

MATLAB DERS NOTLARI

ADİL YÜCEL

SEVGİ PÜSKÜLLÜ

**İTÜ
MAKİNA FAKÜLTESİ**

**İTÜ
İŞLETME FAKÜLTESİ**

2012

Matrix Laboratory

21/06/2010

Interface = (Arayüz) Programın ana ekranı

Kod: Yazılan program

Yüksek Seviyeli Dil: İnsan konuşma diline benzer

Düşük Seviyeli Dil: " " " benzer

Çok Yüksek Seviyeli Dil: Matlab :)

* Matlab her şeyi matris mantığıyla çözer.

Transpoze = Satırı sütun, sütunu satır yapar.

NOT: İki matrisin çarpılabilmesi için 1. nin satır sayısı ile 2. nin sütun sayısı eşit olmalı

Vektör: Tek satırlık veya tek sütünlük matris

$[1 \ 3 \ 4 \ 5]_{1 \times 4}$

yatay vektör
(satri)

(array) = sayı dizisi

$\begin{bmatrix} 8 \\ 6 \\ 7 \\ 3 \end{bmatrix}_{4 \times 1}$

düşey vektör
(kolon)

✓ + } gerek yok, zaten eleman elemana yapar.
✓ - }
✓ / } işlemler matrise yapar. (Matris çarpımı)
✓ * }
✓ ^ }

⊗ . *

⊗ . /

⊗ . ^

• (nokta) = element to element (eleman elemana yapar)

$$* \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} 1 & 2 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 9 & 6 \\ 19 & 14 \end{bmatrix}$$

$$* \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot * \begin{bmatrix} 1 & 2 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 12 & 8 \end{bmatrix}$$

* Nokta çarpımı yapmak için. 1. (m x n) ise 2. de (m x n) olmalı, eşit olmalı.

3000 elemantlı matris şöyle yazılır:

$$x = [\text{---}]_{1 \times 3000}$$

$$y = 3 + x \quad y = [\text{---}]_{1 \times 3000}$$

$$z = x.^2 \quad (\text{her elemanın karesini alır})$$

• (nokta) \odot demek her elemanı ayrı ayrı demek.

* Vektörü sabit bir sayı ile çarpmak demek her elemanı çarpmak demek.

* Başta ki sayı sabitse nokta koyarsan da olur koymazsan da olur.

$$y = 3 \odot * x$$

kübünü alacaksak;

$$y = x \cdot *^3$$

$$y = x \cdot * x \cdot * x$$

$$\text{mat} \quad \text{matlab}$$
$$e^5 = \text{exp}(5)$$

$$\ln 5 = \text{log}(5)$$

clc = ekranı temizler

clear all = workspace 'i temizler

clear 'x' = workspace de x 'i temizler

;' = işlemi yapar ama göstermez (Gizli)

[] = matris yazar.

x = [1 2 3 4 5 6]

1 2 3 } alt satıra geçer.

4 5 6

' = transpozunun alınmış halini verir. Fonksiyonun orijinal hali aynıdır.

$(c = c')$ \Rightarrow c'yi artık c'ne atar.

$\checkmark c+3 \rightarrow$ her elemana 3 ekler, orijinali aynıdır.

$\checkmark c = c+3 \rightarrow$ c artık 3 eklemiş hali olur

İşlem Sırası: $10^{\wedge}(-3) = 1e-3 \Rightarrow$ yazılabilir.

\wedge / * + -

22.06.2010

OTOMATİK DİZİ OLUŞTURMA FONKSİYONLARI

* Matlab'de (:) her zaman birseyden birseye kadar demektir.

2:5 \rightarrow 2'den 5'e kadar

X = 1:2:100

\hookrightarrow \hookrightarrow 100'e kadar
atlamak

* $x = 1:10 \rightarrow 1$ 'er 1 'er atlar

* `linspace` = komut

* $x = \text{linspace}(1, 100, 200)$ \rightarrow 199 aralık
(1 ile 100 ü eşit aralıklı 200 noktaya böl)

! $x = \text{linspace}(0, 10, 5)$ $\xleftrightarrow{\text{aynı}}$ $x = 0:2.5:10$
5 nokta

✓ `logspace(1, 4, 4)` ✓ `linspace(1, 4, 4)`
= 10 100 1000 10000 = 1 2 3 4

✓ `zeros(n, m)` \Rightarrow n satırlı, m sütunlu 0 larla dolu matris

`zeros(1, 3)`
= 0 0 0 \rightarrow vektör olur.

✓ `ones` \Rightarrow bire birlik vektörler ve matris yapar.

`ones(1, 3)` } `ones(4, 4)`
[1 1 1]

✓ `eye(n)` \Rightarrow birim matris
köşegeni bir diğer tüm elemanları sıfır.

