

57534

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İKİ BOYUTLU KÜBİK SPLINE FONKSİYONLAR VE UYGULAMASI

Hasan YÜCEL

F.B.E. Matematik Anabilim Dalında
hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Behiç ÇAĞAL

İSTANBUL - 1996

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
TEZ YERİNE KÖRGEZİLMİŞTİR

İÇİNDEKİLER

TEŞEKKÜR

ÖZET

SUMMARY

BÖLÜM I

KÜBİK SPLINE FONKSİYONLARI	1
1.1. Giriş	1
1.2. Kübik Spline Fonksiyonu'nun Elde Edilişi	2
1.3. Genelleştirilmiş Spline Fonksiyonu	10

BÖLÜM II

ÇİFT SPLINE UYGULAMA	12
1.1. Giriş	12
2.2. Çift Spline Uygulama	12

BÖLÜM III

İKİ KAT KÜBİK SPLINE	14
3.1. Giriş	14
3.2. Coons Yüzeyleri	15

BÖLÜM IV

İKİ BOYUTTA GENELLEŞTİRİLMİŞ SPLINE	18
4.1. Giriş	18
4.2. Temel Tanımlama	18
4.3. Temel Özdeşlik	20
4.4. Spline Fonksiyon Tipleri	22

BÖLÜM V

ÖRNEKLER.....	24
5.1. Kübik Spline Uygulaması.....	24
5.2. Çift Spline Uygulaması.....	25
5.3. İki-Kat Kübik Spline Uygulamaları.....	26

BÖLÜM VI

PROGRAM	29
SONUÇ.....	56
KAYNAKLAR.....	57

ÖZGEÇMİŞ



TEŞEKKÜR

Çalışmalarım sırasında bana destek olan ve yardımını esirgemeyen çok değerli hocam *Prof. Dr. Behiç ÇAĞAL*'a teşekkür ederim.

Hasan YÜCEL

Haziran, 1996



ÖZET

Tezimde, sayısal analizin önemli konularından olan Spline Fonksiyonları'nın, İki-Boyutlu Kübik Spline konusunu araştırdım.

I. Bölüm'de, Kübik Spline fonksiyonları ve bunların elde edilişine ilişkin açıklamalar yer almaktadır.

Spline yaklaşımından yararlanabilmek için kullanılacak apsis değerlerinin artan bir sırada olması gerekmektedir. Tezin II. Bölüm'ünde de apsis değerlerinin artan olmaması halinde kullanılabilecek bir yöntem anlatılmaktadır.

İki-Boyutlu Spline fonksiyonlarının temelini oluşturan İki-Kat Kübik Spline konusu III. Bölüm'de anlatılmıştır.

İki-Boyutlu Spline fonksiyonları ile ilgili genel tanım ve ifadeler IV. Bölüm'de verilmiştir.

VI. Bölüm'de verilen Spline hesaplama programına ait uygulamalar V. Bölüm'de verilmiştir.

Spline fonksiyonlarının hesaplanmasında yapılması gereken bazı rutin hesaplamaların elektronik hesaplayıcılar tarafından yapılması büyük zaman kayıplarını ortadan kaldıracaktır. Bu nedenle, VI. Bölüm'de, Spline yaklaşımının daha hızlı ve kolay bir biçimde hesaplanmasını sağlayacak bir Spline hesaplama programı eklenmiştir.

SUMMARY

In my thesis, I have researched Two-Dimensional Cubic Spline under Spline Functions which is important subject of the numerical analysis.

In the first chapter, there are Cubic Spline functions and their explanation while their obtained.

In order to make use of the spline approximation the values of abscissa should be in increasing order. On the other hand; in the second chapter also a way is explained to use in the situation where the values of abscissa is not in increasing order.

The subject of Doubly Cubic Spline which is the base of Two-Dimensional Spline is mentioned in chapter three.

The definition and expressions of Two-Dimensional Spline functions are indicated in the forth chapter.

The applications of Spline calculation program that is given in chapter six are given in chapter five.

It will eliminate the waste of time to calculate some routine calculations by electronic calculator during the calculation of spline functions. Because of this reason, in chapter six a spline calculation program is added which will cause spline approximation to be calculated fast and easily.

BÖLÜM I

KÜBİK SPLINE FONKSİYONLARI

1.1. Giriş

Spline fonksiyonlarının konu olarak genel başlama noktası *Kübik Spline Fonksiyonlarıdır*.

Spline fonksiyonları ile yaklaşımlarda elemanter işlemlerin algoritmik düzenlemelere olanak vermesi bu yöntemin etkinliğini artırmaktadır.

Bu bölümünde spline fonksiyonları'nın temel denklemleri verilmiştir.

Spline fonksiyonlarında, $a \leq x \leq b$ aralığında

$$\Delta: a = x_0 < x_1 < \dots < x_N = b$$

apsisleri ile bunların

$$Y: y_0, y_1, \dots, y_N$$

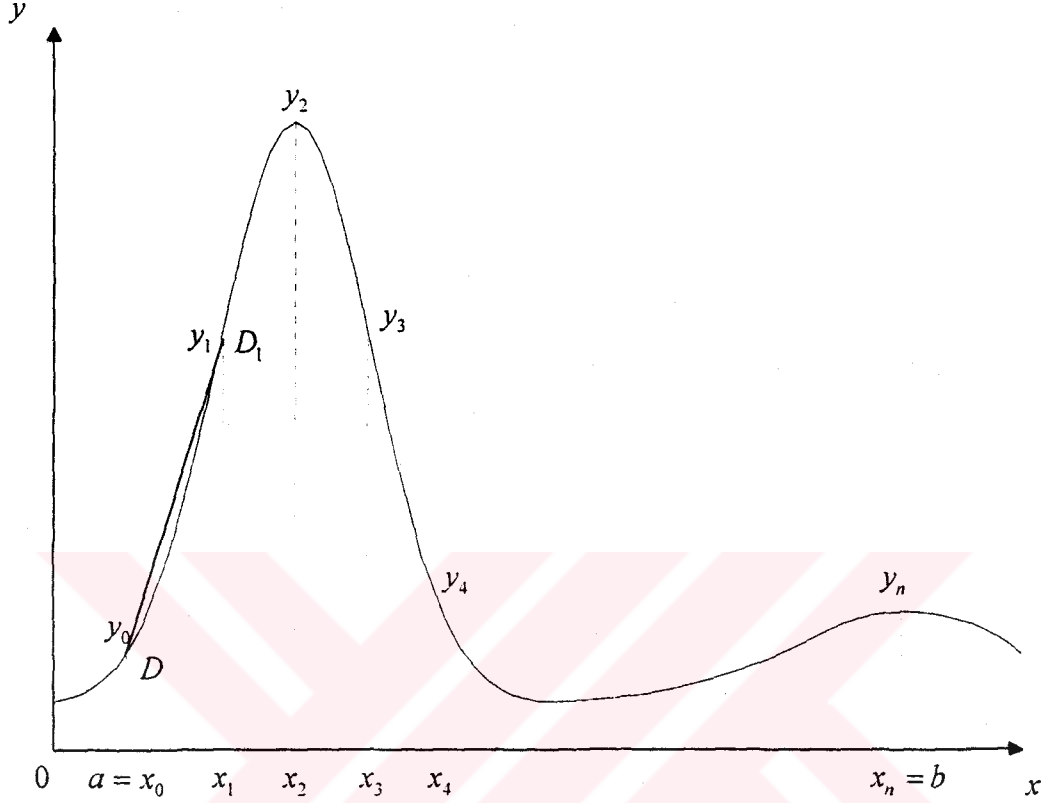
ordinatlarının oluşturduğu ağ ile bölünmesinden oluşan altbölümler göz önüne alınır.

Kendisi, birinci ve ikinci türevleri $[a, b]$ aralığı üzerinde tanımlı ve sürekli olan, her $x_{j-1} \leq x \leq x_j$ ($j = 1, 2, \dots, N$) altaralığında kübik olan ve $S_\Delta(Y; x_j) = y_j$ ($j = 0, 1, \dots, N$) koşulunu sağlayan $S_\Delta(x)$ ya da $S_{\Delta, Y}(x)$ şeklinde gösterilen $S_\Delta(Y; x)$ fonksiyonunu aramaktayız. $S_\Delta(Y; x)$ ya da $S_\Delta(x)$, Δ ağına bağlı spline ya da Δ üzerinde spline denir. Bu fonksiyon ağ bölgelerindeki y_j değerlerine yaklaşım fonksiyonudur.

$$S_\Delta^{(p)}(a+) = S_\Delta^{(p)}(b-) \quad (p = 0, 1, 2)$$

koşulu sağlanırsa spline fonksiyonuna $(b - a)$ periyodunun periyodiği denir.

1.2. Kübik Spline Fonksiyonu'nun Elde Edilişi



Şekil 1

Şekil 1'deki DD_1 doğrusunun denklemi

$$\frac{x - x_0}{y - y_0} = \frac{x_1 - x_0}{y_1 - y_0}$$

dır. Bu ifadeyi düzenlersek;

$$\begin{aligned} (y_1 - y_0)(x - x_0) &= (y - y_0)(x_1 - x_0) \\ (x - x_0)y_1 - (x - x_0)y_0 &= (x_1 - x_0)y - (x_1 - x_0)y_0 \\ (x - x_0)y_1 - (x - x_0)y_0 + (x_1 - x_0)y_0 &= (x_1 - x_0)y \\ (x - x_0)y_1 + (x_1 - x)y_0 &= (x_1 - x_0)y \end{aligned}$$

$$y = \frac{(x_1 - x)}{(x_1 - x_0)} y_0 + \frac{(x - x_0)}{(x_1 - x_0)} y_1 \quad (1.2.1)$$

elde edilir. Bu DD_1 aralığında da fonksiyonumuzu

$$S(x) = ax^3 + bx + c$$

şeklinde bir kübik spline fonksiyonu ile ifade edeceğimiz için $S''(x)$ fonksiyonunun, (1.2.1) denklemi ile ifade ettiğimiz y fonksiyonuna eşit olması gerekir. Bu durumda

$$S''_i(x) = \frac{(x_i - x)}{(x_i - x_{i-1})} y_{i-1} + \frac{(x - x_{i-1})}{(x_i - x_{i-1})} y_i$$

olur.

$S''_\Delta(x_j)$ ifadesini M_j momentleri ile düzenlemek gelenekselleşmiştir. Bununla beraber buradaki M_j momentleri gerçek hayattaki kirişin momenti değildir. Böylece $[x_{j-1}, x_j]$ aralığında $h_j = x_j - x_{j-1}$ olmak üzere ikinci türevin lineerliğinden

$$S''_\Delta(x) = M_{j-1} \frac{x_j - x}{h_j} + M_j \frac{x - x_{j-1}}{h_j} \quad (1.2.2)$$

mevcut olur. Şimdi bu fonksiyonu iki kez integre edip, integrasyon sabitlerini hesaplayalım,

$$S'_\Delta(x) = M_{j-1} \frac{x_j - x}{h_j} + M_j \frac{x - x_{j-1}}{h_j}$$

$$S'_\Delta(x) = -M_{j-1} \frac{(x_j - x)}{2h_j} + M_j \frac{(x - x_{j-1})}{2h_j} + c_1$$

$$S_\Delta(x) = M_{j-1} \frac{(x_j - x)^3}{6h_j} + M_j \frac{(x - x_{j-1})^3}{6h_j} + c_1 x + c_2$$

$S_\Delta(x_i) = y_i$ ($i = 0, 1, 2, \dots, n-1$) olduğundan

$$S_\Delta(x_j) = y_j = M_{j-1} \frac{(x_j - x_j)^3}{6h_j} + M_j \frac{(x_j - x_{j-1})^3}{6h_j} + c_1 x_j + c_2$$

$$S_\Delta(x_{j-1}) = y_{j-1} = M_{j-1} \frac{(x_j - x_{j-1})^3}{6h_j} + M_j \frac{(x_{j-1} - x_{j-1})^3}{6h_j} + c_1 x_{j-1} + c_2 \quad (1.2.3)$$

düzenlersek

$$y_j = M_j \frac{h_j^2}{6} + c_1 x_j + c_2$$

$$y_{j-1} = M_{j-1} \frac{h_j^2}{6} + c_1 x_{j-1} + c_2$$

ifadelerini elde ederiz. Bunları kullanarak c_1 ve c_2 sabitlerini şu şekilde bulabiliriz;

c_1 için;

$$(y_j - y_{j-1}) = \frac{h_j^2}{6} (M_j - M_{j-1}) + c_1 (x_j - x_{j-1})$$

$$c_1 = \frac{(y_j - y_{j-1})}{h_j} - \frac{h_j}{6} (M_j - M_{j-1})$$

c_2 için;

$$y_j x_{j-1} - y_{j-1} x_j = (M_j x_{j-1} - M_{j-1} x_j) \frac{h_j^2}{6} + c_2 (x_{j-1} - x_j)$$

$$c_2 = - \frac{(y_j x_{j-1} - y_{j-1} x_j)}{h_j} + (M_j x_{j-1} - M_{j-1} x_j) \frac{h_j}{6}$$

