

# BIL301 SAYISAL YÖNTEMLER

## 2. Hafta

### Sayısal Yöntemler İçin Matlab

Doç. Dr. Sercan YALÇIN

## 2) Sayısal Hesaplamalarda Gerek Duyulabilecek Matlab İşlemleri

### •2.1. Matlab Temel Komutları

- *Vektörler:*
- `a = [1 2 3 4 5 6 7 8 9]`
- Program sonucu aşağıdaki gibi döndürür
- `a =`
- `1 2 3 4 5 6 7 8 9`

- 0 ile 20 arasında 2'şer 2'şer artan elemanlardan oluşan bir vektör oluşturalım.

t = 0:2:20

(Enter)

t =

0 2 4 6 8 10 12 14 16 18 20

- Şimdi de a kümesinin her elemanına 2 ekleyelim;

b = a+2

Sonuç;

b =

3 4 5 6 7 8 9 10 11

- İki vektörü toplayalım;

$$c=a+b$$

Sonuç;

$$c=$$

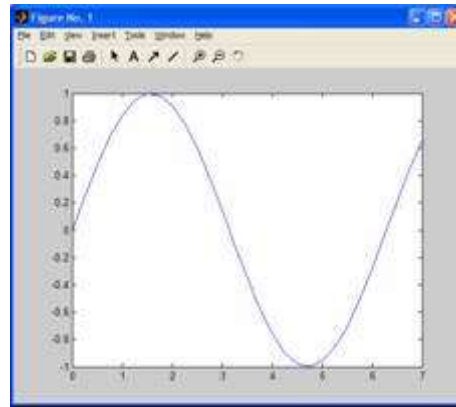
4 6 8 10 12 14 16 18 20

•*Fonksiyonlar*: Matlab, sin, cos, log, exp, sqrt gibi pek çok fonksiyonu içerir. pi gibi sabitler, -1'in karekökünü simgeleyen i veya j gibi sabitler de bulunmaktadır.

$\sin(\pi/4)$

ans=

0,7071



•*Grafik Çizdirme*: `plot(x,y)`

`t =0:0.25:7;`

`y=sin(t);`

`plot(t,y)`

*Polinomlar:*

$x=s^2+3s^3-15s^2-2s+9$  polinomu programa aşağıdaki şekilde yazılır;

$x=[1 \ 3 \ -15 \ -2 \ 9]$

$x =$

$1 \ 3 \ -15 \ -2 \ 9$

Benzer şekilde  $y=s^4+1$ 'in gösterilimi  $y=[1 \ 0 \ 0 \ 0 \ 1]$  şeklindedir.

Polinomun herhangi bir kök için değeri, örneğin  $s^4+1$ 'in  $s=2$  için değeri;

$z=\text{polyval}([1 \ 0 \ 0 \ 0 \ 1],2)$  veya doğrudan  $z=\text{polyval}(y,2)$

$z = \quad 17$

- Polinomun köklerinin bulunması, örneğin

- 

- $s^4+3s^3-15s^2-2s+9$

için;

`roots([1 3 -15 -2 9])`

İki polinomun çarpılması,  $(x+8) (x^2+4x+8) =$   
 $x^3+6x^2+16x+16$

`x=[1 2]`

`y=[1 4 8]`

`z=conv(x,y)`

İki polinomu bölelim  
[xx,R]=deconv(z,y)

*Matrisler:* Matrisler tıpkı vektörler gibi girilir, her satır birbirinden noktalı virgülle ayrılır

B=[1 2 3 4; 5 6 7 8; 9 10 11 12]

veya aşağıdaki şekilde yazılır;

B=[1 2 3 4  
5 6 7 8  
9 10 11 12 ]



Matrisin transpozesi

$$C=B'$$

- İki matrisin çarpımı

$$C = A*B$$

Sadece aynı indisli elemanları çarpmak için;