1 0 0
0 1 0
0 0 1

22.06.2010

NOT:

\cdot $*$ $/$ $//$ $.$ $^$ (element to element)

Fonksiyonlarda Grafik Çizme

Grafiğin daha güzel çizilmesi için başka değerler olması daha iyi.

~~ör~~ $x = -10 : 0.01 : 10$

$$y = 5 * (x.^2) + 3 * x + 4$$

plot(x,y)

tan

sin(30) = 0,5

cos

derece cinsinden yazmada kullanılır.

plot(x,y)

title('super grafik') → başlık alır / grafiğe isim verir

xlabel('güzel xler') → grafiğin x eksenine isim verir

ylabel('tatlı yler') → y eksenine isim verir.

grid(on) → karelere biter

*2 kere üst üste grid yazarsak kareler gider.

✓✓ $x = -2 * \pi : \pi / 100 : 2 * \pi$

$$c = \cos(2 * x)$$

$$s = \sin(2 * x)$$

$$t = \tan(2 * x)$$

plot(x,c,x,s,x,t) ↓ renklendirmek için

c.'r' x s.'a'

(ve) İkisinde true olursa doğru budur
⇒ and & → İki parametrede doğruysa yeterli
⇒ or | → Herhangi birinin doğru olması yeterli
(veya)

M-file:

Komutları yazarız.

Dosya adı verirken:

- Türkçe karakter yok
- Boşluk yok
- Noktalama işareti yok
- Sayı ile başlamayacak

if Döngüleri:

✓ disp('Sevgi')

✓ yaş = input('Lütfen yaşınızı giriniz')

✓ Lütfen yaşınızı giriniz = 20

✓ yaş = 20

* if x < 5

* if x < 5

else

end

end

x 5'ten
büyükse
bu ko-
mutlar
yapılır

değilse
bu
komutlar
yapılır

22/06/2013

```
* if      x < 5  
      _____  
      _____  
      _____  
elseif  x < 6 → koşul kavr  
      _____  
      _____  
      _____  
else    _____ → koşul kavr  
      _____  
      _____  
end
```

* Hangisini sağlarsa onu yapıp bitirir,
diğerine atlamaz

```
yas = input ('lutfen yaşıizi giriniz');
```

```
if yas < 18
```

```
    disp ('bora girmek yasak')
```

```
    disp ('büyü de gel')
```

```
elseif yas < 20
```

```
    disp ('patrona sorulm')
```

```
    disp ('tanıdıkça alın')
```

```
else
```

```
    disp ('buyrun') if ile else arasına  
                    istediğin kadar elseif
```

```
    disp ('ne alırdınız') boy.
```

```
end
```

for Döngüsü

28.06.2010

```
for   k = 1:10  
      _____  
      _____  
      _____  
end
```

} k'ya 1'den 10'a kadar değerler verir hepsi için alttaki komutları yapar.

*for yapısında ASLA k'ya müdahale etme, atama yapma mesela $k = 5$ gibi

⇒ toplam = 0; } en başta toplamı sıfırla ki
estki kalan değere işlem yapmışın.

```
for   k = 1:50  
      _____  
      toplam = toplam + k  
end  
toplam
```

yığıma

*Yığıma yapılıyorsa for'dan önce değişkeni mutlaka sıfırla!

$$\underline{\underline{\text{ör:}}}$$
$$\sum_{n=1}^{100} \left(\frac{1}{n} \right)$$

$$S = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{100}$$

$$S = 0;$$

```
for   n = 1:100  
      _____  
      S = S + (1./n)  
end  
S
```

while Döngüsü

for gibi otomatik arttırmaz!
başlangıç değeri dışarı verilmelidir

```
while k < 5 → sayacı = counter  
    ==  
    ==  
    k = k + 1 → k'yi arttırmayı döngüde  
for'dan farkı: kendimizi veririz, vermezsek  
    sonsuz döngüye girer.  
bunu vermek zorunda- Bunu döngü içinde istediği-  
    mı? yerde verebiliriz.  
end
```

for

while

- * Başlangıç değeri vermez.
- * Başlangıç değerini ver
- * Değişkene m-danele edilmez
- * Değişkenin artımını vermek zorundasın
- * Otomatiktir.

