ONLİNE DERS

2020-2021 BAHAR DÖNEMİ

YMH214
SAYISAL ANALIZ
LAB. DERSİ

10.DERS

Arş. Gör. Alev KAYA

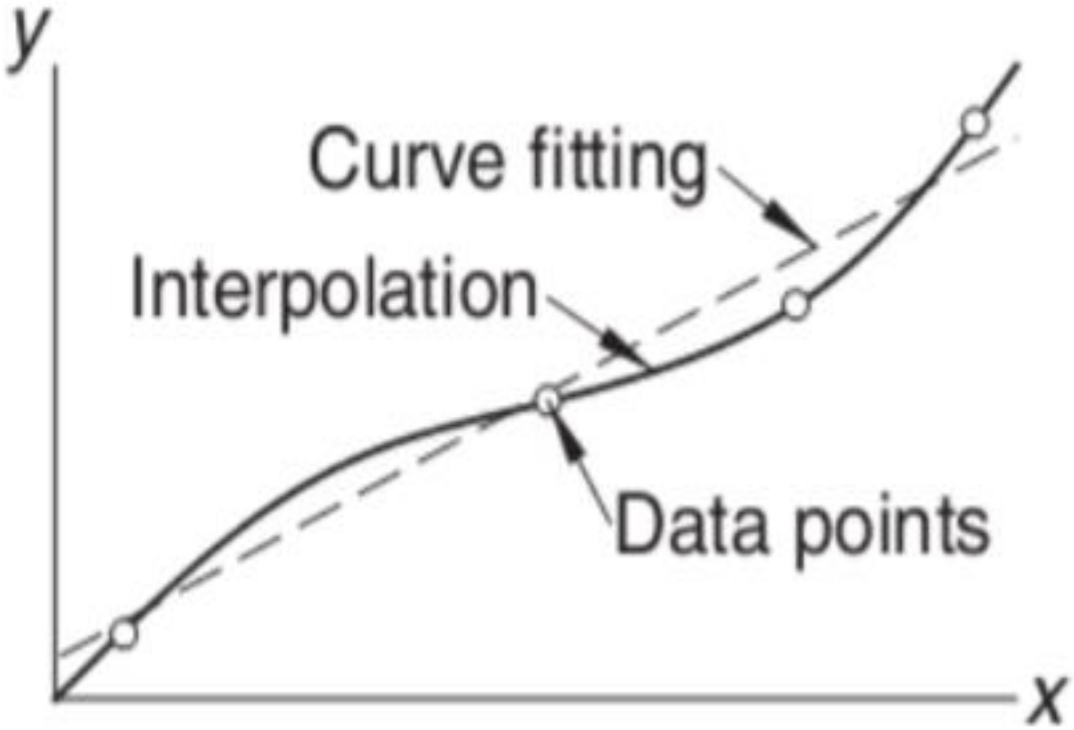
14.05.2021
SAAT:16:00-17:00

Ara Değer Bulma Yöntemleri

- A- Lagrange Polinom İnterpolasyonu
- B- Newton Polinomları
- LAB: Lagrange Polinom İnterpolasyonu yöntemi Matlab örnek programı

BÖLÜM 3 - İNTERPOLASYON VE EĞRİ UYDURMA

Bu bölümde interpolasyon ve eğri uydurma konu başlıklarını irdeleyeceğiz. İnterpolasyon ayrık noktalarda tanımlanan bir veriyi bağlantı kurararak verilen noktaların arasında bir noktadaki değerinin bulunmasına olanak sağlayacaktır. Bunun yanı sıra eğri uydurmada ise, verilen veriler doğrultusunda en iyi eğriyi bulmayı sağlayacaktır. Eğri uydurmada bulunan grafik, verilen verile üzerinden geçmek zorunda değildir.



İnterpolasyon ve eğri uydurma arasındaki fark nedir?

- **1. ÖZELLİK:**
- **Eğri uydurma**, dağınık noktaların bir veri kümesine sahip olduğunuzda ve verilerin genel şekline **en iyi uyan bir çizgi (veya eğri) bulduğunuz zamandır**.
- **İnterpolasyon,** iki veri noktanız olduğunda ve ikisi arasında bir değerin ne olacağını bilmek istediğiniz zamandır. Aralarının yarısı ortalama olabilir, ancak ikisi arasındaki yolun sadece dörtte birini bilmek isterseniz enterpolasyon yapmanız gerekir.