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

$$F=\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

$$G=E.*F$$

$$G=\begin{bmatrix} 2 & 6 \\ 12 & 20 \end{bmatrix}$$

Kare matrisin kuvveti

$E^3$

ans= 37 54  
81 118

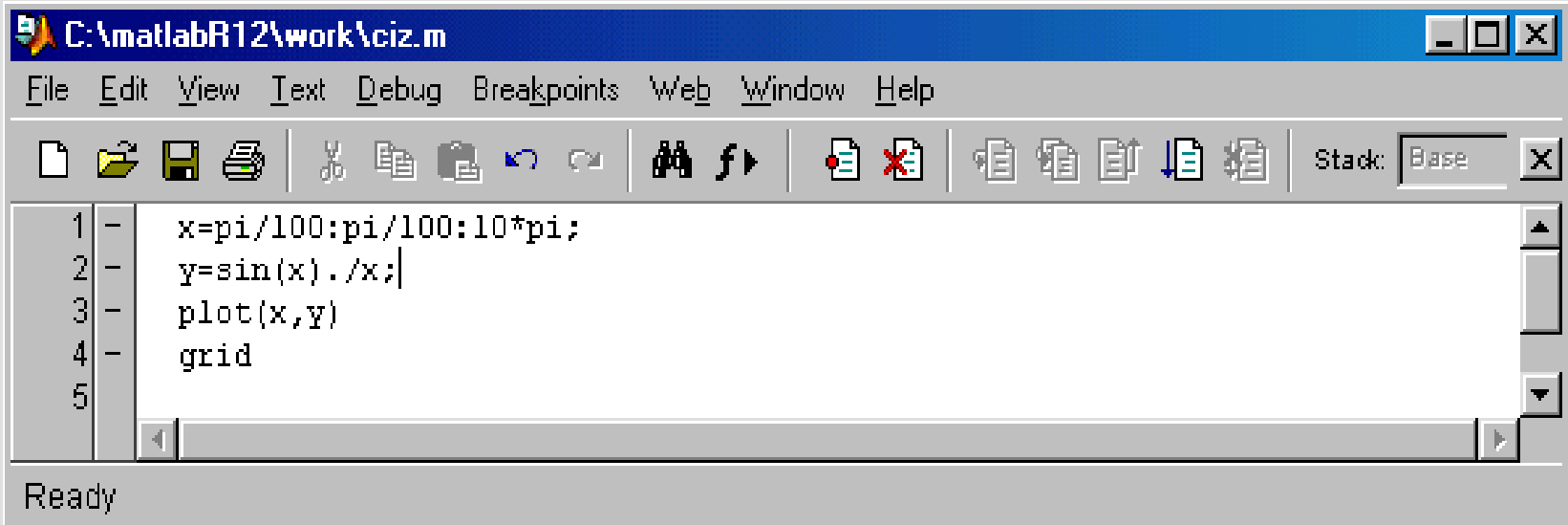
- Matrisin inversi;
- $x = \text{inv}(E)$
- $x = \begin{bmatrix} -2 & 1 \\ 1,5 & -0,5 \end{bmatrix}$

# Matlab'ta açı değerlerini girerken buna dikkat edin !

- Açılar derece değil radyan cinsinden hesaplanmaktadır. Bu nedenle örneğin 180 derecelik açının sinüsü  $\sin(180)$  değil  $\sin(\pi)$  yani  $\sin(3.14)$ 'tür.

## 2.2. Matlab'ta M-dosyalarını Hazırlamak ve Kullanmak:

- Örnek: Sinc işaretini çizdiren düzyazı dosyası. Dosyanın adı ciz.m olsun.

