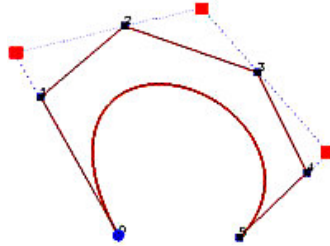
dereceli Bézier eğrileri, işlemlerde daha fazla zaman almasına karşın, şekillerin tasarımı için daha çok esnekliğe sahip olmaları nedeniyle tercih edilir. Bu nedenlerle bir Bézier eğrisinin şekli bozulmadan derecesinin yükseltilmesi yararlı olacaktır. Burada önemli olan eğrinin şeklinin değişmemesidir, (Bkz. Şekil 3.3). Aksi halde derece yükseltmenin hiç bir faydası yoktur (Shene, 2007).

Farz edelim ki,  $(n+1)$  kontrol noktası  $p_0, p_1, \dots, p_n$  ile  $n$ . dereceden bir Bézier eğrisinin derecesini şeklini değiştirmeden  $(n+1)$ . dereceye yükseltmek istiyoruz.  $(n+1)$ . dereceden bir Bézier eğrisi  $(n+2)$ . kontrol noktası ile tanımlandığından, yeni bir kontrol noktası bulunması gereklidir.  $p_0$  ve  $p_n$  yeni kümede bulunmak zorunda çünkü eğri bu noktalardan geçer. Yani,  $n$  kontrol noktasına ihtiyacımız var. Yeni kontrol noktalarının kümesi  $\{q_0, q_1, \dots, q_{n+1}\}$  olsun. Yukarıda belirtildiği üzere  $q_0 = p_0$  ve  $q_{n+1} = p_n$  olur. Diğer kontrol noktaları

$$q_i = \frac{i}{n+1} p_{i-1} + \left(1 + \frac{i}{n+1}\right) p_i \quad 1 \leq i \leq n \quad (3.6)$$

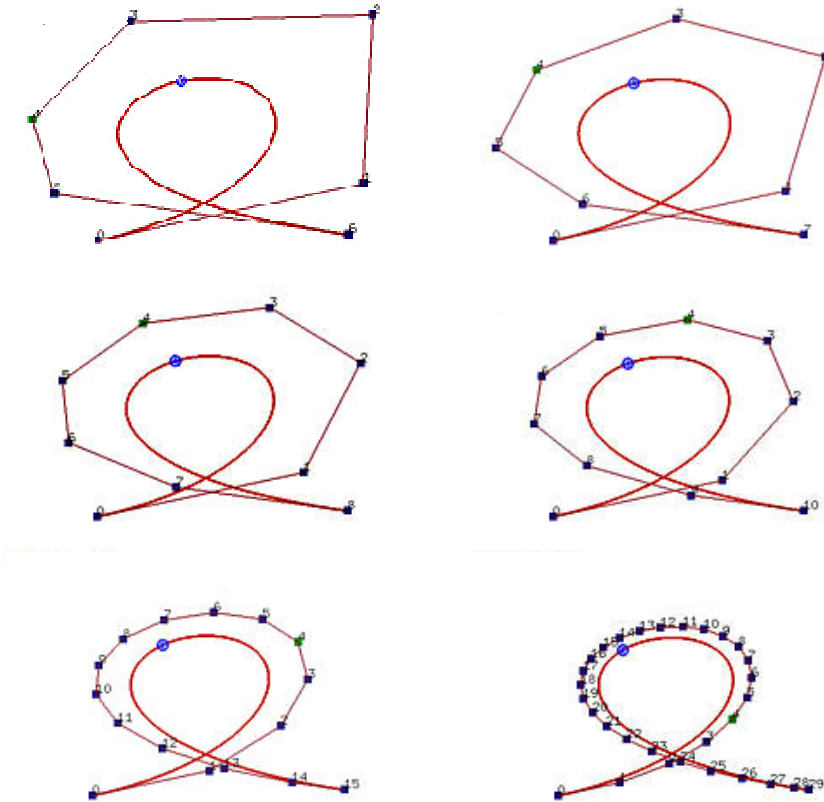
formülü ile hesaplanabilir. Böylece orjinal kontrol çokgeninin her bir kenarı üzerinde yeni bir kontrol noktası bulunarak yeni kontrol noktaları kümesi oluşturulmuş olur (Shene, 2007).

De Casteljau algoritmasını hatırlarsak;  $AB$  doğru parçası üzerideki bir  $C$  noktası, bu doğru parçasını  $u:1-u$  oranında böler; yani,  $C=(1-u)A+uB$  şeklinde ifade edilebilir. Yeni kontrol noktaları için belirtilen formülden  $q_i$  nin  $p_{i-1}p_i$  doğru parçasını  $1-i / (n+1) : i / (n+1)$  oranında böldüğü görülür (Shene, 2007), (Bkz. Şekil 3.2.).



Şekil 3.3. 4. dereceden 5. dereceye yükseltlen Bézier eğrisi

De Casteljau algoritmasından farklı olarak bu oran sabit değildir,  $i$  nin değerlerine göre değişir. Ama yine de hesaplama süreci de Casteljau algoritmasına çok benzer (Shene, 2007).



Şekil 3.4. Çeşitli derecelere yükseltilmiş Bézier eğrisi

## IV. REGLE YÜZEYLER

### IV.1. Regle yüzey

$\alpha = \alpha(u)$  ve  $\tau = \tau(u)$ , yalnız bir  $v$  parametresine bağlı olan iki vektör,  $v$  diğer bir parametre olmak üzere

$$\mathbf{r}(u, v) = \alpha(u) + v\tau(u) \quad (4.1)$$

denklemleri ile bir  $[\mathbf{r}]$  yüzeyi tanımlanabilir. Eğer  $u$  sabit  $v$  değişken olarak kabul edilirse  $y$  noktası  $[\mathbf{r}]$  yüzeyi üzerinde  $\tau$  ya paralel bir  $D$  doğrusu çizer. O halde  $[\mathbf{r}]$  yüzeyinin  $u = \text{sabit}$  koordinat çizgileri doğrulardan oluşmuştur. Bu  $D$  doğrularına yüzeyin ana doğruları (doğuranlar), böyle bir yüzeye de **regle yüzey**, denir. Bu tanıma göre bir parametreye bağlı doğru ailesinin geometrik yerine **regle yüzey**, denir (Biran, 1970).

$[\mathbf{r}]$  regle yüzeyinin  $u$  ve  $v$  değişkenlerine göre kısmi türevleri

$$\mathbf{r}_v = \tau, \mathbf{r}_u = \alpha' + v\tau' \quad (4.2)$$

olduğundan, teğet düzlemi,  $z$  bu düzlemin bir noktası olmak üzere

$$(z - \mathbf{r}, \tau, \alpha' + v\tau') = 0 \quad (4.3)$$

dir. Eğer  $z$  noktası, yüzeyin  $\mathbf{r}$  noktasından geçen anadoğru üzerinde ise,  $\mu$  uygun olarak seçilmiş bir skaler olmak üzere,

$$z - \mathbf{r} = \mu\tau \quad (4.4)$$

dir; bu bağıntı teğet düzlem denkleminde gözönüne alınırsa,  $\mathbf{r}$  noktasından geçen anadoğrunun, bu noktadaki teğet düzlem içinde bulunduğu görülür (Biran, 1970).

## IV.2. Açılabilir Yüzey

Yüzeyin bir  $r$  noktasındaki normal doğrultu,

$$\begin{aligned}
 r_v \times r_u &= \tau \times (\alpha' + v\tau') \\
 &= \tau \times \alpha' + v\tau \times \tau' \\
 &= \frac{1}{v} \tau \times \alpha' + \tau \times \tau'
 \end{aligned} \tag{4.5}$$

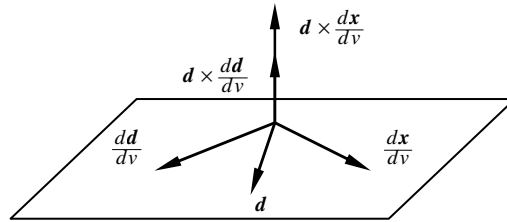
vektörü ile verilmiştir. Yüzeyimizin bir  $D$  anadoğrusu boyunca normal doğrultusunun aynı kalması şartı, normal doğrultu vektörünün  $v$  ye bağlı olmaması ve bunun neticesi olarak,

$$\tau \times \alpha' \quad , \quad \tau \times \tau'$$

vektörlerinin aynı doğrultuya sahip olmaları ile gerçekleşir. Bu takdirde

$$\tau \quad , \quad \alpha' \quad , \quad \tau'$$

vektörlerinin bir düzleme paralel olmaları gerekir.