- **2. ÖZELLİK:**
- **Eğri uydurma:** y'nin x'e karşı bir veri dağılımı verildiğinde, y = f (x) işlevsel bir bağımlılığı olduğu varsayılır. **Örneğin,** doğrusal bir uyum y = a + bx olur
- a, b parametreleri ile veriler en küçük kareler anlamında yerleştirilerek belirlenir.
- **İnterpolasyon:** Takılan eğri, verilen veri noktaları arasındaki noktalardaki y değerlerini okumak için kullanılır.

LAGRANGE POLİNOMLARI İLE İNTERPOLASYON (INTERPOLATION BY LAGRANGE POLYNOMIAL)

$\{(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)\}$, N+1 adet verilen nokta olsun. Bu noktalardan geçen N.dereceden polinomu $p_N(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N$ şeklinde yazarız. Katsayıları bulmak için noktalardan geçtiği bildiğimizden,

$$a_0 + x_0 a_1 + x_0^2 a_2 + \cdots + x_0^N a_N = y_0$$

$$a_0 + x_1 a_1 + x_1^2 a_2 + \cdots + x_1^N a_N = y_1$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \qquad .$$

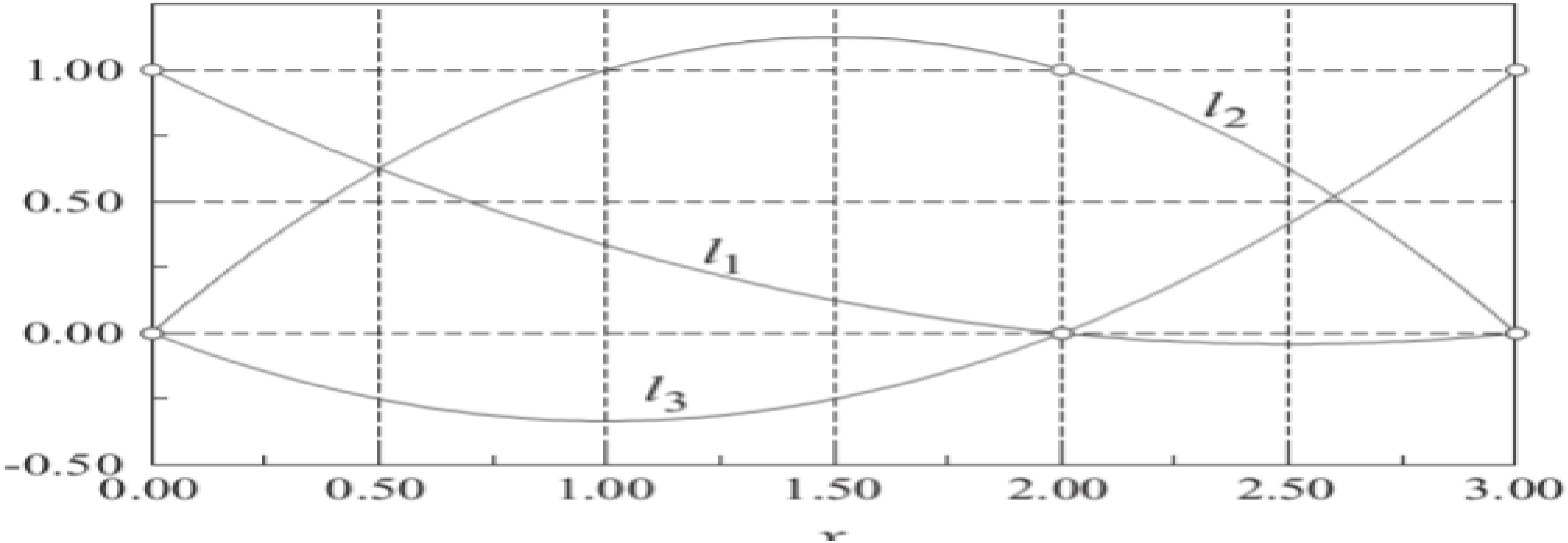
$$a_0 + x_N a_1 + x_N^2 a_2 + \cdots + x_N^N a_N = y_N$$

