

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**BAZI GEOMETRİK YAPILARDAKİ HAREKETLERİN  
BİLGİSAYAR PROGRAMIYLA İZLENMESİ**

**YÜKSEK LİSANS TEZİ**

**Nihal AKGÖKÇE**

**Enstitü Anabilim Dalı : MATEMATİK**

**Tez Danışmanı : Yrd. Doç. Dr. Hüseyin KOCAMAN**

**Temmuz 2008**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

BAZI GEOMETRİK YAPILARDAKİ HAREKETLERİN  
BİLGİSAYAR PROGRAMIYLA İZLENMESİ

YÜKSEK LİSANS TEZİ

Nihal AKGÖKÇE

Enstitü Anabilim Dalı : MATEMATİK

Bu tez 30 /07 /2008 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



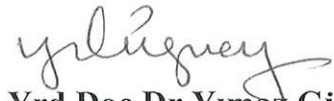
Yrd.Doç.Dr. Hüseyin Kocaman

Jüri Başkanı



Prof.Dr. Abdullah Yıldız

Üye



Yrd.Doç.Dr. Yılmaz Güney

Üye

## **TEŐEKKÜR**

Tez alıŐma aŐamamda beni ynlendirip alıŐmalarıma yardımcı olan Sayın DanıŐman Hocam Yrd. Do. Dr. Hseyin KOCAMAN'a

Geometri ile Bilgisayar destekli tasarım arasındaki iliŐkiyi kavrayıp bu alanda alıŐmama yardımcı olan Sayın Hocam Prof. Dr. Ali ALIŐKAN'a ve yaptığım bu alıŐmanın baŐlangıcından sonuna kadar bana her alanda yardımcı olan deėerli katkılarını benden esirgemeyen, Sayın Hocam Öğr. Görevlisi Filiz GLSOY'a teŐekkrlerimi sunarım.

# İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER .....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ .....	vii
TABLolar LİSTESİ.....	ix
ÖZET.....	x
SUMMARY.....	xi
BÖLÜM 1.	
GİRİŞ.....	1
1.1.Amaçlar.....	1
BÖLÜM 2.	
KOORDİNATLAR.....	3
2.1. Giriş.....	3
2.2. Kutupsal Koordinatlar.....	3
2.3. Silindirik Koordinatlar.....	4
2.4. Küresel Koordinatlar.....	6
BÖLÜM 3.	
BİLGİSAYAR DESTEKLİ PARAMETRİK EĞRİ TASARIMI.....	9
3.1. Temel Kavramlar.....	9
3.2. Bilgisayar Destekli Tasarımda Parametrik Gösterim.....	11
3.2.1. Parametrik gösterimin avantajları.....	12
3.2.2. Bilgisayar destekli tasarımda parametre dönüşümünün uygulanması.....	14

3.3. Eğri Tasarımında Kullanılan Yöntemler.....	16
3.3.1. Serbest biçimdeki eğri ve yüzeylerin matematiksel modellerinin gelişimi .....	17
3.3.2. Bezier eğriler.....	20
3.3.3. Spline eğriler.....	24
3.3.4. Rasyonel eğriler.....	26
3.4. Eğrilerde Dönüşüm İşlemleri.....	28
3.4.1. Öteleme.....	29
3.4.2. Ölçekleme.....	29
3.4.3. Döndürme.....	31
3.4.4. Yansıma.....	31
3.4.5. Eğrilerde dönüşüm işlemleriyle ilgili uygulamalar.....	32
BÖLÜM 4.	
PARAMETRİK YÜZEY TASARIMI.....	34
4.1. Temel Kavramlar.....	34
4.2. Yüzeylerin Uygulamada Kullanımı.....	36
4.2.1. Bezier yüzeyler.....	37
4.2.2. Spline yüzeyler.....	39
4.2.3. Rasyonel yüzeyler.....	41
4.2.4. Dönel yüzeyler.....	42
4.2.5. Bazı özel yüzeyler.....	43
4.2.5.1. Küre yüzeyi.....	43
4.2.5.2. Silindir yüzeyi.....	44
4.2.5.3. Koni yüzeyi.....	46
4.3. Yüzeylerde Dönüşüm İşlemleri.....	48
4.3.1. Öteleme işlemi.....	48
4.3.2. Ölçekleme işlemi.....	48
4.3.3. Döndürme işlemi.....	49
4.4. Dönüşüm İşlemlerinin Yüzelere Uygulanması.....	50

BÖLÜM 5.	
SONUÇLAR VE ÖNERİLER.....	53
KAYNAKLAR.....	54
EKLER.....	56
ÖZGEÇMİŞ.....	59

## SİMGELER VE KISALTMALAR LİSTESİ

$W$	: Ağırlık
$N_{i,k}(s)$	: Baz fonksiyonu
$B_i^n(t)$	: Bernstein polinomu
$\theta$	: Dönme operatörü
$P^*(t)$	: Dönüşümden sonra elde edilmiş eğri
$x_i, y_i$	: Düğüm vektörleri
$\in$	: Elemanıdır
$\mathbb{R}^n$	: n- Boyutlu uzay
$\mathbb{R}$	: Reel sayılar
$\Delta$	: Sonlu fark operatörü
$P(t), \alpha(t)$	: t- parametesine bağlı eğri
$S$	: Ölçekleme
$T$	: Öteleme
$t^*$	: Yeni parametre

## ŞEKİLLER LİSTESİ

Şekil 3.1.	İki yüzeyin arakesitinin eğri olduğunu gösteren örnek.....	10
Şekil 3.2.	İki boyutlu uzayda çemberin parametrik gösterimi.....	11
Şekil 3.3.	Değişen parametrelere göre eğrilerin alabileceği farklı şekillerin Helis üzerinde gösterimi.....	13
Şekil 3.4.	Parametre dönüşümünü gösteren örnek.....	14
Şekil 3.5.	Kapalı Bezier eğrisi ve birinci ile ikinci kontrol noktaları aynı konumda olan eğri.....	22
Şekil 3.6.	Farklı iki bezier eğrisinin birleşimini gösteren bezier eğrisi.....	22
Şekil 3.7.	Fiyonk biçimli bezier eğrisi .....	22
Şekil 3.8.	Üçüncü dereceden bezier eğri örneği.....	23
Şekil 3.9.	Spline eğri örneği.....	26
Şekil 3.10.	Rasyonel eğri örneği.....	28
Şekil 3.11.	Elips ve çemberin x eksenini etrafında dönüşümünün izlenmesi.....	30
Şekil 3.12.	Eğrinin ötelenerek ilerlemesine örnek.....	32
Şekil 3.13.	Eğrinin hem ötelenmesi hem de ölçeklendirilmesine örnek .....	33
Şekil 3.14.	Eğrinin döndürülmesini gösteren örnek.....	33
Şekil 3.15.	Eğrinin ötelenmesi ve döndürülmesini gösteren örnek.....	33
Şekil 4.1.	Yüzeyi oluşturan noktaları ve eğrileri gösteren örnek.....	36
Şekil 4.2.	4x4 Bezier yüzeyi için tanımlanan poligon ağının şeması.....	38
Şekil 4.3.	Bezier eğrileri kullanılarak çizilen bezier yüzey örneği.....	39
Şekil 4.4.	Eğrinin L doğrultusunda döndürülmesiyle oluşan dönel yüzey.....	42
Şekil 4.5.	Dönel yüzey örneği.....	43
Şekil 4.6.	Küre yüzeyi örneği.....	44
Şekil 4.7.	Silindir yüzeyini oluşturan eğri ve ana doğrular.....	45
Şekil 4.8.	Silindir yüzey örneği.....	46



Şekil 4.9.	Koninin dayanak eğrisi ve ana doğrusu.....	46
Şekil 4.10.	Koni yüzey örneği.....	47
Şekil 4.11.	Silindirik koordinatlarda döndürülerek ve ölçeklendirilerek çizilmiş hareketli yüzey .....	50
Şekil 4.12.	Küresel koordinatlarda ölçeklendirilerek çizilmiş hareketli yüzey .....	50
Şekil 4.13.	Küresel koordinatlarda ötelenerek çizilmiş hareketli yüzey.....	50
Şekil 4.14.	Enneper yüzey örneği.....	51
Şekil 4.15.	Catalan yüzey örneği.....	51
Şekil 4.16.	Catenoid yüzey örneği.....	52
Şekil 4.17.	Helisoid yüzey örneği.....	52

## TABLolar LİSTESİ

Tablo 3.1.	Rasyonel olan ve Rasyonel olmayan Bezier ve B-Spline eğrilerinin denklemleri.....	27
Tablo 4.1.	Rasyonel olan ve Rasyonel olmayan Bezier ve B-Spline yüzeylerin denklemleri.....	41

## ÖZET

Anahtar kelimeler: Koordinat, Eğri, Yüzey, Hareketli grafik

Maple programının, geometrik yapılar üzerindeki uygulamaların sunulduğu bu çalışmanın birinci bölümünde, çalışmanın amaçlarından bahsedilmiş. İkinci bölümünde, programlama dilinin temel mantığını oluşturan, koordinat sistemleri, hakkında bilgiler verilmiştir. Verilen bu bilgilerin amacı, ekler kısmında yazılan kodları desteklemektir.

Üçüncü bölümde, tasarım uygulamalarına çok elverişli olan Bezier, Spline...gibi eğri türleri anlatılmış. Eğriler üzerinde gerçekleştirilen dönüşüm işlemlerinden bahsedilmiş ve son olarak verilen örneklerle dönüşüm işlemleri ile geometrik yapıların hareketliliği arasında bağlantı kurulmuştur.

Dördüncü bölümde de üçüncü bölüme paralel olarak eğriler konusunda anlatılanlar, yüzeylere aktarılmıştır ve en son bölümünde de çalışmanın amacı olan hareketli grafiklerle anlatılanlar desteklenmiştir.

# **SOME GEOMETRIC STRUCTURE'S MOVEMENT TO FOLLOW WITH COMPUTER PROGRAMING**

## **SUMMARY**

Keywords: Coordinate, Curve, Surface, Animation

In this thesis's first chapter, explained geometrical structure application with maple programming language. Second chapter, gived information about coordinate systems, wich is the basic logical structure of maple programming language.

In third chapter, explained Bezier and Spline curves, and then explained translation process. Lastly, gived geomerical structure animation with translation process.

And finaly explained Bezier, Spline, Rotational surrrfaces, and also explained ttthis surfaces' animation with animation graphics.

# BÖLÜM 1. GİRİŞ

## 1.1. Amaçlar

Bilgisayarın her alandaki gelişmesinin bir sonucu olarak Bilgisayar destekli çizim ve Tasarım uygulamalarının, kullanımı da yaygınlaşmıştır. Çünkü bu tarz programlama dilleri, tasarımı daha kısa sürede hazırlayabilme imkânı sunar. Bunun yanında görsellikle teorik bilgiyi birleştirdiğimizde anlatmak istediklerimizin verimliliği artar.

Bu anlatılanlara paralel olarak Matematiğin dallarından biri olan geometri ve geometrinin alt dalları olan diferansiyel geometri ve hareket geometrisinin günlük hayata ve uygulamaya çok kolay aktarılabilmesi bilinmektedir. Bu teorik bilgileri, aktarma işlemi de yukarıda da anlatıldığı gibi en ideal bilgisayar teknolojisi ve bu teknolojinin, getirdiği kolaylıklarla gerçekleştirilebilir. Dolayısıyla; kullanılan teorik bilgilerin klasik yaklaşımları ile bu yaklaşımların bilgisayar uygulamaları, bir araya getirilmiş ve hedeflenen davranışlara ulaşılmış olur.

Yeni üretilecek olan ya da önceden üretilmiş olan teorik yaklaşımların, bilgisayar teknolojisine uygun hale getirilmesinin önemi giderek artmaktadır. Bu sebeple, yapılan bu çalışmada da geometrik bilgilerle, tasarım uygulamalarının birlikte kullanımı maple programlama dili aracılığıyla gerçekleştirilmiştir. Bu kullanım sırasında da ilk olarak koordinat bilgilendirmesi yapılmıştır. Çünkü Maple programlama dili komutları, koordinat kullanımını dikkate alır. Dolayısıyla ilk bölüm, Kutupsal, Silindirik ve Küresel koordinat tanımlamalarından oluşturulmuştur.

İkinci bölümde de Parametrik Eğri Tasarımından bahsedilmiş. Parametrik kullanım, bu kullanımın geometriye kattığı avantajlar ve eğri çizim hızını arttıracak yaklaşımlar anlatılmıştır. Ardından da anlatılan bu teorik bilgiler, geometrik tasarımda çok kullanılan eğri tiplerinden olan Bezier, Spline ve Rasyonel eğrilere uygulanmıştır. İkinci bölümün son bölümünde de düzlemde hareketliliği sağlayan Dönüşüm işlemleri, anlatılarak bu işlemlerin, eğriler üzerinde ne gibi hareketlikler, ürettiği sergilenmiştir.

Üçüncü bölümde, Parametrik Yüzey Tasarımı şeklinde yapılandırılmıştır. Yapılan teorik yüzey tanımlamalarından sonra ikinci bölümde olduğu gibi bu tanımlamalar, geometrik tasarım uygulamalarında çok kullanılan Bezier, Spline, Rasyonel ve Dönel yüzeyler ile bilinen bazı geometrik yüzeyler (Küre, Silindir, Koni) üzerine uygulanmıştır. Bu bölümün son alt bölümünde de Uzayda hareketliliği sağlayan dönüşüm işlemleri, anlatılarak bu işlemlerin, yüzeyler üzerinde ne gibi hareketler ürettiği sergilenmiştir.

Bölüm dördte ise anlatılanları desteklemek için verilen şekillerin, Maple programlama dilindeki kodlarına, yer verilmiştir.

## BÖLÜM 2. KOORDİNATLAR

### 2.1. Giriş

Bu bölümde koordinat bilgilerine yer verilmesinin nedeni, kitapta anlatılanları görsellikle destekleyebilmek için kullanılan maple programlama dili kodlarının Kutupsal, Silindirik ve Küresel koordinatları baz alarak yapılandırılmasıdır. Diğer bölümlerde bu bölüm bilgilerinden pek yararlanılmamış sadece son bölümde yer verilen kodlar, bu bölümde anlatılan koordinat bilgileriyle desteklenmiştir.

### 2.2. Kutupsal Koordinatlar

Düzlemde bir AOP açısı ve OP uzunluğu alındığında, OP uzunluğu  $r(P)$  ile gösterilsin ve  $|OP| = r(P)$  olsun. AOP açısının  $[0, 2\pi]$  Aralığındaki ölçüsü de  $\Theta(P)$  ile gösterilsin. Bu durumda, P noktası koordinat olarak  $(r(P), \Theta(P))$  ikilisi şeklinde yazılır ve  $(r(P), \Theta(P))$  ikilisine P noktasının, kutupsal koordinatları (bileşenleri) denir.  $(r(P), \Theta(P))$  ikilisi şeklinde alınan P noktası, aynı noktaya karşılık gelecek şekilde;  $((r(P), \Theta(P)) + 2k\pi), (-r(P), \Theta(P)) + \pi$  ve  $(-r(P), \Theta(P)) + \pi + 2k\pi$  biçimlerinde de yazabilir. Bu ikililerde yine P noktasının, kutupsal koordinatları olarak adlandırılır.

Buradan anlaşılacağı üzere düzlemde alınan bir P noktasının, kutupsal koordinatları tek değildir. Fakat kutupsal koordinatları verilen bir nokta tek olarak belirlenir.  $(r, \Theta)$  fonksiyonlarına, kutupsal koordinat fonksiyonları adı verilir ve  $(r, \Theta)$  ikilisine de kutupsal koordinat sistemi denir.

P noktasının, dik koordinat sistemindeki bileşenleri,  $(p_1, p_2)$  ve kutupsal koordinat sistemindeki bileşenleri,  $(r(P), \Theta(P))$  olmak üzere; bir noktanın, dik koordinat sistemindeki bileşenleri, kutupsal koordinat sistemindeki bileşenleri türünden aşağıdaki gibi yazılabilir.

$$p_1 = r(P) \cdot \cos(\Theta(P))$$

$$p_2 = r(P) \cdot \sin(\Theta(P))$$

$$p_1 = X_1(P)$$

$$p_2 = X_2(P) \text{ olduğundan}$$

$$X_1(P) = r(P) \cdot (\cos \Theta(P))$$

$X_2(P) = r(P) \cdot (\sin \Theta(P))$  eşitlikleri yazılabilir. Dolayısıyla düzlemde alınan her P noktası için:

$$x_1 = r \cdot \cos \Theta$$

$x_2 = r \cdot \sin \Theta$  eşitlikleri elde edilir. Bu eşitliklere de dik koordinat fonksiyonlarını, kutupsal koordinat fonksiyonlarına bağlı olarak veren, fonksiyonlar denir.