Şekil 4.1. Açılabilirlik şartlarını sağlayan vektörlerin durumu

O zaman, bir  $D$  anadoğrusu boyunca yüzeyin normal doğrultusunun aynı kalması şartı,

$$(\tau \quad , \quad \alpha' \quad , \quad \tau') = 0 \tag{4.6}$$

bağıntısı ile ifade edilir. Bu bağıntı gerçekleştiği takdirde regle yüzeye **açılabilir yüzey**, denir (Biran, 1970).

Diđer tanım olarak bir uzay eđrisinin teđetlerinin oluřturduđu yzeye **açılabilir yüzey**, denir. Bu uzay eđrisine de **dönüm ayrıtı** veya **dayanak eđrisi**, denir (Biran, 1970).

## V. REGLE BEZIER YÜZEYLERİ ve AÇILABİLİRLİK ŞARTLARI

n. dereceden iki Bézier eğrisi

$$\alpha(u) = \sum_{i=0}^n B_i^n(u) p_i, \quad u \in [0,1] \quad (5.1)$$

$$q(u) = \sum_{i=0}^n B_i^n(u) q_i, \quad u \in [0,1] \quad (5.2)$$

olsun. Şimdi bu iki fonksiyon karılarak

$$r(u, v) = (1-v)\alpha(u) + vq(u), \quad u, v \in [0,1] \quad (5.3)$$

şeklinde bir yüzey tanımlanır ve yüzeyin denklemi düzenlenirse;

$$\begin{aligned} r(u, v) &= \alpha(u) - v\alpha(u) + vq(u) \\ &= \alpha(u) + v \underbrace{(q(u) - \alpha(u))}_{\tau(u)} \end{aligned}$$

olur. Böylece

$$r(u, v) = \alpha(u) + v\tau(u)$$

yani, regle yüzeyin alışıla gelmiş formu elde edilmiş getirilmiş olur.

Burada

$$\tau(u) = \sum_{i=0}^n B_i^n(u) (q_i - p_i), \quad u \in [0,1] \quad (5.4)$$

olur.

Buradaki  $\alpha(u)$  ve  $\tau(u)$  çizgiler geometrisinde anadoğru ve doğrultman olarak adlandırılır (Zhang ve Wang, 2006; Aumann, 1998; Biran, 1970).

Diferansiyel geometriden biliyoruz ki ,  $\alpha' = \alpha'(u)$ ,  $\tau = \tau(u)$  ve  $\tau' = \tau'(u)$  vektörleri bir  $u \in [0,1]$  için lineer bağımlı ise  $r(u,v)$  regle yüzeyi açılabilir. Başka bir deyişle,

$$\lambda\alpha' + \mu\tau + \gamma\tau' = 0 \quad (5.5)$$

olacak şekilde hepsi birden sıfır olmayan  $\lambda = \lambda(u)$ ,  $\mu = \mu(u)$  ve  $\gamma = \gamma(u)$  skaler fonksiyonları bulunabiliyor ise, regle yüzey açılabilir (Zhang ve Wang, 2006; Aumann, 1998; Biran, 1970).

Spivak açılabilir regle yüzeylerin sadece üç tipinin var olduğunu: silindirler(dülemlerde dahil), koniler ve bir uzay eğrisinin teğetleri tarafından oluşturulan teğet yüzeyleri, söylemiştir. Eşitlik (5.5) sağlayan her bir  $\{\lambda(u), \mu(u), \gamma(u)\}$  fonksiyon sistemi ile oluşan açılabilir regle yüzey, bahsedilen yüzey çeşitlerinden birine tekabül eder (Zhang ve Wang, 2006; Aumann, 1998).

### V.1. Silindir Olma Durumu

$\lambda(u) = 0$  olması durumunda,  $\mu(u)$  ve  $\gamma(u)$  ikisi birlikte sıfır olmamak üzere  $\mu\tau + \gamma\tau' = 0$  olur. Bu  $\tau$  ve  $\tau'$  vektörlerinin lineer bağımlı ve paralel olduklarını gösterir.  $\tau(u)$  vektörüne paralel birim vektör  $e(u)$  olsun. Böylelikle  $w(u)$  bir skaler fonksiyon olmak üzere,  $\tau(u) = w(u)e(u)$  şeklinde ifade edilebilir ve  $\tau' = w'e + we'$  eşitliğin iki tarafı  $\tau$  ile vektörel çarpılırsa

$$\tau \times \tau' = we \times w'e + we \times we' = 0$$

burada  $we$  ve  $w'e$  paralel oldukları için  $we \times w'e = 0$  olur. Böylece

$$\tau \times \tau' = we \times we' = w^2 e \times e' = 0$$



olduğundan  $e \times e' = 0$  olur.  $\|e\|^2 = 1$  olması  $e \cdot e' = 0$  olmasını sağlar.

Lagrange özelliğinden,  $(e \times e)(e' \times e') - (e \times e')^2 = (e \times e')^2$  olur. Bu da  $e' \cdot e' = 0$  yani,  $e' = 0$  olmasını gerektirir. Bunun sonucunda  $e$  birim vektörünün yönü değişmez yani,  $\tau(u)$  yönünü değiştirmeden hareket eder. Böylelikle oluşan yüzey bir silindirdir. Sonuç olarak açılabilir Bézier regle yüzeyi bir silindir olur (Zhang ve Wang, 2006; Aumann, 1998). Eşitlik (5.4) ü kullanarak bir silindir inşa etmek için

$\tau(0) = q_0 - p_0$  böylece  $e = \frac{q_0 - p_0}{\|q_0 - p_0\|}$  olur. Diğer yandan,

$$\frac{d\tau(u)}{du} = \frac{dw(u)}{du} e$$

$$\frac{d\tau(0)}{du} = n\Delta(q_0 - p_0) = \frac{dw(0)}{du} e$$

dir. Burada  $\Delta$  ileri fark operatörüdür yani,

$$\Delta(q_0 - p_0) = (q_1 - p_1) - (q_0 - p_0),$$

$$\Delta^k(q_0 - p_0) = \Delta^{k-1}(q_1 - p_1) - \Delta^{k-1}(q_0 - p_0), \quad k = 2, 3, \dots$$

$(q_1 - p_1)\|e$  olduğu kolaylıkla görülür. Benzer şekilde

$$\frac{d'\tau(0)}{du'} = \frac{n!}{(n-i)!} \cdot \Delta^i(q_0 - p_0) = \frac{d'\omega(0)}{du'} e$$

ifadesinden

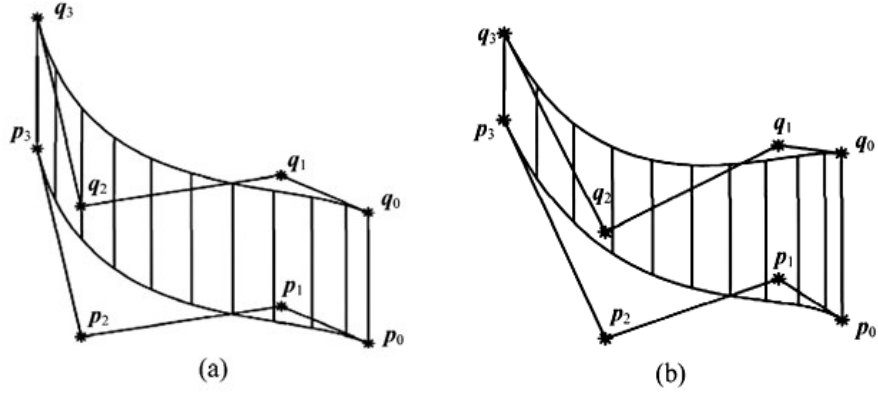
$$(q_i - p_i)\|e, \quad i = 1, 2, \dots, n \quad (5.6)$$

elde edilir.

