

GÖRÜNTÜ İŞLEME MATLAB

DERS-3



MATRİSLER

Matris İşlemleri:

Verilen bir X matrisi için:

$$\gg X = \begin{bmatrix} 2 & -1 \\ 5 & 8 \end{bmatrix}$$

$X =$

$$\begin{matrix} 2 & -1 \\ 5 & 8 \end{matrix}$$

$$\begin{matrix} 5 & 8 \end{matrix}$$

Determinant:

$$\gg \text{DETERMINANT} = \det(X)$$

$$\text{DETERMINANT} =$$

$$21$$

MATRİSLER

B =

$$\begin{pmatrix} 2 & 3 \\ 4 & -2 \\ 2 & 5 \end{pmatrix}$$

Matrisinin determinantını hesaplayınız.



Sadece kare matrislerin determinantı alınabilir.

MATRİSLER

Matris İşlemleri:

Verilen bir X matrisi için:

$$\gg X = \begin{bmatrix} 2 & -1 \\ 5 & 8 \end{bmatrix};$$

Tersini alma işlemi:

$$\gg Y = \text{inv}(X)$$

Y =

$$\begin{array}{cc} 8/21 & 1/21 \\ -5/21 & 2/21 \end{array}$$

Bir Z matrisi için:

$$\gg Z = \begin{bmatrix} 1 & 2 \\ -2 & 4 \end{bmatrix}$$

Bölme işlemi aynı tersini alma işlemi yapıp çarpma işlemi gibi sonuç verir.

$$\gg K = Z/X$$

K =

$$\begin{array}{cc} -2/21 & 5/21 \\ -12/7 & 2/7 \end{array}$$

$$A \equiv \begin{bmatrix} a & b \\ c & d \end{bmatrix};$$

$$A^{-1} = \frac{1}{|A|} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$= \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix};$$

MATRİSLER

Matrislerle Doğrusal Denklem Çözümü:

$$2x+y-z=5$$

$$x-2y+3z=-6$$

$$-x+y-z=2$$

Şeklindeki örnek denklem için öncelikle yapılması gerekenler:

1) x, y, z nin katsayılarını bir A matrisinde yazalım.

$$\begin{aligned} \text{» } A = & \begin{bmatrix} 2 & 1 & -1 \\ 1 & -2 & 3 \\ -1 & 1 & -1 \end{bmatrix}; \end{aligned}$$

2) Eşitliğin ikinci tarafını bir B matrisine yazalım.

$$\begin{aligned} \text{» } B = & \begin{bmatrix} 5 \\ -6 \\ 2 \end{bmatrix}; \end{aligned}$$

$$K = [x \ y \ z]$$

$$A \cdot K = B \Rightarrow \mathbf{K = A^{-1} \cdot B}$$

MATRİSLER

Matrislerle Doğrusal Denklem Çözümü:

3) İşlemimizi uygulayalım...

$$\gg K = \text{inv}(A) * B$$

$$K =$$

1

2

-1

İkinci bir gösterim sola bölme ile de olabilir.

$$\gg K = A \setminus B$$

$$K =$$

1

2

-1

4) Denklemin çözümünün $x=1$, $y=2$ ve $z=-1$ olduğu anlaşılır.



MATRİSLER

Örnek:

$$3x-4y+z=14.85-4i$$

$$x+2y-3z=-10.55+2i$$

$$-2x-y+z=1.85-i$$

Denkleminde x, y ve z değişkenlerini bulunuz.



MATRİSLER

A = Katsayılar matrisi

$$\begin{bmatrix} 3 & -4 & 1 \\ 1 & 2 & -3 \\ -2 & -1 & 1 \end{bmatrix}$$

D = Değişkenler matrisi

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

B = Sonuç matrisi

$$\begin{bmatrix} 14.8500 - 4.0000i \\ -10.5500 + 2.0000i \\ 1.8500 - 1.0000i \end{bmatrix}$$

$$D = \text{inv}(A) * B$$

D =

$$\begin{bmatrix} 1.4000 \\ -2.0000 + 1.0000i \\ 2.6500 \end{bmatrix}$$

ÖRNEKLER

π

- $x^3 - 4.2x^2 + 3.3x - 4$ fonksiyonunun kök ya da köklerini bulunuz.

```
>> x=[-5:0.5:5];
```

```
>> t=x.^3-4.2*x.^2+3.3.*x-4
```

```
t =
```

```
Columns 1 through 10
```

```
-250.50 -195.03 -148.40 -109.88 -78.70 -54.13 -35.40 -21.77 -
```

```
12.50 -6.83
```

```
Columns 11 through 20
```

```
-4.00 -3.28 -3.90 -5.13 -6.20 -6.38 -4.90 -1.03 6.00
```

```
16.93
```

```
Column 21
```

```
32.50
```

```
>> x
```

```
x =
```

```
Columns 1 through 10
```

```
-5.00 -4.50 -4.00 -3.50 -3.00 -2.50 -2.00 -1.50 -1.00 -
```

```
0.50
```

```
Columns 11 through 20
```

```
0 0.50 1.00 1.50 2.00 2.50 3.00 3.50 4.00 4.50
```

```
Column 21
```

```
5.00
```