ör:

clc

clear all;

k = 1;

toplam = 0; ⇒ sıfırla

while k < 11

notum = input('notunuzu giriniz');

toplam = toplam + notum;

k = k + 1;

end

```
clc  
clear all
```

```
toplam = 0;
```

```
notum = input('notu gir');
```

```
while notum ~ = 0
```

```
    toplam = toplam + notum;
```

```
    notum = input('notu gir');
```

```
end
```

```
toplam
```

ÖRNEK .

notu < 50 → FF

50-55 → DD

55-60 → DC

60-65 → CC

65-85 → BB

85-100 → AA

```
notum = input('Notunuzu giriniz');
```

```
if notum <= 50
```

```
    disp('FF aldınız');
```

```
elseif notum <= 55
```

```
    disp('DD aldınız');
```

```
elseif notum <= 60
```

```
    disp('DC aldınız');
```

```
elseif notum <= 65
```

```
    disp('CC aldınız');
```

```
else if notum <= 85
```

```
    disp('BB aldınız');
```

28.06.2010

fprintf (ekranda gösterme)

* `fprintf ('%d yılı için faiz %f dir', (x), (y));`
bunlar birer belirteçtir

⇒ x = 1989

⇒ y = 5000

* 1989 yılı için faiz 5000.000000'dür
ekranda bu şekilde çıkar.

%d → tam sayılar için

%f → ondalıklı sayılar için

yil = 5

para = 10.000

yazdıktan
sonra alt
sıra geçer

`fprintf ('%d.yıl için faiz %f olmaktadır. \n', yil, para)`

5.yıl için faiz 10000.000000 olmaktadır.

`fprintf ('%3d.yıl için faiz %15f olmaktadır. \n', k, para)`
3.kolona dayar 15.kolona dayar (hiçler)

%5d demek:

yazdıracağın sayı 3 karakterliyse 5'le tamamlayar.

— 115 (2 boşluk bıraktı)

2 karakterliyse

— 75 (3 boşluk bıraktı)

Sadece f'le özel olan şey:

%15.③f

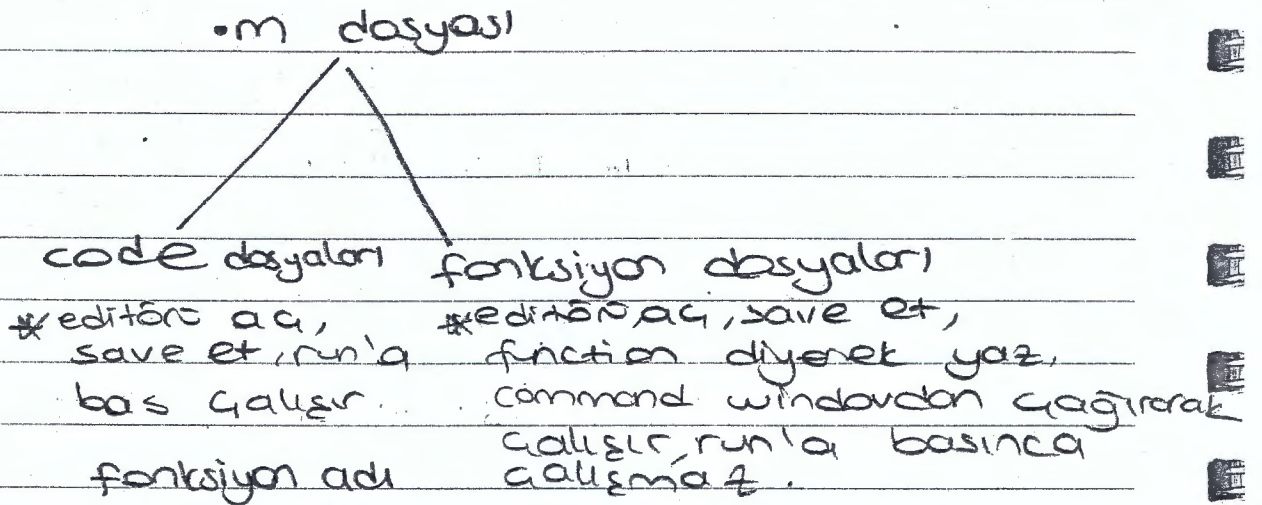
virgülden sonra 3 hane gösterir.

• f in önce koyulan sayı virgülden so

29.06.2010

* Özel Tanımlı Fonksiyonlar *

* "function" terimi kullanılır.



* function $y = \text{sevgi}(x)$ → giriş değişkeni, argüman, sonuç değişkeni parametre

$$y = x^2$$

* sadece fonksiyon içinde, sonucu tanımlamak için kullanılır. Dosya ismi fonk. ismi ile aynı olmak zorunda. ADINI DEĞİŞTİRME

sevgi çağırılır:

* sevgi(5)

* y yerine sonuç yaz daha anlaşılır.

→ function sonuç = topla(a,b)

→ sonuç = a+b;

2 argümanda da bilir.