$\sigma$  sabit olmak üzere  $q_{i+1} - p_{i+1} = \sigma(q_i - p_i)$ ,  $i = 0, 1, \dots, n-1$  olsun.

Böylelikle Bézier silindirinin ifadesini elde etmiş oluruz. Şekil 5.1 iki

farklı 3x1 Bézier silindirini göstermektedir (Zhang ve Wang, 2006; Aumann, 1998).



Şekil 5.1. Bézier silindirleri,  $q_{i+1} - p_{i+1} = \sigma (q_i - p_i), i = 0, 1, \dots, n-1$

(a)  $\sigma = 1.0$ , (b)  $\sigma = 1.8$

## V.2. Koni veya Bir Uzay Eğrisinin Teğetleri Tarafından Oluşturulan Teğet yüzeyi Olma Durumu

$\lambda(u) \neq 0$  olması durumunda  $\alpha' = a\tau + b\tau'$  olacak şekilde

$$a = a(u) - \frac{\mu(u)}{\lambda(u)}, \quad b = b(u) = -\frac{\gamma(u)}{\lambda(u)}$$

iki fonksiyon ortaya çıkar. Böylece  $r(u, v)$  fonksiyonu

$$r(u, v) = \alpha_1(u) + (v + b)\tau \quad (5.7)$$

$$\alpha_1(u) = \alpha - b\tau \quad (5.8)$$

olarak tanımlanır ki burada

$$\alpha'_1 = \alpha' - b'\tau - b\tau' = (a - b')\tau \quad (5.9)$$

olur (Zhang ve Wang, 2006; Aumann, 1998).

### V.2.1. Koni Olma Durumu

$a = b'$  olması durumunda  $\alpha'_i = 0$  olur. Bu da  $\alpha'_i$  nün sabit vektör olduğunu gösterir.

$$\hat{v} = v + b(u) \quad (5.10)$$

parametre dönüşümü yaparsak,  $r(u, v)$  açılabilir regle yüzeyi

$$r(u, v) = \alpha_i + \hat{v}\tau(u) \quad (5.11)$$

olur ki burada  $\hat{v} \in [\mathbf{b}(u), \mathbf{b}(u) + 1]$  olur.

Bu da  $r(u, v)$  yüzeyinin tepe noktası  $\alpha_i$  olan bir koni olmasını sağlar. Eşitlik (5.1) deki  $\alpha(u)$  Bézier eğrisi bilindiği farzedilsin, eşitlik (5.2) ile belirtilen  $q(u)$  Bézier eğrisi belirlenebilmelidir ki eşitlik (5.1) deki  $\alpha(u)$  Bézier eğrisini doğrultman kabul eden eşitlik (5.3) te belirtilen  $r(u, v)$  Bézier regle yüzeyi açılabilir olsun. Bu durumda  $r(u, v)$  regle yüzeyi bir konidir, (Zhang ve Wang, 2006; Aumann, 1998).

#### Örnek 1.

$\rho \neq 1$  ve sabit olmak üzere,  $a = 0, b = 1/(\rho - 1)$  olarak alalım. Bu durumda  $\alpha' = \tau'/(\rho - 1)$  olur. Eşitlik (5.1) ve eşitlik (5.4) i kullanarak

$$n \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta p_i = n \frac{1}{\rho - 1} \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta (q_i - p_i)$$

elde edilir. Bernstein bazının lineer bağımsız olamasından dolayı

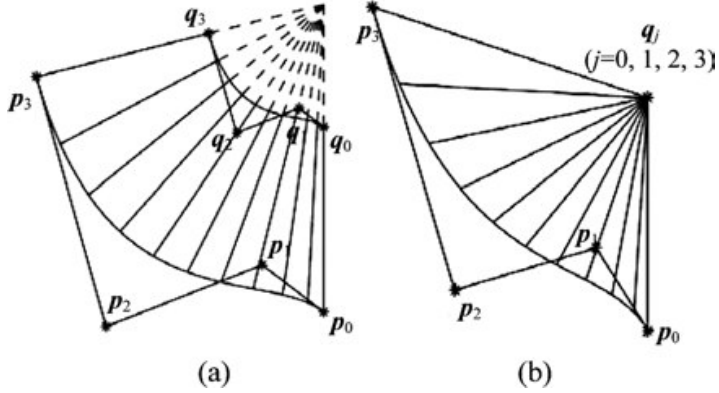
$$(\rho - 1)(p_{i+1} - p_i) = (q_{i+1} - p_{i+1}) - (q_i - p_i), \quad i = 0, 1, \dots, n-1$$

olur ki böylece

$$q_i = q_0 + \rho(p_i - p_0), \quad i = 1, 2, \dots, n \quad (5.12)$$

olur (Zhang ve Wang, 2006). Böylelikle açılabilir Bézier konisinin ifadesi elde edilmiş olur.

Şekil 5.2. iki farklı 3x1 Bézier konisini göstermektedir.



Şekil 5.2. Bézier konileri,  $q_i = q_0 - \rho (q_i - p_i), i = 1, 2, \dots, n$

(a)  $\rho = 0.4$  (b)  $\rho = 0.0$

### Örnek 2.

$\sigma \neq 0$  ve sabit olmak üzere  $a = \sigma$ ,  $b = \sigma u$  olarak alalım. Bu durumda  $\alpha' = a\tau + b\tau'$  olduğundan

$$n \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta p_i = \sigma \sum_{i=0}^n B_i^n(u) (q_i - p_i) + n\sigma u \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta (q_i - p_i)$$

elde edilir.

Yukarıdaki denklemin her iki tarafında bulunan Bézier eğrilerinin derecelerini yükselterek,

$$\begin{aligned} & \sum_{i=0}^n B_i^n(u) ((n-i)\Delta p_i + i\Delta p_{i-1}) \\ &= \sum_{i=0}^n B_i^n(u) \sigma (q_i - p_i) + \sum_{i=1}^n B_i^n(u) (i\sigma \Delta (q_{i-1} - p_{i-1})) \end{aligned}$$

$$\Delta p_{-1} = \Delta p_n = 0$$

elde edilir (Zhang ve Wang, 2006). .

Tekrar Bernstein bazının lineer bağımsızlığına dayanarak,

$$\begin{cases} n\Delta p_0 = \sigma(q_0 - p_0) \\ (n-i)\Delta p_i + i\Delta p_{i-1} = \sigma(q_i - p_i) + i\sigma\Delta(q_{i-1} - p_{i-1}) \\ i = 1, 2, \dots, n-1 \\ n\Delta p_{n-1} = \sigma(q_n - p_n) + n\sigma\Delta(q_{n-1} - p_{n-1}) \end{cases}$$

elde edilir. Bundan dolayı,

$$\begin{cases} q_0 = p_0 + \frac{n}{\sigma}\Delta p_0 \\ q_i = p_i + \frac{i\sigma(q_{i-1} - p_{i-1}) + (n-i)\Delta p_i + i\Delta p_{i-1}}{(i+1)\sigma} \\ i = 1, 2, \dots, n-1 \\ q_n = p_n + \frac{n\sigma(q_{n-1} - p_{n-1}) + n\Delta p_{n-1}}{(n+1)\sigma} \end{cases}$$

olur. Bu da

$$\begin{aligned} (i+1)\sigma(q_i - p_i) - i\sigma(q_{i-1} - p_{i-1}) &= (n-i)\Delta p_i + i\Delta p_{i-1}, \\ i &= 1, 2, \dots, n-1 \end{aligned}$$

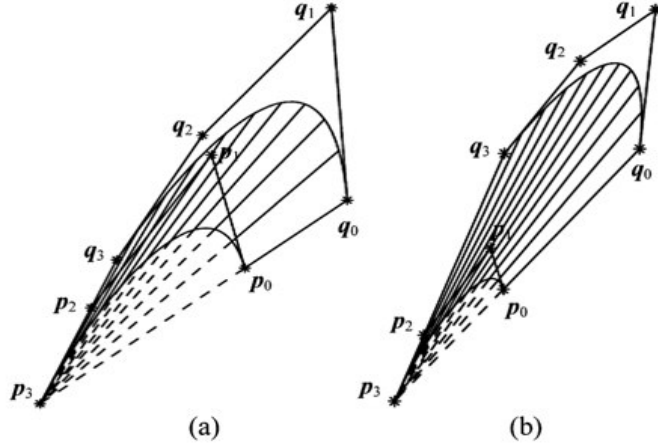
dır.