Sıfır geçişinin
olduğu aralık

ÖRNEKLER

1x5001 double

	909	910	911	912	913	914	915	916	917
1	-0.0053	-0.0041	-0.0029	-0.0017	-5.6158e-04	6.2090e-04	0.0018	0.0030	0.0042
2									

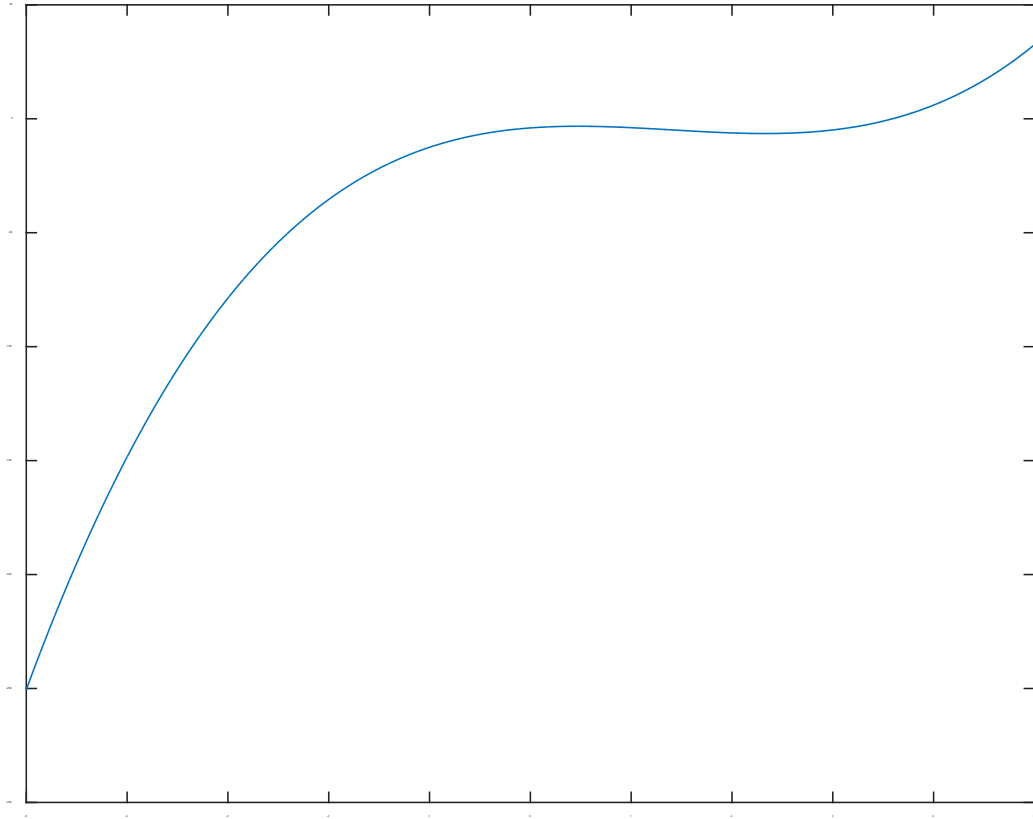
1x5001 double

	909	910	911	912	913	914	915	916	917
1	3.5908	3.5909	3.5910	3.5911	3.5912	3.5913	3.5914	3.5915	3.5916
2									