→ topla(3,5)

→ 8

Lecture 11: Taylor Series:

```
function sonuc = sinus_taylor(x, n) % her zaman  
    toplam = 0; % ilk deđiřken  
    for k = 0:n % x, ikincisi  
        terim = ((-1)^k * x^(2*k+1)) / factorial(2*k+1); % olurl  
        toplam = toplam + terim; % (2*k+1)!  
    end  
    sonuc = toplam;
```

řöyle çağırırız: mesela $\sin 150^\circ$ 'yi çağır

$\text{sinus_taylor}(150 * \pi / 180, 3)$ \rightarrow terim sayısı

Example 2: ln2'nin acılımı olan!

```
function sonuc = logaritma(n) % terim sayısı  
    toplam = 0;  
    for k = 1:n (soruda 1'den başlıyor)  
        terim = (-1)^(k-1) / k;  
        toplam = toplam + terim;  
    end  
    sonuc = toplam;
```

řöyle çağırırız:
 $\text{logaritma}(500)$

Lineer Denklem Takımları

$$\left. \begin{array}{l} 4x - 2y + 6z = 8 \\ 2x + 8y + 2z = 4 \\ 6x + 10y + 3z = 0 \end{array} \right\} \begin{array}{l} \text{Doğrusal (lineer)} \\ \text{denklem takımları} \end{array}$$

Matris formu

$$\begin{bmatrix} 4 & -2 & 6 \\ 2 & 8 & 2 \\ 6 & 10 & 3 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix}$$

katsayılar matrisi bilinmeyen vektörü sonuç vektörü
↓
amaç bunu bulmak

$$\begin{aligned} K * X &= S \\ X &= K^{-1} * S \end{aligned}$$

matrisin tersi (inv ile yazılır)

* det → Determinantını alırlar

$$\boxed{\vec{X} = \text{inv}(K) * \vec{S}} \quad \text{*** Mutlaka sorulur!}$$

bilinmeyenler

$$K = [4 \ -2 \ 6; \ 2 \ 8 \ 2; \ 6 \ 10 \ 3]$$

$$S = [8; \ 4; \ 10]$$

$$\text{inv}(K) * S$$

Kök Bulma

* Sayısal yöntemler = deneme yanılma yöntemi

* Yapılan her adıma "iterasyon" denir

Bir fonksiyonun cesitli yöntemlerle kökünü bulmaya yarayan yapıdır.

Bisection Yöntemi:

Bisection: Tüye bölme demek

* Her yöntemde kullanılacak 2 adım:

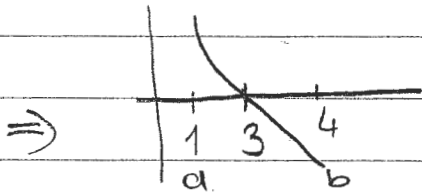
⇒ Fonksiyonun grafiğini çizdirmek

→ Grafikten bakarak kökün nerede olduğunu tahmini olarak görmek.

* Bisection'da yapılacak 3. adım:

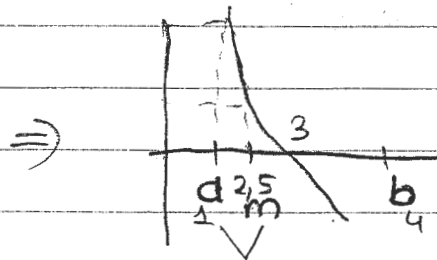
→ 0 kökü içeren bir başlangıç aralığı seçmek

kök 3 ise başlangıç olarak 1-4 aralığını seçebiliriz.



$$(+).(+)=+$$

1 - 2,5 aralığı



$$\frac{1+4}{2} = 2,5$$

2,5 - 4 aralığı

$$(-).(-)=+$$

funk. nun 1 ve 2.5teki değerini uarp sonucu pozitif ise at diyeceğiz. $f(a) \cdot f(m) > 0$

$$f(x) = \cos x - x = 0$$

fonksiyon adı f

function sonuc = f(x);

} f.m diye

sonuc = cos(x) - x;

} kaydet

⇒ clc

function yazoruz

x = -5:0.1:5;

acaba sorulur

y = f(x);

f(x)'i yazmayız

plot(x,y);

sadece f(x) olarak

grid;

yazoruz

* Kökü yaklaşık 0,75 bulunca aralığı 0-1 seçtik

⇒ a=0;

b=1;

yazmamızın nedeni ± 90000

m = (a+b)/2;

olabilir. işareti götürsün diye

while $\text{abs}(f(m)) > 1e-6$ % m noktası kök değil

if $f(a) * f(m) > 0$ → ikisi de aynı taraf-

a = m; → taysa a'nın tarafını at

else

→ yeni a değerini m olur.

b = m; → değilse b tarafını at,
b = m olur.

end

m = (a+b)/2;

→ a ya da b

değiştirdiğinden

tekrar orta noktayı

hesaplarız.

end

m

m = 0.7391

alıyor.

Newton Yöntemi:

Bu yöntemde kök içeren aralık yerine sağlıktan ve soldan nokta almak yeterli!