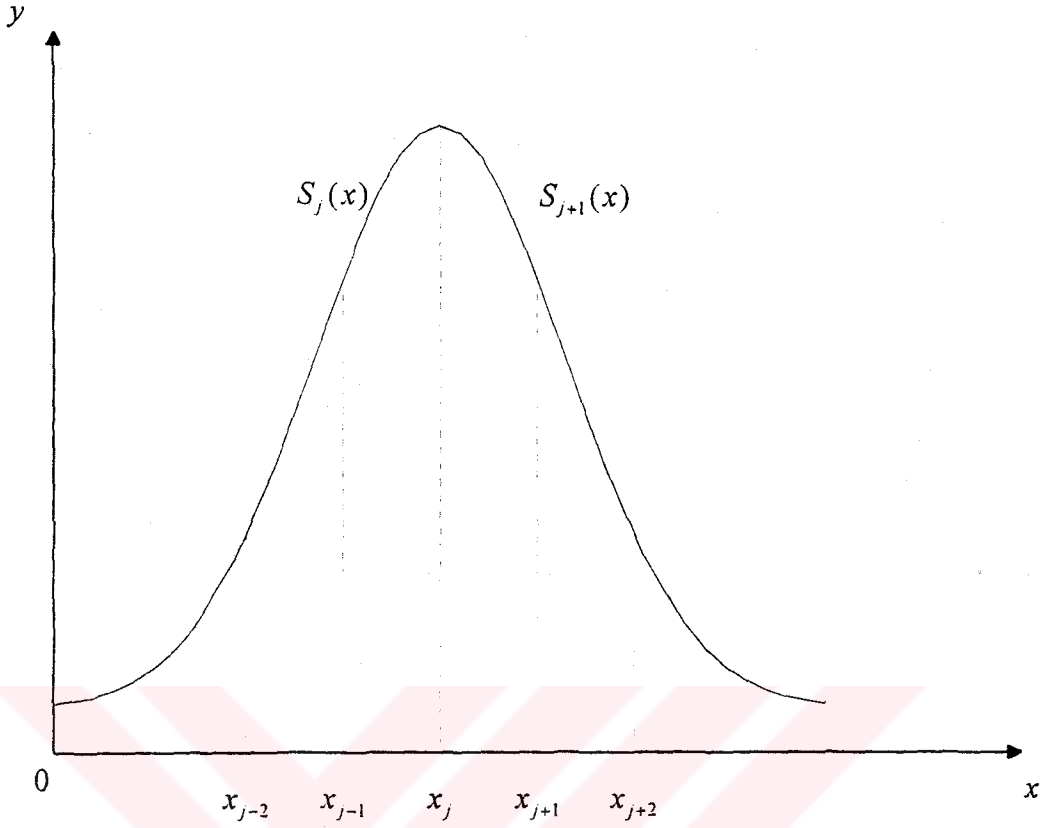
elde edilir.

Yukarıda bulunan c_1 ve c_2 değerlerini de (1.2.3) denkleminde yerine yazıp düzenlersek,

$$S_3(x) = M_{j-1} \frac{(x_j - x)^3}{6h_j} + M_j \frac{(x - x_{j-1})^3}{6h_j} + \left(y_{j-1} - \frac{M_{j-1} h_j^2}{6} \right) \frac{x_j - x}{h_j} + \left(y_j - \frac{M_j h_j^2}{6} \right) \frac{x - x_{j-1}}{h_j} \quad (1.2.4)$$

ifadesini elde ederiz. Bu ifade aradığımız *Spline fonksiyonudur*. Burada M_j ve M_{j-1} bilinmeyenlerdir.

$$S'_3(x) = -M_{j-1} \frac{(x_j - x)^2}{2h_j} + M_j \frac{(x - x_{j-1})^2}{2h_j} + \frac{y_j - y_{j-1}}{h_j} - \frac{M_j - M_{j-1}}{6} h_j \quad (1.2.5)$$



Şekil 2

Şekil 2’de de görüldüğü gibi x_j noktasında $S_j(x)$ ve $S_{j+1}(x)$ fonksiyonları ve bunların türevleri aynı değeri alır.

x_j noktasında 1.2.5 ifadesine sağdan ve soldan yaklaşırsak

$$\begin{aligned} S'_\Delta(x_j-) &= \frac{h_j}{6} M_{j-1} + \frac{h_j}{3} M_j + \frac{y_j - y_{j-1}}{h_j}, \\ S'_\Delta(x_j+) &= -\frac{h_{j+1}}{3} M_j - \frac{h_{j+1}}{6} M_{j+1} + \frac{y_{j+1} - y_j}{h_{j+1}}. \end{aligned} \quad (1.2.6)$$

1.2.2 ve 1.2.4 ifadeleri ışığında $S''_\Delta(x)$ ve $S_\Delta(x)$ fonksiyonlarının $[a, b]$ aralığında sürekli olduğu anlaşılır. $S'_\Delta(x)$ ’in sürekliliğini 1.2.6 ifadesinin yardımıyla

$$\frac{h_j}{6} M_{j-1} + \frac{h_j + h_{j+1}}{3} M_j + \frac{h_{j+1}}{6} M_{j+1} = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j} \quad (1.2.7)$$

koşulu verir.

Periyodik spline fonksiyonu için 1.2.7 eşitliği ($j = 1, 2, \dots, N-1$) aynı anda $M_1, M_2, \dots, M_N$ niceliklerini içeren $N-1$ tane denklem verir. Bu durumda 1.2.7 eşitliğinin $j = N$ için de geçerli olmasını isteriz. Burada $y_N = y_0$, $M_N = M_0$ dır ve $y_{N+1} = y_1$, $M_{N+1} = M_1$, $h_{N+1} = h_1$ olacağı görülmektedir.

Periyodik olmayan spline fonksiyonu için, $N+1$ tane $M_0, M_1, \dots, M_N$ niceliğini belirlemek için ek olarak "sınır koşulları"nı belirlemek gerekir. $S'_\Delta(a) = y'_0$ ve $S'_\Delta(b) = y'_N$ için 1.2.6'dan

$$2M_0 + M_1 = \frac{6}{h_1} \left(\frac{y_1 - y_0}{h_1} - y'_0 \right), \quad M_{N-1} + 2M_N = \frac{6}{h_N} \left(y'_N - \frac{y_N - y_{N-1}}{h_N} \right)$$

ilişkilerini elde ederiz. Bitiş noktasındaki temel yerleştirmeye bağlı olarak $M_0 = 0$ ve $M_N = 0$ alınır.

$$M_0 - \lambda M_1 = 0, \quad 1 > \lambda > 0$$

koşulu, $x_{-1} = (x_0 - \lambda x_1)/(1 - \lambda)$ temel desteklemede yerleştirmekle aynıdır ve $x_{-1} \leq x \leq x_1$ deki bütün eğrilerin, kübiğin yayı olmasını gerektirir. λ 'nın genel seçimi $\frac{1}{2}$ dir.

Genelde bitiş koşulları ile ilgili olarak uygunluk için

$$2M_0 + \lambda_0 M_1 = d_0, \quad \mu_N M_{N-1} + 2M_N = d_N \quad (1.2.8)$$

formunda yazılır.

$$\lambda_j = \frac{h_{j-1}}{h_j + h_{j-1}}, \quad \mu_j = 1 - \lambda_j \quad (j = 1, 2, \dots, N-1)$$

notasyonlarını oluşturalım. 1.2.7 süreklilik gerektirmesi

$$\mu_j M_{j-1} + 2M_j + \lambda_j M_{j+1} = 6 \frac{[(y_{j+1} - y_j) h_{j+1}] - [(y_j - y_{j-1}) h_j]}{h_j + h_{j+1}} \quad (1.2.9)$$

şeklini alır.

Periyodik olmayan spline fonksiyonu için, 1.2.8 ve 1.2.9 denklemleri, d_j ($j = 1, 2, \dots, N-1$) 1.2.9 denkleminde eşitliğin sağ tarafını göstermek üzere

$$\begin{bmatrix} 2 & \lambda_0 & 0 & \cdots & 0 & 0 & 0 \\ \mu_1 & 2 & \lambda_1 & \cdots & 0 & 0 & 0 \\ 0 & \mu_2 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & \lambda_{N-2} & 0 \\ 0 & 0 & 0 & \cdots & \mu_{N-1} & 2 & \lambda_{N-1} \\ 0 & 0 & 0 & \cdots & 0 & \mu_N & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{N-2} \\ M_{N-1} \\ M_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix} \quad (1.2.10)$$

şeklinde yazılır.

Periyodik spline fonksiyonu için tanımlanan denklem $M_0 = M_N$ ve $\lambda_N = h_1/(h_N + h_1)$, $\mu_N = 1 - \lambda_N$ olmak üzere

$$\begin{bmatrix} 2 & \lambda_1 & 0 & \cdots & 0 & 0 & 0 \\ \mu_2 & 2 & \lambda_2 & \cdots & 0 & 0 & 0 \\ 0 & \mu_3 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & \lambda_{N-2} & 0 \\ 0 & 0 & 0 & \cdots & \mu_{N-1} & 2 & \lambda_{N-1} \\ 0 & 0 & 0 & \cdots & 0 & \mu_N & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{N-2} \\ M_{N-1} \\ M_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix} \quad (1.2.11)$$

olur.

Bir çok uygulama için, $m_j = S'_\Delta(x_j)$ eğimleri ile çalışmak M_j momentleri ile çalışmaktan daha uygundur. Burada $[x_{j-1}, x_j]$ üzerinde

$$S_\Delta(x) = m_{j-1} \frac{(x_j - x)^2(x - x_{j-1})}{h_j^2} - m_j \frac{(x - x_{j-1})^2(x_j - x)}{h_j^2} + y_{j-1} \frac{(x_j - x)^2[2(x - x_{j-1}) + h_j]}{h_j^3} + y_j \frac{(x - x_{j-1})^2[2(x_j - x) + h_j]}{h_j^3}, \quad (1.2.12)$$

$$S'_\Delta(x) = m_{j-1} \frac{(x_j - x)(2x_{j-1} + x_j - 3x)}{h_j^2} - m_j \frac{(x - x_{j-1})(2x_j + x_{j-1} - 3x)}{h_j^2} + \frac{y_j - y_{j-1}}{h_j^3} 6(x_j - x)(x - x_{j-1}) \quad (1.2.13)$$

denklemlerini elde ederiz.

Bundan başka, ikinci türev

$$S''_\Delta(x) = -2m_{j-1} \frac{2x_j + x_{j-1} - 3x}{h_j^2} - 2m_j \frac{2x_{j-1} + x_j - 3x}{h_j^2} + 6 \frac{y_j - y_{j-1}}{h_j^3} (x_j + x_{j-1} - 2x), \quad (1.2.14)$$

formunu alır, öyleyse x_j 'nin sağ ve sol limitleri aşağıdaki şekildedir:

$$S''_\Delta(x_j -) = \frac{2m_{j-1}}{h_j} + \frac{4m_j}{h_j} - 6 \frac{y_j - y_{j-1}}{h_j^2}, \quad (1.2.15)$$

$$S''_\Delta(x_j +) = -\frac{4m_{j-1}}{h_{j+1}} - \frac{2m_{j+1}}{h_{j+1}} + 6 \frac{y_{j+1} - y_j}{h_{j+1}^2}$$

x_j 'de ($j = 1, 2, \dots, N-1$) $S''_\Delta(x)$ 'nin süreklilik koşulu kabul edilir. Bu gerektirmenin sonuçları

$$\frac{1}{h_j} m_{j-1} + 2 \left(\frac{1}{h_j} + \frac{1}{h_{j+1}} \right) m_j + \frac{1}{h_{j+1}} m_{j+1} = 3 \frac{y_j - y_{j-1}}{h_j^2} + 3 \frac{y_{j+1} - y_j}{h_{j+1}^2} \quad (1.2.16)$$

ya da daha uygun formda

$$\lambda_j m_{j-1} + 2m_j + \mu_j m_{j+1} = 3\lambda_j \frac{y_j - y_{j-1}}{h_j} + 3\mu_j \frac{y_{j+1} - y_j}{h_{j+1}} \quad (1.2.17)$$

şeklinde ifade edilebilir.