Yukarıdaki rekürsiyon formülünü kullanarak;

$$\begin{cases} q_0 = p_0 + \frac{n}{\sigma}\Delta p_0 \\ q_i = p_i + \frac{(n-i)p_{i+1} + (i+1)p_i - (n+1)p_0}{(i+1)\sigma} \\ i = 1, 2, \dots, n-1 \\ q_n = p_n + \frac{p_n - p_0}{\sigma} \end{cases} \quad (5.13)$$

elde ederiz (Zhang ve Wang, 2006).

Şekil 5.3 iki farklı 3x1 Bézier konisini göstermektedir.



Şekil 5.3. Bézier konileri,  $a = \sigma$ ,  $b = \sigma u$  (a)  $\sigma = 3.0$  (b)  $\sigma = 0.8$

### V.2.2. Bir Uzay Eğrisinin Teğetleri Tarafından Oluşturulan Teğet Yüzeyi Olma Durumu

$a \neq b'$  olması durumunda  $\tau = \alpha_1' / (a - b')$  elde edilir.

$$\hat{v} = \frac{v + b(u)}{a(u) - b'(u)} \quad (5.14)$$

şeklinde yeni bir parametre tanımlayarak açılabilir yüzey  $r(u, v)$  nin denklemi

$$\mathbf{r}(u, v) = \alpha_1(u) + \hat{v} \alpha_1'(u) \quad (5.15)$$

şeklinde yazılabilir ki burada

$$\alpha_1(u) = \alpha(u) - b(u)\tau(u) \quad \text{ve} \quad \hat{v} = v + b(u)$$

dır.  $r(u, v)$   $\alpha_1(u)$  eğrisinin teğet yüzeyidir ve burada  $\alpha_1(u)$  dönüm ayrıtıdır.

Daha önce de hesaplandığı gibi, eşitlik (5.2) ile belirtilen  $q(u)$  Bézier eğrisi belirlenebiliyor ise eşitlik (5.1) deki  $\alpha(u)$  Bézier eğrisini doğrultman kabul eden Eşitlik(5.3) te belirtilen  $r(u, v)$  bir uzay eğrisinin teğetlerinin oluşturduğu teğet yüzeyi olur. Aynı zamanda bu yüzeyin dönüm ayrıtımı da belirleyebiliriz (Zhang ve Wang, 2006; Aumann, 1998). Bazı tipik ve önemli örneklerin incelenmesi faydalı olacaktır.

### Örnek 1.

$a = 2, b = u$  olması durumu

Burada,  $\alpha' = a\tau + b\tau'$  eşitliğine ve Bézier eğrisi için derece yükseltme formülüne dayanarak

$$\begin{aligned} & \sum_{i=0}^n B_i^n(u) ((n-i)\Delta p_i + i\Delta p_{i-1}) \\ &= \sum_{i=0}^n B_i^n(u) [2(q_i - p_i)] + \sum_{i=0}^n B_i^n(u) (i\Delta(q_{i-1} - p_{i-1})) \end{aligned}$$

$$\Delta p_{-1} = \Delta p_n = \Delta p_{-1} q_{-1} = 0$$

elde edilir. Devamında Bernstein bazının lineer bağımsızlığını kullanarak

$$(n-i)\Delta p_i + i\Delta p_{i-1} = 2(q_i - p_i) + i\Delta(q_{i-1} - p_{i-1}) \quad i = 0, 1, \dots, n$$

$$\Delta p_{-1} = \Delta p_n = \Delta(q_{-1} - p_{-1}) = 0$$

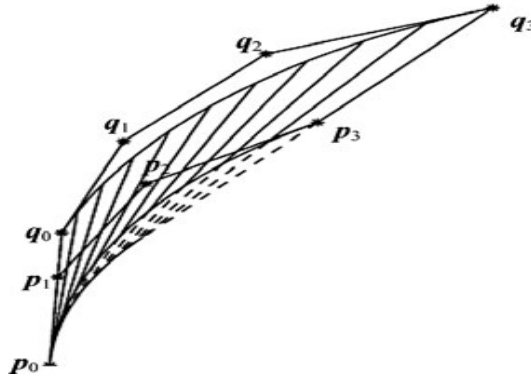
olduğu görülür. Böylece

$$\begin{cases} q_0 = p_0 + \frac{n\Delta p_0}{2} \\ q_i = p_i + \frac{(n-i)\Delta p_i + i\Delta p_{i-1} + i(q_{i-1} - p_{i-1})}{i+2} \\ \quad i = 1, 2, \dots, n-1 \\ q_n = p_n + \frac{n\Delta p_{n-1} + n(q_{n-1} - p_{n-1})}{n+2} \end{cases}$$

yukarıdaki rekürsiyon formülünü kullanarak, kolaylıkla

$$\begin{cases} q_0 = p_0 + nA_0 \\ q_i = p_i + (n-i)A_i + iA_i & i = 1, 2, \dots, n-1 \\ q_n = p_n + nA_{n-1} \\ A_i = \frac{1}{(i+1)(i+2)} \sum_{j=1}^{i+1} j\Delta p_{j-1}, & i = 0, 1, \dots, n \end{cases} \quad (5.16)$$

olduğu anlaşılabilir (Zhang ve Wang, 2006). Şekil 5.4 dönüm ayrıtı  $\alpha_1(u)$  olan  $3 \times 1$  Bézier teğet yüzeyini göstermektedir.



Şekil 5.4. Bézier teğet yüzeyi  $a = 2$ ,  $b = u$

### Örnek 2.

$a = 1, b = 0$  olması durumu

Bu durumda,  $\alpha' = \tau$  dır. Bézier eğrisi için derece yükseltme formülünden

$$\sum_{i=0}^n B_i^n(u) ((n-i)\Delta p_i + i\Delta p_{i-1}) = \sum_{i=0}^n B_i^n(u) (q_i - p_i)$$

$$\Delta p_{-1} = \Delta p_n = 0$$

elde edilir. Daha sonra Bernstein bazının lineer bağımsızlığından dolayı

$$(n-i)\Delta p_i + i\Delta p_{i-1} = (q_i - p_i), \quad i = 0, 1, \dots, n$$

$$\Delta p_{-1} = \Delta p_n = 0$$

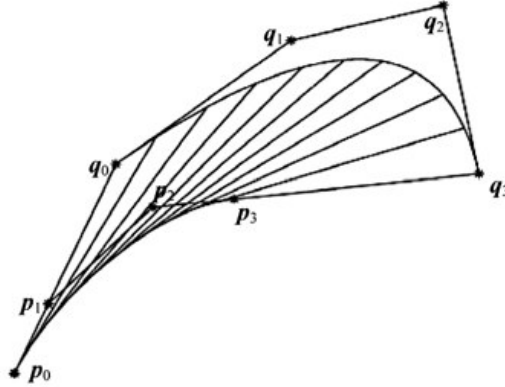


ortaya çıkar. Böylece

$$\begin{cases} \mathbf{q}_0 = \mathbf{p}_0 + n\Delta\mathbf{p}_0 \\ \mathbf{q}_i = \mathbf{p}_i + (n-i)\Delta\mathbf{p}_i + i\Delta\mathbf{p}_{i-1} \\ \quad \quad \quad i = 1, 2, \dots, n-1 \\ \mathbf{q}_n = \mathbf{p}_n + n\Delta\mathbf{p}_{n-1} \end{cases} \quad (5.17)$$

olur (Zhang ve Wang, 2006).

Şekil 5.5 dönüm ayrıtı  $\alpha_I(u)$  olan  $3 \times 1$  Bézier teğet yüzeyini göstermektedir.