Kök: 3.5912

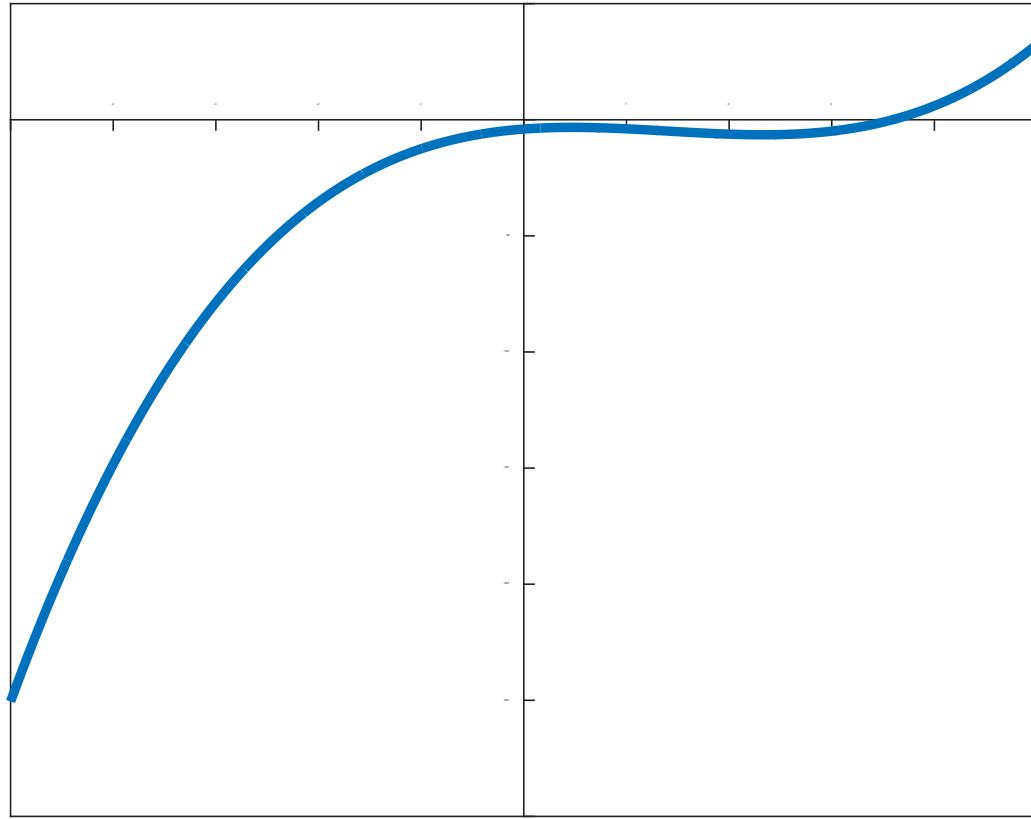
π

ÖRNEKLER



π

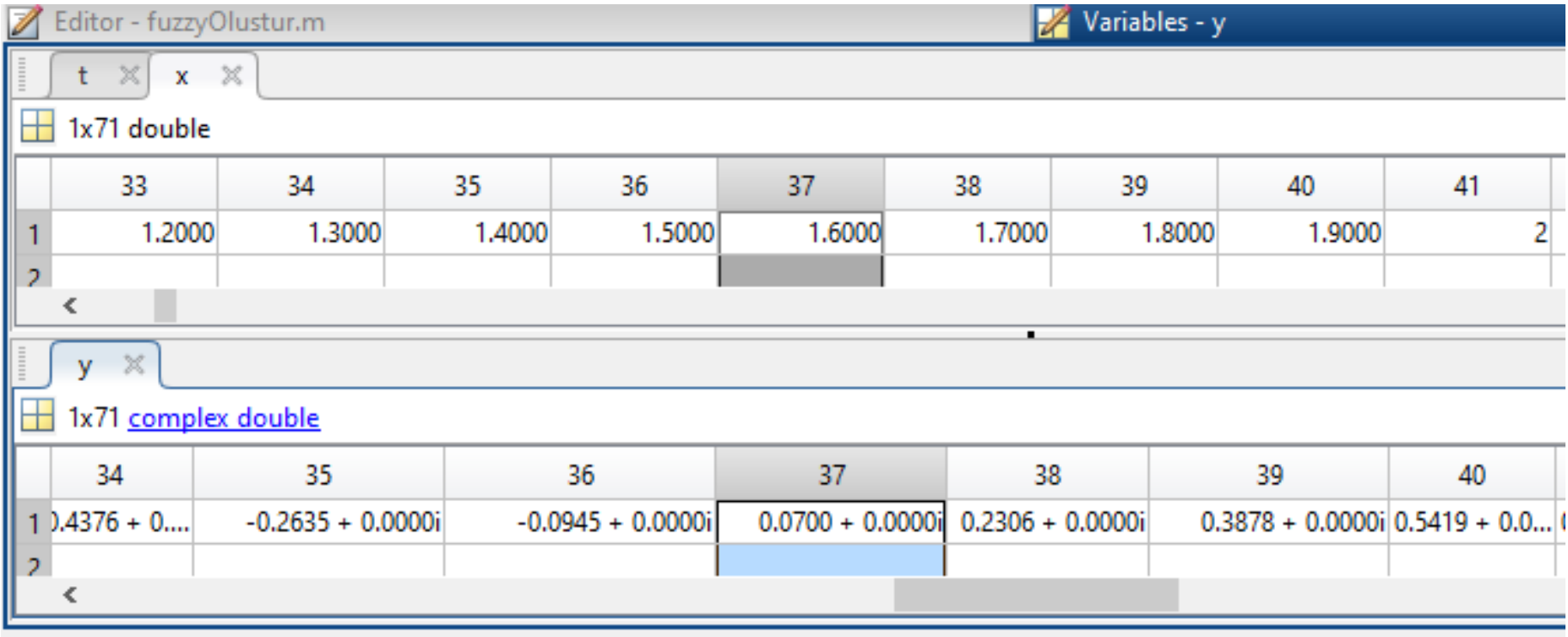
ÖRNEKLER

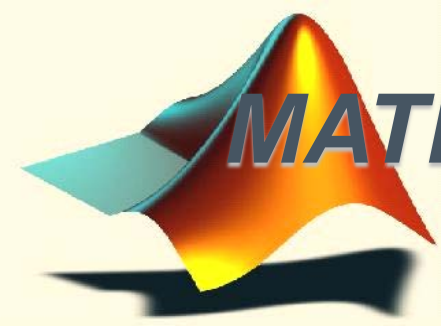


ÖRNEKLER

$y=x-2+\ln(x)$ fonksiyonunun köklerini bulunuz.

$x=[-2:0.1:2]$; aralığında





MATLAB ORTAMINDA KULLANILAN KOMUTLAR

fprintf ('çıkışta gösterilmesi istenen ifade', 'ifadenin gösterilme biçimi', değişken listesi)

'Çıkışta gösterilmesi istenen ifade': Tırnak içine hangi ifade yazılırsa ekranda bu ifade gözüktür.

'ifadenin gösterilme biçimi': Burada, ifadenin MATLAB ortamında aldığı değer hangi formatta ekrana yazılacağını belirten kodlar kullanılır.

Bu kısımda % işaretinin arkasından hassasiyet belirleme ve dönüşüm (c,d,e,E,f,F,g,G, i,o,s,u,x,X) karakterlerinden biri kullanılır.

MATLAB ORTAMINDA KULLANILAN VERİ TÜRLEİ

Örneđin s; deđişken olarak 'karakter' seçildiđi zaman kullanılır. Bu karakterlerin bazılarının ne ifade ettiđi aşıđıdaki örneklerde gösterilmiştir. Diđer karakterler ile ilgili bilgi ise etkileşimli yardım penceresinden öğrenilebilir. Daha sonra kullanılan \n,\r,\t,\b,\f gibi tanımlar ile de (sırası ile); satırbaşı, yeni satır, sekme, geriye doğru silme ve yeni sayfa işlemi gerçekleştirilebilir.

MATLAB ORTAMINDA KULLANILAN VERİ TÜRLERİ

fprintf