### 2.3. Silindirik Koordinatlar

Silindirik koordinat sistemini de yine dik koordinat sisteminden yararlanarak açıklayacağız. Dik koordinat sisteminde herhangi bir noktanın koordinatlarını,  $(x_1, x_2, x_3)$  şeklinde gösterelim ve Uzayda herhangi bir P noktası verildiğinde P noktasının,  $x_1 O x_2$  düzlemindeki dik izdüşümüne, T diyelim.  $x_1 O x_2$  düzleminde,  $[O x_1]$  ışını, kutup eksenini alarak elde edilen kutupsal koordinat sistemine göre, T noktasının kutupsal koordinatları,  $(u_1, u_2, u_3)$  olsun.



Dik koordinat sisteminde, P noktasına karşılık gelen üçlü  $(p_1, p_2, p_3)$  olmak üzere,  $u_3 = p_3$  olarak alınsın. Böylece P noktasına karşılık  $(u_1, u_2, u_3)$  üçlüsü elde edilir ve bu üçlüyü oluşturan  $(u_1, u_2, u_3)$  sayılarına, P noktasının, silindirik koordinatları adı verilir ve  $(u_1, u_2, u_3)$  sayıları da aşağıdaki fonksiyonlar şeklinde yazılır.

$$r = \mathbb{R}^3 \rightarrow \mathbb{R}, \quad \Theta = \mathbb{R}^3 \rightarrow \mathbb{R}, \quad z = \mathbb{R}^3 \rightarrow \mathbb{R} \quad \text{olmak üzere;}$$

$$r(P) = u_1, \quad \Theta(P) = u_2, \quad z(P) = u_3$$

ve bu fonksiyonları oluşturan  $(r, \Theta, z)$  üçlüsüne de silindirik koordinat sistemi, adı verilir. P noktasının,  $p_1, p_2, p_3$  koordinatları ile  $u_1, u_2, u_3$  silindirik koordinatları arasında aşağıdaki eşitlikler bulunur.

$$p_1 = u_1 \cos(u_2)$$

$$p_2 = u_1 \sin(u_2)$$

$$p_3 = u_3$$

$$u_1 = r(P)$$

$$u_2 = \Theta(P)$$

$$u_3 = z(P)$$

olduğundan bu eşitliklerden yararlanarak  $(r, \Theta, z)$  koordinat sistemi ile  $(x_1, x_2, x_3)$  koordinat sistemi arasında:

$x_1 = r \cdot \cos(\Theta)$ ,  $x_2 = r \cdot \sin(\Theta)$ ,  $x_3 = z$  eşitlikleri elde edilir. Aynı açıya ait kosünüs değerinin karesi ile sinüs değerinin karesinin toplamı bir olduğundan,  $x_1$  ve  $x_2$  eşitliklerinde her iki tarafın karesini alınarak eşitliklerin karşılıklı terimleri toplanırsa  $x_1^2 + x_2^2 = r^2$  eşitliği elde edilir. Yapılan işlemler sonucunda  $r$  ve  $\Theta$  koordinat bilgileri, aşağıdaki gibi elde edilir.

$$r = \sqrt{x_1^2 + x_2^2}$$

$$\frac{x_2}{x_1} = \tan(\Theta)$$

$$\arctan \frac{x_2}{x_1} = \Theta \text{ eşitlikleri elde edilir.}$$

Sonuçta silindirik koordinatlar, aşağıdaki gibi yazılabilir.

$$r = \sqrt{x_1^2 + x_2^2}$$

$$\Theta = \arctan \frac{x_2}{x_1}$$

$$z = x_3$$

## 2.4. Küresel Koordinatlar

Uzayda bir P noktası aldığımızda, bu noktanın orijine olan uzaklığı  $|OP|$  olsun ve  $|OP| = h_1$  diye gösterilsin. P noktasının,  $x_1Ox_2$  düzlemindeki dik izdüşümü, Q ve  $A=(1,0,0)$  olmak üzere AOQ yönlü açısının ölçüsü,  $h_2$  olsun.  $B=(0,0,1)$  olmak üzere BOP yönlü açısının ölçüsü de  $h_3$  olsun.  $h_1, h_2, h_3$  sayılarının verilmesiyle P noktası, tek olarak verilmiş olur ve bu üç sayıya, P noktasının, küresel koordinatları adı verilir. Aşağıdaki şekilde görüldüğü gibi O noktasında  $h_1$  sayısı 0'dır.  $h_2$  ve  $h_3$  sayıları istenildiği gibi seçilebilir.

$$\alpha: \mathbb{R}^3 \longrightarrow \mathbb{R}, \quad \Theta: \mathbb{R}^3 \longrightarrow \mathbb{R}, \quad \varnothing: \mathbb{R}^3 \longrightarrow \mathbb{R} \text{ olmak üzere;}$$

$h_1, h_2, h_3$  sayıları,  $\alpha(P) = h_1, \quad \Theta(P) = h_2, \quad \varnothing(P) = h_3$  şeklinde gösterilir ve bu fonksiyonlara, uzayda küresel koordinat fonksiyonları denir.  $(\alpha, \Theta, \varnothing)$  üçlüsüne de uzayda küresel koordinat sistemi adı verilir. Uzayda dik koordinat sistemi,

$(x_1, x_2, x_3)$  olmak üzere, küresel koordinat sistemi,  $\alpha$ ,  $\Theta$  ve  $\emptyset$  türünden aşağıdaki gibi yazılabilir.

$$x_1 = \alpha \cdot \cos(\Theta) \cdot \sin(\emptyset)$$

$$x_2 = \alpha \cdot \sin(\Theta) \cdot \sin(\emptyset)$$

$$x_3 = \alpha \cdot \cos(\emptyset)$$

Bu eşitliklerde  $x_1$ ,  $x_2$  ve  $x_3$ 'te her iki tarafın karesini alırsak;

$$x_1^2 + x_2^2 + x_3^2 = \alpha^2 \cdot \cos^2(\Theta) \cdot \sin^2(\emptyset) + \alpha^2 \cdot \sin^2(\Theta) \cdot \sin^2(\emptyset) + \alpha^2 \cdot \cos^2(\emptyset)$$

$$x_1^2 + x_2^2 + x_3^2 = \alpha^2 \cdot [\cos^2(\Theta) \cdot \sin^2(\emptyset) + \sin^2(\Theta) \cdot \sin^2(\emptyset) + \cos^2(\emptyset)]$$

$$x_1^2 + x_2^2 + x_3^2 = \alpha^2 \cdot [\sin^2(\emptyset) (\cos^2(\Theta) + \sin^2(\Theta)) + \cos^2(\emptyset)]$$

$$x_1^2 + x_2^2 + x_3^2 = \alpha^2 [\sin^2(\emptyset) + \cos^2(\emptyset)] = \alpha^2$$

Dolayısıyla  $\alpha$  koordinatı,  $\alpha = \sqrt{x_1^2 + x_2^2 + x_3^2}$  olur. Yapılan işlemler sonucunda:

$$\frac{x_2}{x_1} = \tan(\Theta)$$

$$\arctan \frac{x_2}{x_1} = \Theta$$

$$\arccos \frac{x_3}{\alpha} = \emptyset \text{ denklemleri elde edilir.}$$

Sonuç olarak küresel koordinatlar, aşağıdaki gibi yazılır.

$$\alpha = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

$$\Theta = \arctan \frac{x_2}{x_1}$$

$$\varnothing = \arccos \frac{x_3}{\sqrt{x_1^2 + x_2^2 + x_3^2}}$$

## BÖLÜM 3.

### BİLGİSAYAR DESTEKLİ PARAMETRİK EĞRİ TASARIMI

#### 3.1. Temel Kavramlar

Bu bölümde, eğrinin parametrik tanımı ve eğri üretme yöntemlerinden bahsedilmiş  $\mathbb{R}^3$ 'te eğri,  $\mathbb{R}$ 'nin bir  $I$  açık aralığından  $\mathbb{R}^3$  uzayına giden sürekli olan bir  $\alpha$ , fonksiyonun  $I$ , aralığındaki görüntüsü olarak tanımlanır.

$\alpha : I \rightarrow \mathbb{R}^3$ , fonksiyonu verildiğinde  $\alpha(I)$  kümesi de verilmiş olacağından ve  $\alpha(I)$  nın özellikleri,  $\alpha$  fonksiyonu yardımıyla incelendiğinden,  $\alpha$  fonksiyonu bir eğri olarak nitelendirilir.  $t \in I$  için  $\alpha(t) \in \mathbb{R}^3$  olduğundan;

$\alpha(t) = (\alpha_1(t), \alpha_2(t), \alpha_3(t))$  biçiminde üç elemanlı olmak zorundadır. Burada  $\alpha_1(t) \in \mathbb{R}$ ,  $\alpha_2(t) \in \mathbb{R}$ ,  $\alpha_3(t) \in \mathbb{R}$  olduğundan, her bir  $\alpha$  eğrisi,

$$\alpha_1 : I \rightarrow \mathbb{R}, \alpha_2 : I \rightarrow \mathbb{R}, \alpha_3 : I \rightarrow \mathbb{R}$$

Şeklinde yazılabilir ve bu fonksiyonlara  $\alpha$  eğrisinin bileşenleri adı verilir.  $\alpha$  eğrisinin sürekli olması,  $\alpha$ 'nın her bir bileşeninin, ayrı ayrı sürekli olması anlamına gelir. Yani  $\alpha_1, \alpha_2, \alpha_3$  fonksiyonları da sürekli olmalıdır.

Bu söylenenler bir başka deyişle, şu şekilde de ifade edebilir. Üç boyutlu uzayda bir  $\alpha$  eğrisi,  $I$  bir aralık,  $\mathbb{R}$ 'de reel eken olmak üzere  $\alpha : I \rightarrow \mathbb{R}^3$  şeklinde tanımlanır.  $t$  alınan  $I$ , açık aralığının bir elemanı olmak üzere ( $t \in \mathbb{R}$ )  $\alpha$  eğrisinin, aşağıdaki gibi üç koordinatı vardır.

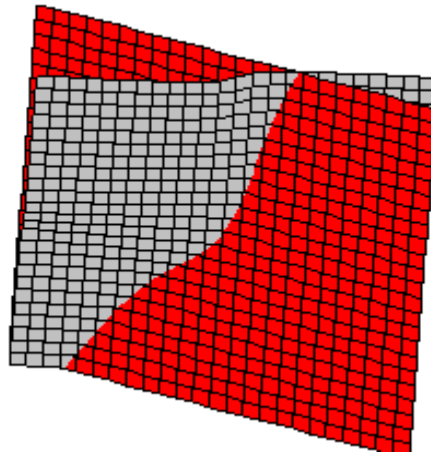
$$\alpha(t) = (\alpha_1(t), \alpha_2(t), \alpha_3(t))$$

Burada  $\alpha_i: I \rightarrow \mathbb{R}^3$  şeklinde tanımlanır.  $t$  değişkeni, zamanı temsil etmek üzere  $\alpha(t)$  aldığı farklı değerlere göre üç boyutlu uzayda bir eğri çizdirir.  $\alpha$  eğrisi, türevlenebilen bir fonksiyon olduğundan koordinatları olan  $\alpha_i$ 'ler de türevlenebilir.

$\mathbb{R}^2$  uzayında ise bir  $\alpha$  eğrisinin, iki tane bileşeni vardır. Yani düzlemde bir  $\alpha$  eğrisi,

$$\alpha: I \rightarrow \mathbb{R}^2, \alpha_1: I \rightarrow \mathbb{R}^2, \alpha_2: I \rightarrow \mathbb{R}^2$$

Sürekli fonksiyonlar olmak üzere,  $\alpha = (\alpha_1, \alpha_2)$  şeklindedir. Uzayda bir doğrudan aslında özel bir eğridir.  $P$  ve  $R$  iki farklı nokta olmak üzere,  $\alpha$  eğrisi;  $\alpha(t) = P + t \cdot PR$  şeklinde yazılabilir. Bu şekilde yazılan  $\alpha$  eğrisi, aslında  $P$  ve  $R$  noktalarından geçen bir doğrudur. Uzayda eğri oluşturmanın bir diğer yolu da iki yüzeyin arakesitini almaktır. Herhangi iki yüzeyin, kesiştikleri bölgede, her zaman için bir eğri oluşur. Bu anlatılanları örnekleyen maple uygulaması, iki yüzeyin kesişimi şeklinde gösterilmiştir.



Şekil 3.1. İki yüzeyin arakesitinin eğri olduğunu gösteren örnek

### 3.2. Bilgisayar Destekli Tasarımda Parametrik Gösterim

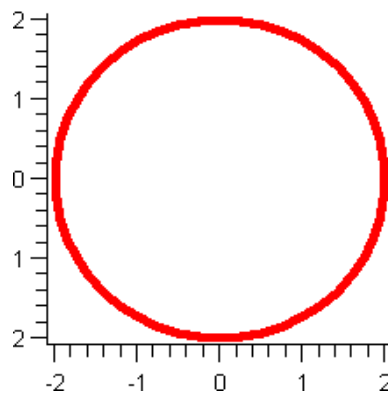
Bu bölümde, bilgisayarda bir eğriyi çizdirilmesinde hassaslığı ve hızı arttıracak bazı yaklaşımlar ve bunların geliştirilmesinde dikkat edilmesi gereken bazı geometrik özellikler anlatılarak Parametrik gösterimin avantajları ve bilgisayar destekli tasarımda parametre dönüşümünün uygulanması işlemlerinden bahsedilmiştir.

Eğri üzerindeki koordinat tanımlamalarında t'yi dördüncü parametre olarak seçmek istediğimizde diğer üç koordinatı, parametrik formda yazarız ve bu yazılımda eğri üzerindeki herhangi bir noktayı,

$$P(t) = (x(t), y(t), z(t))$$

Şeklinde vektörel fonksiyon olarak ifade ederiz. Parametrik denklemler, genellikle 0 ile 1 arasında değişen değerler alabilecek şekilde tanımlanırlar. Örneğin merkez koordinatları, orijin olan ve xy düzleminde bulunan bir çember, parametrik olarak aşağıdaki gibi gösterilir.

$$X(t) = r \cdot \cos(\theta) \quad y(t) = r \cdot \sin(\theta) \quad z(t) = 0;$$



Şekil 3.2. İki boyutlu uzayda çemberin parametrik gösterimi

### 3.2.1. Parametrik gösterimin avantajları

Eğrilerin ya da yüzeylerin, grafik uygulamalarında kullanabileceğimiz, en uygun yöntem, parametrik denklem uygulamalarıdır. Çünkü bu yöntemin getirdiği bir takım avantajlar vardır. Bir eğrinin, parametrik ifadesinin, parametrik olmayan ifadesine göre avantajları, Yamaguchi, tarafından 1989'da belirtilmiştir. Bu avantajlar, daha iyi anlaşılması için verilecek tanımın ardından aşağıdaki gibi sıralanabilir.

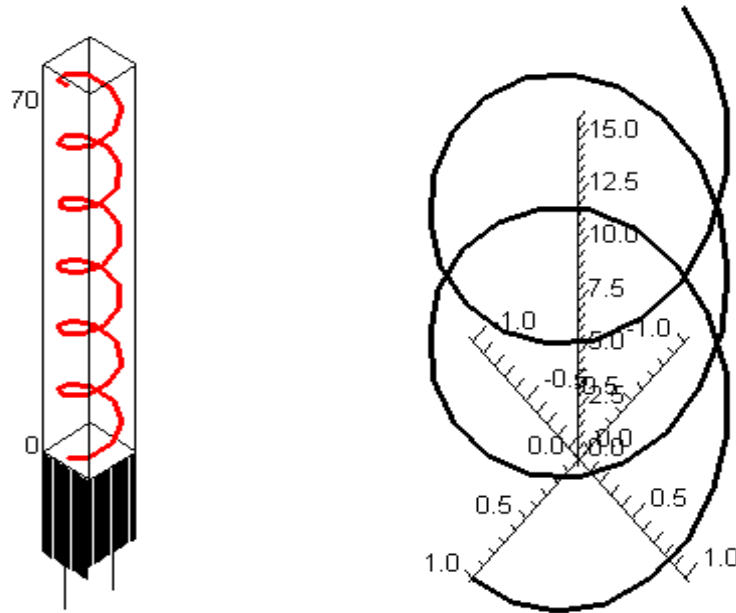
**Tanım 2.1:**  $a \leq t \leq b$  olmak üzere verilen bir  $P(t)$  eğrisini,  $0 \leq t \leq 1$  olacak biçimde  $t$  cinsinden yeniden parametrelendirme işlemine, eğriyi normalize etme işlemi denir. [Yamaguchi(1989), Mortensan(1989), Çalışkan(2000) Diferensiyel geometri ders notları].