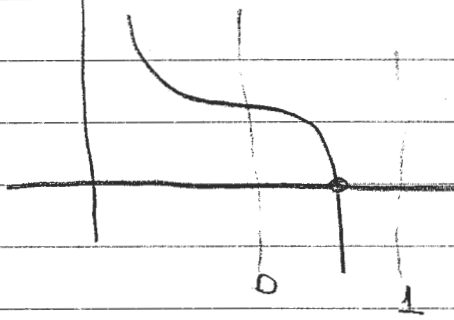
clc

x = -5 : 0.1 : 5 ;

y = f(x) ;

plot (x,y) ;

grid



function sonuc = f(x) ;

sonuc = cos(x) - x ;

function sonuc = fussy(x) ;

sonuc = -sin(x) - 1 ;

clc

x0 = 1 ;

while abs (f(x0)) > 1e-6

x1 = x0 - f(x0) / fussy(x0) ; } yeni x bul

x0 = x1 ; } yeni bulunan x'li x0'a eşitle

end

x0

x0 kök olmadığı
sıçra demek

estki x0'dan

yeni bir x'aret

ve crettigin yeni

x'ir x2 olarak

kabul et, kök m

bak, x0 kök olana

tocker döner *

Fixed-Point Iteration Yöntemi:

While'in her dönmüşü bir iterasyondur.

$$\left. \begin{array}{l} \cos x - x = 0 \\ x = \frac{\cos x}{g(x)} \end{array} \right\} \Rightarrow \text{kökünü bulacağın denklemi} \\ x = \text{--- seklinde yaz.}$$

```
function sonuc = g(x)
```

```
sonuc = cos(x);
```

```
clc
```

```
sayac = 0;
```

```
X0 = 1;
```

→ kökünden sağından veya solundan
sayı alınacaktır, sağından 1'i aldık

```
fark = 100;
```

→ $1e-6$ 'dan büyük herhangi bir sayı
abak
fark milyonda
birden büyük olduğun
süre

```
while abs(fark) > 1e-6
```

```
    X1 = cos(X0);
```

```
    fark = X1 - X0;
```

```
    X0 = X1;
```

```
    sayac = sayac + 1;
```

```
end
```

```
X0
```

kaç defa döndüğünü
gösterir!!! (Önemli-Sınav!)

Polinomlar

$$3x^4 + 2x^3 + 3x^2 + 5x + 7 \quad \text{derecesi} = 4$$

(matlab 'de:

$$\begin{aligned} & [3 \ 2 \ 3 \ 5 \ 7] \quad \text{x}^3 \text{ olmasaydı} \\ & [3 \ 0 \ 3 \ 5 \ 7] \quad \downarrow \end{aligned}$$

$$\underline{\underline{ör:}} \quad p = x^2 - 5x + 6$$

$$\rightarrow p = [1 \ -5 \ 6]$$

polyval(p, s) \rightarrow p polinomunda x yerine s koy

Polyval = 1 x girersen, 1 tane y, 100 tane x girersen 100 tane değer verir. Polinomun x tek "değerini" verir

$$* \text{polyval}(p, [5 \ 10 \ 15 \ 20])$$

$$= 6 \ 56 \ 156 \ 306$$

roots: kökleri verir. (polinomun)

$$* \text{roots}(p)$$

$$\text{ans} = \begin{matrix} 3 \\ 2 \end{matrix}$$

poly: köklerden polinomu verir. (Roots'un tersi)

$$\text{poly}([2 \ 3]) = 1 \ -5 \ 6$$

conv : Tki polinomunu carpip, carpim polinomunu verir.

$$* p1 = [1 -3]$$

$$p2 = [1 -2]$$

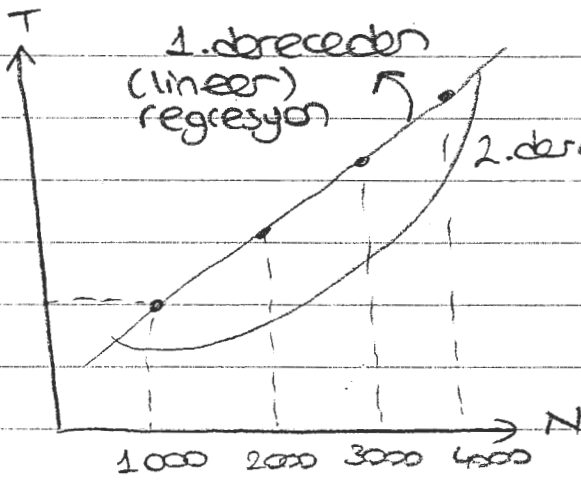
$$\text{conv}(p1, p2)$$

$$= 1 -5 6$$

polyder : pol. un turevini verir.