(N+1)x(N+1) denklem sistemini elde ederiz. Noktalar çoğaldıkça denklem sisteminde bilinmeyenler ve denklemler artacağı için katsayıları bulmak her zaman kolay olmayacaktır. Bu yüzden $\{a_0, a_1, \ldots, a_N\}$ katsayılarını bulmak için Lagrange polinomları adı verilen alternatif bir yol vereceğiz:

$$l_N(x) = y_0 \frac{(x - x_1)(x - x_2)\cdots(x - x_N)}{(x_0 - x_1)(x_0 - x_2)\cdots(x_0 - x_N)} + y_1 \frac{(x - x_0)(x - x_2)\cdots(x - x_N)}{(x_1 - x_0)(x_1 - x_2)\cdots(x_1 - x_N)} + \cdots + y_N \frac{(x - x_0)(x - x_1)\cdots(x - x_{N-1})}{(x_N - x_0)(x_N - x_1)\cdots(x_N - x_{N-1})}$$

$$l_N(x) = \sum_{m=0}^{N} y_m L_{N,m}(x) \quad , \quad L_{N,m}(x) = \frac{\prod_{k\neq m}^{N}(x - x_k)}{\prod_{k\neq m}^{N}(x_m - x_k)} = \prod_{k\neq m}^{N} \frac{x - x_k}{x_m - x_k}$$

Burada $l_N(x_m) = y_m \quad \forall\, m = 0, 1, \ldots, N$ ve $L_{N,m}(x) = \begin{cases} 1, x = x_m \\ 0, x \neq x_m \end{cases}$ koşulu sağlanmaktadır. Örneğin aşağıda $l_1 = L_{2,0}, l_2 = L_{2,1}, l_3 = L_{2,2}$ ile 3 noktalı interpolasyon yapılmıştır.



Editor - C:\Matlab_Dersler\lagrange_polinom.m

regula_falsi.m | sabit_nokta.m | gauss.m | gauss_jordan_elimination.m | jordan_itera...

```matlab
function lagrange_polinom
clc;clear all;
%Giriş:x=[x0 x1...xN], y = [y0 y1 ... yN]
%Çıktılar: l = N.dereceden Lagrange polinom
% L = Lagrange polinom katsayısı
xk = [-2 -1 1 2]; yk = [-6 0 0 6]; % noktalar
n=length(xk);
x=[-2:0.1:2];L=ones(length(x));
for i=1:n
 for j=1:n
 if i~=j
 L(i,:)=L(i,:).*(x-xk(j))/(xk(i)-xk(j));
 end
 end
end
y=0;
for i=1:n
 y=y+yk(i)*L(i,:);
end
plot(xk,yk,'*r',x,y,'k')
```