Periyodik olmayan spline fonksiyonu için sistemi, buna göre, $2m_0 + \mu_0 m_1 = c_0$, $\lambda_N m_{N-1} + 2m_N = c_N$ olmak üzere

$$\begin{bmatrix} 2 & \mu_0 & 0 & \cdots & 0 & 0 & 0 \\ \lambda_1 & 2 & \mu_1 & \cdots & 0 & 0 & 0 \\ 0 & \lambda_2 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & \mu_{N-2} & 0 \\ 0 & 0 & 0 & \cdots & \lambda_{N-1} & 2 & \mu_{N-1} \\ 0 & 0 & 0 & \cdots & 0 & \lambda_N & 2 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{N-2} \\ m_{N-1} \\ m_N \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-2} \\ c_{N-1} \\ c_N \end{bmatrix} \quad (1.2.18)$$

dir.

c_j ($j = 1, 2, \dots, N-1$) nicelikleri, 1.2.17 denkleminin sağ tarafını ifade etmektedir.

Periyodik durumda denklem,

$$\begin{bmatrix} 2 & \mu_1 & 0 & \cdots & 0 & 0 & 0 \\ \lambda_2 & 2 & \mu_2 & \cdots & 0 & 0 & 0 \\ 0 & \lambda_3 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & \mu_{N-2} & 0 \\ 0 & 0 & 0 & \cdots & \lambda_{N-1} & 2 & \mu_{N-1} \\ 0 & 0 & 0 & \cdots & 0 & \lambda_N & 2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_{N-2} \\ m_{N-1} \\ m_N \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{N-2} \\ c_{N-1} \\ c_N \end{bmatrix} \quad (1.2.19)$$

olur.

Sınır koşulları, M_j 'ye göre aşağıdaki gibi sınıflayabiliriz:

$x = a$ 'da;

- (i) $2 M_0 + M_1 = \frac{6}{h_1} \left(\frac{y_1 - y_0}{h_1} - y'_0 \right),$
- (ii) $2 M_0 = 2 y''_0,$
- (iii) $2 M_0 + \lambda_0 M_1 = d_0,$

$x = b$ 'de;

(1.2.20)

$$(i) \quad M_{N-1} + 2 M_N = \frac{6}{h_N} \left(y'_N - \frac{y_N - y_{N-1}}{h_N} \right),$$

$$(ii) \quad 2 M_N = 2 y''_N,$$

$$(iii) \quad \mu_N M_{N-1} + 2 M_N = d_N,$$

ve m_j 'ye göre;

$$x = a' \text{ da;}$$

$$(i) \quad 2m_0 = 2y'_0,$$

$$(ii) \quad 2m_0 + m_1 = 3 \frac{y_1 - y_0}{h_1} - \frac{h_1}{2} y''_0,$$

$$(iii) \quad 2m_0 + \mu_0 m_1 = c_0,$$

$$x = b' \text{ de;} \quad (1.2.21)$$

$$(i) \quad 2m_N = 2y'_N,$$

$$(ii) \quad m_{N-1} + 2m_N = 3 \frac{y_N - y_{N-1}}{h_N} - \frac{h_N}{2} y''_N,$$

$$(iii) \quad \lambda_N m_{N-1} + 2m_N = c_N,$$

Burada y''_0 ve y''_N , spline fonksiyonunun ikinci dereceden türevinin aralığın sınır noktalarındaki değerleri olarak kabul edilmiştir. Bundan başka bu iki küme aynı şekilde

$$\mu_0 = 4(1 - \lambda_0)/(4 - \lambda_0),$$

$$\lambda_N = 4(1 - \mu_N)/(4 - \mu_N),$$

$$c_0 = -\frac{d_0 h_1}{4 - \lambda_0} + 6 \frac{2 - \lambda_0}{4 - \lambda_0} \frac{y_1 - y_0}{h_1},$$

$$c_N = \frac{d_N h_N}{4 - \mu_N} + 6 \frac{2 - \mu_N}{4 - \mu_N} \frac{y_N - y_{N-1}}{h_N}$$

koşullarını sağlar.

1.3. Genelleştirilmiş Spline Fonksiyonu

Genel olarak,

$$L \equiv a_n(x) \cdot D^n + a_{n-1}(x) \cdot D^{n-1} + \dots + a_0(x) \quad (1.3.1)$$

şeklinde tanımlanmış bir L lineer diferansiyel operatörü olsun. Burada her $a_j(x)$ ($j = 0, 1, \dots, n$), $C^n[a, b]$ de yer almaktadır ve $a_n(x)$, $[a, b]$ üzerinde mevcuttur.

L^* , L 'nin formal bileşkesi olsun, öyle ki,

$$L^* = (-1)^n \cdot D^n \{a_n(x)\} + (-1)^{n-1} \cdot D^{n-1} \{a_{n-1}(x)\} + \dots - D \{a_1(x)\} + a_0(x). \quad (1.3.2)$$

Eğer $\Delta: a = x_0 < x_1 < \dots < x_n = b$, $[a, b]$ üzerinde bir ağ ise $S_\Delta(x)$ fonksiyonu, Δ 'ya bağlı, $\mathcal{H}^{2n-k}(a, b)$ de yer alan ve Δ 'nın her açık aralığında

$$L^* L S_\Delta = 0 \quad (1.3.3)$$

denklemini sağlayan k ($0 \leq k \leq n$) eksikli *genelleştirilmiş spline fonksiyonudur*.

BÖLÜM II

ÇİFT SPLINE UYGULAMA

1.1. Giriş

Bilindiği gibi spline yaklaşımının uygulanabilmesi için kullanılacak apsis değerlerinin artan bir sırada olması gerekir. Yani bir $[a, b]$ aralığı ele alındığı zaman

$$\Delta: a = x_0 < x_1 < \dots < x_N = b$$

olmak durumundadır. Bunun dışında bir durum söz konusu olduğu zaman, yani kullanılan x değerleri artan bir sırada olmadığı zaman I. Bölümde anlatılan spline yaklaşımını kullanmak mümkün değildir. Bu yüzden, x değerlerinin artan olmadığı durumlar için şu yöntem önerilebilir;

2.2. Çift Spline Uygulama

Verilen x değerlerinin değişiminin, bir f fonksiyonuna, y değerlerinin değişiminin de bir g fonksiyonuna bağlı olduğunu düşünelim. Bu durumda f ve g fonksiyonlarının ayrı ayrı spline yaklaşım fonksiyonlarını elde etmemiz mümkündür.

$$\Delta_1: a = t_0 < t_1 < \dots < t_N = b \quad (2.2.1)$$

$$\Delta_2: c = s_0 < s_1 < \dots < s_N = d \quad (2.2.2)$$

(2.2.1) ifadesi ile verilen $[a, b]$ aralığında artan bir t ele alalım. Ayrıca (2.2.2) ifadesi ile verilen $[c, d]$ aralığında artan bir s ele alalım (Burada s ve t 'nin artımları ne kadar küçük olursa yaklaşım o kadar iyi olur). Bu durumda $[a, b]$ aralığı boyunca t değişkenine bağlı bir $x = f(t)$ fonksiyonu mevcut olur. Aynı şekilde $[c, d]$ aralığı boyunca s değişkenine bağlı bir $y = g(s)$ fonksiyonu mevcut olur. Bu durumda $f(t)$

fonksiyonuna yaklaşımlar olarak $S_{\Delta_1}(t)$ spline fonksiyonunu elde ederiz. Aynı şekilde $g(s)$ fonksiyonuna yaklaşımlar olarak da $S_{\Delta_2}(s)$ spline fonksiyonunu elde ederiz.

Bulunan bu iki spline fonksiyonu kullanarak $[a,b]$ aralığındaki gerçekte fonksiyonun değerlerine yaklaşımlar bulmak mümkündür. Buradan da anlaşılacağı gibi esas yaklaşımlar fonksiyonumuz $y = \varphi(x)$ ise, burada $x = S_{\Delta_1}(t)$ ve $y = S_{\Delta_2}(s)$ olacaktır. Elde edilecek olan $[x,y]$ ikilileri φ fonksiyonunun yaklaşımlar değerleridir. Bu konuyla ilgili uygulama 5. Bölüm'de mevcuttur.

BÖLÜM III

İKİ KAT KÜBİK SPLINE

3.1. Giriş

Şu ana kadar anlatılan spline teorisi bir-boyutlu idi. Bu bölümde anlatılacak olan iki kat kübik spline, *iki-boyutlu kübik spline fonksiyonlarının* temelini oluşturmaktadır. İki-boyutlu kübik spline fonksiyonlarının daha genel teorisi IV. Bölüm'de anlatılmaktadır.

Bir düzlemin $\mathcal{R}: a \leq t \leq b; c \leq s \leq d$ şeklindeki dikdörtgen bölgesi verilmiş olsun. Eğer $\Delta_t: a = t_0 < t_1 < \dots < t_N = b$ ve $\Delta_s: c = s_0 < s_1 < \dots < s_M = d$ gibi iki boyutlu ağları ele alırsak, $P_{ij} = (t_i, s_j)$, $\mathcal{R}$ 'yi

$$\left\{ \mathcal{R}_{ij}: t_{i-1} \leq t \leq t_i; s_{j-1} \leq s \leq s_j \right\} \quad (i = 1, 2, \dots, N; j = 1, 2, \dots, M)$$

altdikdörtgen ailelerine bölen olmak üzere iki-boyutlu çözümleme ağı $\pi = \{P_{ij}\}$ ($i = 0, 1, \dots, N; j = 0, 1, \dots, M$) dir.

$\mathcal{R}$ üzerinde ve π 'ye göre basit iki kat kübik spline fonksiyonu $S_\pi(t, s)$ (1) $\mathcal{R}_{ij}$ dikdörtgeninde çifte kübiktir ve (2) tek değişkene göre n . dereceden kısmi türevini hesaplamak için r tane diferansiyelden fazla gerektirmeyen sürekli $\mathcal{R}$ üzerindeki $f(t, s)$ fonksiyonlarının ailesi $C_r^n(\mathcal{R})$ olmak üzere $C_2^1(R)$ 'nin elemanıdır.