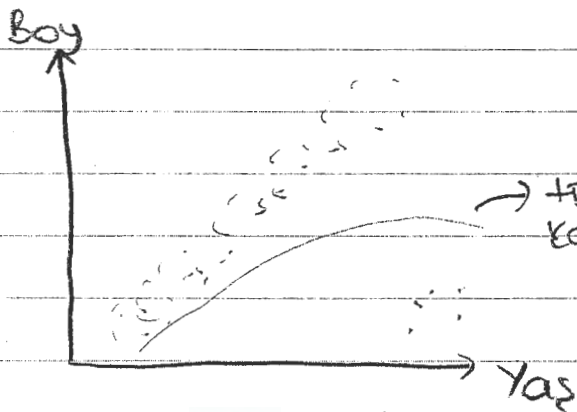
$$* \text{polyder}(p) = 2 -5$$

Egri Uyduurma (Curve Fitting)



* Asil ismi
"regresyon"

* Egri noktalarin geirmek zorunda degil!



→ tum noktalar katarsan boye bir egri cikar!

* Abitosis
sonuclari at,

* $\text{polyfit}(x, y, n)$ = x ve y noktalarına n. dereceden bir polinom uyduruyor

→ enterpolasyon
* Örneğimizi noktaların hepsinin üzerinden geçen bir eğri istiyorsak $(n-1)$. dereceden polyfit yapmalısn.

* `clc`

`x = [0.9 1.5 3 4 6 8 9.5];`

`y = [0.9 1.5 2.5 5.1 4.5 4.9 6.3]`

`plot(x, y, 'o');` → 3. derece polinom

`p = polyfit(x, y, 3);`

`yenix = 0: 0.01: 10;`) grafiği çizdirmek için başka

`yeniy = polyval(p, yenix);` } x değerleri

`plot(x, y, 'o', yenix, yeniy)` } \downarrow yenixlerin polinom değerleri

netice görmek için bunları da

• boyduk, kaymasakta olur.

*

→ hep 1. dereceden
 $\text{polyfit}(x, y, 1)$

* $\text{polyfit}(\log(x), \log(y), 1) \Rightarrow$ Power function $y = bx^m$

* $\text{polyfit}(x, \log(y), 1) \Rightarrow$ Exponential function $y = be^{mx}$

* $\text{polyfit}(\log(x), y, 1) \Rightarrow$ logarithmic function

$$y = m \ln(x) + b$$

$$y = m \log(x) + b$$

deneyerek bakarsınız, isimize hangisi yarıyor; onunla
çalışırsınız.

- * plot \rightarrow iki eksen normal (x, y)
- * loglog \rightarrow iki eksen logaritmik
- * semilogx \rightarrow x eksen logaritmik, y normal
- * semilogy \rightarrow y eksen logaritmik, x normal

Bunlara karar verdikten sonra orka sayfa-
dan polyfit'i (uygun olanı) seçersiniz.

ör:

normal(x = []);

normal(y = []);

p = polyfit(x, log(y), 1);

Semilogy
ile çizdiğimizden
bunu yazdık

normal(yenix = 0:0.01:5);

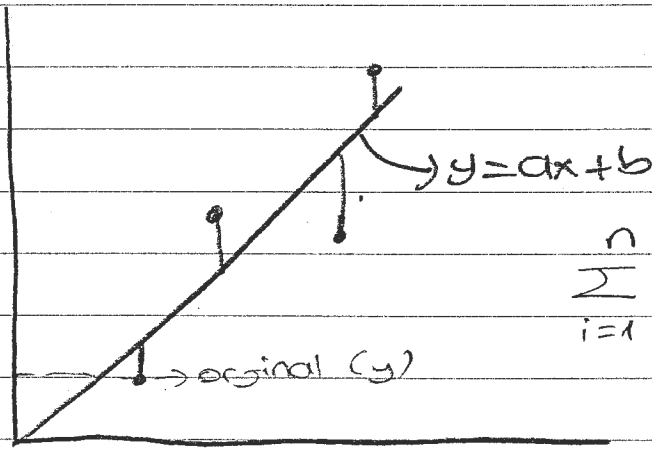
logaritma(yeni_y = polyval(p, yenix));

plot(x, y, 'o', yenix, exp(yeni_y))

* Logaritmik sayıyı normal sayıya çevirmenin
yolu e üzeri almak

Doğru Uydurma (Linear Regresyon)
polyfit(x, y, 1) uydurmak → regresyon

Original y değer ile farklı xleri koyduğumuzda elde ettiğimiz yer arasındaki farkların karelerinin toplamı minimum olacak. Buna en küçük kareler yöntemi denir.



$$\sum_{i=1}^n (y_i - (ax_i + b))^2$$

$$* \boxed{S = \sum (y_i - ax_i - b)^2} \rightarrow \text{min olmak}$$

✓ x_i ve y_i bilinen
✓ a ve b bilinmeyen

* Pozitif olsun diye
koresini alıyoruz!