Şekil 5.5. Bézier teğet yüzeyi  $a = 1.0$ ,  $b = 0.0$

### Örnek 3.

$a = 1/u, b = 1$  olması durumu.

$\alpha' = a\tau + b\tau'$  ten dolayı

$$n \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta \mathbf{p}_i = \frac{1}{u} \sum_{i=0}^n B_i^n(u) (\mathbf{q}_i - \mathbf{p}_i) + n \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta (\mathbf{q}_i - \mathbf{p}_i)$$

veya

$$n u \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta \mathbf{p}_i = \sum_{i=0}^n B_i^n(u) (\mathbf{q}_i - \mathbf{p}_i) + n u \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta (\mathbf{q}_i - \mathbf{p}_i)$$

olur ve eşitliğin her iki tarafındaki Bézier eğrisinin dereceleri yükseltirse

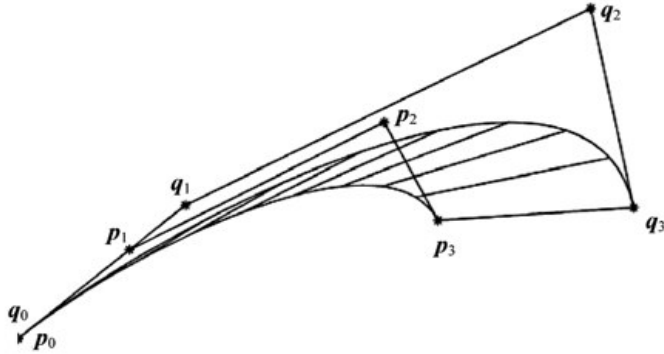
$$\sum_{i=1}^n B_i^n(u)(i\Delta p_{i-1}) = \sum_{i=0}^n B_i^n(u)(q_i - p_i) + \sum_{i=1}^n B_i^n(u)(i\Delta(q_{i-1} - p_{i-1}))$$

elde edilir. Bernstein bazının lineer bağımsızlığından dolayı

$$\begin{cases} q_0 = p_0 \\ q_i = p_i + \frac{1}{i+1}(p_i + q_{i-1} - 2p_{i-1}), \quad i = 1, 2, \dots, n. \end{cases} \quad (5.18)$$

olur (Zhang ve Wang, 2006) .

Şekil 5.6 dönüm ayrıtı  $\alpha_1(u)$  olan  $3 \times 1$  Bézier teğet yüzeyini göstermektedir.



Şekil 5.6. Bézier teğet yüzeyi  $a = 1/u$ ,  $b = 1.0$

#### Örnek 4.

$$\alpha = \frac{n(1-\sigma)}{\rho-1}, \quad b = \frac{1-u+\sigma u}{\rho-1}, \quad \sigma \neq 1, \quad \rho \neq 1 \text{ olması durumu.}$$

$$\alpha' = a\tau + b\tau'$$

eşitliğinden dolayı

$$\begin{aligned}
& n \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta \mathbf{p}_i \\
&= \frac{n(1-\sigma)}{\rho-1} \sum_{i=0}^n B_i^n(u) (\mathbf{q}_i - \mathbf{p}_i) + \frac{n(1-u+\sigma u)}{\rho-1} \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta (\mathbf{q}_i - \mathbf{p}_i)
\end{aligned}$$

olur ve Bézier eğrileri için olan derece yükseltme formülünü kullanarak;

$$\begin{aligned}
& (\rho-1) \sum_{i=0}^n B_i^n(u) ((n-i)\Delta \mathbf{p}_i + i\Delta \mathbf{p}_{i-1}) \\
&= n(1-\sigma) \sum_{i=0}^n B_i^n(u) (\mathbf{q}_i - \mathbf{p}_i) + \sum_{i=0}^n B_i^n(u) ((n-i)\Delta (\mathbf{q}_i - \mathbf{p}_i) \\
&\quad + \sigma i \Delta (\mathbf{q}_{i-1} - \mathbf{p}_{i-1})) \quad i = 0, 1, \dots, n
\end{aligned}$$

$$\Delta \mathbf{p}_{-1} = \Delta \mathbf{p}_n = \Delta (\mathbf{q}_{-1} - \mathbf{p}_{-1}) = \Delta (\mathbf{q}_n - \mathbf{p}_n) = 0$$

elde edilir. Benzer şekilde

$$\begin{aligned}
& (\rho-1) ((n-i)\Delta \mathbf{p}_i + i\Delta \mathbf{p}_{i-1}) \\
&= n(1-\sigma) (\mathbf{q}_i - \mathbf{p}_i) + (n-i)\Delta (\mathbf{q}_i - \mathbf{p}_i) + \sigma i \Delta (\mathbf{q}_{i-1} - \mathbf{p}_{i-1})
\end{aligned}$$

$$i = 0, 1, \dots, n$$

$$\Delta \mathbf{p}_{-1} = \Delta \mathbf{p}_n = \Delta (\mathbf{q}_{-1} - \mathbf{p}_{-1}) = \Delta (\mathbf{q}_n - \mathbf{p}_n) = 0.$$

olur. Düzenlersek

$$\begin{aligned}
& (\rho-1) ((n-i)\Delta \mathbf{p}_i + i\Delta \mathbf{p}_{i-1}) \\
&= (n-i) ((\mathbf{q}_{i+1} - \mathbf{p}_{i+1}) - \sigma (\mathbf{q}_i - \mathbf{p}_i)) + i ((\mathbf{q}_i - \mathbf{p}_i) - \sigma (\mathbf{q}_{i-1} - \mathbf{p}_{i-1}))
\end{aligned}$$

$$i = 0, 1, \dots, n$$

sadeleştirme yaparsak

$$\begin{aligned}
& (n-i) ((\rho-1)\Delta \mathbf{p}_i - ((\mathbf{q}_{i+1} - \mathbf{p}_{i+1}) - \sigma (\mathbf{q}_i - \mathbf{p}_i))) \\
&= i ((\rho-1)\Delta \mathbf{p}_{i-1} - ((\mathbf{q}_i - \mathbf{p}_i) - \sigma (\mathbf{q}_{i-1} - \mathbf{p}_{i-1}))) \quad (5.19)
\end{aligned}$$

$$i = 0, 1, \dots, n$$

elde ederiz.