Spline fonksiyonlarının I. Tip gösterimi, spline fonksiyonlarının tanım değerleri olan aşağıdaki kısmi türevleri içerir;

$$\left. \begin{aligned} (a) \quad & \frac{\partial S_\pi}{\partial t}, \{P_{ij}\} \text{ noktalarında } (i = 0, N; j = 0, 1, \dots, M) \\ (b) \quad & \frac{\partial S_\pi}{\partial s}, \{P_{ij}\} \text{ noktalarında } (i = 0, 1, \dots, N; j = 0, M) \\ (c) \quad & \frac{\partial^2 S_\pi}{\partial t \partial s}, \{P_{ij}\} \text{ noktalarında } (i = 0, N; j = 0, M) \end{aligned} \right\} \quad (3.1.1)$$

$S_\pi(t,s)$ 'yi göstermek için diğer parametrelere ihtiyaç vardır, bazen bunlar belirlenemez, fakat π üzerindeki interpolasyonun spline fonksiyonu için, π 'nin ağ noktalarındaki $S_\pi(t,s)$ 'in değerleridir. Benzer şekilde II. Tip gösterimi de spline fonksiyonun aşağıdaki tanım değerlerini içerecek şekilde belirlenir.

$$\left. \begin{aligned} (a) \quad & \frac{\partial^2 S_\pi}{\partial t^2}, \{P_{ij}\} \text{ noktalarında } (i = 0, N; j = 0, 1, \dots, M) \\ (b) \quad & \frac{\partial^2 S_\pi}{\partial s^2}, \{P_{ij}\} \text{ noktalarında } (i = 0, 1, \dots, N; j = 0, M) \\ (c) \quad & \frac{\partial^4 S_\pi}{\partial t^2 \partial s^2}, \{P_{ij}\} \text{ noktalarında } (i = 0, N; j = 0, M) \end{aligned} \right\} \quad (3.1.2)$$

3.2. Coons Yüzeyleri

Coons [1964] $\mathcal{R}: a \leq t \leq b; c \leq s \leq d$ dikdörtgensel alanını $\mathcal{R}_{ij}: t_{i-1} \leq t \leq t_i; s_{j-1} \leq s \leq s_j$ ($i = 1, 2, \dots, N; j = 1, 2, \dots, M$) alt dikdörtgenlerine bölerek bir metod geliştirmiştir. Sonra her $\mathcal{R}_{ij}$ için bir $S(t,s)$ yüzeyi tanımlamıştır. Bu $S(t,s)$ yüzeyi, R 'yi $\mathcal{R}_{ij}$ alt dikdörtgenlerine bölen şebeke çizgileri üzerinde tanımlı, verilmiş olan $f_i(s)$ ($i = 0, 1, \dots, N$), $g_j(t)$ ($j = 0, 1, \dots, M$) eğrilerinin bir kümesinin kenarları ile sınırlı $\mathcal{R}_{ij}$ 'nin her dört kenarı boyunca uygun yüzeydir. $S(t,s)$ 'in, $\mathcal{R}_{ij}$ 'nin kenarlarındaki normal yöndeki birinci dereceden kısmi türevi sadece, kenarların sonlarındaki sınır değerlerindeki değerine ve sınır değerlerindeki ilk türev değerine bağlıdır. Özellikle her $\mathcal{R}_{ij}$ için $S(t, s_{j-1}) = g_{j-1}(t)$, $t_{j-1} \leq t \leq t_i$ ve $\partial S(t,s)/\partial s|_{s=s_{j-1}}, t_{j-1} \leq t \leq t_i$ sadece $f_{j-1}(s_{j-1})$, $f_{j-1}(s_i)$, $f'_{j-1}(s_{j-1})$, $f'_{j-1}(s_i)$ niceliklerine bağlıdır. Sonuç olarak, $g_{j-1}(t)$, $t_{j-1} \leq t \leq t_i$ aralığında $S(t,s)$ 'nin normal türevini etkilemeden tanımlanabilir.

Bu amaçlar için birden fazla metod vardır. Burada, Spline teorisi ile ilişkili bir metodun ana hatları verilmeye çalışılmıştır.

$g_{j-1}(t)$ ve $g_j(t)$, $s = s_{j-1}$ ve $s = s_j$ boyunca $S(t,s)$ 'yi tanımlasın. Benzer şekilde, $f_{i-1}(s)$ ve $f_i(s)$, $t = t_{i-1}$ ve $t = t_i$ boyunca $S(t,s)$ 'yi tanımlasın. Bu dört fonksiyon, $\mathcal{R}_{ij}$ 'deki $S(t,s)$, $\partial S(s,t)/\partial s$ ve $\partial S(s,t)/\partial t$ 'nin değerlerini belirler. Son olarak $\mathcal{R}_{ij}$ 'deki keyfi belirlenmiş noktalardaki $\partial^2 S(s,t)/\partial s \partial t$ 'nin değerini belirler. Her $\mathcal{R}_{ij}$ için sadece bir tek $\hat{S}(t,s)$ çift kübiği vardır ve şu 16 denklikle belirlenir;

$$\left. \begin{aligned} S_1(t,s) &= \left\{ g_{j-1}(t) - \hat{S}(t, s_{j-1}) \right\} \left\{ -2 \left(\frac{s_j - s}{s_j - s_{j-1}} \right)^3 + 3 \left(\frac{s_j - s}{s_j - s_{j-1}} \right)^2 \right\} \\ S_2(t,s) &= \left\{ g_j(t) - \hat{S}(t, s_j) \right\} \left\{ -2 \left(\frac{s - s_{j-1}}{s_j - s_{j-1}} \right)^3 + 3 \left(\frac{s - s_{j-1}}{s_j - s_{j-1}} \right)^2 \right\} \\ S_3(t,s) &= \left\{ f_{j-1}(t) - \hat{S}(t_{j-1}, s) \right\} \left\{ -2 \left(\frac{t_j - t}{t_j - t_{j-1}} \right)^3 + 3 \left(\frac{t_j - t}{t_j - t_{j-1}} \right)^2 \right\} \\ S_4(t,s) &= \left\{ f_j(s) - \hat{S}(t_j, s) \right\} \left\{ -2 \left(\frac{t - t_{j-1}}{t_j - t_{j-1}} \right)^3 + 3 \left(\frac{t - t_{j-1}}{t_j - t_{j-1}} \right)^2 \right\} \end{aligned} \right\} \quad (3.2.1)$$

sonra

$$S(t,s) = \hat{S}(t,s) + \sum_{i=1}^4 S_i(t,s) \quad (3.2.2)$$

dir.

$\hat{S}(t,s)$ 'yi belirleyelim. $\mathcal{R}_j$, birim kenar uzunluklu bir kare olsun,

$$\begin{aligned} \hat{S}(t,s) &= (s - s_{j-1})^3 \{ a_{11}(t - t_{j-1})^3 + a_{12}(t - t_{j-1})^2 + a_{13}(t - t_{j-1}) + a_{14} \} \\ &+ (s - s_{j-1})^2 \{ a_{21}(t - t_{j-1})^3 + a_{22}(t - t_{j-1})^2 + a_{23}(t - t_{j-1}) + a_{24} \} \\ &+ (s - s_{j-1}) \{ a_{31}(t - t_{j-1})^3 + a_{32}(t - t_{j-1})^2 + a_{33}(t - t_{j-1}) + a_{34} \} \\ &+ \{ a_{41}(t - t_{j-1})^3 + a_{42}(t - t_{j-1})^2 + a_{43}(t - t_{j-1}) + a_{44} \} \end{aligned} \quad (3.2.3)$$

olur. Buradan hareketle,

$$a_{44} = S(t_{j-1}, s_{j-1}), \quad a_{34} = \partial S / \partial s \big|_{t_{j-1}, s_{j-1}}, \quad a_{43} = \partial S / \partial t \big|_{t_{j-1}, s_{j-1}}, \quad a_{33} = \partial^2 S / \partial s \partial t \big|_{t_{j-1}, s_{j-1}}$$

$$a_{14} + a_{24} = S(t_{i-1}, s_j) - a_{34} - a_{44}$$

$$3a_{14} + 2a_{24} = \partial S / \partial s \big|_{t_{i-1}, s_j} - a_{34}$$

$$a_{13} + a_{23} = \partial S / \partial t \big|_{t_{i-1}, s_j} - a_{33} - a_{43}$$

$$3a_{13} + 2a_{23} = \partial^2 S / \partial s \partial t \big|_{t_{i-1}, s_j} - a_{33}$$

$$a_{41} + a_{42} = S(t_i, s_{j-1}) - a_{43} - a_{44}$$

$$3a_{41} + 2a_{42} = \partial S / \partial t \big|_{t_i, s_{j-1}} - a_{43}$$

$$a_{31} + a_{32} = \partial S / \partial s \big|_{t_i, s_{j-1}} - a_{33} - a_{34}$$

$$3a_{31} + 2a_{32} = \partial^2 S / \partial s \partial t \big|_{t_i, s_{j-1}} - a_{33}$$

$$a_{11} + a_{12} + a_{21} + a_{22} = S(t_i, s_j) - a_{13} - a_{14} - a_{23} - a_{24} - a_{31} - a_{32} \\ - a_{33} - a_{34} - a_{41} - a_{42} - a_{43} - a_{44}$$

$$3a_{11} + 3a_{12} + 2a_{21} + 2a_{22} = \partial S / \partial s \big|_{t_i, s_j} - 3a_{13} - 3a_{14} - 2a_{23} - 2a_{24} - a_{31} - a_{32} - a_{33} - a_{34}$$

$$3a_{11} + 2a_{12} + 3a_{21} + 2a_{22} = \partial S / \partial t \big|_{t_i, s_j} - a_{13} - a_{23} - 3a_{31} - 2a_{32} - a_{33} - 3a_{41} - 2a_{42} - a_{43}$$

$$9a_{11} + 6a_{12} + 6a_{21} + 4a_{22} = \partial^2 S / \partial s \partial t \big|_{t_i, s_j} - 3a_{13} - 2a_{23} - 3a_{31} - 2a_{32} - a_{33}$$

bulunur.

BÖLÜM IV

İKİ BOYUTTA GENELLEŞTİRİLMİŞ SPLINE

4.1. Giriş

Burada dikkatimizi, her $\mathcal{R}_y$ alt dikdörtgeninde iki diferansiel denklemi sağlayan $S_x(t,s)$ spline fonksiyonları üzerinde yoğunlaştıracamız. Bu diferansiel denklemlerden birincisi t , ikincisi ise s değişkenine bağlıdır (birbirine bağlı olmayan). İkinci denklem, değişkenlerden birinin tutulduğu kabul edilerek bir boyutlu spline fonksiyonlarının basit spline fonksiyonuna dönüştürülmesini sağlar. Bu tutulan değişkene parametre gibi bakarsak, bütün bir parametrelili spline ailesindeki spline fonksiyonlar ortak ağı paylaşırlar; ek olarak, iç ağ noktalarında devamlılık özellikleri olduğunu ve aynı tipteki bitiş koşullarını sağladığını kabul edelim.

Bir boyutlu spline fonksiyonların iki Hilbert uzayının direkt oluşum formunun tartışması altındaki spline fonksiyonlar için bir Hilbert uzay teorisi elde edilir.

4.2. Temel Tanımlama

$a \leq t \leq b; c \leq s \leq d$ ile sınırlı $\mathcal{R}$ dikdörtgeni aşağıdaki iki diferansiel operatörle verilmiş olsun;

$$L_t = a_n(t)D_t^n + a_{n-1}(t)D_t^{n-1} + \dots + a_0(t), \quad (4.2.1)$$

$$L_s = b_m(s)D_s^m + b_{m-1}(s)D_s^{m-1} + \dots + b_0(s).$$

Bu diferansiyel operatörlerde, $a_i(t)$ ($i = 0, 1, \dots, n$) katsayıları n . dereceden türeve sahip ve $a_n(t)$, $[a, b]$ aralığındadır; $b_j(s)$ ($j = 0, 1, \dots, m$) katsayıları m . dereceden türeve sahip ve $b_m(s)$, $[c, d]$ aralığındadır ve $D_t \equiv d/dt$, $D_s \equiv d/ds$ dir. Ayrıca L_t^* ve L_s^* da sırasıyla L_t ve L_s 'nin formal bileşkesidir. Son olarak, $\mathcal{R}$ 'yi $N \cdot M$ içine $\mathcal{R}_y$

$(i = 1, 2, \dots, N; j = 1, 2, \dots, M)$ alt dikdörtgenlere bölen π ağını belirleyen $\Delta_t: a = t_0 < t_1 < \dots < t_N = b$ ve $\Delta_s: c = s_0 < s_1 < \dots < s_M = d$ dir.

Tanım :

π 'ye göre ve L_t, L_s operatörleri ile ilişkilendirilmiş $S_\pi(t, s)$ 'nin (iki boyutlu genellemede) $\mathcal{R}$ üzerinde spline fonksiyon olabilmesi için; (a) L_t operatörü ile ilişkilendirilmiş Δ_t 'ye göre $[c, d]$ aralığındaki her s için $S_\pi(t, s)$ $[a, b]$ aralığı üzerinde farklı bir spline fonksiyon ve L_s operatörü ile ilişkilendirilmiş Δ_s 'ye göre $[a, b]$ aralığındaki her t için $S_\pi(t, s)$ $[c, d]$ aralığı üzerinde farklı bir spline fonksiyon olması yeterlidir.

Burada önemli olan, her $\mathcal{R}_{ij}$ dikdörtgeninde $L_t * L_t S_\pi = 0$ ve $L_s * L_s S_\pi = 0$ gereksinimlerinin önemini incelenmesidir.