$P(t) = (x(t), y(t), z(t))$  eğrisi için, eğrisi için,  $x, y, z$ 'nin ve  $\dot{x}, \dot{y}, \dot{z}$  nün ve eğer gerekirse  $\ddot{x}, \ddot{y}, \ddot{z}$  v.b nin  $t$ 'ye göre grafikleri çizilip, karakterleri belirlenebilir. Böylece teğet ve eğrilik süreklilikleri daha kolay belirlenebilir. Ayrıca, bunlara uygun parametre aralıkları tespit edilebileceği gibi, bunların nerelerde ekstremum değerler aldığı, hangi  $t$  parametre değerleri için hangi koordinat fonksiyonunun eğrinin şekli üzerinde daha fazla etkisi olduğu belirlenebilir. Yani parametre aralıklarında bileşenlerin artış veya azalış miktarları ve bunların her bir veriyle kıyaslanmaları kolaylaşır. Bir sonraki sayfada bu söylenenler, değişen  $t$ , parametresi için gerçekleştirilmiş birinci örnekte, 0 ile  $12\pi$  aralığı için çizilen bir helis eğrisi, ikinci örnekte ise 0 ile  $5\pi$  aralığı için çizilen başka bir helis eğrisi gösterilmiştir.

Başka bir örnekte de eğri çizim hızına etki edebilecek farklılıklara bakalım ve  $P(t) = (1-t^2 + t)P_0 + (t^2 - t)P_1$  Eğrisini ele alalım.  $P(0) = P_0$  ve  $P(1) = P_0$  dır. Burada  $P_0$ ın seçimi eğriyi değiştirebilir. Örneğin  $P_0 = 0$  alırsak,  $P(t) = (t^2 - t)P_1$  şekline döner. Bu ifade de  $P_1 = (P_{1x}, P_{1y})$  dersek



$P(t) = ( (t^2 - t) P_{1x}, (t^2 - t) P_{1y} )$  olur ve koordinatlar,  
 $x(t) = (t^2 - t) P_{1x}$  ve  $y(t) = (t^2 - t) P_{1y}$  dir. Örneğin  $x(t)$  nin T tepe noktasının koordinatları,  $T = (\frac{1}{2}, \frac{-P_{1x}}{4})$  ve y eksenini kesim noktalarına karşılık gelen t parametre değerleri,  $x(t) = (t^2 - t) P_{1x} = 0$  için,  $t = \{0, 1\}$  olarak bulunur. Bu veriler yardımıyla eğrinin grafiğini çizersek  $0 < t < 1$  arasında,  $0 < t < \frac{1}{2}$  için ulaşılan  $x(t)$  değerlerinin  $\frac{1}{2} < t < 1$  için tekrarlandığı görülür. Öyleyse eğriye yaklaşım için  $0 < t < \frac{1}{2}$  ya da  $\frac{1}{2} < t < 1$  seçmek yeterlidir. Aksi takdirde eğri aynı noktalardan ikişer defa geçecek ve bu da eğri çiziminde yavaşlığa neden olacaktır.



Şekil 3.3. Değişen parametrelere göre eğrilerin alabileceği farklı şekillerin Helis üzerinde gösterimi

### 3.2.2. Bilgisayar destekli tasarımda parametre dönüşümünün uygulanması

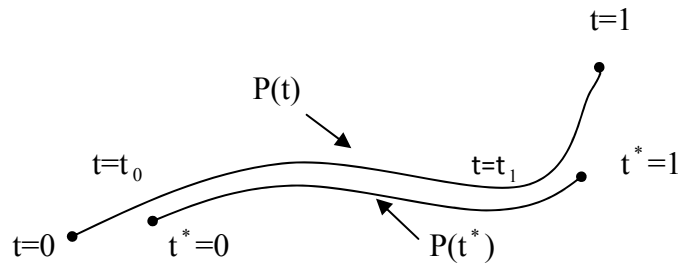
Tek bir parametrik denklem kümesiyle tanımlanması güç olan eğrilerde işlem yapılırken, üzerinde işlem yapılan eğrinin tümünün değil de belli bir kısmının kullanılmasıyla istenilen sonuca daha net ulaşılabilir. Bu işleme, parametre dönüşümü adı verilir ve bu durumda eğrinin,  $t$  parametresinin tanım aralığı  $[0,1]$  kapalı aralığı olmak üzere eğri yeniden parametrelendirilebilir. Bu parametrelendirme işlemi, parametrik gösterilimin sağladığı bir avantajdır. Bu anlatılanların daha iyi anlaşılabilmesi için öncelikle parametre dönüşüm tanımı verilecek ve ardından parametre dönüşüm uygulaması gösterilecektir.

**Tanım 2.2:**  $P: I \rightarrow E^n$  bir eğri olsun  $J$ , bir açık aralık olmak üzere  $h: J \rightarrow I$  fonksiyonu ile,  $h$ 'nin tersi,  $C^k$  ( $k \geq 1$ ) sınıfından ise  $h$ , fonksiyonuna,  $P$  eğrisinin, parametre dönüşümü,  $\beta = P \circ h$  olmak üzere,  $\beta: J \rightarrow E^n$  fonksiyonuna da,  $P$  eğrisinin,  $h$  ile yeniden parametrelendirilişi denir. [Hacısalıhoğlu (1982), Spivak (1970), Akbulut(1970)].

**Örnek 3.1:** Bir eğri, uç noktalarında  $t=0$  ve  $t=1$  parametre değerlerini alacak şekilde,

$$t = (1 - t^*)t_0 + t^*t_1 = t_0 + (t_1 - t_0)t^* \quad (1)$$

Şeklinde parametre dönüşümü ile derecesi değişmeyecek biçimde yeniden parametrelendirilebilir.



Şekil 3.4. Parametre dönüşümünü gösteren örnek

Bu da eğrinin tamamının yerine, sadece belli bir kısmı üzerinde işlem yapmamızın yeterli olduğunu gösterir. Bölüm 3.2.1' de anlatıldığı gibi gerçekleştirilen bu işlemler sonrasında eğri çiziminde hız kazanılmış olur. (1) bağıntısı eğri denkleminde, yerine konulursa, ( $0 \leq t^* \leq 1$ ) için  $t^*$  cinsinden normalize eğrinin, yeni parametrelendirilmiş denkleminin elde edileceği çok kolay gözükür. Bölüm 3.3'de bahsedileceği gibi parametrisasyonu en iyi kübik fonksiyonlar sağlamaktadır. Bu sebeple örnek olarak kübük eğrileri alırsak;  $t_0 \leq t \leq t_1$  aralığında normalize eğrinin denklemini

$$\begin{aligned}
 P(t) &= [t^3, t^2, t] [A \ B \ C \ D]^T \\
 P(t^*) &= \left[ \{t_0 + (t_1 - t_0)t^*\}^3 \quad \{t_0 + (t_1 - t_0)t^*\}^2 \quad \{t_0 + (t_1 - t_0)t^*\} \quad 1 \right] \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \\
 &= \left[ (t_1 - t_0)^3 t^{*3} + 3(t_1 - t_0)^2 t_0 t^{*2} + 3(t_1 - t_0)t_0^2 t^* + t_0^3 \right. \\
 &\quad \left. (t_1 - t_0)^2 t^{*2} + 2(t_1 - t_0)t^* + t_0^2 \quad (t_1 - t_0)t^* + t_0 \quad 1 \right] \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \\
 &= [t^{*3} \ t^{*2} \ t^* \ 1] \begin{bmatrix} (t_1 - t_0)^3 & 0 & 0 & 0 \\ 3(t_1 - t_0)^2 & (t_1 - t_0)^2 & 0 & 0 \\ 3(t_1 - t_0)t_0^2 & 2(t_1 - t_0)t_0 & (t_1 - t_0) & 0 \\ t_0^3 & t_0^2 & t_0 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \\
 &= [t^{*3} \ t^{*2} \ t^* \ 1] \begin{bmatrix} (t_1 - t_0)^3 A \\ 3(t_1 - t_0)^2 A + (t_1 - t_0)^2 B \\ 3(t_1 - t_0)t_0^2 A + 2(t_1 - t_0)t_0 B + (t_1 - t_0)C \\ t_0^3 A + t_0^2 B + t_0 C + D \end{bmatrix}
 \end{aligned}$$

$$= \begin{bmatrix} t^* 3 & t^* 2 & t^* 1 \end{bmatrix} \begin{bmatrix} A^* \\ B^* \\ C^* \\ D^* \end{bmatrix}$$

Sonucunun, şeklinde olduğu gözükür ki bu da bize yapılan işlemler sonrasında eğrinin değişmeden kalabildiğini göstermektedir.

### 3.3. Eğri Tasarımında Kullanılan Yöntemler

Bu bölümde, eğri uyarlama yöntemlerinin gelişimi ve eğri uyarlamada kullanılan farklı yöntemlerin tanıtımı yapılarak bu yöntemlerin, geometrik tasarımda çok kullanılan Bezier, Spline ve Rasyonel eğriler üzerine olan uygulamaları anlatılmıştır.

Eğri tasarımında kullanılan yöntemlerden biri, interpolasyon tekniğidir. İnterpolasyon, kelime anlamı olarak bir fonksiyonun, tablo halinde verilmiş değerlerinden hareketle, fonksiyonun tanımlı olduğu aralıkta bilinmeyen değerlerinin, hesaplanması işlemidir. Bir başka deyişle ele alınan fonksiyonun, daha basit bir  $P(x)$  polinomu ile gösterilmesi veya onun yerine kullanılması işlemidir.

Polinomlarla işlem yapmak kolay olduğundan ve polinomların bir çok özelliği bilindiğinden fonksiyonların yerine, onları temsil edebilecek polinomların araştırılıp kullanılması her zaman için büyük kolaylıklar sağlamaktadır. Çünkü bazı durumlarda, teğetlerin, normallerin ya da eğriliklerin belirlenmesi için türev alma ya da parametrelendirme işlemleri gerekir. Bu işlemlerin kullanımı sırasında da en uygun seçim, polinom fonksiyonlarıdır. Polinom fonksiyonları içerisinde de parametrizasyonu en iyi, kübik olan fonksiyonların sağladığı görülmüştür. Bunun örneğine de bölüm 3.2.2' de yer verilmiştir.

### 3.3.1. Serbest biçimdeki eğri ve yüzeylerin matematiksel modellerinin gelişimi

1960'ların başında Boeing uçak şirketinden, J.C. Ferguson, eğri parçalarını vektörel olarak tanımlayan bir metot geliştirmiştir. Ferguson eğri parçalarının başlangıç ve bitiş noktalarını, belirleyerek her parçayı  $[0,1]$  Aralığında tanımlayarak eğri parçalarını, kübik vektör fonksiyonları şeklinde ifade etmiştir. Yapılan çalışmalarda, gerçek eğri, bu eğri parçalarının düzgünce birleştirilmesiyle elde edilir. Buna ek olarak Ferguson, bu eğri parçalarını kullanarak eğrinin dört köşesine dâhil edilen konum ve teğet vektör şartlarını sağlayan bir yüzey parçası (yaması) oluşturmuştur. Fergusonun bu çalışmasından önce eğri tanımlamaları,  $y=f(x)$  ve  $f(x,y)=0$  biçimindeydi. Yapılan bu çalışmayla,  $y=f(x)$  ve  $f(x,y)=0$  biçimindeki eğrilerin, dezavantajları da ortadan kaldırılmıştır.

Ferguson eğrileri, aşağıdaki üstünlüklere sahiptir.

- 1) Sadece düzlemdeki değil uzaydaki eğriler de basit fonksiyonlarla ifade edilebilirler.
- 2) Eğrinin istenen kısmı, parametre aralığıyla kolayca belirlenebilir, yani eğri parametrelendirilebilir.
- 3) Eğrinin, öteleme ve döndürme işlemleri, dönüşüm matrisleri ile çarpılarak elde edilebilir.

Daha sonra 1964 yılında, MIT( Massachuset Institute of Technology) den S.A. Conns, yüzey yamasının, dört köşe noktalarındaki konum vektörleri ve sınır eğrilerinin göz önüne alındığı ve sınır şartlarını sağlayan matematiksel tanımı veren bir yüzey tanımlama metodunu geliştirmiştir. 1967 yılında ise bu kavramın (yüzey yamalarının) geliştirilen versiyonları ortaya çıkarılmıştır. Bu versiyonlara göre Pratikte, yüzey yamasının dört köşesindeki konum vektörleri, iki yöndeki teğet vektörler, karşılıklı kısmi türev büküm vektörleri, dört sınır eğrisinin konumları ve eğriler boyunca teğet vektör fonksiyonlarını vermek yeterlidir. Daha sonra yapılan

çalıřmalarda da sürekli bir yüzey oluşturmak ve yüzey yamalarının birleřtirilmesi için gerekli olan řartlar da ortaya koyulup yamaları birleřtirmede, sınır eğrileri boyunca n. Dereceden türevin sürekli olacağı fikri geliřtirilmiřtir. (B.A. Vera,1992) Genel olarak Coons yüzey yaması modeli, Ferguson yüzey yaması modeliyle örtüřür. Bir bařka deęiřle Ferguson eğrileri, Coons eğrilerinin, özel bir halidir denilebilir. Yalnız bu tekniklerde, yüzeylerin řeklini kontrol etme zorluğu ortaya çıkar. Bu kontrol etme zorluğu ise řöyle tanımlanır.

Eęri parçaları veya yüzey yamaları birleřtirildięinde, birleřme noktalarındaki matematiksel süreklilik yetmemektedir. Dolayısıyla yeni matematiksel düzenlemelere ihtiyaç duyulur. Bu zorluğu ortadan kaldırmak ve problemi çözmek için Spline eğrileri geliřtirilmiřtir.

Spline; çelik, plastik veya odundan yapılmıř esnek bir banttır. Spline'lar, gemi, uçak, otomobil gibi serbest řekilli yüzey örneklerinde sıklıkla kullanılırlar. Özellikle de spline řekilleri, bilgisayar destekli çizim ortamlarında çok kullanılır. Spline eğriler, birleřim noktalarında matematik olarak sürekli olan sonsuz sayıdaki eğrilerdir. Teęet vektörünün, deęiřtirilmesiyle elde edilirler.

Daha sonra Renault řirketinden, P. Bezier, bir çokgen yardımıyla eğriyi tanımlamıřtır. Bezier eğrisi adı verilen bu eğri, verilen çokgenin köřelerini düzelterek elde edilen bir eğri olarak görülebilir. Bezier eğrileri, Renault otomobil gövdesi tasarımında pratięe uygulanmıřtır. Bezier eğri parçaları ve yüzey yamaları, sadece çokgen köřelerindeki konum vektörleriyle tanımlanmıřtır. Dolayısıyla, bu metodla, Ferguson ve Coons metodlarının aksine, teęet ve büküm vektörleri gibi kavranmasında zorluk çekilen analitik verilere ihtiyaç duyulmamıřtır. Bezier eğri parçaları, eğriyi tanımlayan köře konum vektörlerinin, bir konveks birleřimi olarak ifade edilebilirler. Sonuç olarak eğrinin řeklinin, yaklařık olarak çokgenin řekline benzemesi umulur. Ayrıca yapılan bu çalıřmayla, polinom eğrisinin derecesini yükseltmek de mümkündür; eğri parçası, řeklini deęiřtirmeden iki parçaya bölünebilir veya eğrinin derecesi, řekil deęiřtirmeden yükseltilebilir, [H.M. Osinga, R. Rokni, 2003].