Bu toplamın minimum olması demek, karesinin "0" olması demektir!

$$\frac{\partial S}{\partial a} = 0 \quad \frac{\partial S}{\partial b} = 0 \quad * a \text{ ve } b'ye \text{ göre kısmi} \\ \text{trev "0" !}$$

→ O da esit olacağından giderler.

$$-2 \sum (y_i - ax_i - b) \cdot x_i = 0 \rightarrow a'ya göre türev$$

$$-2 \sum (y_i - ax_i - b) = 0 \rightarrow b'ye göre türev$$

↓ * sigmoidayı dağıtalım:

$$-\sum x_i y_i - a \sum x_i^2 - b \sum x_i = 0$$

$$-\sum y_i - a \sum x_i - \sum b = 0$$

$$a \sum x_i^2 + b \sum x_i = \sum x_i y_i$$

$$a \sum x_i + \sum b \cdot n = \sum y_i$$

Linear denklem takımına döndü!

$$* \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

ör: $x = [1 \ 2 \ 3 \ 4 \ 5]$;

$$y = [0.7 \ 2.2 \ 2.8 \ 4.4 \ 4.9]$$
;

sum ile x^2 arasına sakın boşluk koyma!!!

$$K = [\text{sum}(x.^2) \ \text{sum}(x) ; \text{sum}(x) \ 5]$$
;

$$S = [\text{sum}(x .* y) ; \text{sum}(y)]$$
;

$$\text{inv}(K) * S$$

$$= 1,0600 \rightarrow a$$

$$-0,1800 \rightarrow b$$

→ çıkar!

(5'inci deger için)

Sorunun cevabı ⇒

$$a \text{ ve } b = \text{inv}(K) * S;$$

$$a = a \text{ ve } b(1); \rightarrow 1. \text{ elemanı } a$$

$$b = a \text{ ve } b(2); \rightarrow 2. \text{ elemanı } b$$

$$\text{yenix} = 1:0.01:5;$$

$$\text{yeniy} = a * \text{yenix} + b;$$

$$\text{plot}(x, y, 'o', \text{yenix}, \text{yeniy});$$

original
x aralığına göre
seçtik

Interpolation (interpolasyon):

Polyfit ve polinomialin birleşmiş şekli

interp1 ⇒ komutu

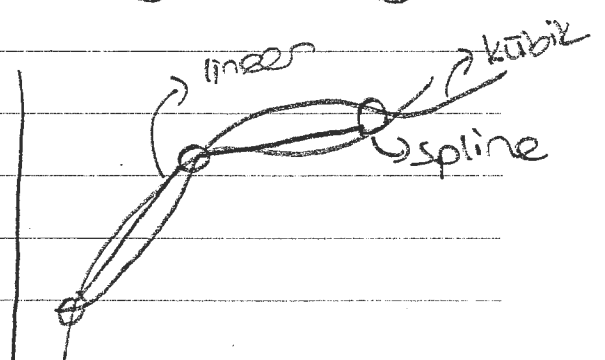
$$\text{Yeniy} = \text{interp1}(x, y, \text{yenix}, \text{method})$$

Tom aalımı, x ve y yi kullanarak methoda göre interpolasyonu yap ve yenix'e karşılık gelen yeniy yi ver

* bir tane x olmak zorunda değil, dizi gerebilirsin.

method yerine;

- ✓ linear
 - ✓ spline
 - ✓ cubic
- } noktalar, ortaya
aitken resme ve
yarma bağlı



ör :

clc

x = [0 1 2 3 4 5];

y = [1.0 -0.6242 -1.4707 3.2406 -0.7366 6.3713];

interp1(x,y,2.5,'linear')

interp1(x,y,[2.5 3.5],'linear')

yenix = 0:0.01:5;

yeny = interp1(x,y,yenix,'cubic');

plot(x,y,'o',yenix,yeny);

linear
spline

3 ünü de aynı grafikte çizdirelim:

yeny - LIN = interp1(x,y,yenix,'linear');

yeny - SPL = " " " " 'spline';

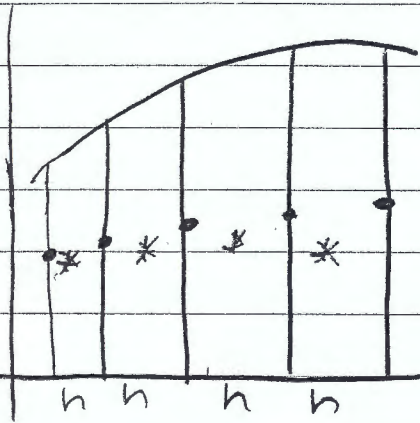
yeny - CUB = " " " " 'cubic';

plot(x,y,'o',yenix,yeny - LIN,yenix,yeny - SPL,
yenix,yeny - CUB);

(yamuk)

*TRAPEZOID ALGORİTHM

Fonk'u biliyorsan kullanılır. Fonk'un altındaki alanı hesaplatırken alanı yamuklara böleriz.