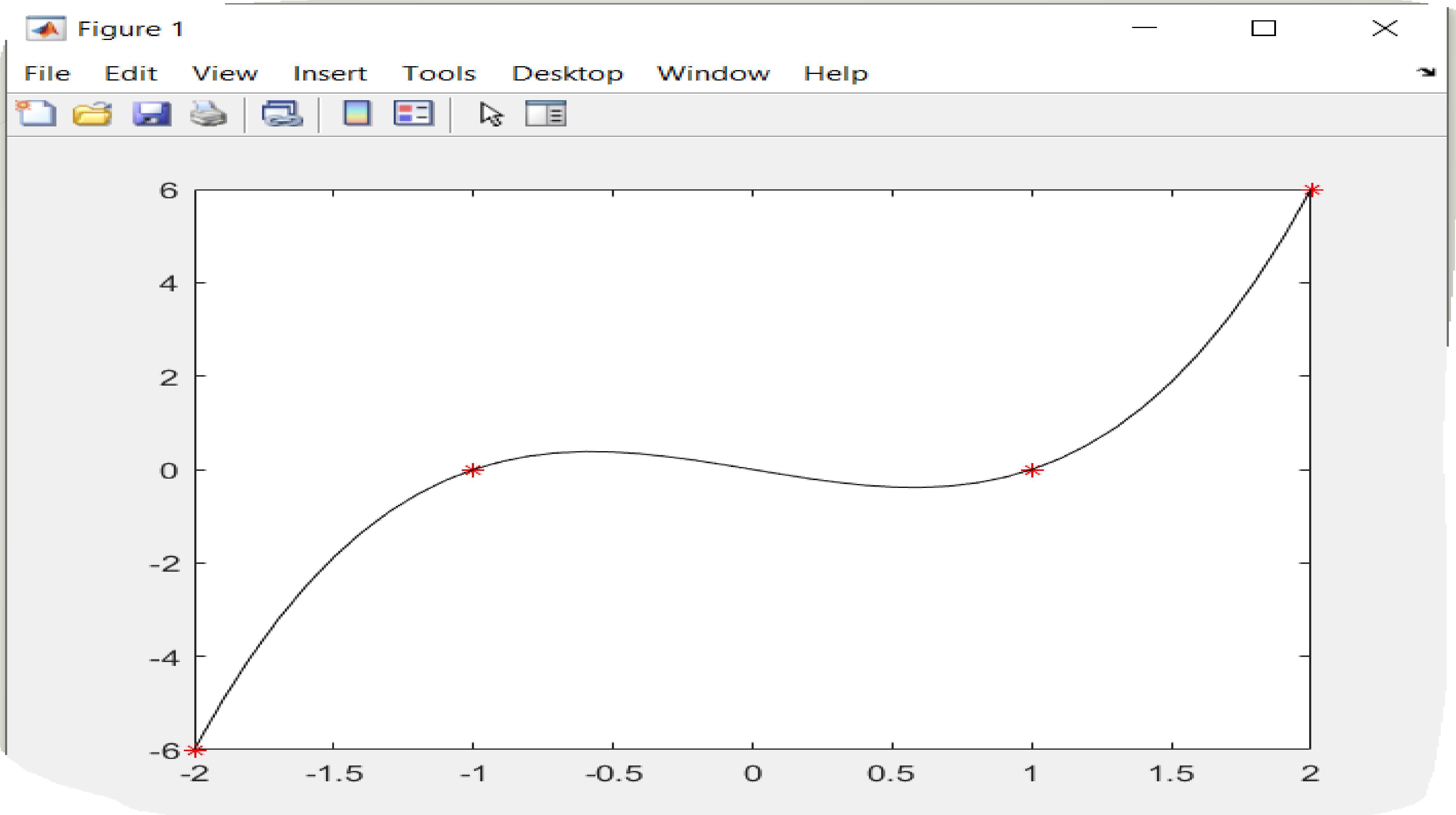


Figure 1
File Edit View Insert Tools Desktop Window Help
6
4
2
0
-2
-4
-6
-2 -1.5 -1 -0.5 0 0.5 1 1.5 2

NEWTON POLİNOMLARI İLE İNTERPOLASYON (INTERPOLATION BY NEWTON POLYNOMIAL)

$\{(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)\}$ noktaları verilen Newton polinomları N.derecedendir ve her polinom bir önceki polinom cinsinden yazılabilir. Lagrange polinomlaronda ise iki polinom arasında bir bağlantı kurulamaz.

$$\begin{aligned} n_N(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots \\ &= n_{N-1}(x) + a_N(x - x_0)(x - x_1)\cdots(x - x_{N-1}) \end{aligned} \quad , \quad n_0(x) = a_0$$

Olarak verilir. Burada $\{a_0, a_1, \ldots, a_N\}$ katsayıları aşağıdaki şekilde hesaplanır:

$$n_1(x) = n_0(x) + a_1(x - x_0)$$

$$n_1(x_0) = a_0 + a_1(x_0 - x_0) = y_0$$
$$n_1(x_1) = a_0 + a_1(x_1 - x_0) = y_1$$

$$a_0 = y_0, \qquad a_1 = \frac{y_1 - a_0}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \equiv Df_0$$

Tekrar 2.dereceden polinomu yazarsak ve noktadaki değerini yerine koyarsak,

$$n_2(x) = n_1(x) + a_2(x - x_0)(x - x_1) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$

$$n_2(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \equiv y_2$$

$$\begin{aligned} a_2 &= \frac{y_2 - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} = \frac{y_2 - y_0 - \dfrac{y_1 - y_0}{x_1 - x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{y_2 - y_1 + y_1 - y_0 - \dfrac{y_1 - y_0}{x_1 - x_0}(x_2 - x_1 + x_1 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{\dfrac{y_2 - y_1}{x_2 - x_1} - \dfrac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0} = \frac{Df_1 - Df_0}{x_2 - x_0} \equiv D^2 f_0 \end{aligned}$$

Ve bu işlemi genelleştirirsek,

$$a_N = \frac{D^{N-1} f_1 - D^{N-1} f_0}{x_N - x_0} \equiv D^N f_0$$

sonucunu elde ederiz.