Sonuç olarak bezier eğri ve yüzeyleri, kontrol açısından Bilgisayar destekli tasarım ortamında kolaylıklar sağlamaktadır. Sağladığı kolaylıklar ise, Bezier tasarımcılarının daha çok esneklik verebilmek için daha çok matematik kullanmalarından kaynaklanır. Tasarımcılar, eğrileri eğriye yaklaşım sağlayan sıralı noktalar kümesi olan  $(v_0, v_1, \dots, v_n)$  kontrol noktalarını, kullanarak tanımlamıştır. Bu noktalar grafik ekranda temsil edilebilirler ve kullanıcıya eğrinin şeklini kontrol etme imkânı verirler.

Daha sonra Spline eğrilerinden esinlenen, Gordon ve Riesenfeld ise baz spline karma fonksiyonlar olarak kullanılan eğri ve yüzeyleri tanıtmıştır. Beta Spline eğri ve yüzeyleri denilen bu özel eğri ve yüzeyler, problemleri çözmeye daha da kolaylıklar sağlamıştır. Beta Spline eğrileri de Bezier eğrileri gibi çokgen köşeleri tarafından tanımlanır. Yani Beta Spline eğrileri de çokgen köşe konum vektörlerinin, konveks kombinasyonu olarak ifade edilirler. Eğri şekilleri, çokgen şekillerinin düzeltilmiş bir versiyonudur. Eğri yaklaşık olarak çokgenin şeklinden tahmin edilebilir. Bezier eğrileri ile Beta Spline eğrileri arasındaki tek fark ise Bezier eğrileri, tüm köşe noktalarının konum vektörlerinin birleşimi olmasına karşılık Beta Spline eğrileri, sadece çok yakınlardaki köşe konum vektörlerinin, konveks kombinasyonudurlar. Bundan dolayıdır ki çokgenin şekil değiştirme özellikleri, Beta Spline eğrilerinde daha açık görülür. Verilen çokgen sayısı  $n$  ve spline mertebesi  $m$  olmak üzere, Beta Spline eğrisi  $(n-m+2)(m-1)$  dereceden eğri parçalarının, düzgünce birleşmesinden oluşur. Beta Spline, düğüm vektörü ve  $m$  mertebesi ile tanımlanır. Beta Spline eğrisini oluşturan eğri parçaları, her birinin yapımındaki  $m$  çokgen köşeleri ile tanımlanır. Sonuç olarak bir çokgen köşesinin konumu değiştirildiğinde eğrinin tümü değil sadece belli bir kısmı (local kontrol) değişir. Yani yerel değişim gerçekleşir. Sıralanan bu özellikler eğri ve yüzey tasarımında çok önemlidirler. Çünkü eğri ve yüzeyler, parametreye bağlı polinomlarla çok kolay ifade edilebilirler. Ancak endüstride çok kullanılan yaylar ve çemberler, bu sıradan polinomlarla ifade edilemezler. Bu sebeple de Yamaguchi, 1998' de eğri ve yüzeylerin rasyonel tanımlamalarını yapmıştır.

### 3.3.2. Bezier eğriler

Günlük hayatta kullandığımız çoğu cismin, basit bir şekli olmadığı için bu şekilleri, basit bir analitik fonksiyon olarak ifade edemeyiz. Bu yüzden eğrileri, eğri parçaları (yaylar) şeklinde tanımlarız. Bu tanımlamaya uyan serbest şekilli eğri örneklerinden olan Bezier eğrileri, daha öncede belirtildiği gibi Renault otomobil gövdelerinde kullanılmak üzere Pierre Bezier tarafından geliştirilmiştir. Forrest, Gorden ve Riesenfield da elde edilen bu sonuçları, Bernstein kuralları şeklinde ifade etmişlerdir. Bezier eğrileri, girdi olarak kullanılan kontrol noktaları ve bir dizi polinom fonksiyonlarıyla tanımlanırlar. N. Dereceden, n+1 kontrol noktasına sahip Bezier eğrisinin parametrik denklemi,

$$P(t) = \sum_{k=0}^n P_k B_{k,n}(t) \quad (2)$$

Şeklinde yazılır. Denklemdaki  $B_{k,n}(t)$  katsayıları, Bernstein polinomları ve  $P_k$  'ler kontrol noktalarıdır. Bernstein polinomları,

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

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

Olarak tanımlanır. (2) denklemi olarak verilen eğri üzerindeki bir noktanın koordinatları da aşağıdaki şekilde formülüne edilir.



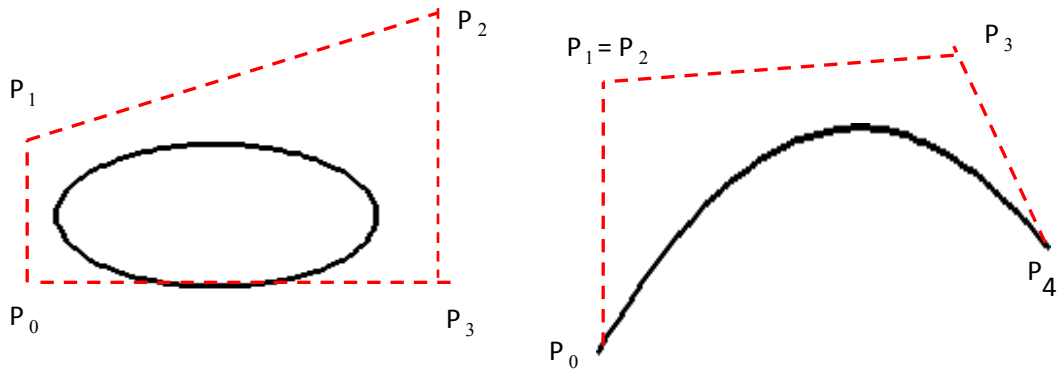
$$x(t) = \sum_{k=0}^n x_k B_{k,n} P(t)$$

$$y(t) = \sum_{k=0}^n y_k B_{k,n} P(t)$$

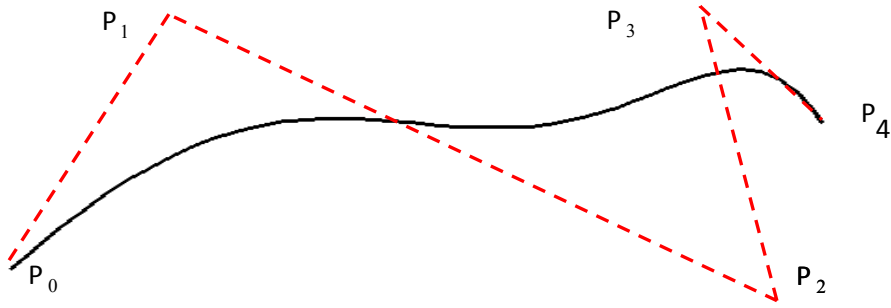
$$z(t) = \sum_{k=0}^n z_k B_{k,n} P(t)$$

Kontrol noktaları, karma fonksiyon şeklinde yazıldığı için  $B_{k,n}(t)$  polinomu, karma fonksiyon olarak adlandırılır. Bu karma fonksiyonun polinom derecesi, kullanılan kontrol noktaları sayısından bir eksiktir. Örneğin paraboller üç, küp dört kontrol noktası kullanılarak çizilir dolayısıyla bu çokyüzlülerin dereceleri, sırasıyla iki ve üç olur. Yukarıda anlatılanlar ışığında Bezier uygulamalarına ve bu uygulamalardan elde edilebilecek sonuçlara bakalım;

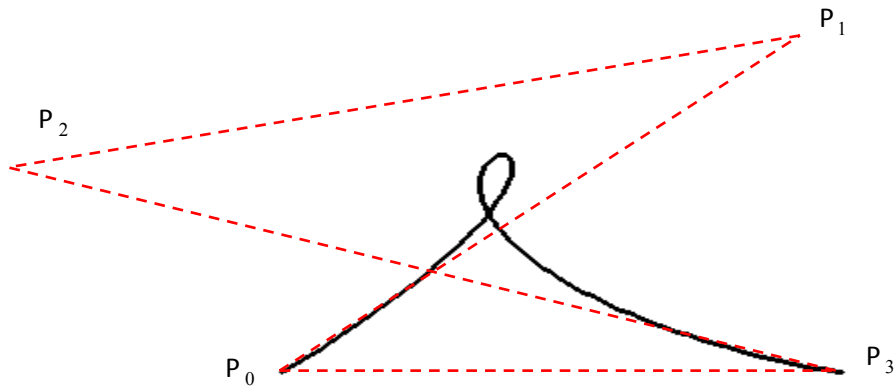
Bezier eğri uygulamaları sırasında, kapalı bir eğri çizmek istersek eğrimizin başlangıç ve bitim noktalarını aynı konumda almamız gerekir. Bunun yanında eğri üzerindeki iki kontrol noktasına aynı değeri vererek yani kontrol noktalarımızı, aynı konumdan seçerek farklı bir Bezier eğrisi daha elde edebiliriz. Karmaşık yapıda bir Bezier eğrisi çizmek istediğimizde de yüksek mertebeden bir polinom fonksiyonu tanımlamamız gerekir. Bu şekilde işlem yapmak güç olduğu için Bezier eğrimizi daha küçük mertebeden eğri kısımlarına ayırırız ve daha sonra düşük mertebeden fonksiyona sahip olan bu eğri parçalarını bir araya getirerek istediğimiz gibi karmaşık yapıda olan Bezier eğrisini elde edebiliriz. Yukarıda anlatılanlar şekil 3.5, 3.6 ve 3.7'de örneklendirilmiştir.



Şekil 3.5. Kapalı Bezier eğrisi ve birinci ile ikinci kontrol noktaları aynı konumda olan eğri



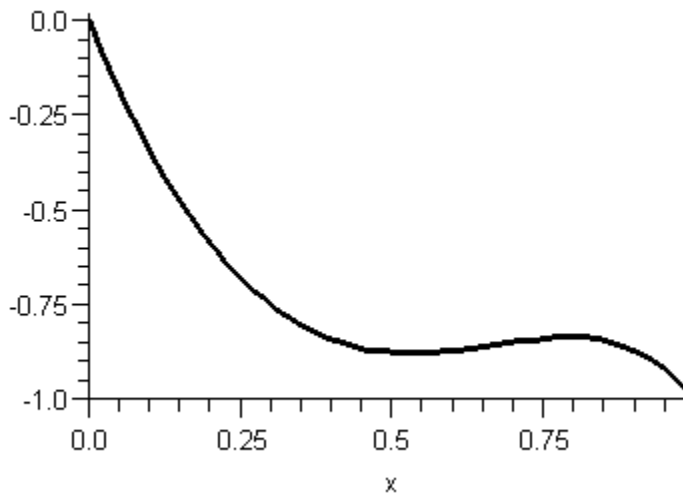
Şekil 3.6. Farklı iki bezier eğrisinin birleşimini gösteren bezier eğrisi



Şekil 3.7. Fiyonk biçimli bezier eğrisi

Şekil 3.5. ve 3.7’de gözüktüğü gibi  $P_0P_1$ ,  $P_1P_2$ ,  $P_2P_3$  doğruları, eğrinin karakteristik çokgeni denilen şekli oluştururlar. Bu şekil genellikle kapalı bir şekil değildir. Şekillerden de anlaşılacağı gibi bir eğri tasarımı için eğrinin geçmesini istediğimiz  $P_0$  ve  $P_3$  noktalarını seçer ve  $P_1$  ve  $P_2$ ’yi  $P_0$  ve  $P_3$ ’de ki istenen eğriler üzerine yerleştiririz. O halde Bezier eğrileri, tamamen  $P_0P_1P_2P_3$  çokgeninin içinde uzanacaktır. Eğrinin karakteristik çokgeni ile olan ilişkisine, dışbükey tekne özelliği denir ve bu bir çokgen cinsinden tanımlanan herhangi bir eğri için çok kullanışlı bir özelliktir, [Faux and Pratt(1979), Farin(1990), Tantai(1992)].  
Sonuç olarak Bezier eğrilerinin, özellikleri şöyle sıralanabilir:

- 1) Bir baz fonksiyonu vardır.
- 2) Eğri parçalarını tanımlayan polinomun derecesi, tanımlanan kontrol nokta sayısından bir eksiktir.
- 3) Eğri tanımlanan, karakteristik çokgenin şeklini izler.
- 4) Eğri üzerindeki ilk ve son noktalar, tanımlanan karakteristik çokgenin ilk ve son noktalarıyla aynı konumdadır. Eğri bu noktalar arasında düzenli salınım hareketleriyle gidip gelir. [Mastin, C.,1992)].



3.8. Üçüncü dereceden Bezier eğri örneği

### 3.3. 3. Spline eğriler

Çok sayıdaki veri noktalarına, bir tek eğri ile yaklaşmak büyük kolaylıklar sağlasa da bazı hallerde bu durum bazı hatalara neden olabilir ve ayrıca bu amaçla kullanılan interpolasyon polinomlarıyla yapılan işlemler zorlaşır. Bu durumu ortadan kaldırmak için de peş peşe iki veri arasında birinci (Lineer), ikinci (Kuadratik) ya da üçüncü (Kübik) dereceden fonksiyonlarla yaklaşımın yapıldığı Spline Interpolasyon yöntemi kullanılır. Bu yöntem, veri noktalarını çeşitli aralıklara bölerek her bir aralıkta daha küçük dereceden polinomlarla yaklaşım yapma esasına dayanır ve bu yöntemde eğri, tanımlanan poligon köşeleri yani karakteristik çokgen tarafından üretilir.

Spline eğrileri, matematik dilinde, düzgün bir eğri elde etmek için kontrol noktaları kullanılarak çizilen esnek ve kolay bükülebilen, şeritler olarak tanımlandıklarından birçok kolaylıklar getirir. Spline eğrileri ve Spline fonksiyonları, özel kontrol noktaları verilerek görüntülenmek istenen eğri teknikleri kullanılarak çizilen son eğriler olarak bilinirler ve bu eğriler, uygulamada yukarıda da bahsedildiği gibi birinci, ikinci ve üçüncü mertebeden kübik polinomlara, parça parça yaklaşım olarak tanımlanırlar. Geri kalan birçok fonksiyona yaklaşımda yine Spline eğrileri kullanılarak gerçekleştirilir. B-Spline eğrileri ise özellikle grafik uygulamalarında kullanılan Spline eğrilerinin bir sınıfıdır. B-Spline eğrilerinde değişiklikler, yerel (lokal) bölgelerde gerçekleştirilir ve B-Spline eğrileri, farklı sayıdaki kontrol noktaları için eğrinin derecesini arttırmadan çizilebilir.  $k$  sıfırdan  $n$ 'ye kadar değişen değerler olmak üzere  $(n+1)$  tane kontrol noktası kullanılarak çizilen bir B-Spline Eğrisi,  $P(t)$ - $t$  parametresine bağlı bir fonksiyon olmak üzere aşağıdaki gibi tanımlanır.

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \quad ; \quad x_{\min} \leq t \leq x_{\max}$$

Burada  $B_i$ , (n+1) karakteristik poligonun, köşelerinin konum vektörü ve  $N_{i,k}(t)$  de normalize edilmiş baz (temel) fonksiyonlardır. k. Mertebeden (k-1) dereceden i. Normalize edilmiş B-Spline eğrisi için  $N_{i,k}(t)$  baz fonksiyonu, Coox-de-Boor tarafından aşağıdaki gibi tanımlanmıştır.

$$N_{i,k}(t) = \left\{ \begin{array}{ll} 1, & x_i \leq t \leq x_{i+1} \\ 0, & \text{diğer durumlarda} \end{array} \right\}$$

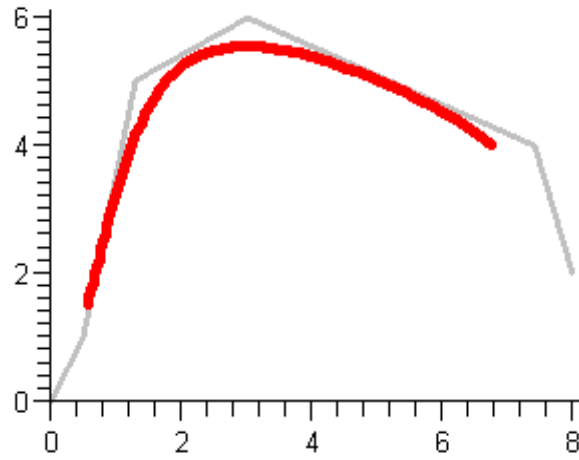
$$N_{i,k}(t) = \frac{(t-x_i)N_{i,k-1}(t)}{x_{i+k-1}-x_i} + \frac{(x_{i+k-1}-t)N_{i,k-1}(t)}{x_{i+k-1}-x_i}$$

$x_i$  değeri,  $x_i \leq x_{i+1}$  bağıntısından elde edilen düğüm vektörünün bir elemanıdır.

t parametresinin değeri,  $x_{\min}$  ile  $x_{\max}$  arasında değişirken bir B-Spline eğrisi, k. mertebeden (k-1 dereceden) polinom fonksiyonları olarak tanımlanır.