$L_t * L_t S_\pi = 0$ oluşundan, $\mathcal{R}_{ij}$ içinde ve $u_i(t)$ ($i = 1, 2, \dots, 2n$) fonksiyonları, $L_t * L_t u = 0$ 'ın temel çözüm kümesi olmak üzere

$$S_\pi(t, s) = a_1(i, j; s)u_1(t) + a_2(i, j; s)u_2(t) + \dots + a_{2n}(i, j; s)u_{2n}(t)$$

elde ederiz. Bunun sonucu olarak $L_s * L_s S_\pi = 0$ da S_π 'yi yerine koyarsak

$$0 = L_s * L_s a_1(i, j; s)u_1(t) + L_s * L_s a_2(i, j; s)u_2(t) + \dots + L_s * L_s a_{2n}(i, j; s)u_{2n}(t)$$

olur.

Bundan sonra $L_s * L_s a_k(i, j; s) = 0$ ($k = 1, 2, \dots, 2n$) olur. Bunu, $v_l(s)$ ($l = 1, 2, \dots, 2m$) fonksiyonları, $L_s * L_s v = 0$ 'ın temel çözüm kümesi olmak üzere

$$a_k(i, j; s) = c_{k1}(i, j)v_1(s) + c_{k2}(i, j)v_2(s) + \dots + c_{k,2m}(i, j)v_{2m}(s)$$

takip eder. Bu oluşumlar $\mathcal{R}_{ij}$ 'deki (t, s) için

$$S_\pi(t, s) = \sum_{k=1}^{2n} \sum_{l=1}^{2m} c_{kl}(i, j)u_k(t)v_l(s) \quad (4.2.2)$$

yi ifade eder.

(4.2.2) ifadesinden, eğer $\partial^\alpha S_\pi(t, s) / \partial t^\alpha$, $t = t_i$ 'de sürekli ise, $\partial^{\beta+\alpha} S_\pi(t, s) / \partial s^\beta \partial t^\alpha$ ($\beta = 0, 1, \dots, 2m$) de $t = t_i$ 'de olası istisna (t_i, s_j) ($j = 0, 1, \dots, m$)

noktalarına rağmen sürekli. Benzer sonuç t ile s 'nin rolleri değiştirilerek elde edilebilir.

4.3. Temel Özdeşlik

Lineer diferansiel operatörü L_t ile ilişkilendirilmiş $P(u, v)$

$$\beta_j(L_t, v; t) = \sum_{i=0}^j (-1)^i \{a_{n-j-i}(t)v(t)\}^{(i)} \quad (4.3.1)$$

olmak üzere şu şekilde ifade edilebilir;

$$P[u, v] = \sum_{j=0}^{n-1} u^{(n-j-1)}(t) \beta_j(L_t, v; t). \quad (4.3.2)$$

Tartışma t ve s operatörlerine bağlı olduğundan ve L_t ve L_s operatörler de var olduğundan, şu değiştirilmiş notasyonlar kullanılmıştır;

$$P_{L_t}[u, v] = \sum_{j=0}^{n-1} D_t^{n-j-1} u(t, s) \cdot \beta_j(L_t, v, s; t), \quad (4.3.3)$$

$$\beta_j(L_t, v, s; t) = \sum_{i=0}^j (-D_t)^i \{a_{n-j-i}(t)v(t, s)\}. \quad (4.3.4)$$

$$P_{L_s}[u, v] = \sum_{j=0}^{m-1} D_s^{m-j-1} u(t, s) \cdot \beta_j(L_s, v, t; s), \quad (4.3.5)$$

$$\beta_j(L_s, v, t; s) = \sum_{i=0}^j (-D_s)^i \{b_{m-j-i}(t)v(t, s)\}. \quad (4.3.6)$$

$f(t, s)$, $\mathcal{R}: a \leq t \leq b; c \leq s \leq d$ dikdörtgeninde tanımlı ve t 'ye göre kısmi türevini hesaplarken n den fazla ve s 'ye göre kısmi türevini hesaplarken de m den fazla diferansiel alma gerektirmeyen ve $\mathcal{R}$ üzerinde sürekli bir fonksiyon olsun. $\Delta_t: a = t_0 < t_1 < \dots < t_n = b$ ve $\Delta_s: c = s_0 < s_1 < \dots < s_m = d$, $\mathcal{R}$ üzerinde π ağını tanımlasın; $S_\pi(t, s)$, L_t ve L_s operatörleri ile ilişkilendirilmiş, π 'ye göre $\mathcal{R}$ üzerinde bir spline fonksiyon olsun ve

$$I = \int_c^d \int_a^b L_t L_s \{f(t, s) - S_\pi(t, s)\} L_t L_s S_\pi(t, s) dt ds$$

gözönüne alınsın.

Her $\mathcal{R}_{ij}$ altıkdörtgeninde $L_t * L_t S_\pi(t, s) = 0$ ve $L_s * L_s S_\pi(t, s) = 0$ olduğundan

$$\begin{aligned} I &= \int_c^d \sum_{i=1}^N \sum_{j=0}^{n-1} D_t^{n-j-1} L_s \{f(t, s) - S_\pi(t, s)\} \cdot \beta_j(L_t, L_s L_t S_\pi, s; t) \Big|_{t_{i-1}}^{t_i} ds \\ &= \sum_{i=1}^N \sum_{k=1}^M \sum_{j=0}^{n-1} \sum_{l=0}^{m-1} D_s^{m-l-1} D_t^{n-j-1} \{f(t, s) - S_\pi(t, s)\} \\ &\quad \cdot \beta_l \left[L_s, \beta_j(L_t, L_s L_t S_\pi, s; t), t; s \right] \Big|_{t_{i-1}}^{t_i} \Big|_{s_{k-1}}^{s_k} \end{aligned} \quad (4.3.7)$$

dir. P yı

$$\begin{aligned} &\int_c^d \int_a^b \{L_t L_s \{f(t, s) - S(t, s)\}\}^2 dt ds \\ &= \int_c^d \int_a^b \{L_t L_s f(t, s)\}^2 dt ds \\ &\quad - 2 \int_c^d \int_a^b \{L_t L_s \{f(t, s) - S(t, s)\}\} L_t L_s S_\pi(t, s) dt ds \\ &\quad - \int_c^d \int_a^b \{L_t L_s S(t, s)\}^2 dt ds \end{aligned}$$

özdeşliğine katarsak, iki boyutta genelleştirilmiş spline fonksiyonlar için temel özdeşlik olan

$$\begin{aligned} \int_c^d \int_a^b \{L_t L_s f(t, s)\}^2 dt ds &= \int_c^d \int_a^b \{L_t L_s S_\pi(t, s)\}^2 dt ds \\ &\quad + \int_c^d \int_a^b \{L_t L_s f(t, s) - L_t L_s S(t, s)\}^2 dt ds \\ &\quad - 2 \sum_{i=1}^N \sum_{k=1}^M \sum_{j=0}^{n-1} \sum_{l=0}^{m-1} D_s^{m-l-1} D_t^{n-j-1} \{f(t, s) - S_\pi(t, s)\} \\ &\quad \cdot \beta_l \left[L_s, \beta_j(L_t, L_s L_t S_\pi, s; t), t; s \right] \Big|_{t_{i-1}}^{t_i} \Big|_{s_{k-1}}^{s_k} \end{aligned} \quad (4.3.8)$$

elde edilir. Ağ noktalarında veya şebeke çizgileri boyunca β_j 'de devamsızlıklar olabilir, bunlar basit sekme devamsızlıklarıdır. Bunların değerleri, P 'dan sağ ve sol limitlerle belirlenmelidir.

4.4. Spline Fonksiyon Tipleri

$\Delta: a = t_0 < t_1 < \dots < t_N = b$ 'ye bağlı olarak, $[a, b]$ aralığı üzerinde L lineer diferansiyel operatörü ile birleştirilmiş çok çeşitli spline fonksiyonlar vardır. Herhangi bir özel $S_\Delta(t)$ spline fonksiyonu, tanımlama değerlerini ifade ettiğimiz sonlu sayıda bağımsız parametreye bağlıdır; $S_\Delta(t)$ bu parametreler için, nümerik değerlerin özel atamalarının çözümüdür. Eğer burada k tane parametre varsa ve bunların her biri reel sayılar alanında daha fazla değişime izin veriyorsa, sonuç Δ ya bağlı spline fonksiyonların k -parametrelili F_Δ ailesidir. F_Δ ailesi, bu parametrelere nümerik değer atama yönünden farklı spline fonksiyonlar ve bundan dolayı, ağ noktalarında spline fonksiyonların benzer devamlılık özelliklerinden ibarettir. Bir bakıma F_Δ , spline fonksiyonunun tipini tanımlar. Böylece F_Δ ile k 'yı, doğrusal olarak bağımsız $e_i(t)$ ($i = 1, 2, \dots, k$) spline fonksiyonu ile birleştirebiliriz. F_Δ , $e_i(t)$ bazlı bir lineer uzaydır. Sonuç olarak eğer $S_\Delta(t)$, F_Δ 'da yer alıyorsa

$$S_\Delta(t) = \sum_{i=1}^k a_i e_i(t) \quad (4.4.1)$$

dir.

Tanım :

$S_\Delta(t)$, F_Δ 'da yer alsın. Eğer (4.4.1)'deki her a_i , Δ 'nın bazı ağ noktalarındaki bazı α ($\alpha = 0, 1, \dots, n-1$) için $S_\Delta^{(\alpha)}(t)$ türev değerleri yerine kullanılabilirse, F_Δ , apaçık tipteki türlü türlü spline fonksiyonların ailesidir. Bundan başka, eğer çok sayıdaki $S_\Delta^{(\alpha)}(t_i)$ 'den biri ($\alpha = 0, 1, \dots, n-1; i = 0, 1, \dots, N-1$) bu yolla türetilemezse $\beta_\alpha(L, LS_\Delta; t)$, $t = t_i$ ($i = 1, 2, \dots, N-1$) de sürekli ya da $\beta_\alpha(L, LS_\Delta; t)$, $t = t_i$ ($i = 0, N$) de ortadan yok olmuştur.

Tanım :

F_π , π 'ye göre $\mathcal{R}$ üzerinde k tabanlı $h_i(t, s)$ ($i = 1, 2, \dots, k$) spline fonksiyonlarının iki boyutlu spline ailesi olsun. F_π ailesi, eğer

$$S_\pi(t, s) = \sum_{i=1}^k a_i h_i(t, s)$$

her a_i , bazı $\alpha + \gamma$ ($\alpha = 0, 1, \dots, n-1; \gamma = 0, 1, \dots, m-1$) için π 'nin bazı ağ noktalarında $\partial^{\alpha+\gamma} S_\pi(t, s) / \partial t^\alpha \partial s^\gamma$ kısmi türevinin değeri ile aynı sayılabildiğini ifade ederse apaçık tipteki çeşitli spline fonksiyonlardan oluşur. Ayrıca çok sayıdaki $\partial^{\alpha+\gamma} S_\pi(t_i, s_j) / \partial t^\alpha \partial s^\gamma$ ($\alpha = 0, 1, \dots, n-1; \gamma = 0, 1, \dots, m-1; i = 0, 1, \dots, N; j = 0, 1, \dots, M$) den biri bu şekilde

türetilemezse, $\beta_\gamma[L_s, \beta_\alpha(L_t, L_t L_s S, s; t), t; s]$, t veya s den biri için (t_i, s_j) noktasında sürekli, ya da $\beta_\gamma[L_s, \beta_\alpha(L_t, L_t L_s S_\pi, s; t), t; s]$ ortadan kaldırılmıştır veya (t_i, s_j) deki sınır boyunca sürekli (bu sınır ağ noktaları, köşe noktaları değildir) ya da $\beta_\gamma[L_s, \beta_\alpha(L_t, L_t L_s S_\pi, s; t), t; s]$, (t_i, s_j) (köşe noktası) noktasında ortadan kalkmıştır.

$S_\pi(t, s) = f(t, s)$ olduğunu iddia eder. Şimdi güçlü bir kavrama ihtiyacımız vardır.

Tanım :

F_π , açık spline fonksiyonlardan oluşsun ve $S_\pi(f; t, s)$, F_π 'de yer alsın. Eğer $S_\pi(f; t, s)$ 'nin kısmi türevleri bir önceki tanımda belirtildiği gibi, π 'nin ağ noktalarında $f(t, s)$ fonksiyonunun kısmi türevlerine yakınsarsa, $S_\pi(f; t, s)$, $f(t, s)$ fonksiyonuna güçlü yaklaşımın spline fonksiyonudur denir.