Bunu tablo halinde aşağıdaki şekilde gösteririz:

x_k	y_k	Df_k	$D^2 f_k$	$D^3 f_k$	—
x_0	y_0	$Df_0 \equiv \dfrac{y_1 - y_0}{x_1 - x_0}$	$D^2 f_0 \equiv \dfrac{Df_1 - Df_0}{x_2 - x_0}$	$D^3 f_0 \equiv \dfrac{D^2 f_1 - D^2 f_0}{x_3 - x_0}$	—
x_1	y_1	$Df_1 \equiv \dfrac{y_2 - y_1}{x_2 - x_1}$	$D^2 f_1 \equiv \dfrac{Df_2 - Df_1}{x_3 - x_1}$	—	
x_2	y_2	$Df_2 \equiv \dfrac{y_3 - y_2}{x_3 - x_2}$	—		
x_3	y_3	—			

Editor - C:\Matlab_Dersler\newton_polinom.m

regula_falsi.m | sabit_nokta.m | gauss.m | gauss_jordan_elimination.m | jordan_iterasyonu.m | lu_ayrim.m | lu_matlab_soltion.m

```
function newton_polinom
clc;clear all;
%Giriş:x=[x0 x1...xN], y = [y0 y1 ... yN]
%Çıktılar: l = N.dereceden Newton polinom
xk = [-2 -1 1 2]; yk = [-6 0 0 6]; % noktalar
n=length(xk);
Df(:,1)=(yk(2:n)-yk(1:n-1))./(xk(2:n)-xk(1:n-1));
for i=2:n-1
 Df(1:n-i,i)=(Df(2:n-i+1,i-1)-Df(1:n-i,i-1))'./((xk(2+i-1:n)-xk(1:n-i)));
end
a(1)=yk(1);a(2:n)=Df(1,1:n-1);
x=[-2:0.1:2];nx=length(x);
Newton(1,1:nx)=a(1)*ones(1,nx);
for i=2:n
 c=1;
 for j=1:i-1
 c=c.*(x-xk(j));
 end
 Newton(i,1:nx)=Newton(i-1,1:nx)+a(i)*c;
end
save ('newton_polinom.mat','Newton','xk','yk','x')
plot(xk,yk,'*r',x,Newton(n,:),'k')
```