* panel sayısı = 4
• nokta sayısı = 5
nokta = panel + 1
h = her panel genişliği

$$h = \frac{b-a}{\text{nokta sayısı} - 1}$$

panel = ---; \rightarrow saklından ver, arttıka a fonk
nokta = panel + 1; gszellesin

a =

b =

$$h = (b-a) / (\text{nokta} - 1);$$

$$x = a:h:b;$$

y = ---

$$I = (h/2) * (y(1) + 2 * \text{sum}(y(2:\text{nokta}-1)) + y(\text{nokta}));$$

↓
integral

$$S = \frac{y_1 + y_2}{2} h + \frac{y_2 + y_3}{2} h + \frac{y_3 + y_4}{2} h + \frac{y_4 + y_5}{2} h$$

$$S = \frac{h}{2} (y_1 + 2y_2 + 2y_3 + 2y_4 + y_5)$$

KURAL: $\frac{h}{2}$ * baştaki ve sondakinden 1'er tane ortadakilere 2'er tane

SIMPSON ALGORİTİMİ:

panel =

nokta = 2 * panel + 1;

a =

b =

$h = (b - a) / (nokta - 1);$

$x = a : h : b;$

y =

$I = (h/3) * (y(1) + 4 * \text{sum}(y(2:2:nokta-1))) + 2 * \text{sum}(y(3:2:nokta-2)) + y(nokta));$

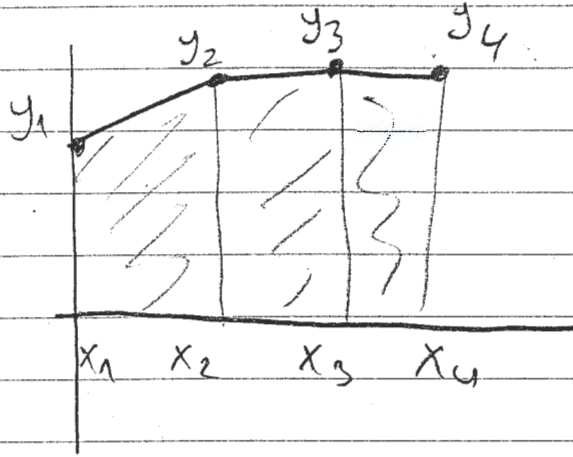
*Yanuk yerine 2. derece polinomları birleştirir.

$I = \text{quad}('function', a, b)$

fonksiyon
402.

integral sınırları
b

*NOT: Eger bize fonk. verilmemisse sadece noktalarimiz belliyse;



$$I = \frac{y_1 + y_2}{2} (x_2 - x_1) + \dots$$

clc

$$x = [3 \ 8 \ 12 \ 15 \ 16];$$

$$y = [27 \ 512 \ 1928 \ 3375 \ 4096];$$

funk belli degil
5 tane (x,y) nok
verildi

$$\text{toplam} = 0$$

for k = 1:4 → (nok. sayisi - 1) / k = 1:length(x)-1

$$\text{alan} = (y(k) + y(k+1) / 2) * (x(k+1) - x(k));$$

$$\text{toplam} = \text{toplam} + \text{alan};$$

end

toplam

trapz(x,y) → x ve y noktaları belli olduğunda
integrali hesaplar. Matlab'in hazır komutu!

Sembolik Matematik

* syms x

$$f = x^3 + 3*x^2 + 5*x + 6 \quad \left. \vphantom{f} \right\} \rightarrow \text{nokta kullanılmaz!}$$

* expand (f)

f'in açılmış halini verir

* $f = \text{expand}(f)$ \rightarrow dersek f bozuktur.

* factor \Rightarrow açık halde olanı parçalarına ayırır.

* $\text{subs}(f, a, 5)$ \Rightarrow f'de a yerine 5 koy

* $\text{poly2sym}([1 \ 5 \ 6])$ \rightarrow polinom vektörün sembolize çevirir.

* $\text{sym2poly}(x^2 + 5*x + 6)$

\downarrow tam tersi

$$= [1 \ 5 \ 6]$$

* $y = x^2 + 5*x$

* $\text{ezplot}(y)$ \sim y'yi -2π 'den $+2\pi$ 'ye çizer

$\text{diff}(y, x)$ \rightarrow y'nin x'le göre türevini alır

$\text{int}(y, x)$ \rightarrow " " " integralini alır