Sonuç olarak Bezier eğrilerinin, özellikleri şöyle sıralanabilir:

- 1)  $[x_i, x_{i+1}]$  aralığında polinomun derecesi, (k-1) i geçmez.
- 2) 1'den (k-2)'ye kadar olan bütün türevler süreklidir.
- 3) Kübik bir B-Spline eğrisi için k=4 ve derece (k-1)=3'e eşittir.



Şekil 3.9. Spline eğri örneği

### 3.3.4. Rasyonel eğriler

Yukarıdaki eğri tanımlamalarında serbest şekilli eğrilerin modellenmesi anlatıldı. Bununla birlikte, mühendislik tasarımları çoğu kez doğrular, çemberler, konikler gibi standart analitik şekillerin kullanımını da gerektirir. Eğrilerin her iki çeşidinin yani hem serbest biçimli hem de konik biçimli olarak modellenmesinin, birleşimini sağlayan eğri tasarım yöntemi, rasyonel polinom fonksiyonlarının kullanımındadır.

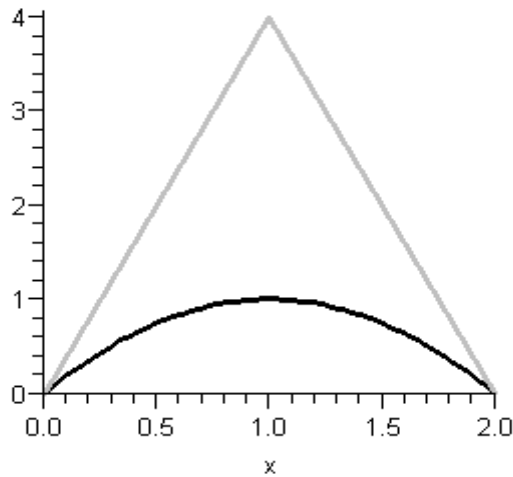
Bu polinomlar için Rasyonel teriminin kullanılmasının sebebi, elde edilen fonksiyonun iki polinomunun birbirine oranlanmasından kaynaklanmaktadır. Rasyonel fonksiyonlar, izdüşüm dönüşümler altında invaryanttır. Örneğin, bir rasyonel eğrinin, perspektif izdüşümü de yine bir rasyonel eğridir.

Rasyonel eğrilerin, Bilgisayar Destekli Tasarım uygulamalarında, homojen koordinatlar kullanılır. Yani,  $(n-1)$  boyutlu uzaydaki noktalar,  $n$ -boyutlu uzaydaki noktaların izdüşümü olarak düşünülür.

$P(x,y,z)$  gibi üç boyutlu Öklid uzayında tanımlanan bir noktanın, dört boyutlu homojen uzayda  $Q=(wx,wy,wz,w)$  şeklinde gösterimi vardır. Buradaki  $w_i$  koordinatları, bire eşitse ifade rasyonel olmayan forma indirgenir. Rasyonel olmayan ifadelerle ait olan dönme gibi bazı dönüşümler, rasyonel eğriler için de geçerlidir. Bezier ve B-Spline eğrilerinin, her ikisi de rasyonel forma sahiptir. Bu ifadeler aşağıdaki tabloda olduğu gibi açıklanabilir.

Tablo 3.1. Rasyonel olan ve Rasyonel olmayan Bezier ve B-Spline eğrilerinin denklemleri

	Bezier	B-Spline
Rasyonel olmayan	$P(t) = \sum_{i=0}^n B_i J_{n,i}(t)$	$P(t) \sum_{i=1}^{n+1} B_i N_{i,k}(t)$
Rasyonel	$P(t) = \frac{\sum_{i=0}^n B_i w_i J_{n,i}(t)}{\sum_{i=0}^n w_i J_{n,i}(t)}$	$P(t) = \frac{\sum_{i=1}^{n+1} B_i w_i N_{i,k}(t)}{\sum_{i=1}^{n+1} w_i N_{i,k}(t)}$



Şekil 3.10. Rasyonel eğri örneği

### 3.4. Eğrilerde Dönüşüm İşlemleri

Nesnelerin dış görünüşlerini ve niteliklerini göstermek için onların çeşitli grafik biçimlerini oluştururuz. Bazı uygulamalarda ise, nesneleri sıralamaya işlenmiş görüntülerine, bazen de nesnenin büyüklüğünü küçültmeye veya nesneyi daha büyük ölçülerde görüntülemeye veya modelin görünümünü, parçalarının büyüklüğünü yeniden şekillendirmeye ihtiyaç duyarız.

Hareketli uygulamalar için ise hareketin devamını, sürekliliğini sağlamalı ve nesneyi ekranda o şekilde görüntülemeliyiz. Nesne üzerinde yapılacak bütün bu değişiklikler, belirli noktalar kullanılarak uygun geometrik dönüşümlerle görüntülenir. Kullanılan bu basit dönüşümler, öteleme, ölçekleme, dönme ve yansımadır. Bu bölümde de yukarıda sıralanan dönüşümlerin, nasıl yapıldığına ve dönüşüm fonksiyonlarının grafik uygulamalarıyla nasıl birleştirileceğine bakılacaktır.



### 3.4.1. Öteleme

Öteleme, bir nesneyi bir doğru etrafında bir yerden başka bir yere hareket ettirmediir. Öteleme işlemi sırasında  $T_x, T_y$  olan ilk nokta çiftine, öteleyeceğiz uzaklıkları ekleyerek yeni nokta çifti olan  $(x', y')$  yü elde ederiz ve yeni koordinatlarımız aşağıdaki gibi yazılır.

$$x' = x + T_x \qquad y' = y + T_y$$

Çokgenlerin dönüşümünde ise, nesne üzerindeki her doğrunun bitim noktasına uygulanacak öteleme uzaklıkları eklenerek yeni koordinatlar bulunur. Öteleme uzaklıkları, pozitif, negatif veya sıfır gibi reel sayılar olabilir. Öteleme işlemi sırasında, limitin üzerinde bir sayı ile işlem yapılırsa bilgisayar limitin üzerini aşmayın veya grafiğinzin şekli bozuk gibi hata mesajları verir.

### 3.4.2. Ölçekleme

Nesnenin büyüklüğünde değişiklik yapmaya ölçekleme denir. Bu işlemde  $(x,y)$  koordinat değerleri, ölçeklendirmeyi sağlayacak olan  $S_x, S_y$  değerleri ile çarpılarak elde edilmek istenen sonuç değerlerine ulaşılır.

$$x' = x \cdot S_x \qquad y' = y \cdot S_y \text{ olur.}$$

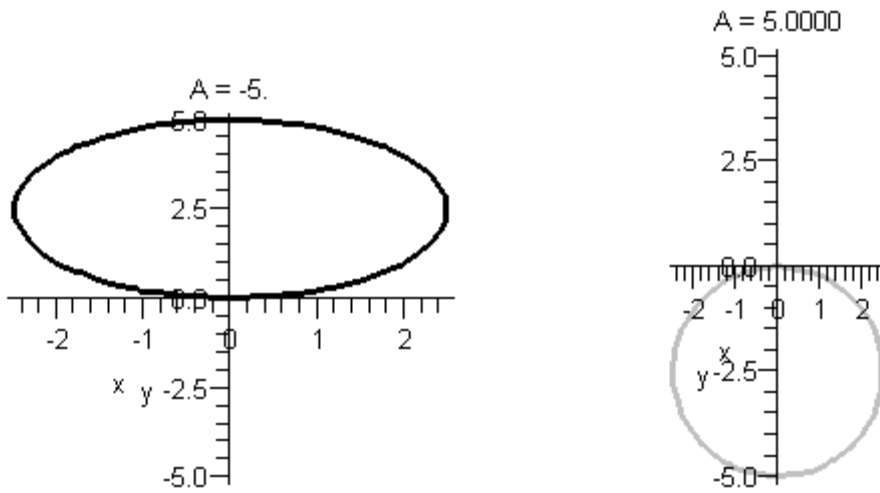
Ölçeklendirmeyi sağlayacak olan  $S_x$  değeri, nesnenin x eksenini doğrultusundaki  $S_y$  değeri ise nesnenin y eksenini doğrultusundaki ölçeklendirilmesini sağlar. Ölçeklendirmeyi sağlayan  $S_x$  ve  $S_y$  değerleri, birden küçük seçilirse nesnenin büyüklüğünde azalma, birden büyük seçilirse de nesnenin büyüklüğünde artma olur. Özel olarak  $S_x$  ve  $S_y$  değerleri, bir olarak alınırsa nesnenin büyüklüğünde bir değişiklik olmaz.

$S_x$  ve  $S_y$  ölçeklendirme değerleri, aynı alınırsa da tek cins bir ölçekleme yapılmış olur. Yani nesnenin ölçeklendirme oranında hiç bir değişiklik olmaz. Küçültülen nesneler, orijine yaklaşırken büyütülen nesneler orijinden uzaklaşır. Ölçeklendirme işlemi sırasında ölçeklendirilecek nesne için önce bir konum seçilir. Bu konuma sabit nokta denir. Bu nokta, yapılan bütün dönüşüm işlemlerinden sonra değişmeyen noktadır.  $(X_F, Y_F)$  sabit noktası, ölçeklendirilmek istenen nesnenin, bir köşe noktası, merkezi ya da herhangi bir noktası olabilir.

$(X_F, Y_F)$  sabit nokta,  $(x', y')$  dönüştürülmüş koordinatlar olmak üzere nesnenin ölçeklendirildikten sonraki koordinatları, aşağıdaki gibi elde edilir.

$$x' = x \cdot S_x + X_F(1 - S_x) \qquad y' = y \cdot S_y + Y_F(1 - S_y)$$

Aşağıdaki örnekte elips ve dairenin önce ölçekleme sonra yansıma işlemlerinden sonraki durumları gösterilmek istenmiştir.



Şekil 3.11. Elips ve Çemberin x eksenine etrafındaki dönüşümünün izlenmesi

### 3.4.3. Döndürme

Dönüşümü yapılacak olan nesnenin noktalarının, bir yol boyunca dairesel hareketlerle çevrilmesine döndürme denir. Bu tür özel dönüşümlerde bir döndürme açısı seçilir ve seçilen bu açı çokgenin köşe noktalarına, döndürme değeri olarak eklenerek döndürme işlemi gerçekleştirilir.  $(x,y)$  noktasının  $x$  eksenine ile yaptığı açı,  $\varphi$  olmak üzere, bu noktaya  $\theta$  açısı altında orijin koordinatları temel alınarak döndürme işlemi yapılırsa  $x'$ ,  $y'$  noktasına dönüştürülür. Kullanılan trigonometrik ifadeler sonucunda dönüşüm denklemleri aşağıdaki gibi yazılır.

$$\begin{aligned}x' &= r \cdot \cos(\varphi + \theta) = r \cdot \cos\varphi \cdot \cos\theta - r \sin\varphi \sin\theta \\y' &= r \cdot \sin(\varphi + \theta) = r \cdot \sin\varphi \cdot \cos\theta + r \cos\varphi \sin\theta\end{aligned}$$

Burada  $\theta$  açısını, pozitif bir değer olarak alırsak saatin ters yönünde, negatif bir değer olarak alırsak saat yönünde döndürme işlemi yapmış oluruz. Seçtiğimiz dönme noktası, nesnenin içinde veya sınırları dışında olabilir. Dönme noktası dışarıda olduğunda diğer bütün noktalar bu nokta boyunca dairesel yollar çizerek yer değiştirir.

### 3.4.4. Yansıma

Yansıma nesnenin görüntüsünün, aynaya izdüşürülmüş halidir. Nesnenin  $x$  eksenine üzerine yansıtılması işleminde,  $x$  eksenine koordinatları sabit kalıp  $y$  eksenine koordinatları yansıtılır.  $y$  eksenine üzerine yapılan yansıma işleminde ise,  $y$  koordinatı sabit kalıp  $x$  koordinatı yansıtılır. Bu her iki yansımanın matris şekli aşağıda ki gibi yazılabilir.

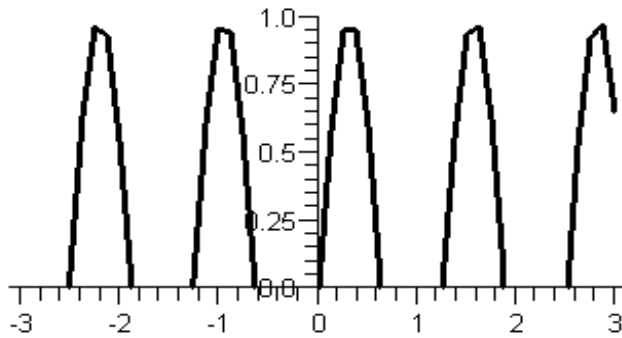
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Diğer bir tip yansıma ise hem x hem de y eksen koordinatlarının orjin baz alınarak yansıtılması ve  $y=x$  eksenini kullanarak yapılan yansımadır. Bu yansıma durumlarının matris gösterilişleri ise aşağıda gösterildiği gibidir.

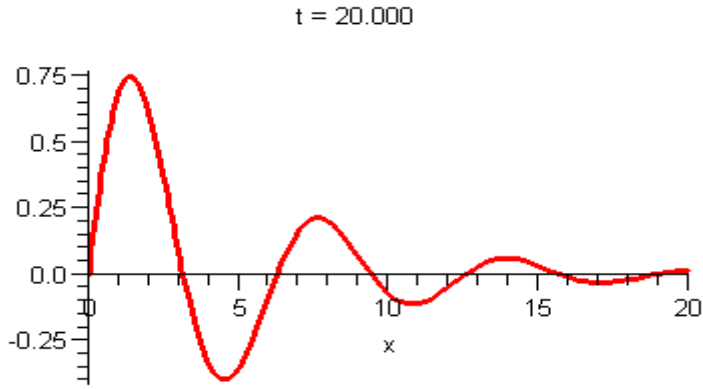
$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

### 3.4.5. Eğrilerde dönüşüm işlemleriyle ilgili uygulamalar

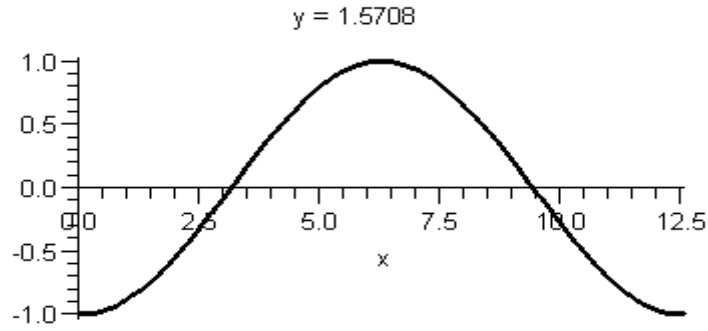
Aşağıdaki örneklerin birincisinde eğrinin ötelenmesi, ikincisinde hem ötelenmesi hem de ölçeklenmesi, üçüncüsünde sadece döndürülmesi ve sonuncusunda hem döndürülmesi hem de ötelenmesi gösterilmiştir.