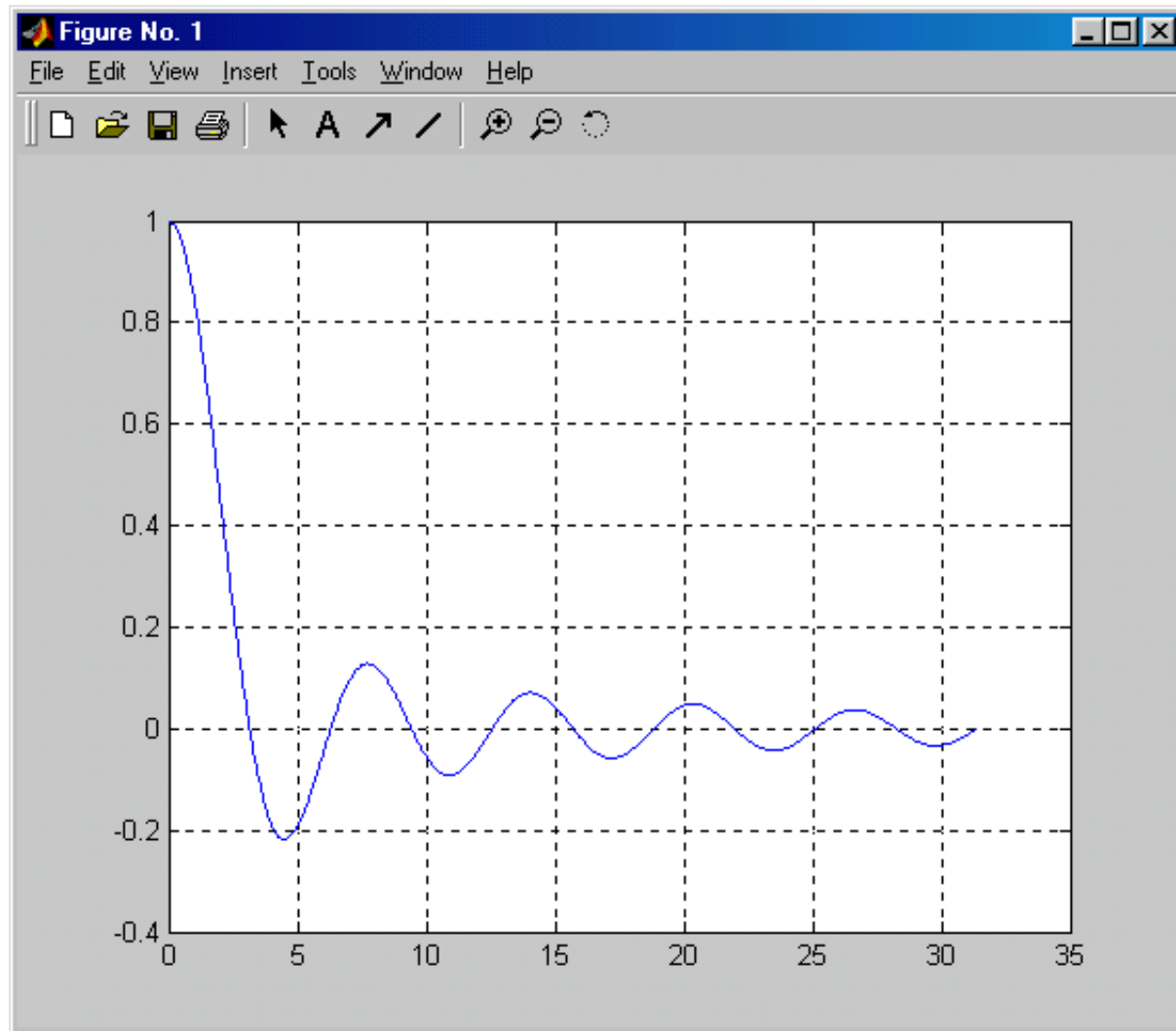


The screenshot shows the MATLAB R12 editor window with the file 'C:\matlabR12\work\ciz.m' open. The window has a menu bar (File, Edit, View, Text, Debug, Breakpoints, Web, Window, Help) and a toolbar with various icons for file operations and execution. The script content is as follows:

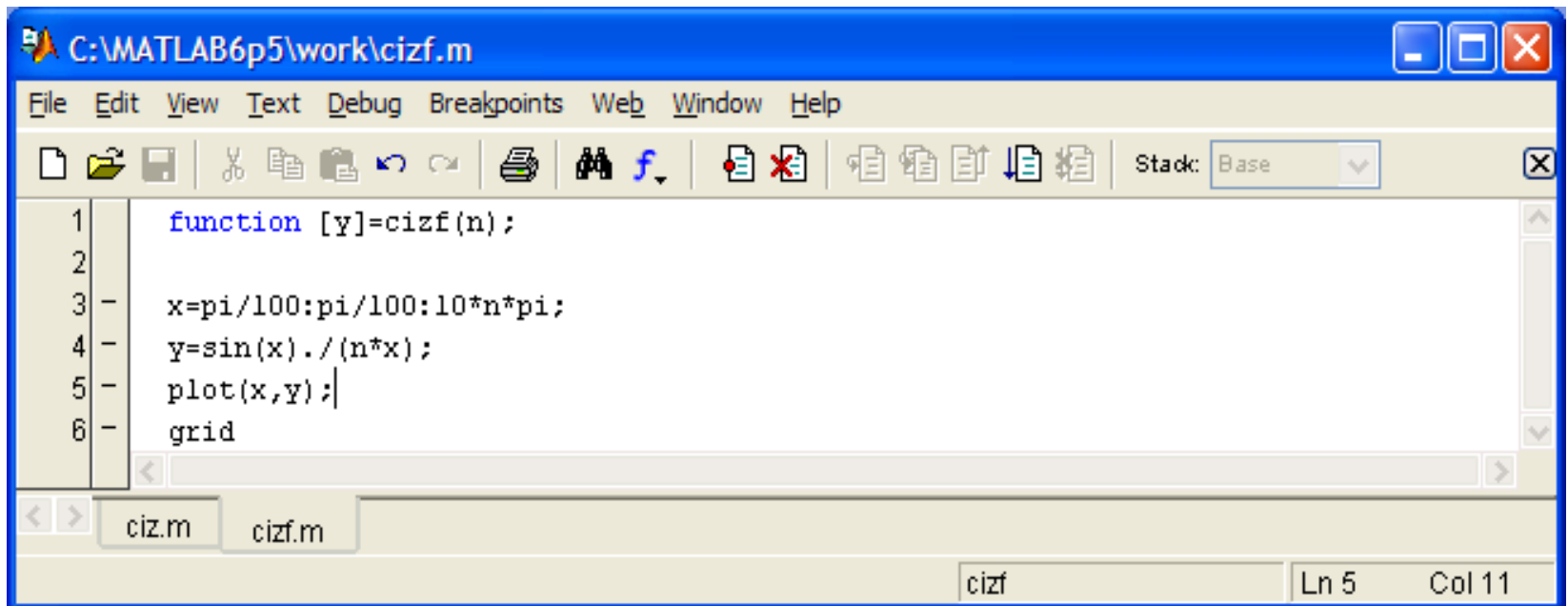
```
1 - x=pi/100:pi/100:10*pi;  
2 - y=sin(x)./x;  
3 - plot(x,y)  
4 - grid  
5
```

The status bar at the bottom indicates 'Ready'.

>> ciz



Örnek: Çizim yapacak olan m. dosyası, bir n değişkenine bağlı olarak farklı sonuçlar bulan cizf.m isimli bir fonksiyon olsun.



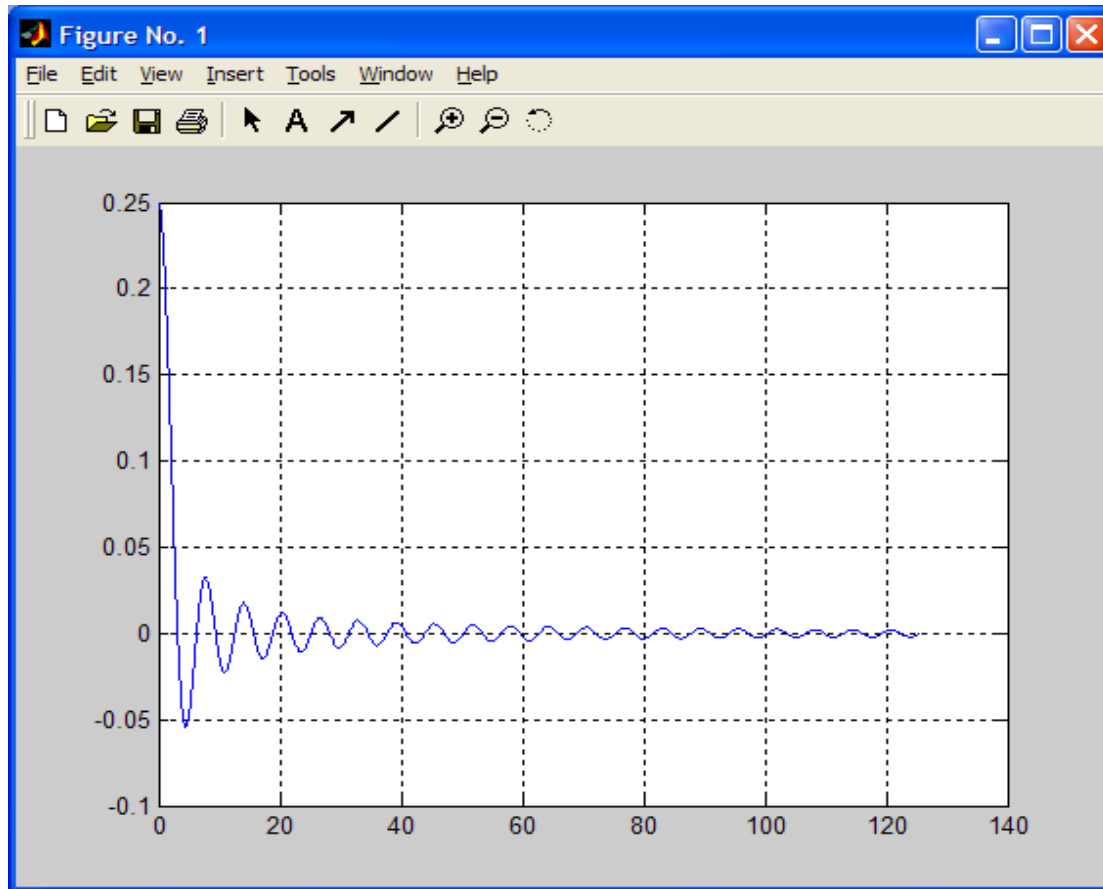
The image shows a MATLAB editor window titled 'C:\MATLAB6p5\work\cizf.m'. The window has a menu bar with 'File', 'Edit', 'View', 'Text', 'Debug', 'Breakpoints', 'Web', 'Window', and 'Help'. Below the menu bar is a toolbar with various icons for file operations, editing, and debugging. The main text area contains the following MATLAB code:

```
1 function [y]=cizf(n);  
2  
3 x=pi/100:pi/100:10*n*pi;  
4 y=sin(x)./(n*x);  
5 plot(x,y);  
6 grid
```