Özel olarak, yukarıdaki eşitlikte  $i$  tam sayısını sıfır alırsak ve

$$n(\rho - 1)\Delta p_0 - n((q_1 - p_1) - \sigma(q_0 - p_0)) = 0$$

kabul edersek

$$(\rho - 1)\Delta p_i - ((q_{i+1} - p_{i+1}) - \sigma(q_i - p_i)) = 0$$

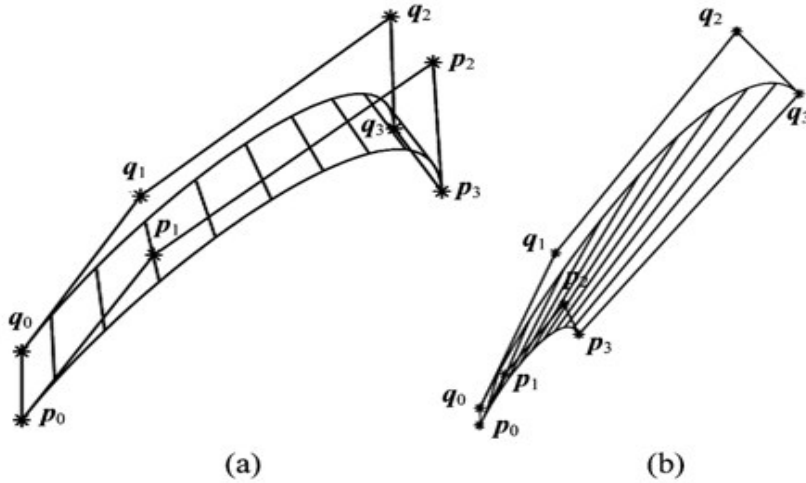
$$i = 0, 1, \dots, n-1$$

veya

$$q_{i+1} = p_i + \rho(p_{i+1} - p_i) - \sigma(q_i - p_i) \quad (5.20)$$

$$i = 0, 1, \dots, n-1$$

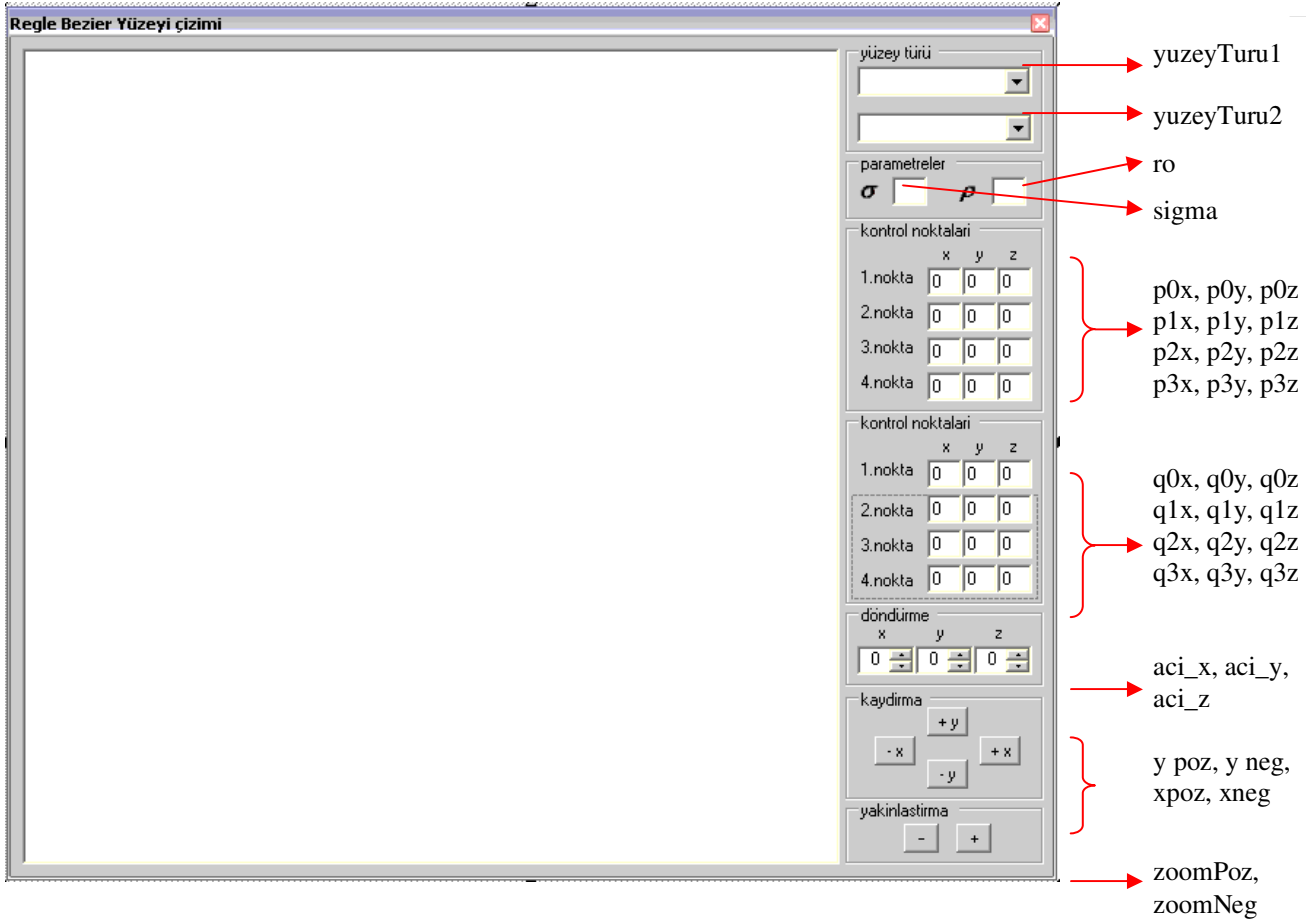
Bézier teğet yüzeyinin ifadesi elde edilir (Zhang ve Wang, 2006). Şekil 5.7. iki farklı  $3 \times 1$  Bézier tanjant yüzeyini göstermektedir.



Şekil 5.7. Bézier teğet yüzeyleri (a)  $\rho = 0.9$ ,  $\sigma = 1.1$ ; (b)  $\rho = 3$ ,  $\sigma = 1.1$

## VI. BİLGİSAYAR UYGULAMASI

Bu kısımda yaptığımız bu çalışma için C# yazılım dili ile gdi+ grafik kütüphanesini kullanarak geliştirdiğimiz windows tabanlı



bilgisayar programının kodları bulunmaktadır. Ayrıca Şekil 6.1 bu programda kullanılan kontrolleri göstermektedir.

Şekil 6.1. Kullanıcı arayüzü ve kontrollerin isimleri

```

static void Main()
{
    Application.Run(new Form1());
}

private void Form1_Load(object sender, System.EventArgs e)
{
    yuzeyTurul.SelectedIndex=0;
}

//temel deęişken tanımları

public static float
y_otele=280,x_otele=280,zoom_par=100,sig=0,rho=0,cx=1,sx=0,
                                     cy=1,sy=0,cz=1,sz=0;
public static float[,]
don={{1,0,0,0},{0,1,0,0},{0,0,1,0},{0,0,0,1}},
p={{0,0,0,1},{0,0,0,1},{0,0,0,1},{0,0,0,1}},
q={{0,0,0,1},{0,0,0,1},{0,0,0,1},{0,0,0,1}};

// ilk parametredeki objenin matrisine ikinci parametredeki
dönüşüm matrsini carparak uygulayan prosedür.

public void donusum_uygula(float[,]obj,float[,]don)
{
    int satsay=obj.GetLength(0);
    float[,] don_obj= new float[satsay,4];

    for (int i=0;i<satsay;i++)
    {
        for (int j=0;j<4;j++)
        {
            float t =0;
            for(int k=0;k<4;k++)
            {
                t=t+obj[i,k]*don[k,j];
            }
            don_obj[i,j]=t;
        }
    }
    Array.Copy(don_obj,obj,obj.Length);
}

```

// 3.boyuttaki bir noktayı düzleme noktası haline getiren  
ve ekran yapılandırmasını ayarlayan görüntüleme dönüşümleri

```
public void viewingTrans(float[, ]s)
{
    float[,] projeksiyon={{1,0,0,0},{0,1,0,0},{(float)
    Math.Cos(Math.PI/4),(float)Math.Sin(Math.PI/4),0,0},
    {0,0,0,1}};

    float[, ]zoom={{zoom_par,0,0,0},{0,-zoom_par,0,0},
    {0,0,0,0},{0,0,0,1}};
    float[, ]oteleme2d={{1,0,0,0},{0,1,0,0},{0,0,1,0},
    {x_otele,y_otele,0,1}};

    donusum_uygula(s,don);
    donusum_uygula(s,projeksiyon);
    donusum_uygula(s,zoom);
    donusum_uygula(s,oteleme2d);
}
```