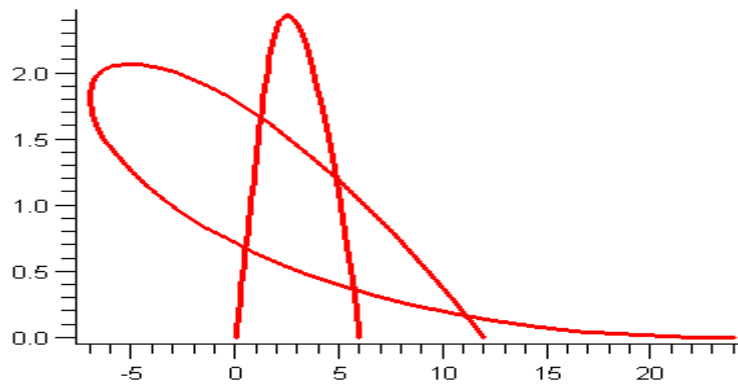
Şekil 3.12. Eğrinin ötelenerek ilerlemesine örnek



Şekil 3.13. Eğrinin hem ötelenmesi hem de ölçeklendirilmesine örnek



Şekil 3.14. Eğrinin döndürülmesini gösteren örnek



Şekil 3.15. Eğrinin ötelenmesi ve döndürülmesini gösteren örnek

## BÖLÜM 4. PARAMETRİK YÜZEY TASARIMI

### 4.1. Temel Kavramlar

Günlük hayatta, yüzey örnekleri pek çok yerde görülebilir. Balon, konserve kutusu, sabun köpüğü gibi... Birçok alanda görülen bu nesnelere, geometri alanında da çalışılabilir.

Bunun için belirli koordinat hesaplamalarına ihtiyaç vardır. Bu yüzeyler, üç boyutlu uzaydadır, fakat her zaman için üç boyutlu uzayda çalışma zorunluluğu yoktur. Örneğin, üç boyutlu uzayda ele alınan silindir yüzeyi, uzunlamasına ya da enlemesine kesilip düz bir zemin üzerine serildiğinde, aslında iki boyutlu uzayda yer almış olur. Bu sayede geometrik yüzeylere yaklaşım mantığında ipuçları elde edilmiş olur. Çünkü yüzeyin şeklini açıp düzlem üzerine sermek koordinat hesaplamalarını, kolaylaştırır.

Bu şekilde tanımlanan yüzeylere, regle yüzeyler denir. Regle yüzey, bir parametreye bağlı olan bir doğru ailesinin, geometrik yeridir. Yani regle yüzey, doğrunun oluşturduğu yüzeydir. Zaten doğru, bir parametreye bağlı olarak hareket ettirildiğinde, düzlem oluşur. Düzlem de en basit yüzeydir. Yüzeyler, mühendislik tasarımı problemlerinde ve üretimde önemli yere sahiptir. Eğri tasarımları için kullanılan parametrik gösterimler, yüzey tasarımları içinde geçerlidir. Parametrik biçimde bir eğri,

$$P(t) = [ x(t), y(t), z(t) ]$$

Biçiminde tanımlanıyor idi. Tanımlanan bu gösterim yardımıyla yüzey tanımlamaları da kolaylaşır ve yüzeyin parametrik gösterimi de

$$P(s,t) = [ x(s,t), y(s,t), z(s,t) ]$$

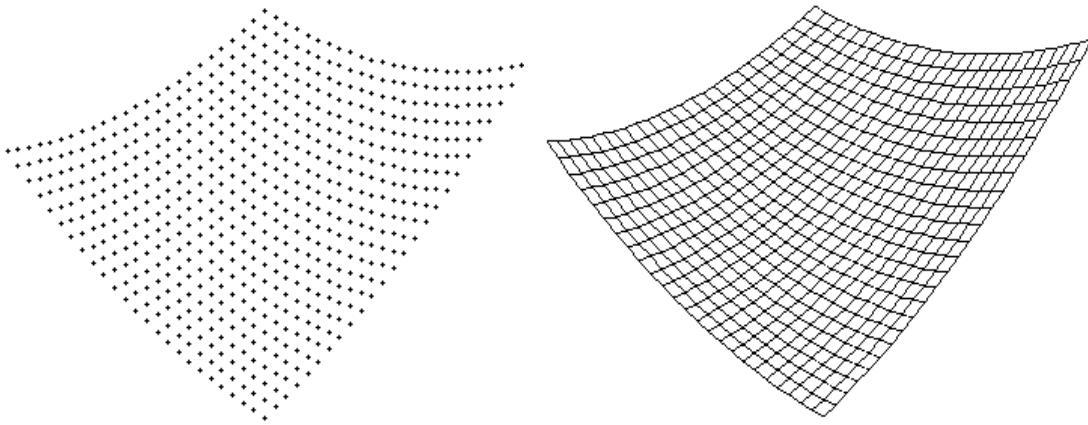
Biçiminde yazılır. Parametrelerin biri sabit tutulup diğeri değiştirilerek yüzeyin parametre eğrileri elde edilir.  $s$  ve  $t$  parametrelerini 0 ile 1 arasında alırsak  $t$  doğrultusundaki eğriler,  $P(0,t), P(0.1,t), \dots, P(0.9,t), P(1,t)$  ve  $s$  doğrultusundaki diğer eğriler,  $P(s,0), P(s,0.1), \dots, P(s,0.9), P(s,1)$  şeklinde olur. Yüzeyin üretilmesi o yüzeyi üretecek eğrilerin bulunması demektir. [Adams, D.F. and Mastin C.W. 1990]

Diğer bir ifadeyle de yüzey tanımını şu şekilde verebiliriz.  $D, \mathbb{R}^2$  'de açık aralık olmak üzere;

$$P: D \longrightarrow \mathbb{R}^3$$

$$(s,t) \longrightarrow (x_1(s,t), x_2(s,t), x_3(s,t))$$

Burada  $x_i(s,t), x(s,t), y(s,t), z(s,t)$ 'ler  $x$  fonksiyonunun bileşenleri olmak üzere;  $t=t_0$  gibi sabit bir değer ve  $s$ , değişen değerler alırken  $x$ , bir parametreye bağlı bir eğri belirtir ve  $s$ , parametrelili eğri, diye adlandırılır. Aynı şekilde  $s=s_0$  gibi sabit bir değer,  $t$  değişen değerler alırken de  $t$ , parametrelili eğri oluşur. Her iki eğride  $(s_0, t_0)$  noktasından geçmektedir.  $s = s(s_0, t)$  ve  $t = t(s, t_0)$  şeklinde tanımlanan, parametrelili eğriler, olmak üzere bir yüzeyin her noktasından en az iki eğri geçer. Dolayısıyla, kesişen bu eğrilerin, kesişim yerlerinde oluşan noktalar, yüzey üzerindeki noktaları, yüzey üzerinde oluşan bu noktalar da yüzeyi belirler. Aşağıdaki örneklerde, yüzeyi oluşturan noktalar ve bu noktaların üzerinde bulunduğu eğriler, ayrı ayrı şekiller üzerinde gösterilmiştir.



Şekil 4.1. Yüzeyi oluşturan noktaları ve eğrileri gösteren örnek

#### 4.2. Yüzeylerin Uygulamada Kullanımı

Yüzeyleri, grafik olarak görüntülemenin bir kaç yolu vardır. Elimizde görüntülenmesini istediğimiz nesneyle, ilgili ne kadar iyi bilgi varsa nesneyi görüntülememizde o kadar iyi olur.

Örneğin küp, altı tane düz yüzeyden, silindir bir eğri yüzey, iki düz yüzeyden küre ise sadece bir eğri yüzeyden oluşmaktadır. Daha karmaşık nesnelere tanımlarken de yine bu şekilde bir anlatma metodunu kullanırız ve nesneyi tam olarak tanımlayamadığımızda da tanımlayabildiğimiz yüzey örneklerinden hareketle nesnemizi, tanımlamaya çalışırız.

Dolayısıyla yüzeyleri, grafik olarak görüntülemenin yollarından biri, nesneyi düz ve çokgen yüzey kümelerine ayırmaktır. Bunun dışında yüzeyler, parametrik eğri denklemleri ve kesirli (fractal) yaklaşımlarla da temsil edilebilirler. Bazı uygulamalarda da nesne, basit şekillere bölünerek nesnenin, bu basit şekiller üzerinden görüntülenmesi yoluna gidilir. yüzeyleri, oluşturmanın bir diğer yolu da eğrileri kullanarak yeni nesne yığınları elde etmektir. Diğer bir yapılandırma da blok tekniği denilen piramit, silindir gibi basit üç boyutlu yüzey örneklerini kullanarak y daha karmaşık üç boyutlu yüzey örnekleri, oluşturmaktır.



#### 4.2.1. Bezier yüzeyler

Conns yüzeylerinin, tasarım ortamında güçlü bir yüzey aracı olduğu görülmüş fakat bu yüzey üzerinde kullanılan eğrilerin kullanımı sırasında, konum, teğet ve burulma vektörleri gibi matematik bilgilere ihtiyaç duyulduğundan uygulamada güçlüklerle karşılaşmıştır. Bu zorluklar da Bezier eğrilerinin, yüzeye genişletilmesiyle ortadan kaldırılmış ve elde edilen yüzey örneklerine Bezier yüzey adı verilmiştir.

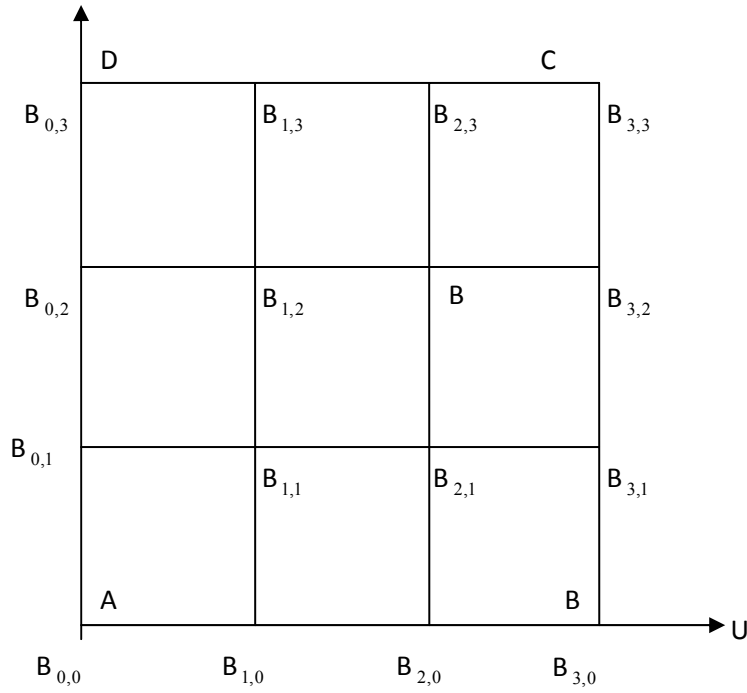
$$Q(u,v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,j} J_{n,i}(u) K_{m,j}(w)$$

Bu denklemde,  $J_{n,i}(u)$  ve  $K_{m,j}(w)$  fonksiyonları, u ve w parametrelerine bağlı Bernstein baz fonksiyonlarıdır. Bu baz fonksiyonları,

$$J_{n,i}(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

$$J_{m,j}(w) = \binom{m}{j} w^j (1-w)^{m-j} \quad \binom{m}{j} = \frac{m!}{j!(m-j)!}$$

Olarak tanımlanır, buradaki n ve m indis değerleri, u ve w yönlerindeki poligon köşelerinden, yani karakteristik çokgen sayılarından bir eksiktir. Kuadratik yüzey yamaları için tanımlanan poligon ağları, topolojik yapı olarak dikkörtgendir. Ağlar her bir sütunda aynı sayıya sahip olmalıdır.

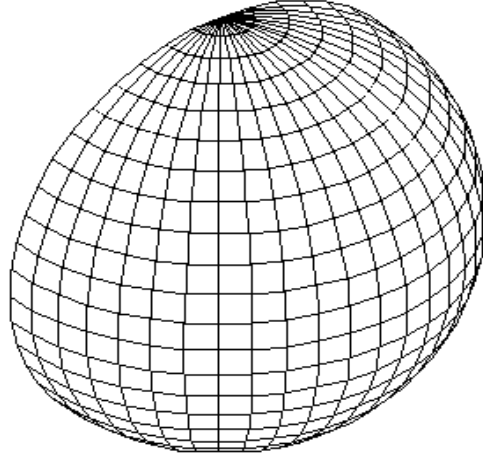


Şekil 4.2. 4x4 Bezier yüzeyi için tanımlanan poligon ağının şeması

Bir Bezier yüzeyinin, sınır eğrilerinin her biri Bezier eğrisidir. Şekil 4.2 de görülen Bezier yüzeyi için tanımlanan bir poligon ağını içerir. Bu ağa göre A noktası üzerinde bulunan u, w yönündeki teğet vektörleri, köşeleri,  $B_{0,1}$  ve  $B_{1,0}$  olan poligon ağları tarafından kontrol edilir. Aynı şekilde B noktası,  $B_{2,0}$  ve  $B_{3,1}$  Köşeleri, C noktası,  $B_{3,2}$  ve  $B_{2,3}$  köşeleri ve D noktası da köşeleri,  $B_{1,3}$  ve  $B_{0,2}$  olan ağlar tarafından kontrol edilir.  $B_{1,1}$ ,  $B_{2,1}$ ,  $B_{2,2}$ ,  $B_{1,2}$  poligon ağ köşeleri de A, B, C, D köşelerindeki büküm vektörlerinin, büyüklüklerini ve yönünü etkiler. Bu nedenle kullanıcı yüzey yamasının şekli kontrol edebilir.

Bezier yüzeyler uygulamada da Bezier eğrilerinde olduğu gibi üzerlerinde bulunan kontrol noktaları tarafından temsil edilirler. Şekil 4.3 de ki yüzey örneğinde, kesikli çizgilerle, yüzey üzerindeki kontrol noktaları, kesin çizgilerle de u ve v sabit parametreleri kullanılarak çizilen eğri yüzeyleri gösterilmektedir.

Eđri zerinden seilen her  $u$  sabit parametre deęerlerine, 0 ile 1 arasında deęiŐen  $v$  parametre deęerleri karŐılık gelir. Eđri zerinden alınan her  $v$  sabit parametre deęeri in de yine aynı Őekilde 0 ile 1 arasında deęer alabilen  $u$  parametreleri vardır. Bezier yzeyler, Bezier eđrilerine gre tasarım uygulamalarında daha etkili olmalarının dıŐında birbirinin aynısıdır.



Őekil 4.3. Bezier eđrileri kullanılarak zilen bezier yzey rneęi

#### 4.2.2. Spline yzeyler

Bezier Yzeylerin geniŐletilmiŐi olan B-Spline yzeyleri,

$$Q(u,v) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M_{j,l}(w)$$

olarak tanımlanır. Bu formlde,  $N_{i,k}(u)$ ,  $M_{j,l}(w)$  baz fonksiyonlardır.

$$N_{i,k}(t) = \left\{ \begin{array}{ll} 1, & x_i \leq u \leq x_{i+1} \\ 0, & \text{diğer durumlarda} \end{array} \right\}$$

$$N_{i,k}(u) = \frac{(u-x_i)N_{i,k-1}(u)}{x_{i+k-1}-x_i} + \frac{(x_{i+k-1}-u)N_{i,k-1}(u)}{x_{i+k-1}-x_i} \quad \text{ve}$$

$$M_{j,l}(t) = \left\{ \begin{array}{ll} 1, & y_i \leq w \leq y_{i+1} \\ 0, & \text{diğer durumlarda} \end{array} \right\}$$

$$M_{i,k}(u) = \frac{(w-y_i)M_{i,l-1}(w)}{y_{i+k-1}-y_i} + \frac{(y_{i+l}-w)M_{j+1,l-1}(w)}{y_{i+l}-y_{i+1}}$$

Olarak tanımlanır. Denklemdaki  $x_i$ ,  $y_i$  ler, düğüm vektörleridir.  $B_{i,j}$  ler de tanımlanan poligon ağının köşeleridir. Alt indis olan  $n$  ve  $m$  değerleri, tanımlanan poligon köşelerinin sayısından en az bir eksik olmalıdır.

B-Spline yüzeylerinin, karakteri ve şekli B-Spline eğrilerinde olduğu gibi düğüm vektörler tarafından belirlenir. Dolayısıyla B-Spline yüzeyleri, bileşenleri olan B-Spline eğrileri ile aynı özelliklere sahiptir. Düğüm vektörler, genelde iki parametrik yönde ve aynı tip seçilirler. Fakat bu şart değildir. Bir parametrik yönde açık düğüm vektörler kullanılırken, diğer parametrik yönde periyodik düğüm vektörler kullanılabilir. [Rogers, D.F. and Adams, J.A., 1990)].

### 4.2.3. Rasyonel yüzeyler

Rasyonel yüzeyleri meydana getirmek için uygulanan yöntem, rasyonel eğriler için kullanılan yöntemin bir genişletilmiştir. Homojen koordinat kavramı burada da uygulanmaktadır.