BÖLÜM V

ÖRNEKLER

5.1. Kübik Spline Uygulaması

Kübik Spline kullanılarak, $\sin(x)$ fonksiyonunun x değerlerine karşılık gelen y değerleri yardımı ile x_i değerlerine karşılık gelen y_i değerleri ve gerçek değerler Tablo 1'de verilmiştir. Bulunan değerler açılı derecesi 2 artırılarak elde edilen değerlerdir.

Tablo 1.

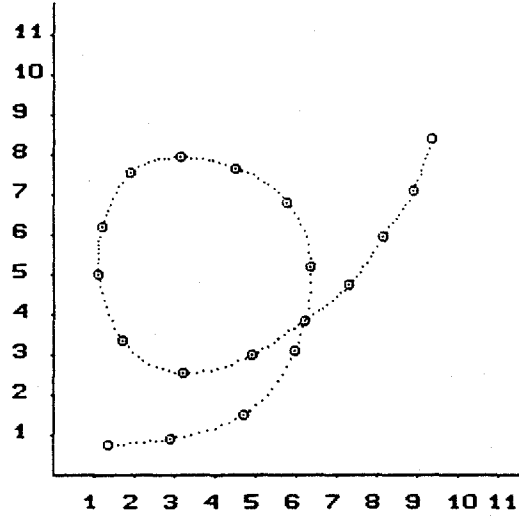
x	y	x_i	y_i	$\sin(x_i)$	Hata
10	0.173648	12	0.20766000	0.20791169	0.000251690
		14	0.24158760	0.241921895	0.000334295
		16	0.27534640	0.275637355	0.000290955
		18	0.30885200	0.309016994	0.000164994
20	0.342020	22	0.37476726	0.374606593	-0.00016066
		24	0.40701571	0.406736643	-0.00027906
		26	0.43868853	0.438371146	-0.00031738
		28	0.46970890	0.469471562	-0.00023733
30	0.5				
40	0.642787				

5.2. Çift Spline Uygulaması

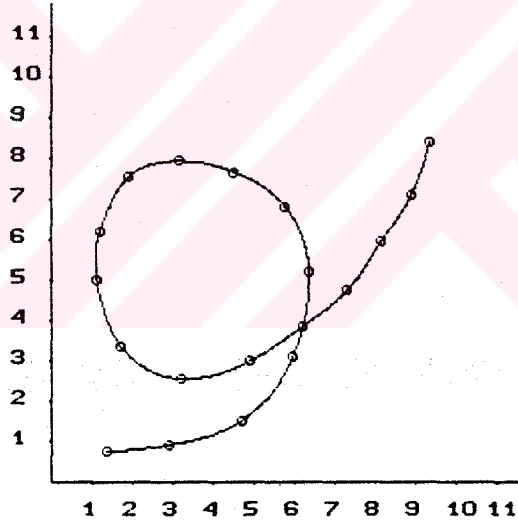
Tablo 2’de verilen x ve y değerlerine 0.1 ve 0.02 artımlarla Çift Spline uygulanarak elde edilen noktaların grafikleri sırasıyla Şelil 3 ve Şekil 4’de verilmiştir.

Tablo 2.

t	x	y
1	1.3506	0.7591
2	2.9091	0.9241
3	4.7013	1.5182
4	5.9740	3.1023
5	6.3896	5.2145
6	5.7922	6.8317
7	4.5455	7.6898
8	3.1688	7.9538
9	1.9221	7.5578
10	1.2208	6.2376
11	1.1429	5.0495
12	1.7403	3.3663
13	3.2208	2.5743
14	4.9091	3.0033
15	6.2078	3.8614
16	7.3247	4.7525
17	8.1818	5.9736
18	8.9351	7.1287
19	9.3766	8.4488
20	9.6364	9.6700



Şekil 3. Artım 0.1 alınarak yapılan hesaplama ile bulunan grafik.



Şekil 4. Artım 0.02 alınarak yapılan hesaplama ile bulunan grafik.

5.3. İki-Kat Kübik Spline Uygulamaları

a) İki-Kat Kübik Spline uygulayarak, $Z(x, y) = xy$ fonksiyonunun, Tablo 3'de verilen x , y , $Z(x, y)$ ve Z'_x , Z'_y , Z''_{xy} değerleri yardımıyla x_i , y_i noktalarında bulunan z_i değerleri ve fonksiyonun bu noktadaki gerçek değerleri Tablo 4'de verilmiştir.

Tablo 3.

x	y	$Z(x,y)$	Z'_x	Z'_y	Z''_{xy}
1	1	1	1	1	1
1	2	2	2	1	1
1	3	3	3	1	1
2	1	2	1	2	1
2	2	4	2	2	1
2	3	6	3	2	1
3	1	3	1	3	1
3	2	6	2	3	1
3	3	9	3	3	1

Tablo 4.

x_i	y_i	z_i	$Z(x_i, y_i)$	Hata
1.1	1.5	1.65	1.65	0
1.2	1.9	2.28	2.28	0
2.2	2.9	6.38	6.38	0
2.5	2.7	6.75	6.75	0
2.897	2.1876	6.3374772	6.3374772	0
2.9	2.9	8.41	8.41	0

b) Bu kez fonksiyonumuz $Z(x,y) = x^3 + y^3$ olsun. Tablo 5'de verilen x , y , $Z(x,y)$ ve Z'_x , Z'_y , Z''_{xy} değerleri yardımıyla x_i , y_i noktalarında bulunan z_i değerleri ve fonksiyonun bu noktalardaki gerçek değerleri Tablo 6'da verilmiştir.

Tablo 5.

x	y	$Z(x, y)$	Z'_x	Z'_y	Z''_{xy}
1	1	2	3	3	0
1	2	9	3	12	0
1	3	28	3	27	0
2	1	9	12	3	0
2	2	16	12	12	0
2	3	35	12	27	0
3	1	28	27	3	0
3	2	35	27	12	0
3	3	54	27	27	0

Tablo 6.

xi	yi	zi	$Z(xi, yi)$	Hata
1.1	1.5	4.706	4.706	0
1.2	1.9	8.587	8.587	0
2.2	2.9	35.037	35.037	0
2.5	2.7	35.308	35.308	0
2.897	2.1876	34.78235318	34.78235318	0
2.9	2.9	48.778	48.778	0

BÖLÜM VI

PROGRAM

Bu bölümde, Spline uygulamalarını içeren, Pascal programlama dilinde yazılmış bir program verilmiştir. Bu program içerisinde, oluşan lineer denklem sistemini çözmek için *Gauss Eliminasyon* metodu kullanılmıştır.