// Şekli döndürecek şekilde dönüşüm uygular

```
public void dondur()
{
    float[, ]birim={{1,0,0,0},{0,1,0,0},{0,0,1,0},
    {0,0,0,1}};
    float[, ]donx={{1,0,0,0},{0,cx,sx,0},{0,-sx,cx,0},
    {0,0,0,1}};
    float[, ]dony={{cy,0,-sy,0},{0,1,0,0},{sy,0,cy,0},
    {0,0,0,1}};
    float[, ]donz={{cz,sz,0,0},{-sz,cz,0,0},{0,0,1,0},
    {0,0,0,1}};

    donusum_uygula(birim,donx);
    donusum_uygula(birim,dony);
    donusum_uygula(birim,donz);

    Array.Copy(birim,don,don.Length);
}
```

```
//verilen u değeri için bezier egrisinin üzerindeki noktayı hesaplar
```

```
public void bezier_curve(float u, float[, ]points,
                        float[, ]curve)
{
    float[, ] bernstein={{-1,3,-3,1},{3,-6,3,0},{-3,3,0,0}
                        ,{1,0,0,0}};

    float [, ]T={{u*u*u,u*u,u,1}};

    donusum_uygula(T,bernstein);
    donusum_uygula(T,points);
    Array.Copy(T, curve, T.Length);
}
```

```
//çizimin yapılmasını sağlayan ana bölüm
```

```
private void panell_Paint(object sender, System.Windows.
                        Forms.PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.SmoothingMode = System.Drawing.Drawing2D.
                        SmoothingMode.AntiAlias;

    Pen pn = new Pen(Color.Red, 1);
    Pen pn1 = new Pen(Color.Yellow, 1);
    Pen pn2 = new Pen(Color.Green, 1);
    Pen pn3 = new Pen(Color.Blue, 1);

    //yüzey türüne göre q matrisinin değerlerini bulur
    switch(yuzeyTuru1.SelectedIndex)
    {
        case 3:
        {
            switch(yuzeyTuru2.SelectedIndex)
            {
                case 0:
                {
                    for (int n=0;n<3;n++)
                    {
                        q[0,n]=p[0,n]+3*(p[1,n]-p[0,n])/2;
                        q[1,n]=p[1,n]+(2*(p[2,n]-p[1,n])+(p[1,n]-
                            p[0,n])+(q[0,n]-p[0,n]))/3;
                        q[2,n]=p[2,n]+(2*(p[3,n]-p[2,n])+(p[2,n]-
                            p[1,n])+(q[1,n]-p[1,n]))/4;
                        q[3,n]=p[3,n]+((p[3,n]-p[2,n])+(q[2,n]-
                            p[2,n]))/5;
                    }
                }break;
            }
        }
    }
}
```



```

case 1:
{
  for (int n=0;n<3;n++)
  {

q[0,n]=p[0,n]+3*(p[1,n]-p[0,n]);
q[1,n]=p[1,n]+2*(p[2,n]-p[1,n])+(p[1,n]-
p[0,n]);
q[2,n]=p[2,n]+(p[3,n]-p[2,n])+2*(p[2,n]-
p[1,n]);
q[3,n]=p[3,n]+3*(p[3,n]-p[2,n]);

  }
}break;

case 2:
{
q[0,0]=p[0,0];
q[0,1]=p[0,1];
q[0,3]=p[0,3];
for (int i=1;i<4;i++)
{
  for (int n=0;n<3;n++)
  {
    q[i,n]=p[i,n]+(p[i,n]+q[i-1,n]-2*p[i-1,n])
/(i+1);
  }
}
}break;

case 3:
{
for (int i=1;i<4;i++)
{
  for (int n=0;n<3;n++)
  {
    q[i,n]=p[i-1,n]+rho*(p[i,n]+p[i-1,n])-
sig*(q[i-1,n]-p[i-1,n]);
  }
}
}break;
}
}break;

```

```

case 2:
{
switch(yuzeyTuru2.SelectedIndex)
{
case 0:
{
for (int i=1;i<4;i++)
{
for (int n=0;n<3;n++)
{
q[i,n]=rho*(p[i,n]-p[0,n])+q[0,n];
}
}
}break;

case 1:
{
for (int n=0;n<3;n++)
{
q[0,n]=p[0,n]+3/sig*(p[1,n]-p[0,n]);
q[1,n]=p[1,n]+(2*p[2,n]+2*p[1,n]-4*p[0,n])
/(2*sig);

q[2,n]=p[2,n]+(1*p[3,n]+3*p[2,n]-4*p[0,n])
/(2*sig);

q[3,n]=p[3,n]+(p[3,n]-p[0,n])/sig;
}
}break;
}
}break;

case 1:
{
for (int i=1;i<4;i++)
{
for (int n=0;n<3;n++)
{
q[i,n]=sig*(q[i-1,n]-p[i-1,n])+p[i,n];
}
}
}break;

case 0:
{
}break;
}

```

```

//koordinat eksenlerini çizdirir

float[,] m={{-2,0,0,1},{2,0,0,1},{0,2,0,1},
            {0,-2,0,1},{0,0,3,1},{0,0,-3,1}};

viewingTrans(m);
g.DrawLine(pn1, m[0,0], m[0,1], m[1,0], m[1,1]);
g.DrawLine(pn2, m[2,0], m[2,1], m[3,0], m[3,1]);
g.DrawLine(pn3, m[4,0], m[4,1], m[5,0], m[5,1]);

//hesaplanan noktaları tutan matrisler

float[,] p0=new float[1,4];
float[,] p1=new float[1,4];
float[,] q0=new float[1,4];
float[,] q1=new float[1,4];
int say=0;

// yüzeyi çizdiren döngü

for (int i=0 ;i<100;i++)
{
    float t=i;
    t=t/100;
    bezier_curve(t,p,p1);
    viewingTrans(p1);
    if (i!=0) g.DrawLine(pn, p0[0,0], p0[0,1], p1[0,0],
                        p1[0,1]);

    Array.Copy(p1,p0,p1.Length);

    bezier_curve(t,q,q1);
    viewingTrans(q1);
    if (i!=0)g.DrawLine(pn2, q0[0,0], q0[0,1], q1[0,0],
                        q1[0,1]);

    Array.Copy(q1,q0,q1.Length);

    if (say==3) say=0;
    if (say==0) g.DrawLine(pn3, q0[0,0], q0[0,1],p0[0,0],
                        p0[0,1]);

    say++;
}
g.DrawLine(pn3, q0[0,0], q0[0,1], p0[0,0], p0[0,1]);

}

```

```
// yüzey türünün değişmesi ile oluşacak durumlar

private void yuzeyTuru1_SelectedIndexChanged(object sender,
                                             System.EventArgs e)
{
    switch(yuzeyTuru1.SelectedIndex)
    {
        case 3:
            {
                yuzeyTuru2.Visible=true;
                yuzeyTuru2.Items.Clear();
                yuzeyTuru2.Items.Add("1.durum");
                yuzeyTuru2.Items.Add("2.durum");
                yuzeyTuru2.Items.Add("3.durum");
                yuzeyTuru2.Items.Add("4.durum");
                yuzeyTuru2.SelectedIndex=0;
            }break;

        case 2:
            {
                yuzeyTuru2.Visible=true;
                yuzeyTuru2.Items.Clear();
                yuzeyTuru2.Items.Add("1.durum");
                yuzeyTuru2.Items.Add("2.durum");
                yuzeyTuru2.SelectedIndex=0;
            }break;

        case 1:
            {
                yuzeyTuru2.Visible=false;
                kontrol_q.Visible=true;
                panel2.Visible=false;
                sigma.Visible=true;
                ro.Visible=false;

            }break;

        case 0:
            {
                yuzeyTuru2.Visible=false;
                kontrol_q.Visible=true;
                sigma.Visible=false;
                panel2.Visible=true;
                ro.Visible=false;
            }break;
    }
    panell1.Invalidate();
}
```

```
private void yuzeyTuru2_SelectedIndexChanged(object sender,
                                             System.EventArgs e)
{
    switch(yuzeyTuru1.SelectedIndex)
    {
        case 3:
        {
            switch(yuzeyTuru2.SelectedIndex)
            {
                case 0:
                {
                    sigma.Visible=false;
                    ro.Visible=false;
                    kontrol_q.Visible=false;
                }break;

                case 1:
                {
                    sigma.Visible=false;
                    ro.Visible=false;
                    kontrol_q.Visible=false;
                }break;

                case 2:
                {
                    sigma.Visible=false;
                    ro.Visible=false;
                    kontrol_q.Visible=false;
                }break;
                case 3:
                {
                    sigma.Visible=true;
                    ro.Visible=true;
                    kontrol_q.Visible=true;
                    panel2.Visible=false;
                }break;
            }
        }break;

        case 2:
        {
            switch(yuzeyTuru2.SelectedIndex)
            {
                case 0:
                {
```

```

        sigma.Visible=false;
        ro.Visible=true;
        kontrol_q.Visible=true;
        panel2.Visible=false;
    }break;

    case 1:
    {
        sigma.Visible=true;
        ro.Visible=false;
        kontrol_q.Visible=false;
    }break;
    }
}break;
}
panell1.Invalidate();
}

//öteleme değerlerindeki değişiklikleri uygulama

private void yPoz_Click(object sender, System.EventArgs e)
{
    y_otele+=10;
    panell1.Invalidate();
}

private void yNeg_Click(object sender, System.EventArgs e)
{
    y_otele-=10;
    panell1.Invalidate();
}

private void xNeg_Click(object sender, System.EventArgs e)
{
    x_otele-=10;
    panell1.Invalidate();
}

private void xPoz_Click(object sender, System.EventArgs e)
{
    x_otele+=10;
    panell1.Invalidate();
}