Tablo 4.1. Rasyonel olan ve rasyonel olmayan Bezier ve B-Spline yüzeylerin denklemleri

$Q(s, t) = \sum_{i=0}^n \sum_{j=0}^m N_{i,n}(s) N_{j,m}(t) V_{i,j}$	$P(s, t) = \sum_{i=0}^n \sum_{j=0}^m N_{i,k}(s) N_{j,t}(t) V_{i,j}$
$Q(s, t) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} N_{i,n}(s) N_{j,m}(t) V_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} N_{i,n}(s) N_{j,m}(t)}$	$P(s, t) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{i,k} N_{i,k}(s) N_{j,t}(t) V_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m w_{i,k} N_{i,k}(s) N_{j,t}(t)}$

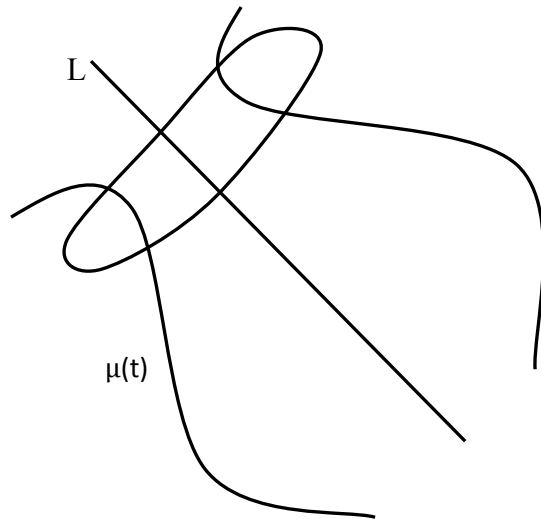
Bu tabloda birinci satır, birinci sütunda rasyonel olmayan Bezier yüzeyin, ikinci sütunda rasyonel olmayan B-Spline yüzeyin denklemleri verilmiş olup ikinci satırda da yine aynı şekilde bu yüzeylerin, rasyonel olanlarının denklemleri verilmiştir.

Burada  $N_{i,k}(s)$  ve  $N_{j,t}(t)$  baz fonksiyonlar,  $V_{i,k}$  kontrol çokyüzlüsü ve  $w_{ij}$  her bir noktadaki ağırlıktır. Rasyonel B-Spline yüzeyler, analitik ve geometrik özellikleri, rasyonel olmayan B-Spline yüzeylerininkine benzerdir. Benzeyen bu özellikler, aşağıdaki gibi sıralanabilir. Yüzeyin karma fonksiyonu ya pozitifdir veya sıfırdır. Yüzeyin en büyük mertebe değeri, her bir parametrik yönünde tanımlanan poligon köşelerinin sayısına eşittir.

#### 4.2.4. Dönel yüzeyler

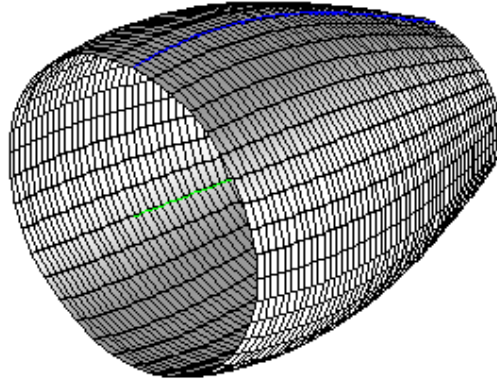
Üç boyutlu yüzey üretme yöntemlerinden biri, bir doğru veya düzlem eğrisinin, bir eksen etrafında döndürülmesidir. Bu biçimde elde edilen yüzeye, dönel yüzey denir. Dönel yüzey üzerindeki herhangi bir nokta,  $t$  ve  $\theta$  parametrelerinin, bir fonksiyonudur.  $t$ , parametresi, döndürülecek değişkeni ve  $\theta$  parametresi de döndürme açısının göstermek üzere doğru parçası üzerinde bir noktanın koordinatı,  $[x(t), 0, z(t)]$  olur ve nokta  $\theta$  açısı kadar döndürüldüğünde de  $[x(t)\cos(\theta), x(t)\sin(\theta), z(t)]$  biçimini alır. Genelde dönel yüzey üzerinde bulunan ve  $z$  eksenini etrafında döndürülerek elde edilen bir nokta,  $[x(t)\cos(\theta), x(t)\sin(\theta), z(t)]$  biçiminde yazılır ve aşağıdaki matrisle gösterilir.

$$P(t, \theta) = \begin{bmatrix} x(t) & 0 & z(t) & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Şekil 4.4. Eğrinin  $L$  doğrultusunda döndürülmesiyle oluşturulan dönel yüzey

Yukarıdaki şekle göre,  $\mu(t)$  eğrisinin L doğrusu çevresinde döndürülmesiyle elde edilen şekle, döneel yüzey denir. Eğri üzerindeki bir noktanın, L doğrusu çevresinde döndürülmesiyle elde edilen çembere, döneel yüzeyin o noktadan geçen paraleli, L doğrusuna, döneel yüzeyin, dönme eksenini, dönme ekseninden geçen bir düzlemlle, döneel yüzeyin arakesiti olan eğriye de, döneel yüzeyin bir meridyeni denir.



Şekil 4.5. Döneel yüzey örneđi

#### 4.2.5. Bazı özel yüzeyler

##### 4.2.5.1. Küre yüzeyi

Küre yüzeyi üzerinde bulunan bir P noktasının koordinatları,  $P(P_1, P_2, P_3)$  ve  $\theta$  ile  $\phi$ , P noktasının eksenlerle yapmış olduđu açılar olmak üzere; P noktasının, parametrik denklemi:

$$P_1 = r \cdot \cos\theta \cos\phi$$

$$P_2 = r \cdot \sin\theta \cos\phi$$

$$P_3 = r \cdot \sin\theta$$

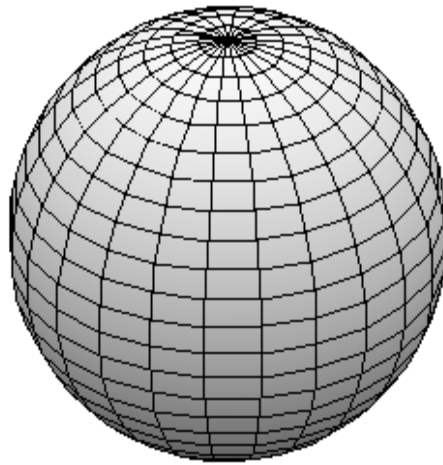
Şeklinde olur. Böylece, küre üzerindeki her bir P noktası için, koordinatlar,

$P = (r \cdot \cos\theta \cdot \cos\phi, r \cdot \sin\theta \cdot \cos\phi, r \cdot \sin\phi)$  şeklinde yazılabilir. Bu yazılanları, düzleme taşımak istersek düzlemde, dik koordinat fonksiyonları  $x_1$  ve  $x_2$  için aşağıdaki gibi yazılabilir ve oluşan eşitliklere de kürenin, parametrik denklemleri denir.

$$y_1 = r \cos x_1 \cdot \cos x_2$$

$$y_2 = r \sin x_1 \cdot \cos x_2$$

$$y_3 = r \sin x_2$$

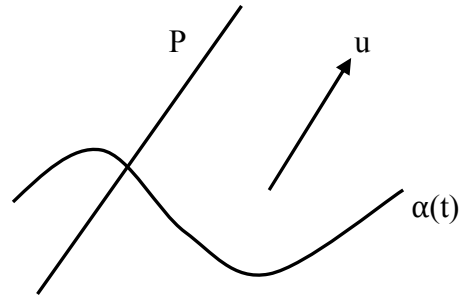


Şekil 4.6. Küre yüzeyi örneği

#### 4.2.5.2 Silindir yüzeyi

$I, \mathbb{R}$ 'nin bir alt aralığı olmak üzere, uzayda  $\alpha : I \rightarrow \mathbb{R}^3$  şeklinde tanımlanan bir  $\alpha$ , eğrisi ile  $u$  vektörü verilsin. Eğrinin her bir  $\alpha(t)$  noktasında,  $u$  vektörüne paralel olan bir ve yalnız bir  $L(t)$  doğrusu vardır. Bu  $L(t)$  doğrularının birleşimi olan yüzeye, silindir yüzeyi adı verilir.  $\alpha$  eğrisine, silindir yüzeyinin dayanak eğrisi,  $u$  vektörüne de silindir yüzeyin doğrultman vektörü denir.  $L(t)$  doğrusu ise silindir yüzeyinin,  $\alpha(t)$  noktasındaki ana doğrusudur. Silindir yüzeyi üzerinde bir  $P$  noktası verildiğinde  $P = \alpha(t) + \mu u$  olacak biçimde  $I \times \mathbb{R}$  kümesinin, bir  $(t, \mu)$  elemanı vardır.





Şekil 4.7. Silindir yüzeyini oluşturan eğri ve ana doğrular

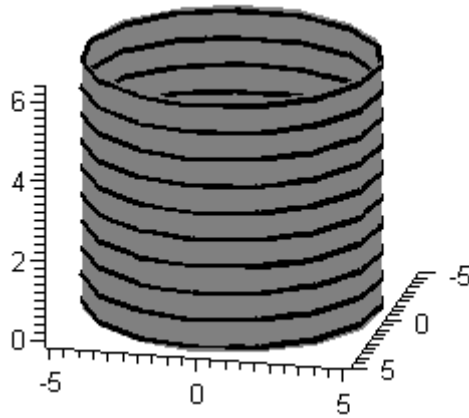
$A=(\alpha_1, \alpha_2, \alpha_3)$   $u=(u_1, u_2, u_3)$  ve  $P=(p_1, p_2, p_3)$  olmak üzere,  $P= \alpha(t) + \mu u$ , eşitliğinden, silindirin parametrik denklemi denilen aşağıdaki eşitlikler elde edilir.

$$p_1 = \alpha_1(t) + \mu u_1$$

$$p_2 = \alpha_2(t) + \mu u_2$$

$$p_3 = \alpha_3(t) + \mu u_3$$

Bu üç eşitlik arasında,  $t$  ve  $\mu$  yok edilirse,  $p_1, p_2, p_3$  arasında,  $F(p_1, p_2, p_3)=0$  biçiminde bir ilişki elde edilir. Bu önermeden de uzaydaki dik koordinat fonksiyonlarına bağlı olarak,  $F(x_1, x_2, x_3)=0$  biçiminde, silindir yüzeyinin denklemi elde edilir.

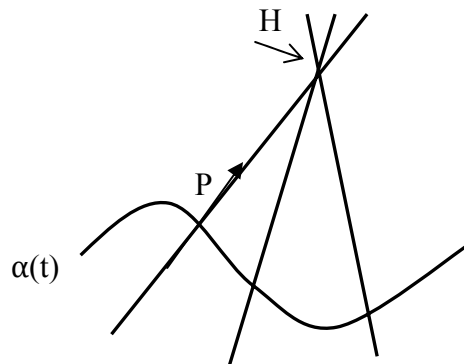


Şekil 4.8. Silindir yüzey örneği

#### 4.2.5.3. Koni yüzeyi

$I, R$ 'nin bir alt aralığı olmak üzere, uzayda  $\alpha : I \rightarrow R^3$  şeklinde tanımlanan bir  $\alpha$ , eğrisi ile bu eğri üzerinde bulunmayan bir  $H$  noktası verilsin. Eğrinin her bir  $\alpha(t)$  noktası için  $\alpha(t)$  ve  $H$  noktası, bir  $L(t)$  doğrusu belirler. Bu  $L(t)$  doğrularının birleşimi olan yüzeye, koni yüzeyi adı verilir.  $\alpha$  eğrisine, koni yüzeyinin dayanak eğrisi,  $H$  noktasına, yüzeyinin tepe noktası denir.  $L(t)$  doğrusu ise koni yüzeyinin,  $\alpha(t)$  noktasındaki ana doğrusudur. Silindirik yüzeyi üzerinde bir  $P$  noktası verildiğinde, bu nokta bir  $L(t)$  doğrusu üzerindedir.  $\Delta(t) = H - \alpha(t)$  olmak üzere,  $P = \alpha(t) + \mu \Delta(t)$  olacak biçimde,  $I \times R$  kümesinin, bir  $(t, \mu)$  elemanı vardır. Burada  $\Delta(t)$  yerine eşitini yazarsak;

$$P = (1 - \mu) \alpha(t) + \mu H$$



Şekil 4.9. Koninin dayanak eğrisi ve ana doğrusu

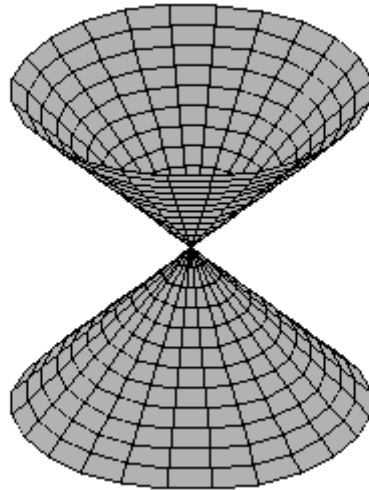
$\alpha = (\alpha_1, \alpha_2, \alpha_3)$ ,  $H = (h_1, h_2, h_3)$  ve  $P = (p_1, p_2, p_3)$  olmak üzere,  
 $P = \alpha(t) + \mu \Delta(t)$  eşitliğinden, koninin parametrik denklemleri elde edilir.

$$p_1 = (1-\mu)\alpha_1(t) + \mu h_1$$

$$p_2 = (1-\mu)\alpha_2(t) + \mu h_2$$

$$p_3 = (1-\mu)\alpha_3(t) + \mu h_3$$

bu üç eşitlik arasında,  $t$  ve  $\mu$  yok edilirse,  $p_1, p_2, p_3$  arasında,  $F(p_1, p_2, p_3) = 0$  biçiminde bir ilişki elde edilir. Bu önermeden de uzaydaki dik koordinat fonksiyonlarına bağlı olarak,  $F(x_1, x_2, x_3) = 0$  biçiminde, koni yüzeyin denklemleri elde edilir.



Şekil 4.10. Koni yüzey örneği

### 4.3. Yüzeylerde Dönüşüm İşlemleri

#### 4.3.1. Öteleme işlemi

Üç boyutlu uzayda aldığımız  $(x,y,z)$  noktasını homojen koordinatlar kullanarak  $(x^1,y^1,z^1)$  noktasına getirebilmek için aşağıdaki dönüşüm matrisini kullanırız.

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

Buradaki  $x'$ ,  $y'$ ,  $z'$  uzaklık koordinatları, herhangi bir reel sayı olabilirler ve ötelenmiş uzaklıklar, yukarıdaki matris kullanılarak aşağıdaki gibi yazılabilir.

$x'$ ,  $y'$ ,  $z'$  uzaklık koordinatları, herhangi bir reel sayı olabilirler ve ötelenmiş uzaklıklar yukarıdaki matris kullanılarak aşağıdaki gibi yazılabilir ve üç boyutlu uzayda nesne üzerinde öteleme işlemi uygulanırken nesne üzerinde tanımlanan her nokta için ayrı ayrı öteleme işlemi uygulanır. Böylece nesnenin konumu da değiştirilmiş olur.

$$x' = x + T_x \qquad y' = y + T_y \qquad z' = z + T_z$$

#### 4.3.2. Ölçekleme işlemi

Üç boyutlu geometrik uzayda ölçekleme işlemi,  $S_x$ ,  $S_y$ ,  $S_z$  ölçekleme parametreleri, olmak üzere aşağıdaki matrisle temsil edilir ve  $(x, y, z)$  koordinat değerlerinin ölçekleme işleminden sonraki değerleri,  $(x', y', z')$ 'de aşağıdaki gibi yazılır.

$$[x' \ y' \ z' \ 1]s = [x \ y \ z \ 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x' = x \cdot S_x$$

$$y' = y \cdot S_y$$

$$z' = z \cdot S_z$$

nesne üzerine yukarıdaki ölçekleme matrisi uygulandığında nesnenin büyüklüğünde değişme olduğu gibi orjine göre öteleme işlemi de olur. Bu ölçekleme işlemi, ölçekleme parametreleri birbirine eşit olmadığında gerçekleşir eğer ölçekleme parametreleri, birbirine eşit ise yani ( $S_x = S_y = S_z$ ) ise nesnenin büyüklüğünde hiçbir değişiklik olmaz.