Program içerisinde Çift Spline Uygulama konusuna da bir örnek verilmiştir.

```
Program spline_and_spline2d_menu;
Uses Spline,Spline2d,Crt,Graph;

var
  xi,xi0,xin,artim,sp : real;
  yi                  : real;
  sec,ch              : char;
  n,i,j,sat,nl       : integer;
  xi2,yi2             : array[0..1000] of real;
  xil,yil,ti          : array[0..30] of real;
  title1,title2,yazi  : string[70];

  grDriver : Integer;
  grMode    : Integer;
  ErrCode   : Integer;

begin
  repeat
    textcolor(7);
    textbackground(0);
```

```
clrscr;
textcolor(15);
gotoxy(23,2);writeln('SPLINE VE İKİ BOYUTLU SPLINE HESAPLAMA');
textcolor(14);
gotoxy(23,3);writeln('-----');
gotoxy(7,5);writeln('1 - Tek boyutlu spline ile bir noktanın
ordinatını hesaplama');
gotoxy(7,6);writeln('2 - Tek boyutlu spline ile bir aralığın ordinat
kumesini hesaplama');
gotoxy(7,7);writeln('3 - Çift spline uygulayarak yaklaşım hesaplama');
gotoxy(7,8);writeln('4 - Örnek çift spline uygulaması');
gotoxy(7,9);writeln('5 - İki boyutlu spline ile bir noktanın z
değerini hesaplama');
gotoxy(7,10);writeln('6 - Çıkış [ESC]');
for i:=1 to 30 do
begin
    title1:='Copyright (c) Hasan YÜCEL, 1996';
    gotoxy(9,24);write(' '+copy(title1,length(title1)-i,i+1));
    title2:='Yıldız Teknik Üniversitesi';
    gotoxy(80-i,24);write(copy(title2+' ',1,i));delay(15);
end;

gotoxy(10,12);write('Seçiminiz :');

sec:=readkey;
if (sec = #27) or (sec = '6') then
begin
    textcolor(7);
    clrscr;
    exit;
end;

clrscr;
if (sec > '0') and (sec < '6') then
begin
```

```
repeat
    write('n =');readln(n);
    if n < 3 then
        begin
            writeln('girilecek nokta sayısı en az 3 olmalıdır. Menü
için ESC ...');
            ch:=readkey;
            if ch = #27 then
                begin
                    sec := '9';
                    n := 3;
                end;
            clrscr;
        end;
until n > 2;
n:=n-1;
end;

if sec = '1' then
begin
    for j:=0 to n do
        begin
            write('x(',j+1,')=');readln(x[j]);
        end;
    for j:=0 to n do
        begin
            write('y(',x[j]:3:5,')=');readln(y[j]);
        end;
end;

writeln;
repeat
    write('xi =');readln(xi);
    if xi<>0 then
        begin
            sp := xSpline(x,y,xi,n,false);
```

```
writeln;
writeln('menüye dönmek için xi`yi 0 giriniz');
writeln('Spline(',xi:3:5,')= ',Sp:5:8);
end;
until xi=0;
end;

if sec = '2' then
begin
    n1:=n;
    for j:=0 to n do
    begin
        write('x(',j+1,')=');readln(x[j]);
        xil[j]:=x[j];
    end;
    for j:=0 to n do
    begin
        write('y(',x[j]:3:5,')=');readln(y[j]);
        yil[j]:=y[j];
    end;

    writeln;
    repeat
        write('xi0 ( >= x[1] ) =');readln(xi0);
        write('xin ( < x[n] ) =');readln(xin);
        write('artım ( <> 0 ) =');readln(artim);
    until (xi0 >= x[0]) and (xin < x[n]) and (artim <> 0);

    xi:=xi0+artim;
    sat :=1;j:=0;
    repeat
        sp := xSpline(x,y,xi,n,false);
        xi2[j]:=xi;
        yi2[j]:=sp;
        j:=j+1;
```

```
write(xi:5:5);writeln('      Spline(',xi:5:5,')=',Sp:5:8);
xi:=xi+artim;
sat := sat + 1;
if sat = 20 then
begin
    writeln;
    write('Devam etmek için bir tuşa basınız ...');
    ch:=readkey;
    writeln;
    writeln;
    sat := 1;
end;
until xi > xin;
n:=j-1;
writeln;
writeln('Grafik için bir tuşa basınız ...');
ch:=readkey;
grDriver := Detect;
InitGraph(grDriver,grMode,'');
ErrCode := GraphResult;
if ErrCode = grOk then
begin
    { Do graphics }
    setcolor(14);
    setbkcolor(0);
    settextstyle(1,horizdir,1);
    setusercharsize(1,2,1,2);
    outtextxy(100,10,'TEK-BOYUTLU   SPLINE   UYGULANARAK   BULUNAN
NOKTALARIN GRAFiGi');
    setcolor(15);
    line(100,400,600,400);
    line(100,400,100,50);
    settextstyle(5,horizdir,1);
    for i:=1 to 16 do
    begin
```

```
        str(i,yazi);
        putpixel(100+20*i,401,14);
        outtextxy(100+20*i-5,410,yazi);
        putpixel(99,400-20*i,14);
        outtextxy(80,400-20*i-5,yazi);
    end;
    for j:=1 to n do
    begin
        putpixel(100+trunc(xi2[j]*20),400-trunc(yi2[j]*20),14);
        if j <= n1 then circle(100+trunc(xi1[j-1]*20),400-
trunc(yi1[j-1]*20),2);
        end;
        readln;
        CloseGraph;
    end
end;

if sec = '3' then
begin
    n1:=n;
    for j:=0 to n do
    begin
        write('x(',j+1,')=');readln(xi1[j]);
    end;
    for j:=0 to n do
    begin
        write('y(',xi1[j]:3:5,')=');readln(yi1[j]);
    end;

    writeln;
    repeat
        write('artim [ornek 0.1] = ');readln(artim);
    until artim <> 0;

    for j:=0 to n do
```

```
begin
    y[j]:=xi1[j];
    x[j]:=j+1;
end;
xi:=1+artim;
j:=1;
repeat
    xi2[j] := xSpline(x,y,xi,n,false);
    xi:=xi+artim;
    j:=j+1;
until xi > n;

for j:=0 to n do
begin
    y[j]:=yi1[j];
    x[j]:=j+1;
end;
xi:=1+artim;
j:=1;
repeat
    yi2[j] := xSpline(x,y,xi,n,false);
    xi:=xi+artim;
    j:=j+1;
until xi > n;

n:=j-1;
sat:=1;
for j:=1 to n do
begin
    write(xi2[j]:5:5);writeln(' ',yi2[j]:5:8);
    sat := sat + 1;
    if sat = 20 then
    begin
        writeln;
        write('Devam etmek için bir tusa basınız ...');
```

```
        ch:=readkey;
        writeln;
        writeln;
        sat := 1;
    end;
end;
writeln;
writeln('Grafik için bir tuşa basınız ...');
ch:=readkey;
grDriver := Detect;
InitGraph(grDriver,grMode,'');
ErrCode := GraphResult;
if ErrCode = grOk then
    begin
        { Do graphics }
        setcolor(14);
        setbkcolor(0);
        setttextstyle(1,horizdir,1);
        setusercharsize(1,2,1,2);
        outtextxy(100,10,'ÇİFT SPLINE UYGULANARAK BULUNAN NOKTALARIN
GRAFiGi');
        setcolor(15);
        line(100,400,600,400);
        line(100,400,100,50);
        setttextstyle(5,horizdir,1);
        for i:=1 to 16 do
            begin
                str(i,yazi);
                putpixel(100+20*i,401,14);
                outtextxy(100+20*i-5,410,yazi);
                putpixel(99,400-20*i,14);
                outtextxy(80,400-20*i-5,yazi);
            end;
        for j:=1 to n do
            begin
```

```
        putpixel(100+trunc(xi2[j]*20),400-trunc(yi2[j]*20),14);
        if j<=n1 then circle(100+trunc(xil[j-1]*20),400-
trunc(yil[j-1]*20),2);
        end;
        readln;
        CloseGraph;
    end
end;
```

```
if sec = '4' then
begin
```

```
    n1:=n;
```

```
    yil[0] := 0.7591;
```

```
    yil[1] := 0.9241;
```

```
    yil[2] := 1.5182;
```

```
    yil[3] := 3.1023;
```

```
    yil[4] := 5.2145;
```

```
    yil[5] := 6.8317;
```

```
    yil[6] := 7.6898;
```

```
    yil[7] := 7.9538;
```

```
    yil[8] := 7.5578;
```

```
    yil[9] := 6.2376;
```

```
    yil[10] := 5.0495;
```

```
    yil[11] := 3.3663;
```

```
    yil[12] := 2.5743;
```

```
    yil[13] := 3.0033;
```

```
    yil[14] := 3.8614;
```

```
    yil[15] := 4.7525;
```

```
    yil[16] := 5.9736;
```

```
    yil[17] := 7.1287;
```

```
    yil[18] := 8.4488;
```

```
    yil[19] := 9.6700;
```

```
    xil[0] := 1.3506;
```

```
xil[1] := 2.9091;
xil[2] := 4.7013;
xil[3] := 5.9740;
xil[4] := 6.3896;
xil[5] := 5.7922;
xil[6] := 4.5455;
xil[7] := 3.1688;
xil[8] := 1.9221;
xil[9] := 1.2208;
xil[10] := 1.1429;
xil[11] := 1.7403;
xil[12] := 3.2208;
xil[13] := 4.9091;
xil[14] := 6.2078;
xil[15] := 7.3247;
xil[16] := 8.1818;
xil[17] := 8.9351;
xil[18] := 9.3766;
xil[19] := 9.6364;

for j:=0 to n do
begin
    writeln('y(',xil[j]:3:5,',') =',yil[j]:8:5);
end;

writeln;
repeat
    write('artim [ornek 0.1] = ');readln(artim);
until artim <> 0;

for j:=0 to n do
begin
    y[j]:=xil[j];
    x[j]:=j+1;
```

```
end;
xi:=1+artim;
j:=1;
repeat
    xi2[j] := xSpline(x,y,xi,n,false);
    xi:=xi+artim;
    j:=j+1;
until xi > n;

for j:=0 to n do
begin
    y[j]:=yi1[j];
    x[j]:=j+1;
end;
xi:=1+artim;
j:=1;
repeat
    yi2[j] := xSpline(x,y,xi,n,false);
    xi:=xi+artim;
    j:=j+1;
until xi > n;

n:=j-1;
sat:=1;
for j:=1 to n do
begin
    write(xi2[j]:5:5);writeln(' ',yi2[j]:5:8);
    sat := sat + 1;
    if sat = 20 then
    begin
        writeln;
        write('Devam etmek için bir tusa basınız ...');
        ch:=readkey;
        writeln;
        writeln;
```

```
        sat := 1;
    end;
end;
writeln;
writeln('Grafik için bir tuşa basınız ...');
ch:=readkey;
grDriver := Detect;
InitGraph(grDriver,grMode,'');
ErrCode := GraphResult;
if ErrCode = grOk then
    begin
        { Do graphics }
        setcolor(14);
        setbkcolor(0);
        settextstyle(1,horizdir,1);
        setusercharsize(1,2,1,2);
        outtextxy(100,10,'ÇİFT SPLINE UYGULANARAK BULUNAN NOKTALARIN
GRAFiGi');
        setcolor(15);
        line(100,400,600,400);
        line(100,400,100,50);
        settextstyle(5,horizdir,1);
        for i:=1 to 16 do
            begin
                str(i,yazi);
                putpixel(100+20*i,401,14);
                outtextxy(100+20*i-5,410,yazi);
                putpixel(99,400-20*i,14);
                outtextxy(80,400-20*i-5,yazi);
            end;
        for j:=1 to n do
            begin
                putpixel(100+trunc(xi2[j]*20),400-trunc(yi2[j]*20),14);
                if j<=n1 then circle(100+trunc(xil[j-1]*20),400-
trunc(yil[j-1]*20),2);
```

```
        end;  
        readln;  
        CloseGraph;  
    end  
end;
```

```
if sec = '5' then
```

```
begin
```

```
    n:=n+1;
```

```
    for j:=1 to n do
```

```
    begin
```

```
        write('x(',j,')=');readln(t[j]);
```

```
    end;
```

```
    for j:=1 to n do
```

```
    begin
```

```
        write('y(',j,')=');readln(s[j]);
```

```
    end;
```

```
    for i:=1 to n do
```

```
    for j:=1 to n do
```

```
    begin
```

```
        write('z(',t[i]:3:5,',',s[j]:3:5,')=');readln(z[i,j]);
```

```
    end;
```

```
    for i:=1 to n do
```

```
    for j:=1 to n do
```

```
    begin
```

```
        write('dZx(',t[i]:3:5,',',s[j]:3:5,')=');readln(dzt[i,j]);
```

```
    end;
```

```
    for i:=1 to n do
```

```
    for j:=1 to n do
```

```
    begin
```

```
        write('dZy(',t[i]:3:5,',',s[j]:3:5,')=');readln(dzs[i,j]);
```

```
    end;
```

```
for i:=1 to n do
for j:=1 to n do
begin
    write('dZxy(',t[i]:3:5,',',s[j]:3:5,')=');readln(dzts[i,j]);
end;

writeln;
repeat
    write('xi =');readln(xi);
    if xi<>0 then
    begin
        write('yi =');readln(yi);
        sp := xSpline2d(t,s,z,xi,yi,n,false);
        writeln;
        writeln('menüye dönmek için xi`yi 0 giriniz');
        writeln('Spline2d(',xi:3:5,',',yi:3:5,')= ',Sp:5:8);
    end;
until xi=0;
end;

writeln;
write('Menüye dönmek için <enter> tuşuna basınız');
readln;
until false;
end.
```

***** Spline Unit *****

```
unit spline;
```

```
interface
```

```
type
```

```
    dizi = array[0..40] of real;
```

const

n_ = 40;

m_ = 40;

var

x,y : dizi;

n : integer;

goster : boolean;

procedure init;

procedure farkhesapla(var x,y : dizi);

procedure matolustur;

procedure siscoz;

function xspline(var x,y : dizi ; xi : real; n_size : integer; visible
: boolean):real;

implementation

var

h,d,mat,bb : dizi;

a : array[1..n_+3,1..m_+3] of real;

C : array[0..n_,0..n_] of real;

procedure init;

var

j,i : integer;

begin

for j:=0 to n_ do

begin

h[j]:=0;d[j]:=0;mat[j]:=0;bb[j]:=0;

end;

for i:=1 to n_+3 do

for j:=1 to m_+3 do

a[i,j]:=0;

```
    for i:=1 to n_ do
    for j:=1 to n_ do
    C[i,j]:=0;
end;
```

```
{***** farkhesapla *****}
```

```
procedure farkhesapla(var x,y : dizi);
```

```
{h(j) ve d(j) farklar dizisi oluşturuluyor (j=0,1,2,...,n).}
```

```
    h(j) = [x(1) - x(0), x(2) - x(1), ... , x(n) - x(n-1)]
```

```
    d(j) = [y(1) - y(0), y(2) - y(1), ... , y(n) - y(n-1)]}
```

```
var
```

```
    j : integer;
```

```
begin
```

```
    for j:=0 to n-1 do
```

```
    begin
```

```
        h[j]:=x[j+1]-x[j];
```

```
        d[j]:=y[j+1]-y[j];
```

```
    end;
```

```
end;
```

```
{***** matolustur *****}
```

```
procedure matolustur;
```

{M momentlerinin katsayılar matrisi ve sistemin sağ yanını oluşturacak olan

B matrisi oluşturuluyor. Burada sistem $A \cdot M = B$ şeklindedir ve bilinmeyen M dir.

A matrisinin boyutu $(n-1) \times (n+1)$. Bu A matrisi bir üçlü band matristir.}

```
var
```

```
    j,k : integer;
```

```
begin
  for j:=1 to n-1 do
    begin
      a[j,0+j]:=h[j-1];
      a[j,1+j]:=2 * (h[j]+h[j-1]);
      a[j,2+j]:=h[j];
      bb[j]:=6 * (-<math>d[j-1]/h[j-1]</math>)+<math>d[j]/h[j]</math>))
    end;

  for j:=1 to n-1 do
    begin
      for k:=1 to n+1 do
        begin
          if goster then write(a[j,k]:4:0,'M',k);
        end;
        if goster then writeln(' = ',bb[j]:4:5);
      end;
      if goster then writeln;

      for j:=1 to n-1 do
        for k:=1 to n do
          begin
            C[j,k]:=a[j,k+1];
          end;
        end;
      end;
    end;

    {***** siscoz *****}
    procedure siscoz;

    {Oluşturulan A ve B matrislerine bağlı olan A*M=B sistemi çözülüyor.
    Çözüm
    matrisi MAT matrisidir.}

  var
    i,j,k,n1 : integer;
```

```

r      : real;
begin
n1:=n-1;
  for k:=1 to n1 do
    begin
      while (C[k,k]=0) do
        begin
          for j:=2 to n1 do
            begin
              r:=C[k,j];C[k,j]:=C[k+1,j];C[k+1,j]:=r;
              r:=bb[k];bb[k]:=bb[k+1];bb[k+1]:=r;
            end;
          end;
          for i:=k+1 to n1 do
            begin
              for j:=k+1 to n1 do
                begin
                  C[i,j]:=C[i,j]-((C[i,k]*C[k,j])/C[k,k]);
                end;
              bb[i]:=bb[i]-C[i,k]*bb[k]/C[k,k];
            end;
          end;
        end;
      mat[n1]:=bb[n1]/C[n1,n1];
      for i:=2 to n1 do
        begin
          mat[n1-i+1]:=bb[n1-i+1];
          for k:=n1-i+2 to n1 do
            begin
              mat[n1-i+1]:=mat[n1-i+1]-C[n1-i+1,k]*mat[k];
            end;
          mat[n1-i+1]:=mat[n1-i+1]/C[n1-i+1,n1-i+1];
        end;
      for i:=1 to n1 do
        begin
          if goster then writeln('mat[' ,i ,']= ',mat[i]:5:10);

```

```
end;  
if goster then readln;  
end;
```

```
{***** Spline *****}
```

```
function xSpline(var x,y : dizi; xi:real; n_size : integer; visible :  
boolean):real;
```