```

```
// yakınlaştırma değerlerindeki değişimi uygulama

private void zoomNeg_Click(object sender, System.EventArgs e)

    {
        zoom_par-=10;
        panell.Invalidate();
    }

private void zoomPoz_Click(object sender, System.EventArgs e)

    {
        zoom_par+=10;
        panell.Invalidate();
    }

// döndürme açılarındaki değişimi uygulama

private void aci_x_ValueChanged(object sender,
                                System.EventArgs e)
    {
        cx=(float)Math.Cos(Math.PI*(double)aci_x.Value/180);
        sx=(float)Math.Sin(Math.PI*(double)aci_x.Value/180);
        dondur();
        panell.Invalidate();
    }

private void aci_y_ValueChanged(object sender,
                                System.EventArgs e)
    {
        cy=(float)Math.Cos(Math.PI*(double)aci_y.Value/180);
        sy=(float)Math.Sin(Math.PI*(double)aci_y.Value/180);
        dondur();
        panell.Invalidate();
    }

private void aci_z_ValueChanged(object sender,
                                System.EventArgs e)
    {
        cz=(float)Math.Cos(Math.PI*(double)aci_z.Value/180);
        sz=(float)Math.Sin(Math.PI*(double)aci_z.Value/180);
        dondur();
        panell.Invalidate();
    }
```

```
//p ve q matrislerindeki deęişimi uygulama

private void plx_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(plx.Text,0,0);
    panell.Invalidate();
}

private void ply_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(ply.Text,0,1);
    panell.Invalidate();
}

private void plz_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(plz.Text,0,2);
    panell.Invalidate();
}

private void p2x_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p2x.Text,1,0);
    panell.Invalidate();
}

private void p2y_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p2y.Text,1,1);
    panell.Invalidate();
}

private void p2z_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p2z.Text,1,2);
    panell.Invalidate();
}

private void p3x_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p3x.Text,2,0);
    panell.Invalidate();
}

private void p3y_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p3y.Text,2,1);
    panell.Invalidate();
}
```



```
private void p3z_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p3z.Text, 2, 2);
    panell.Invalidate();
}

private void p4x_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p4x.Text, 3, 0);
    panell.Invalidate();
}

private void p4y_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p4y.Text, 3, 1);
    panell.Invalidate();
}

private void p4z_Leave(object sender, System.EventArgs e)
{
    ver_gir_p(p4z.Text, 3, 2);
    panell.Invalidate();
}

private void q1x_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q1x.Text, 0, 0);
    panell.Invalidate();
}

private void q1y_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q1y.Text, 0, 1);
    panell.Invalidate();
}

private void q1z_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q1z.Text, 0, 2);
    panell.Invalidate();
}

private void q2x_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q2x.Text, 1, 0);
    panell.Invalidate();
}
```

```
private void q2y_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q2y.Text, 1, 1);
    panell.Invalidate();
}

private void q2z_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q2z.Text, 1, 2);
    panell.Invalidate();
}

private void q3x_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q3x.Text, 2, 0);
    panell.Invalidate();
}

private void q3y_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q3y.Text, 2, 1);
    panell.Invalidate();
}

private void q3z_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q3z.Text, 2, 2);
    panell.Invalidate();
}

private void q4x_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q4x.Text, 3, 0);
    panell.Invalidate();
}

private void q4y_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q4y.Text, 3, 1);
    panell.Invalidate();
}

private void q4z_Leave(object sender, System.EventArgs e)
{
    ver_gir_q(q4z.Text, 3, 2);
    panell.Invalidate();
}
```

```
// sigma değerindeki değişimi uygulama
private void sigma_Leave(object sender, System.EventArgs e)
{
    try
    {
        if (sigma.Text=="")
        {
            sig=0 ;
        }
        else
        {
            sig=(float)Double.Parse(sigma.Text);
        }
    }
    catch(Exception ex)
    {
        MessageBox.Show("LÜTFEN SAYISAL BİR DEGER
                        GIRIN!" + ex.Message);
    }
    panell.Invalidate();
}

//ro değerindeki değişimi uygulama
private void ro_TextChanged(object sender,
                            System.EventArgs e)
{
    try
    {
        if (ro.Text=="")
        {
            rho=0 ;
        }
        else
        {
            rho=(float)Double.Parse(ro.Text);
        }
    }
    catch(Exception ex)
    {
        MessageBox.Show("LÜTFEN SAYISAL BİR
                        DEGER GIRIN!" + ex.Message);
    }
    panell.Invalidate();
}
```



## **VII. SONUÇ**

Bu çalışma ile açılabilir Bézier yüzeyleri konusunda bir birikim oluşturmuş ve bunların kullanımındaki kolaylığı yaptığımız bilgisayar programı ile görmüş olduk. Bilgisayar destekli geometrik tasarım alanında yaptığımız çalışmanın, bu alanda çalışan ve çalışacak olan kişiler için faydalı olmasını istemekteyiz. Bu amaçla küçük bir adım attığımızı inanmaktayız. İleri ki çalışmalarda programın Bézier Yüzeyleri dışında NURBS benzeri yöntemleri de kapsayacak şekilde genişletilmesinin faydalı olacağı kanısındayız.

## KAYNAKLAR DİZİNİ

- Aumann, G., 2003** A Simple algorithm for designing developable Bézier surfaces. *Computer Aided Geometric Design*, 20(8-9): 621-632
- Biran, L., 1970** Diferensiyel Geometri Dersleri, İstanbul Üniversitesi Fen Fak. Yayınları, İstanbul.
- Farin, G., 1990** Curves and Surfaces for Computer-Aided Geometric Design, A Practical Guide, Academic Press Inc.
- Gallier, J., 1999** Curves and Surfaces in Geometric Modeling, Theory and Algorithms, Morgan Kauffman.
- Joy K., 2000** On-Line Geometric Modeling Notes, Computer Science Department, University of California, Davis
- Salomon, D., 2006** Curves and Surfaces for Computer Graphics, Springer.
- Shene, C.-K., 2007** Introduction to Computing with Geometry Notes, Michigan Technological University
- Tantay, B., 1998** Bilgisayar Destekli Geometrik Tasarım ve Hareket Geometrisi, Doktora Tezi.
- Vera, B. A., 1992,** Computer Graphics and Geometric Modeling for Engineers, John Wiley and Sons, Inc.
- Yamaguchi, F., 1989,** Curves and Surfaces in Computer Aided Geometric Design, Springer-Verlag.
- Zhang, X.-W. ve Wang, G.-J., 2006** A new algorithm for designing developable Bézier surface. *Journal of Zhejiang University*, 7(12):2050-2056

## ÖZGEÇMİŞ

Türkiye Cumhuriyeti vatandaşı olan Abdullah Güngöz 1982 yılında Eskişehir’ de doğdu. İlkokulu, Alanya Hayate Hanım İlkokulunda okudu. İlk öğretim ve Lise eğitimini Alanya Ayşe Melahat Erkin Anadolu Lisesi’ nde tamamlayarak 2000 yılında Ege Üniversitesi Fen Fakültesi Matematik Bölümü’ nü kazandı. İngilizce sınavı ile hazırlık sınıfından muaf oldu. 2005 yılında Bilgisayar Ağırlıklı Matematik’ den “iyi” dereceyle mezun olarak aynı yıl Geometri anabilim dalında yüksek lisans eğitimine başladı. Şuan özel bir eğitim kurumunda öğretmenlik yapmakta ve eğitim hayatına devam etmektedir.