### 4.3.3. Döndürme işlemi

Üç boyutlu yapılarda döndürme işlemi yapmadan önce hangi eksen üzerine döndürme işlemi uygulayacağımızı belirlemeli ve döndürme açımızı seçmeliyiz. Döndürme işlemi uygulamalarında, en ideal yöntem, koordinat eksenlerine paralel döndürme işlemi yapmaktır. Bunun dışında kendimiz tanımladığımız herhangi bir eksene göre de döndürme yapabiliriz. İki boyutlu yapılarda kullanılan döndürme denklemleri baz alınarak  $\theta$ , döndürme parametresi olmak üzere, sırasıyla  $x$ ,  $y$  ve  $z$  eksenleri üzerine uygulanan döndürme denklemleri aşağıda olduğu gibidir.

$$x' = x$$

$$x' = x \cos \theta - z \sin \theta$$

$$x' = x \cos \theta - y \sin \theta$$

$$y' = y \cos \theta - z \sin \theta$$

$$y' = y$$

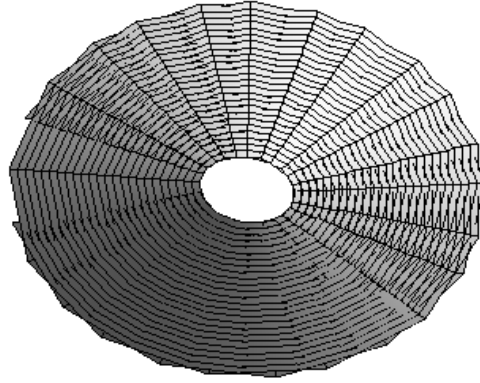
$$y' = x \sin \theta + y \cos \theta$$

$$z' = y \sin \theta + z \cos \theta$$

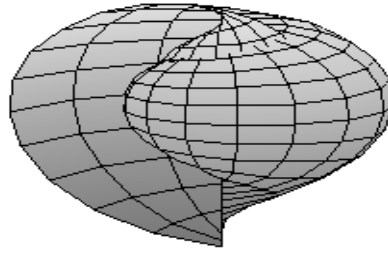
$$z' = z \cos \theta - x \sin \theta$$

$$z' = z$$

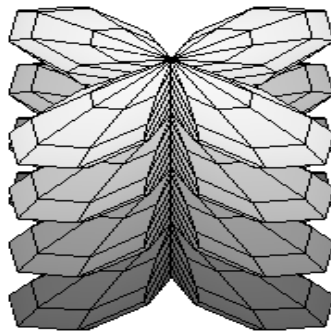
#### 4.4. Dönüşüm İşlemlerinin Yüzelere Uygulanması



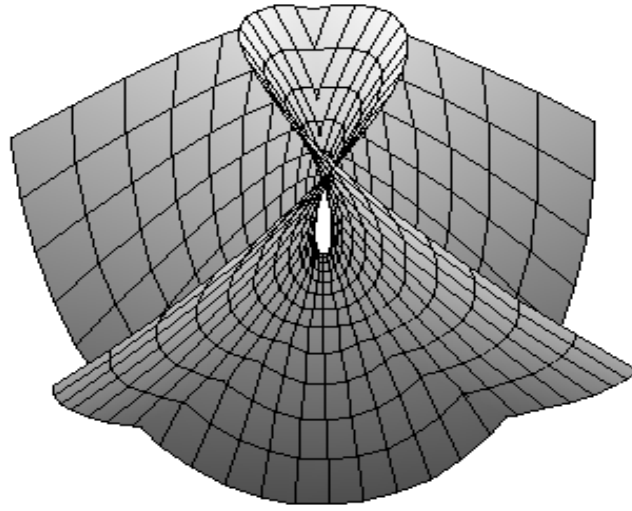
Şekil 4.11. Silindirik koordinatlarda döndürülerek ve ölçeklendirilerek çizilmiş hareketli



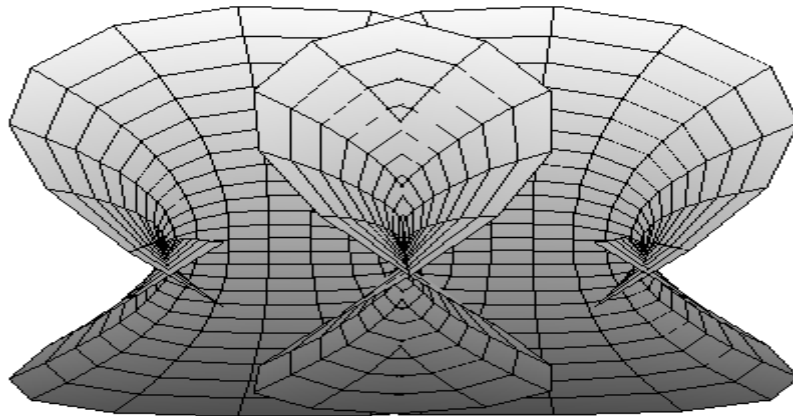
Şekil 4.12. Küresel koordinatlarda ölçeklendirilerek çizilmiş hareketli yüzey



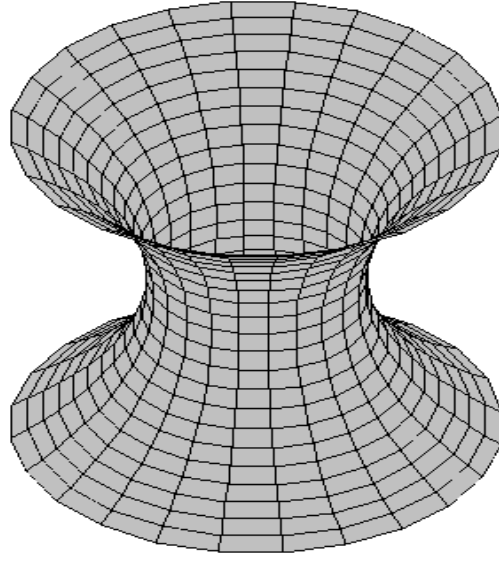
Şekil 4.13. Küresel koordinatlarda ötelenerek çizilmiş hareketli yüzey



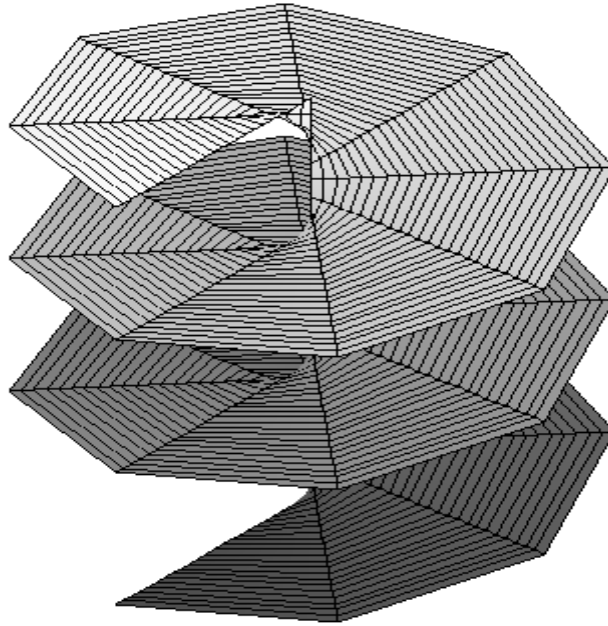
Şekil 4.14. Enneper yüzey örneği



Şekil 4.15 Catalan yüzey örneği



Şekil 4.16. Catenoid yüzey örneği



Şekil 4.17. Helisoid yüzey örneği



## **BÖLÜM 5. SONUÇLAR VE ÖNERİLER**

Bilgisayarın her alandaki gelişmesinin bir sonucu olarak Bilgisayar destekli çizim ve Tasarım uygulamalarının, kullanımı da yaygınlaşmıştır. Çünkü programlama dilleri, tasarımı, daha kısa sürede hazırlayabilme imkanı sunar. Ayrıca görsellikle klasik yaklaşımları birleştirdiğimizde anlatmak istediklerimizin verimliliği artar.

Bu çalışmada da hedeflenenler; geometrideki teorik ve klasik yaklaşımların, programlama diliyle uygulamaya dökülmesidir. Uygulama aracı olarak eğriler ve yüzeyler üzerinde durularak teorik bilgilerle, görsel yaklaşımlar arasında geçişler yapılmış ve bu sayede anlatımı desteklemek amaçlanmıştır.

Yapılmak istenen bu çalışmalar, çok farklı programlama dilleri, üzerine geliştirilebilir. Burada Maple programlama dilinin seçilmesinin, amacı görselliğe hitap etmesinin yanında hareketli uygulamalara da imkan vermesidir. Sonuçta anlatılmak istenenleri ifade etme derecesi bakımından başarılı olmuş, bir dildir. Bundan sonra aynı doğrultuda çalışmak isteyenler için farklı teorik bilgiler üzerine de tavsiye edilebilir.

## KAYNAKLAR

- [1] EDWARDS, C. H. and PENNY, D.E., Calculus with Analytical Geometry, Prentice Hall,1998
- [2] ÇALIŞKAN, A., Diferansiyel Geometri ders notları, 2000
- [3] ÇELİK, B., Maple ve Maple ile Matematik, Nobel Yayıncılık, 2004
- [4] FARİN, G.,Curves and Surface for Computer Aided geometrik design, 1998
- [5] HACISALİHOĞLU, H., ve SABUNCUOĞLU, A., Diferansiyel Geometri, Milli eğitim bakanlığı, 1982
- [6] DONALD, H., and PAULINE, B., Computer Graphics, Prentice Hall, International Editions.
- [7] NEWNES, H., Computer Graphics of Henemen Professional, Ltd.
- [8] BRADLEY, G.L. and SMİTH, K.J., Calculus, Prentice Hall, 1999
- [9] OPREA, J., Differential Geometry and its Applications. Prentice Hall,1997
- [10] MASTIN, C.W., Three Dimensional Bezier Interpolasyon in Solid Modeling and grid generation, Computer Aided Geometrik Design, 1997
- [11] MAO, V.W., and ZHAO, L.H., Free Knot Polynomial Spline Confidence Intervals, 2003
- [12] SPİVAK, M. A.,Comprehensive Introduction to Differential Geometry, 2<sup>nd</sup> edition, 1970
- [13] TANTAY,B., Tasarı Geometri ders notları, 2001
- [14] VERA,B. A., Computer Graphics and Geometric Modeling for Engineers', WILLEY, J. and SOONS, Inc, New York, 1992.

- [15] YAMAGUCHI, F., Curves and Surface in Computer Aided Geometric Design, Springer Verlag, 1989
- [16] WEISA, V., ANDORB, L., RENNERRA, G. and VARADYA, T., Advanced Surface Fitting Technique, Computer Aided Geometric, 2001
- [17] SABUNCUOĞLU, A., Diferansiyel Geometri, Nobel Yayıncılık, 2003

## EKLER

Bu çalışmada kullanılan şekiller, maple programlama dili aracılığı ile çizdirilmiştir. Bu bölümde de çizdirilen şekillerin kodlarına yer verilmiştir.

with (plots):

```
plot3d([arctan(x*y), x + 2*y], x=-Pi..Pi, y=-Pi..Pi, color=[gray,black]); (Şekil 3.1)
```

```
animatecurve([2*cos(t),2*sin(t),t=0..2*Pi],frames=80,thickness=3,color=black,  
axes = framed ); (Şekil 3.2)
```

```
spacecurve([4*cos(t),4*sin(t),2*t],t=0..12*Pi,scaling=constrained,thickness=3,  
color =black,axes=boxed); (Şekil 3.3)
```

```
spacecurve([cos(t),sin(t),t],t=0..5*Pi,axes=normal,color=black,thickness=2);  
(Şekil 3.3)
```

```
plot(-x^2-3*x+4,x=-3..-0.4,thickness=2,color=black,axes=none); (şekil 3.5)
```

```
plot(-12*x^6+4*x^4-24*x^5+36*x^3-4*x,x=-1..1, color=black, axes=framed,  
thickness=2,axes=none); (şekil3.6)
```

```
plot(-2*x^5+3*x^4-4*x^3+6*x^2- 4*x, x=0..1, thickness=2, color=black,  
axes=framed); (şekil3.8)
```

with(CurveFitting):

```
points3:=[[0,0],[0.5,1],[1.3,5],[3,6],[7.4,4],[8,2]]:
```

```
plot( {bsplcurve,points3},color=[gray,red],thickness=[2,3]) (Şekil 3.9)
```

```
u:=plot([[0,0],[1,4],[2,0]],thickness=2,color=gray)
```

```
v:=plot(-x^2+2*x,x=0..2,thickness=2,color=black) display({u,v}); (Şekil 3.10)
```

```
animate(implicitplot,[x^2+A*y+y^2, x=-3..3,y=-8..8, thickness=2,color=black],
A=- 5..5) (Şekil 3.11)
```

```
animate(implicitplot,[x^2+A*y+y^2,x=-3..3,y=-8..8, scaling=constrained,
thickness=2,color=gray], A=-5..5); (Şekil 3.11)
```

```
animatecurve(sin(5*x),x=-3..3,view=0..1,color=black,thickness=2); (şekil 3.12)
```

```
animate( plot, [sin(x)*exp(-x/5), x=0..t], t=0..20, frames=50,thickness=2,
color=black ) (Şekil 3.13)
```

```
animate(plot,[cos(x/2)*sin(3*y),x=0..4*Pi,color=black,thickness=2],y=0..Pi)
( Şekil 3.14)
```

```
animatecurve({[4*t*(1-t)^3+8*t^2*(1-t)^2+6*t^3*(1-t)+6*t^4,6*t*(1-t)^3+
```

```
6*t^2*(1-t)^2+20*t^3*(1-t),t=0..1],[12*(1-t)^3-36*(1-t^2)*t+16*(1-t)*t^2+
```

```
24*t^3,14*t*(1-t)^2,t=0..1]}, thickness=2,axes=framed,color=black); (Şekil 3.15)
```

```
with(plots):
```

```
plot3d([x^2-2*x*y-4*x+y^2+4*y-100],x=-5..5,y=-5..5)
```

```
plot3d([x^2-2*x*y-4*x+y^2+4*y-100],x=-5..5,y=-5..5) (Şekil 4.1)
```

```
plot3d(5,x=0..Pi, y=0..Pi, coords=spherical,scaling=constrained); (Şekil 4.3)
```

```
plot3d(height,angle=0..2*Pi,height=-5..5, coords=cylindrical); (Şekil 4.6)
```

```
plot3d(5,r=0..10,t=0..2*Pi, coords=cylindrical,axes=frame,style=patch,
thickness=2); (Şekil 4.8)
```

```
plot3d(sin(x)+sin(y),x=0..2*Pi, y=0..2*Pi, axes=boxed); (Şekil 4.10)
animate3d([x*u,u*v,u*sin(v*u)],x=1..5,v=1..4,u=3..4,coords=cylindrical);
(Şekil 4.11)
```

```
animate3d(r*t*cos(u),r=0..10,t=0..2*Pi,u=2..4,coords=spherical); (Şekil 4.12)
animate3d(cos(u*v)*sin(u*t),v=-Pi..Pi,t=-Pi..Pi,u=1..2,coords=cylindrical);
(Şekil 4.13)
```

with(plots):

```
plot3d([u-u^3/3+u*v^2,v-v^3/3+v*u^2,u^2-v^2],u=-3..3,v=- 3..3); (Şekil 4.14)
```

with(plots):

```
plot3d([u-sin(u)*cosh(v), 1-cos(u)*cosh(v),4*sin(u/2)*sinh(v/2)] ,u=-2.1*Pi..2.1*Pi,
v=-2.4..2.4); (Şekil 4.15)
```

```
plot3d([u*cos(v),u*sin(v),v],u=0..15,v=0..6*Pi); (Şekil 4.17)
```

## **ÖZGEÇMİŞ**

1978 Sakarya doğumluyum. 2001 yılında Ege Üniversitesi Fen Fakültesi Bilgisayar Bilimleri ağırlıklı Matematik opsiyonundan, 2003 yılında da Ege Üniversitesi Eğitim Fakültesi, Fen ve Matematik alanlar Tezsiz Yüksek Lisans bölümünden mezun oldum. 2001-2006 yılları arası çeşitli dersanelerde çalıştım halen Sakarya Fatih Endüstri Meslek Lisesinde, Matematik öğretmeni olarak görev yapmaktayım.