%a.bf : f ifadesi sayının **sabit noktalı** olduğu anlamına gelir (default olarak short e formatındadır). f sayısı gösterim olarak 'a.b' gibi iki sayıdan meydana gelir. 'a' sayısı değişkenin aldığı değer in işareti, tam kısmı ve varsa noktayı da dahil ederek noktadan sonraki rakam sayısını içerir. 'b' sayısı ise sabit noktalı sayının noktadan sonra ekrana yazılması istenen rakam sayısını gösterir. Eğer tamsayı için kullanılacak rakam sayısı 'a', olması gerekenden az olsa bile ekrana yazılan sonuç hatalı olarak ortaya çıkmaz.

```
>> sicaklik=1056.789432;
('enter')

>>fprintf('sicaklik=%4.0fderece',sicaklik)
('enter')

sicaklik=1057derece

>>fprintf('sicaklik = %4.1f derece',sicaklik) ('enter')

sicaklik = 1056.8 derece
```

MATLAB ORTAMINDA KULLANILAN VERİ TÜRLERİ

fprintf

Örnek olarak 'Ahmet Sadık' adındaki bir öğrencinin adı, soyadı, okul numarası, ilgili dersi, bu dersten aldığı not alta alta ekrana yazdırılsın. Bunun için fprintf komutundan yararlanılabilir;

```
>>Ad='Ahmet';                               ('enter')
>>Soyad='Sadık';                             ('enter')
>>Numara='307';                              ('enter')
>>Ders='Matematik';                          ('enter')
>>Not=45.75;
('enter')
>>fprintf('Ad:%s\nSoyad: %s\nNumara:%s\nDers: %s\nNot:%3.1f',...
          Ad,Soyad,Numara,Ders,Not)          ('enter')
```

Ad:Ahmet

Soyad: Sadık

Numara:307

Ders: Matematik

Not:45.8

İLİŞKİSEL OPERATÖRLER

- › **8. KONTROL AKIŞ YAPILARI**
- › MATLAB bir programlama dilidir. Bu nedenle diğer programla dillerindeki temel yapılar benzer şekilde kullanılmaktadır. if-else-end, switch, for, while, continue ve break yapılarını burada da göreceğiz.