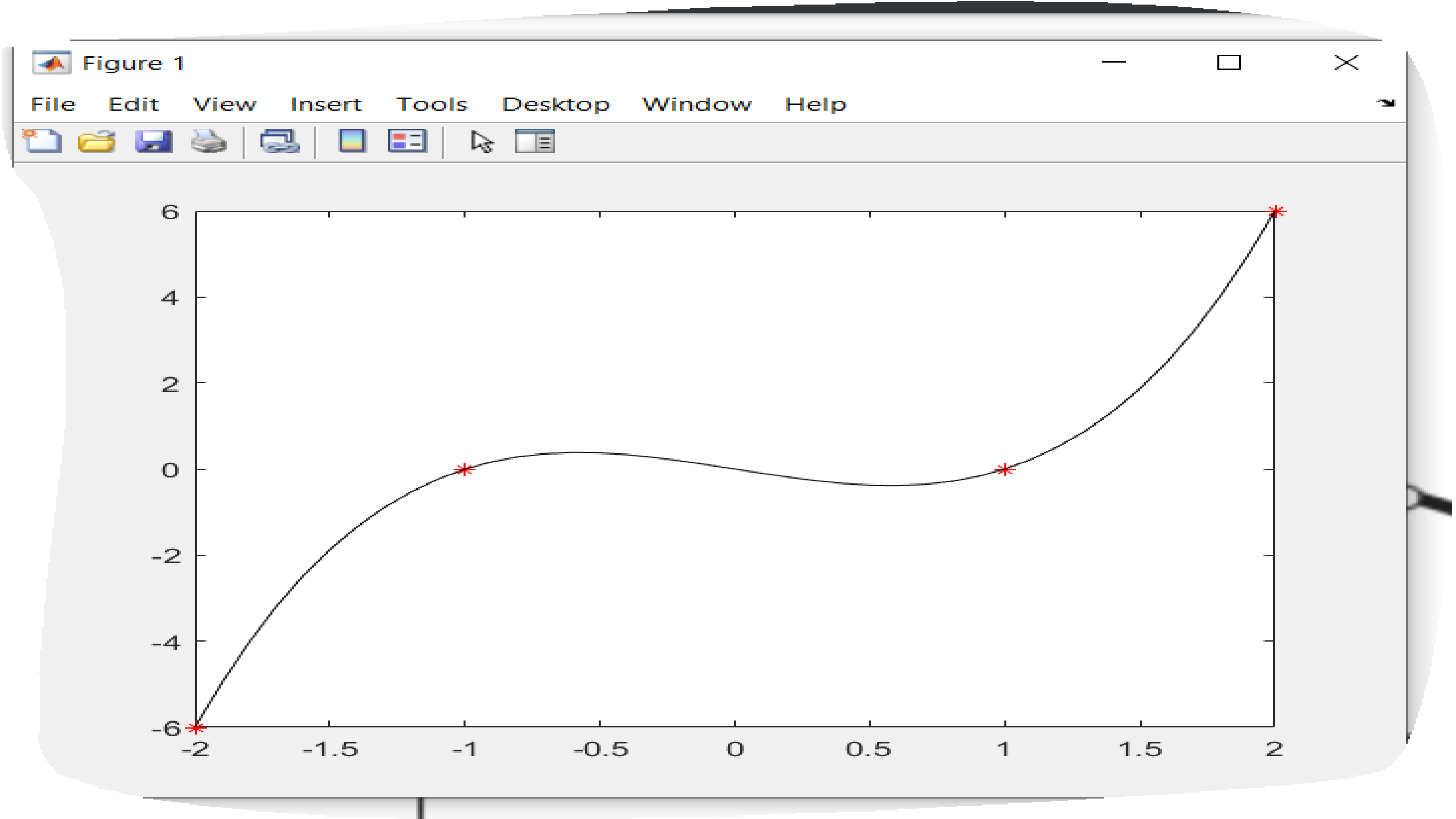
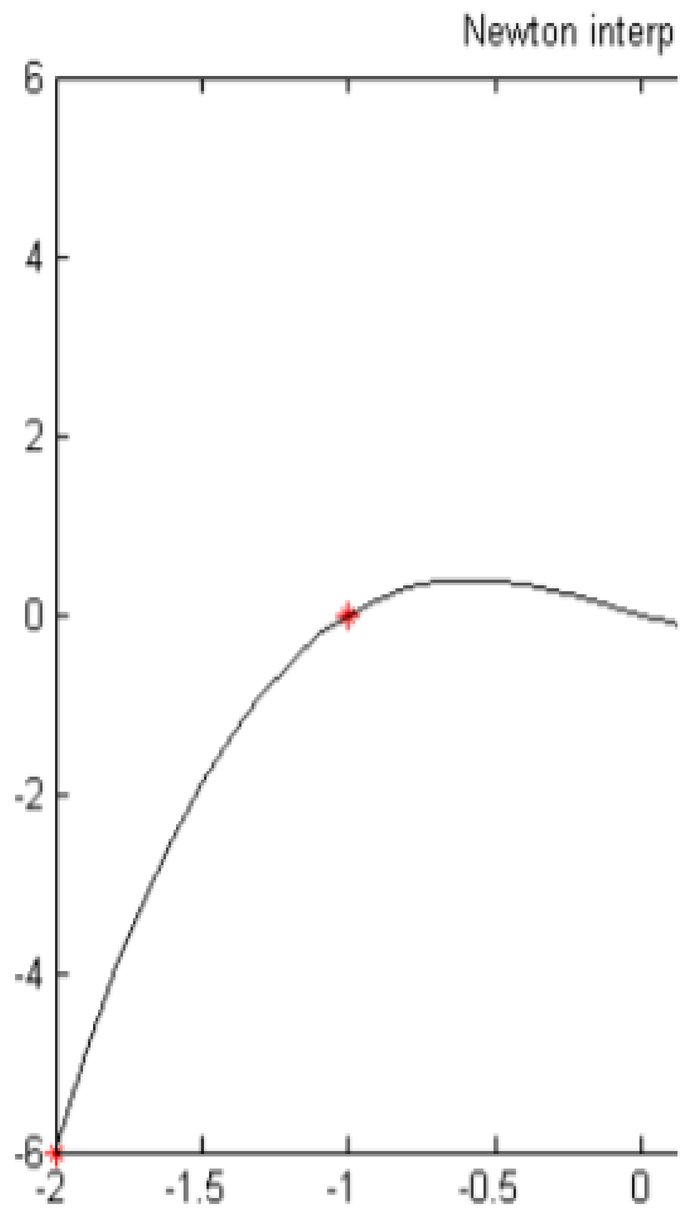
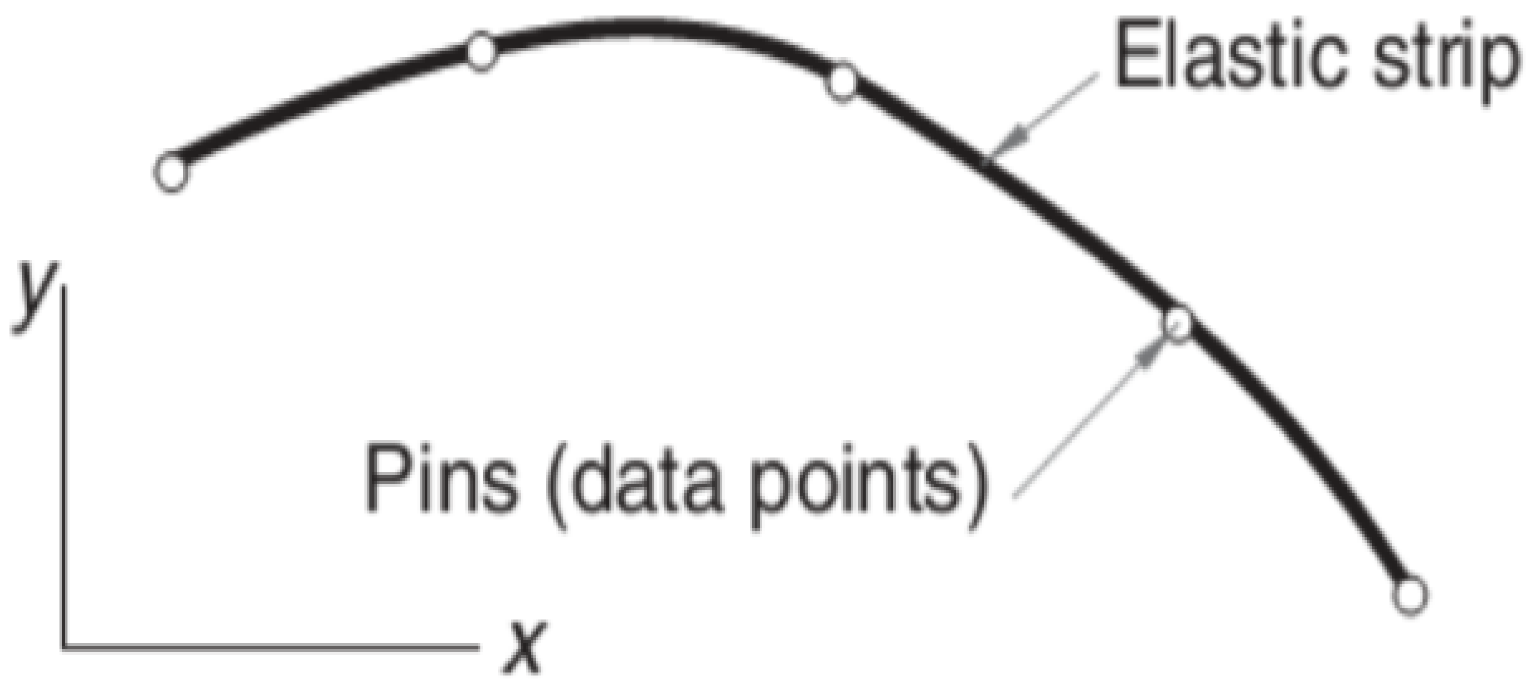


Figure 1
File Edit View Insert Tools Desktop Window Help
6
4
2
0
-2
-4
-6
-2 -1.5 -1 -0.5 0 0.5 1 1.5 2


Newton interp
6
4
2
0
-2
-4
-6
-2
-1.5
-1
-0.5
0


Elastic strip
y
Pins (data points)
x