{Verilen X,Y matrisleri yardımıyla xi noktasındaki yi değerini bulmaya
yarayan

fonksiyon. Spline hesaplamasındaki son aşama.}

```
var
```

```
  i,m      : integer;  
  s1,s2,s3,s4 : real;
```

```
begin
```

```
  n := n_size;
```

```
  goster := visible;
```

```
  init;           {matris ve diziler sıfırlanıyor}
```

```
  farkhesapla(x,y); {h(j) ve d(j) fark dizileri hesaplanıyor}
```

```
  matolustur;      {katsayılar matrisi oluşturuluyor}
```

```
  siscoz;          {oluşan lineer denklem sistemi çözülüyor}
```

```
  if (xi <= x[0]) or (x[n-1] < xi) then
```

```
  begin
```

```
    writeln('Verilen nokta aralık dışında.');
```

```
    writeln('Bu noktadaki değerin hesaplanabilmesi için daha  
fazla noktaya ihtiyaç var.');
```

```
    xSpline:=0;
```

```
  end
```

```
  else
```

```
  begin
```

```
    for m:=0 to n-1 do
```

```
begin
  if (x[m] < xi) and (xi <= x[m+1]) then
    begin
      i:=m+1;
      m:=n-1;
    end;
  end;
  s1:=(x[i]-xi)*(x[i]-xi)*(x[i]-xi)*mat[i-1]/(6*h[i]);
  s2:=(xi-x[i-1])*(xi-x[i-1])*(xi-x[i-1])*mat[i]/(6*h[i]);
  s3:=(y[i-1]-(h[i]*h[i]*mat[i-1]/6))*(x[i]-xi)/h[i];
  s4:=(y[i]-(h[i]*h[i]*mat[i]/6))*(xi-x[i-1])/h[i];
  xSpline:=s1+s2+s3+s4;
end;
end;

begin
end.

***** Spline2d Unit *****unit spline2d;

interface

type
  dizi   = array[0..30] of real;
  matris = array[0..30,0..30] of real;

const
  n_ = 30;
  m_ = 30;

var
  t,s                : dizi;
  z,dzt,dzs,dzts    : matris;
  n,i,j              : integer;
  a11, a12, a13, a14,
```

```
a21, a22, a23, a24,  
a31, a32, a33, a34,  
a41, a42, a43, a44      : real;  
goster                   : boolean;  
si,ti                    : real;  
s1,s2,s3,s4              : real;
```

```
procedure init;  
procedure katsayilar(i,j : integer);  
procedure siscoz;  
function xspline2d(var t,s : dizi; z : matris; xi, yi : real; n_size :  
integer; visible : boolean):real;
```

```
implementation
```

```
var  
    mat,bb : dizi;  
    C      : array[1..4,1..4] of real;
```

```
procedure init;
```

```
var
```

```
    j,i : integer;
```

```
begin
```

```
    for j:=0 to n_ do
```

```
    begin
```

```
        mat[j]:=0;bb[j]:=0;
```

```
    end;
```

```
    for i:=1 to 4 do
```

```
    for j:=1 to 4 do
```

```
        C[i,j]:=0;
```

```
end;
```

```
{***** siscoz *****}
```

```
procedure siscoz;
```

{Oluşturulan C ve bb matrislerine bağlı olan $C \cdot A = bb$ sistemi çözülüyor.

Çözüm

matrisi MAT matrisidir.}

var

i,j,k,n1 : integer;

r : real;

begin

n1:=4;

for k:=1 to n1 do

begin

while (C[k,k]=0) do

begin

for j:=2 to n1 do

begin

r:=C[k,j];C[k,j]:=C[k+1,j];C[k+1,j]:=r;

r:=bb[k];bb[k]:=bb[k+1];bb[k+1]:=r;

end;

end;

for i:=k+1 to n1 do

begin

for j:=k+1 to n1 do

begin

$C[i,j] := C[i,j] - ((C[i,k] * C[k,j]) / C[k,k]);$

end;

$bb[i] := bb[i] - C[i,k] * bb[k] / C[k,k];$

end;

end;

mat[n1]:=bb[n1]/C[n1,n1];

for i:=1 to n1 do

begin

mat[n1-i+1]:=bb[n1-i+1];

for k:=n1-i+2 to n1 do

begin

```
        mat[n1-i+1]:=mat[n1-i+1]-C[n1-i+1,k]*mat[k];
    end;
    mat[n1-i+1]:=mat[n1-i+1]/C[n1-i+1,n1-i+1];
end;

a11:=mat[1];
a12:=mat[2];
a21:=mat[3];
a22:=mat[4];

for i:=1 to n1 do
begin
    if goster then writeln('mat[' ,i, ']=' ,mat[i]:5:10);
end;
if goster then readln;
end;

function kup(deger : real):real;
begin
    kup := deger * deger * deger;
end;

function kare(deger : real):real;
begin
    kare := deger * deger;
end;

{function dZs(t,s : real):real;
begin
    dZs := t;
end;
```

```
function dZt(t,s : real):real;
begin
    dZt := s;
end;
```

```
function dZst(x,y : real):real;
begin
    dZst := 1;
end;}
```

```
{***** katsayilar *****}
```

```
procedure katsayilar(i,j : integer);
```

```
begin
```

```
    a44:=z[i-1,j-1];
```

```
    a34:=dZs[i-1,j-1];
```

```
    a43:=dZt[i-1,j-1];
```

```
    a33:=dZts[i-1,j-1];
```

```
    a14:=dZs[i-1,j] - 2*z[i-1,j] + a34 + 2*a44;
```

```
    a24:=z[i-1,j] - a34 - a44 - a14;
```

```
    a13:=dZts[i-1,j] - 2*dZt[i-1,j] + a33 + 2*a43;
```

```
    a23:=dZt[i-1,j] - a33 - a43 - a13;
```

```
    a41:=dZt[i,j-1] - 2*z[i,j-1] + a43 + 2*a44;
```

```
    a42:=z[i,j-1] - a43 - a44 - a41;
```

```
    a31:=dZts[i,j-1] - 2*dZs[i,j-1] + a33 + 2*a34;
```

```
    a32:=dZs[i,j-1] - a33 - a34 - a31;
```

```
C[1,1]:=1; C[1,2]:=1; C[1,3]:=1; C[1,4]:=1;
C[2,1]:=3; C[2,2]:=3; C[2,3]:=2; C[2,4]:=2;
C[3,1]:=3; C[3,2]:=2; C[3,3]:=3; C[3,4]:=2;
C[4,1]:=9; C[4,2]:=6; C[4,3]:=6; C[4,4]:=4;
```

```
bb[1]:=z[i,j] - a13 - a14 - a23 - a24 - a31 - a32 - a33 - a34 -
a41 - a42 - a43 - a44;
```

```
bb[2]:=dZs[i,j] - 3*a13 - 3*a14 - 2*a23 - 2*a24 - a31 - a32 - a33
- a34;
```

```
bb[3]:=dZt[i,j] - a13 - a23 - 3*a31 - 2*a32 - a33 - 3*a41 - 2*a42
- a43;
```

```
bb[4]:=dZts[i,j] - 3*a13 - 2*a23 - 3*a31 - 2*a32 - a33;
```

```
if goster then
```

```
begin
```

```
  writeln(bb[1]:3:5);
```

```
  writeln(bb[2]:3:5);
```

```
  writeln(bb[3]:3:5);
```

```
  writeln(bb[4]:3:5);
```

```
  readln;
```

```
end;
```

```
end;
```

```
{***** S şapka'nın hesaplanması *****}
```

```
function ssapka(ti,si : real):real;
```

```
var
```

```
  s1,s2,s3,s4 : real;
```

```
  k1,k2,k3 : real;
```

```
begin
```

```
  k1:=kup(si - s[j-1]);
```

```
  k2:=kup(ti - t[i-1]);
```

```
  k3:=kare(ti - t[i-1]);
```

```
  s1 := k1 * (a11 * k2 + a12 * k3 + a13 * (ti - t[i-1]) + a14);
```

```
s2 := kare(si - s[j-1]) * (a21 * k2 + a22 * k3 + a23 * (ti - t[i-1]) + a24);
s3 := (si - s[j-1]) * (a31 * k2 + a32 * k3 + a33 * (ti - t[i-1]) + a34);
s4 := (a41 * k2 + a42 * k3 + a43 * (ti - t[i-1]) + a44);
ssapka := s1 + s2 + s3 + s4;
end;
```

```
{***** İki Boyutlu Spline *****}
```

```
function xSpline2d(var t,s : dizi; z : matris; xi, yi : real; n_size : integer; visible : boolean):real;
```

{Verilen X,Y,Z,dZx,dZy,dZxy matrisleri yardımıyla xi, yi noktasındaki zi değerini bulmaya yarayan fonksiyon. İki Boyutlu Spline hesaplamasındaki son aşama.}

```
var
```

```
    k,l      : integer;
```

```
begin
```

```
    n := n_size;
```

```
    ti := xi;
```

```
    si := yi;
```

```
    goster := visible;
```

```
    init;           {matris ve diziler sıfırlanıyor}
```

```
    if ((xi <= t[1]) or (t[n] <= xi)) or ((yi <= s[1]) or (s[n] <= yi)) then
```

```
        begin
```

```
            writeln('Verilen nokta aralık dışında.');
```

```
            writeln('Bu noktadaki değerin hesaplanabilmesi için daha fazla noktaya ihtiyaç var.');
```

```
            xSpline2d:=0;
```

```
        end
```

```
else
begin
  for k:=1 to n-1 do
  begin
    if (t[k] <= xi) and (xi < t[k+1]) then
    begin
      i:=k+1;
      k:=n-1;
    end;
  end;

  for l:=1 to n-1 do
  begin
    if (s[l] <= yi) and (yi < s[l+1]) then
    begin
      j:=l+1;
      l:=n-1;
    end;
  end;

  katsayilar(i,j);          {katsayilar oluřturuluyor}
  siscoz;                   {oluřan lineer denklem sistemi cözölüyor}
  xspline2d := ssapka(xi,yi);
end;

end;

begin
end.
```

SONUÇ

Tezimde, günümüzde en çok kullanılan yaklaşım yöntemlerinden biri olan Spline ve İki-Boyutlu Spline konusunu vermeye çalıştım.

Spline ve İki-Boyutlu Spline yaklaşımları yapılan uygulamalardan da anlaşılacağı gibi hata mertebesi çok düşük yaklaşım yöntemleridir.

Özellikle İki-Boyutlu Spline uygulamalara açık bir konudur. Bu açıdan, bu tezin, İki-Boyutlu Spline uygulamalarına bir temel oluşturacağı düşüncesindeyim.

Tezimle birlikte hazırlamış olduğum Pascal programlama dili ile yazılmış olan Spline hesaplama programının, konu ile ilgilenen araştırmacılara faydalı olacağını ümit ederim. Bu program sayesinde, örneğin; 20×20 boyutunda bir lineer denklem sistemi kolaylıkla çözülebilmekte ve büyük bir zaman kazanımı meydana gelmektedir.

Bu tezin ileride yapılacak olan İki-Boyutlu Spline çalışmalarına bir kaynak olmasını ümit ediyorum.



KAYNAKLAR

1. **J.H. Ahlberg, E.N. Nilson, J.L. Walsh.** “*The Theory of Spline and Their Applications*”. Academic Press Inc. 1967.
2. **Matlab 4.0 with Simulink 1.2c.** Copyright (c) by the MathWorks, Inc.
3. **B.D. Bojanov, H.A. Hakopian, A.A. Sahakian.** “*Spline Functions and Multivariate Interpolations*”. 1993.
4. **Behiç ÇAĞAL.** “*Sayısal Analiz*”. Seç Yayın Dağıtım, 1989, s. 281-288.



ÖZGEÇMİŞ

Adı Soyadı : Hasan YÜCEL

Doğum Tarihi : 7 Temmuz 1972

Doğum Yeri : Lefkoşa/KIBRIS

İlk Öğrenimi : 1984 yılında Hisarköy İlkokulu'ndan mezun oldu.

Orta Öğrenimi : 1990 yılında 20 Temmuz Lisesi'nden mezun oldu.

Yüksek Öğrenimi : 1990 yılında Yıldız Üniversitesi Fen-Edebiyat Fakültesi Matematik Bölümünde başladığı eğitimini 1994 yılında tamamladı.