- › **İlişkisel ve mantıksal işlemler**

- › MATLAB diğer programlar gibi ilişkisel imleçleri kullanmaktadır.

- ›
- › < Küçük
- › <= Küçük ya da eşit
- › > Büyük
- › >= Büyük ya da eşit
- › == Eşit
- › ~= Eşit değil

Operatörler:

- & and operatörü
- | or operatörü
- ~ not operatörü

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

IF DEYİMİ

if deyimi, Matlab dilinde **sartli dallanma(conditional branching)** adi verilen islemi gerçekleştiren bir deyimdir. Sartli dallanma, herhangi bir programlama dili için temel kontrol yapısıdır. Sartli dallanma islemi sayesinde, bir program, kararlar alma imkanına kavuşur; bir ifadenin sonucuna göre, bir komutlar dizisinin icra edilip edilmeyeceğine karar verebilir. Ifadenin değeri, bir icradan diğerine değişebileceği için, bu özellik bir programa farklı verilere karşı farklı şekillerde davranma imkanı sağlar. Matlab dilinde sartli dallanma if ve else anahtar sözcükleri ile gerçekleştirilir. if deyiminin en basit şekli için yazılış biçimi aşağıdaki gibidir:

if ifade

```
deyim1;
```

end

```
deyim2;
```

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

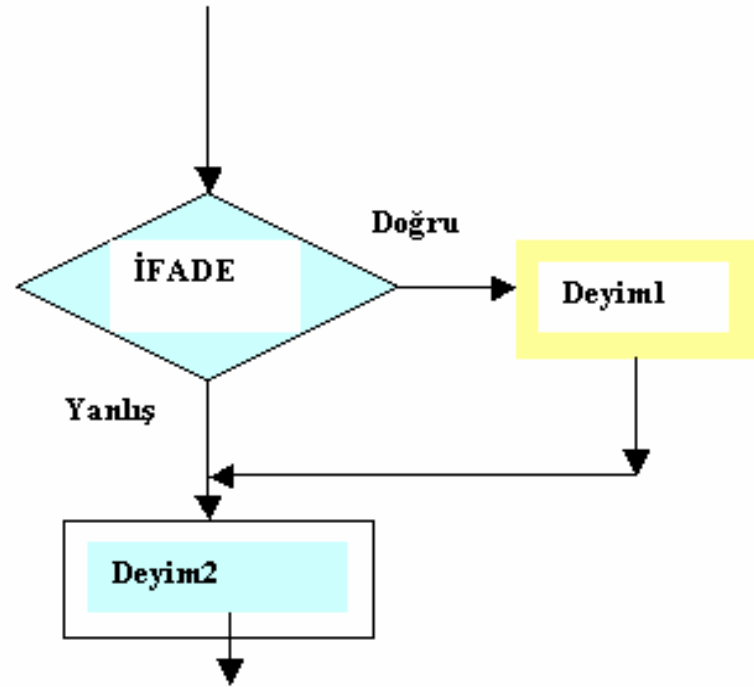
Burada ifadenin degeri dogru(true) ise deyim1 icra edilir sonra icra deyim2'ye geçer.Ifadenin degeri yanlis(false) ise bu durumda da dogrudan deyim2'ye geçilir. Ifadenin degeri yanlis ise deyim1 icra edilmeyecektir.Asagida if yapisini açıklayan bir akis diyagrami verilmistir.

if ifade

deyim1;

end

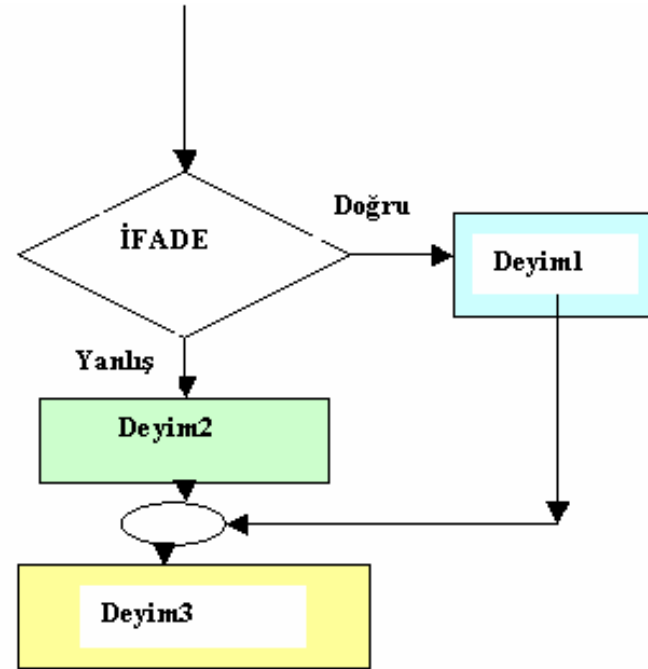
deyim2;



SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

If deyimi else sözcüğü ile birlikte kullanılırsa aşağıdaki yazılış biçimi kullanılır:

```
if ifade  
deyim1 ;  
else  
deyim2;  
end  
deyim3;  
...
```



if else deyimi için akış diyagramı(flow chart)

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

- › if-else-end
- ›
 - » A=[1 2; -3 6];
 - » if det(A)>0
 - Ainv=inv(A);
 - disp(Ainv)
 - end
- › En basit gösterim şekli;
- ›
- › **if expression**
- › **commands**
- › ...
- › **end**
- ›
- › Örneğin;
- ›
 - 0.5000 -0.1667
 - 0.2500 0.0833

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

```
r = round(-10 + (20+10).*rand(1,8))
```

```
r = [ 5  7 -3  4 19  6  6 -3]
```

Dizinin ortalaması pozitif ise pozitif dizi yazan ifadeyi yazalım.

```
if (mean(r)>0)
    fprintf('Pozitif Dizi')
End
```

Else yapısı ekleyelim.

```
r = round(-10 + (-1+10).*rand(1,8))
```

```
if (mean(r)>0)
    fprintf('Pozitif Dizi')
else
    fprintf('Negatif Dizi')
end
```


SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

- › if-else-end yapısının en genel hali aşağıdaki gibidir.
- ›
- › **if expression-1**
- › **commands-1**
- › **elseif expression-2**
- › **commands-2**
- › .
- › .
- › .
- › **elseif expression-(n-1)**
- › **commands-(n-1)**
- › **else**
- › **commands-n**
- › **end**
- ›

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

```
a=input('a değerini giriniz:');

if a == 10
    % IF şartı doğruysa aşağıdakini yaz
    fprintf('Girilen değer= 10\n' );
elseif( a == 20 )
    % ELSEIF şartı doğruysa aşağıdakini yaz
    fprintf('Girilen değer= 20\n' );
elseif a == 30
    % ELSEIF şartı doğruysa aşağıdakini yaz
    fprintf('Girilen değer= 30\n' );
else
    % Hiçbir şart doğru değil ise '
    fprintf('Hiçbir değer eşleşmedi\n');
fprintf('Girilen değer: %d\n', a );
end
```

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

- › switch
- › Bir seçim söz konusu olduğunda bu yapı kullanılabilir. Genel gösterimi:
- ›
- › **switch (selector)**
- › **case label-1**
- › **commands-1**
- › **case label-2**
- › **commands-2**
- › .
- › .
- › .
- › **case label-n**
- › **commands-n**
- › **otherwise**
- › **commands-m**
- › **end**

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

› Örnek:

›

› `>> a=[4 5;2 3];`

› `>> switch (det(a))`

› `case 1`

› `b=a';`

› `disp(b)`

› `case 2`

› `b=a*a;`

› `disp(b)`

› `end`

›

› a matrisi determinanı 2 ye eşit olduğu için çıktısı aşağıdaki gibi olacaktır.

›

› `26 35`

› `14 19`

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

- › **for döngüsü:**
 - › Bir grup bilginin birkaç defa değerlendirilmesi
 - › gerekiyorsa bu yapı kullanılır.
 - › **for x=array**
 - › **commands**
 - › **end**
 - › Komutlar x'in bütün değerleri için işler. Örneğin;
 - ›
 - › **» k=1;**
 - › **» for num=[6 37 23 -1]**
 - › **disp([num2str(k), ' inci elementi ', num2str(num)])**
 - › **k=k+1;**
 - › **end**
- çıktısı;
- 1 inci elementi 6
2 inci elementi 37
3 inci elementi 23
4 inci elementi -1

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

Rastgele bir dizideki pozitif ve negatif sayıları yazdırınız.

```
r = round(-10 + (20+10).*rand(1,20)) % random neaktif, pozitif sayılar
i=1;
for i=1:length(r)
    if (r(i)>0)
        fprintf('r(%d) pozitif. Deđeri=%d\n',i,r(i));
    else
        fprintf('r(%d) negatif. Deđeri=%d\n',i,r(i));
    end
end
end
```

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

Dizinin maksimumunu bulma örneği:

For ile çözüm.

```
rnd= round(-10 + (20+10).*rand(1,20));
maks=rnd(1);
j=1;
for i=1:20
    if (rnd(i)>maks)
        maks=rnd(i);
        j=j+1;
    end
end
fprintf('maksimum=%4.2f\n\n',maks);
fprintf('indis=%d\n',j)
```

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

- › **while döngüleri**
- › Genel gösterimi;
- › **while expression**
- › **commands**
- ›
- › **end**
- ›
- › şeklindedir.

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

Dizinin maksimumunu bulma örneği:

While ile çözüm.

```
rnd= round(-10 + (20+10).*rand(1,20));
maks=rnd(1);
j=1;i=1;
while (i<21)
    if (rnd(i)>maks)
        maks=rnd(i);
        j=j+1;
    end
    i=i+1;
end
fprintf('maksimum=%4.2f\n\n',maks);
fprintf('indis=%d\n',j)
```

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

- › **break** komutu
- › break yapısı döngünün dışına ulaşmak için kullanılır.
- ›
- › » `y=zeros(1,2);`
- › » `x=[2 3];`
- › » **while** `x>=0`
- › `y=y+x;`
- › `x=x-1;`
- › **if** `y>=3`
- › `break`
- › `end`
- › `disp(y)`
- › `end`

çıktısı;

2 3

(3, 5) vektörü görünmedi çünkü `y>=3` ifadesi doğru bir ifade.

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

› continue

›

çıktısı;

› Bu yapı kontrol durumunu geçip sıradaki

› iterasyona ulaşmak için for ve while döngülerinde kullanılır.

3 5

›

3 6

› >> y=zeros(1,2);

› >> x=[2 3];

› >> while x>=0

› y=y+x;

› x=x-1;

› if y<=3

› continue

› end

› disp(y)

› end

(2, 3) vektörü görüntülenmedi. $y \leq 3$ ifadesi doğru olduğu için sıradaki iterasyonu atladı.

27.2.2017

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

For ve İf örnekleri

$X^3 - 4.2x^2 + 3.3x - 4$ fonksiyonunun köklerini bulunuz.

```
tic
for x=-5:0.0001:5
    t=x.^3-4.2*x.^2+3.3.*x -4;
    %   if(t==0)
    if (t>=-0.001 && t<=0.001)
        fprintf('Kök=%2.5f',t);
        fprintf(' değer=%2.5f\n',x);
    %       disp(t);
    end
end
plot(x,t,'LineWidth',3)
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
toc
```

SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

$$f(x) = 2x - \cos x$$

Fonksiyonun köklerini döngüler kullanarak bulunuz.

```
tic
clc
clear
for x=-5:0.0001:5
    t=2*x-cos(x);

    if (t>=-0.0001 && t<=0.0001)
        fprintf('Kök=%2.5f',x);
        fprintf(' deđer=%2.5f\n',t);
    end
end
```

```
toc
```

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

1'den 999'a kadar olan sayılardan asal sayı olanların toplamını bulunuz.

1. yöntem:

```
total = 0;
for k = 1:999
    if(isprime(k))
        total = total + k;
    end
end
disp(total)
```

2. yöntem

```
total = 0;
for k = primes(999)
    total = total + k;
end
disp(total)
```

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

1'den 200'e kadar olan sayılardan kendisi ve 2 fazlası asal olan sayıları bulunuz.

```
for x = 1:2:200
if(isprime(x) & isprime(x+2))
fprintf('%.0f ve %.0f ikisi de asal sayıdır\n',x,x+2)
end
end
```

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

İç içe geçmiş döngüler (Nested Loops)

```
for x = 1:3
    for y = 1:2
        fprintf('x= %.0f and y= %.0f\n',x,y)
    end
end
```

```
x= 1 and y= 1
x= 1 and y= 2
x= 2 and y= 1
x= 2 and y= 2
x= 3 and y= 1
x= 3 and y= 2
```


SEÇME(SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

Bir matrisi satır vektöre dönüştüren döngüyü oluşturunuz.

```
[m n] = size(A);  
k=1;  
for i=1:m  
    for j=1:n  
        v(k) = A(i,j);  
        k = k+1;  
    end  
end
```

SEÇME (SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

CELL ARRAY

Hem nümerik hem de nümerik olmayan değerler tek bir dizi ya da matriste ifade edilebilir mi?

Bazı durumlarda nümerik ve string ifadeler aynı dizi ya da matriste bulunabilmelidir.

```
w=[1 4 'aku' 5.3; 'afyon' -2.6 teknoloji' 1e2 ]
```

```
w={1 4 'aku', 5.3; 'afyon' -2.6 'teknoloji' ' 1e2 }
```

```
>>w(1,:)
```

```
>>isnumeric(w(1,1))
```

```
>>cell2mat(w(1,1))
```

```
>>[a b c ]=w{1,1:3}
```

M-Fonksiyon Yapısı

M-Fonksiyonlar kullanılırken dikkat edilecek hususlar:

- 1- Kullanıcılar kendi fonksiyonlarını yazmak için m-fonksiyonlarını kullanabilirler.
- 2- **Function** alt programı ve ana program şeklinde iki program yazılarak bu iki program ayrı ayrı kaydedilir.
- 3- Alt programdaki *fonksiyon_adi*, m-dosyasına verilen isimle aynı olmalıdır.
- 4- Ana programdan alt program, function adı kullanılarak çağrılır.
- 5- Alt programdan da ana programa geçiş yapılabilir fakat genelde tercih edileni tersidir.
- 6- Parametre aktarımı olması durumunda alt ve ana programda eşit sayıda parametre ve giriş değişkeni olmalıdır.

function cikis_ifadesi1, 2,..., n =*fonksiyon_adi* (giris_ifadesi1, 2, ...n)

Örnek: İki nokta arasındaki uzaklığı bulan programı m-fonksiyon (alt program) kullanarak yazınız.

x1=1.noktanın x koordinati; x2=2.noktanın x koordinati

y1=1.noktanın y koordinati; y2=2.noktanın y koordinati

FUNCTION ALT PROGRAMI (uzak.m):

```
function uzaklik =uzak(x1,y1,x2,y2)
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);
```

Bu function alt programı uzak.m olarak kaydedilir.

ANA PROGRAM:

```
ax=3; ay=4; bx=1; by=2;
```

```
uzaklik = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor
```

```
fprintf('iki nokta arasındaki uzaklık=%f',uzaklik);
```

Adım adım gerçekleştirilen işlemler:

- Ana program herhangi bir isimle kaydedilir ve koşturulur.
- Program, **function** adına (**uzak**) geldiği zaman alt program çağrılır ve ax, ay, bx, by parametreleri sırasıyla x1, y1, x2, y2 giriş değişkenlerine aktarılır.
- **Function** alt programında hesaplama gerçekleştirilir.
- **Function**'daki çıkış değişkeni olan **uzaklik** hem alt programda hem de ana programda hesaplanan sonuç değerinin aktarıldığı değişken olarak kullanılır.
- Alt programdan ana programa parametre aktarımı zorunlu değildir. İstenirse değişkenlerin değerleri alt programda da girilebilir ve sonuç alt programda yazdırılabilir.

Uygulama: Yukarıdaki örneği ana programdan alt programa parametre aktarımı yapmadan yeniden yazınız.

(Değişkenlerin girilmesi, sonucu hesaplama ve yazdırma işlemi alt programda yapılacaktır)

FUNCTION ALT PROGRAMI:

```
function uzaklik = uzak(x1,y1,x2,y2)
x1=3; y1=4; x2=1; y2=2;
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);
fprintf('iki nokta arasindaki uzaklik=%f', uzaklik);
```

ANA PROGRAM:

```
uzaklik = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor
```



FONKSİYONLAR

Rastgele sayı üreten bir fonksiyon yazınız.

```
function r = RastgeleSayi( altSinir, ustSinir, satir,  
sutun )  
%UNTÍTLED Summary of this function goes here  
% Detailed explanation goes here  
r = round(altSinir + (ustSinir-  
altSinir).*rand(satir,sutun)); % random neagtif, pozitif  
sayýlar  
  
end
```

SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

Bir dizinin maksimum ve minimumunu bulan bir fonksiyon yazınız.

```
function [ maks, minimum ] = MaxMinBul( vektor )
```

```
maks=max(vektor);  
minimum=min(vektor);  
end
```

```
>> j=RastgeleSayi(-40,50,1,20);  
>> [maksimum, minimum]=MaxMinBul(j)
```

```
maksimum =
```

```
45
```

```
minimum =
```

```
-29
```


SEÇME(SELECTION) TIPİNDEKİ KONTROL DEYİMLERİ

Bir vektör ya da matrisi [0-1] arasına normalize eden bir fonksiyon yazınız.

```
function y = NormalizeEt01(x)
%b alt sýnýr, a ust sýnýr
[satir,sutun]=size(x);
i=1; j=1;
for i=1:satir
    for j=1:sutun
        y(i,j)=[(x(i,j)-min(x(:)))]/[max(x(:))-min(x(:))];
    end
end
```