The status bar at the bottom shows 'cizf' and 'Ln 5 Col 11'.

Örneğin  $n=4$  için

```
>> cizf(4)
```



## 2.3. İşletmenler ( Operatörler):

### i) **Tablo.2.1.** Aritmetik İşletmenler

+	Toplama	.\	Sola doğru bölme
-	Çıkarma	:	Sütun oluşturma
.*	Çarpma	.^	Kuvvet alma
./	Sağa doğru bölme		

**Tablo.2.2.** İlişki İşletmenleri

<	...den küçük	>=	...den büyük veya eşit
<=	...den küçük veya eşit	=	...e eşit
>	...den büyük	~=	...e eşit değil



i) **Tablo.2.3.** Mantık İşletmenleri

$\&$	VE
$ $	VEYA
$\sim$	DEĞİL

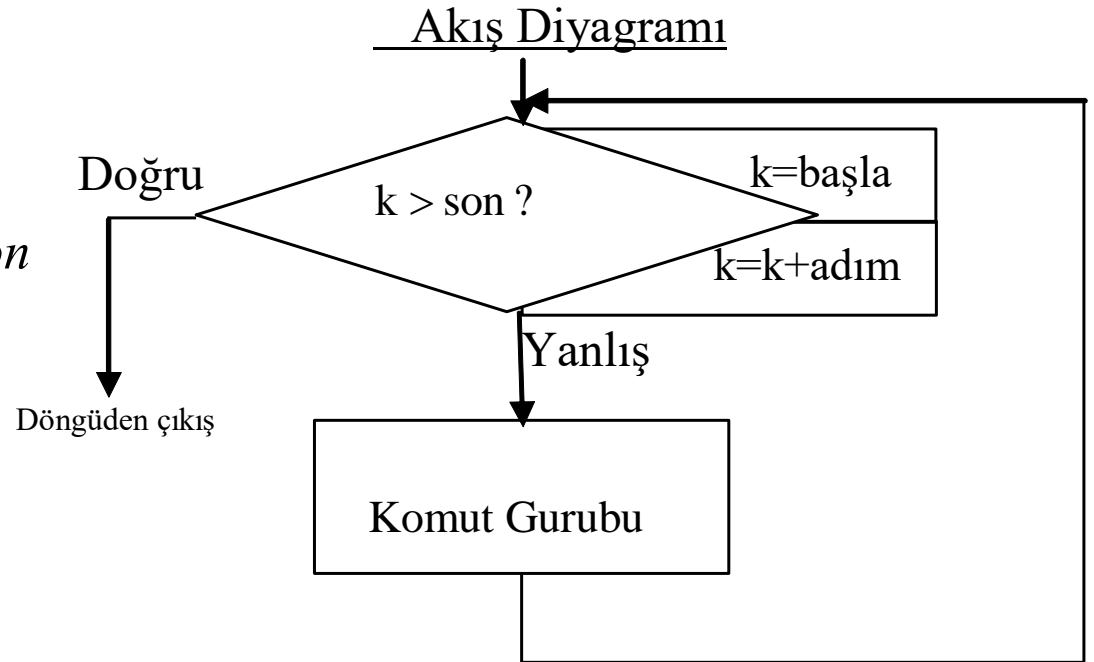
## 2.4. Komut Akışının Denetimi:

Programlama dillerinde akış denetimi döngüler (for,while) ve şartlı deyimler (if,switch..) ile sağlanır.

For döngüleriyle yineleme:

### Yazılış Şekli

```
for k=başlangıç:adım:son  
    komutlar  
end
```



## 2.4. Komut Akışının Denetimi:

- For döngüleriyle yineleme:

**Örnek.1)** Aşağıdaki program döngü içindeki hesaplamayı 5 kez yürütür.  $y$ 'nin ilk değeri 1 olsun.

```
y(1)=1  
for i=2:6  
     $y(i)=2*y(i-1);$   
end
```

## 2.4. Komut Akışının Denetimi:

- For döngüleriyle yineleme:

*Örnek.2) 3X4'lük bir matrisin elemanları şu formüle göre hesaplınsın;*

*m=3; n=4;*

*for i=1:m*

*for j=1:n*

*A(i,j)=1/(i+j-1)*

*end*

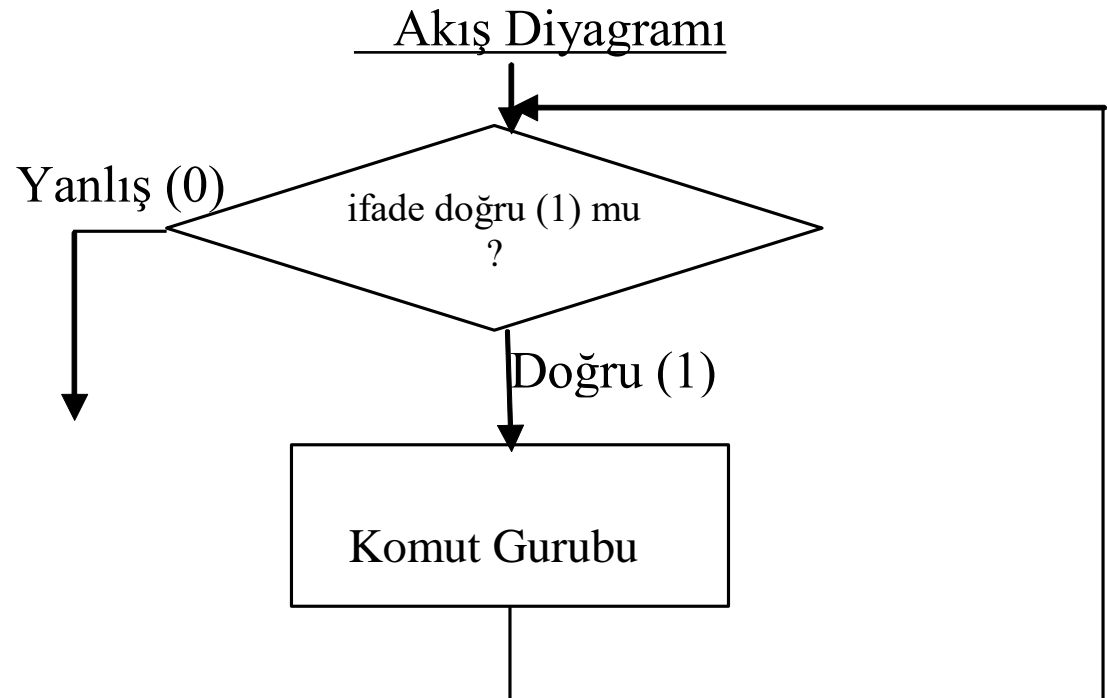
*end*

## 2.4. Komut Akışının Denetimi:

- While döngüleriyle yineleme:

### Yazılış Şekli

*while ifade  
komutlar  
end*



## 2.4. Komut Akışının Denetimi:

- While döngüleriyle yineleme:
- Örnek.1) Bu örnekte ilk değeri 3.14 olan bir  $q$  değişkeninin değeri, 0.01'den küçük olana kadar sürekli olarak yarılanır.
- $q=pi;$
- while  $q>0.01$
- $q=q/2$
- end
- Burada  $q$ 'nın sondan bir önceki değeri olan 0.0123, 0.01'den büyük olduğu için döngünün içindeki komut son kez yürütülmüş ve  $q$ , 0.0061 bulunmuştur.  $q$ 'nın son değeri 0.01'den küçük olduğundan döngü içindeki komut bundan sonra yürütülmemiştir

## 2.4. Komut Akışının Denetimi:

- if-elseif-else-end şartlı deyimleri
- **Temel if-else yapısı:** if ile başlayan bir koşul sağlanıyorsa, yani doğruysa altındaki komutlar yürütülür, sağlanmıyorsa, yani yanlışsa, else kelimesinin altındaki komutlar yürütülür.
- Yazılış Şekli
- if Koşul
- Komut Gurubu1
- else
- Komut Gurubu2
- end

## 2.4. Komut Akışının Denetimi:

- Çok seçenekli if-elseif-else yapısı:
- Eğer çok sayıda alternatif varsa bu yapı kullanılır

Yazılış Şekli

if Koşul1

Komut G-1

elseif Koşul2

Komut G-2

elseif Koşul3

Komut G-3

elseif.....

.....

.....

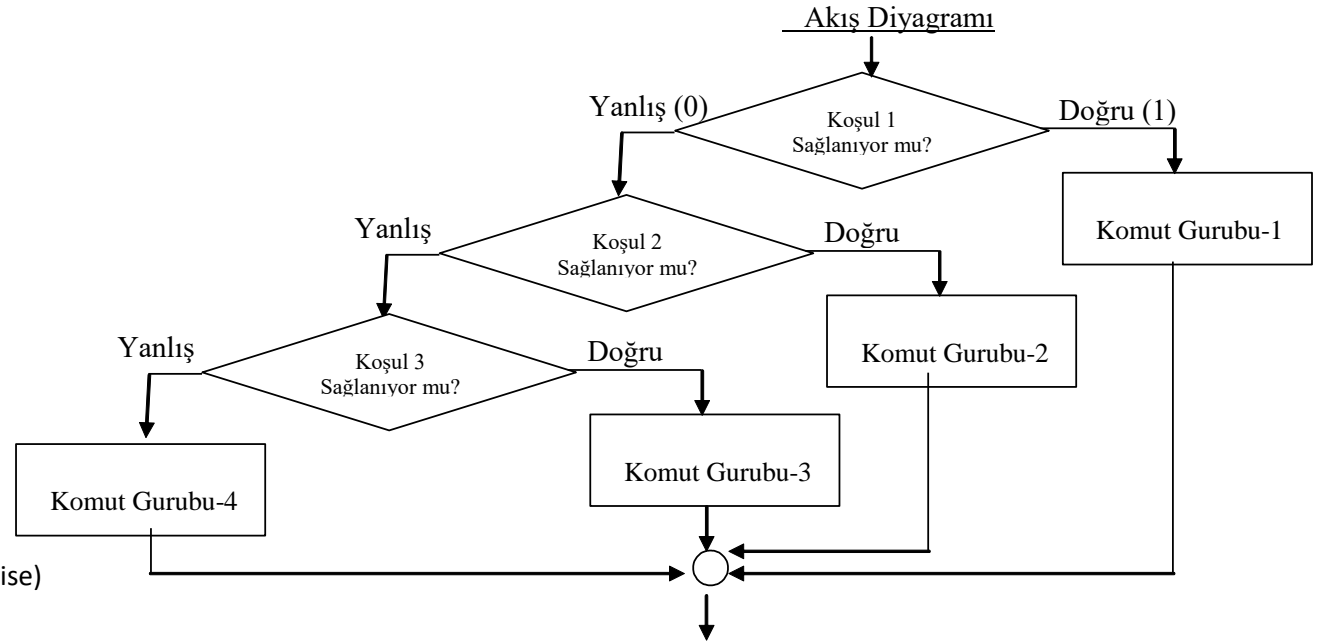
...

else

(Tüm diğer koşullar yanlış ise)

Komut G-Son

end

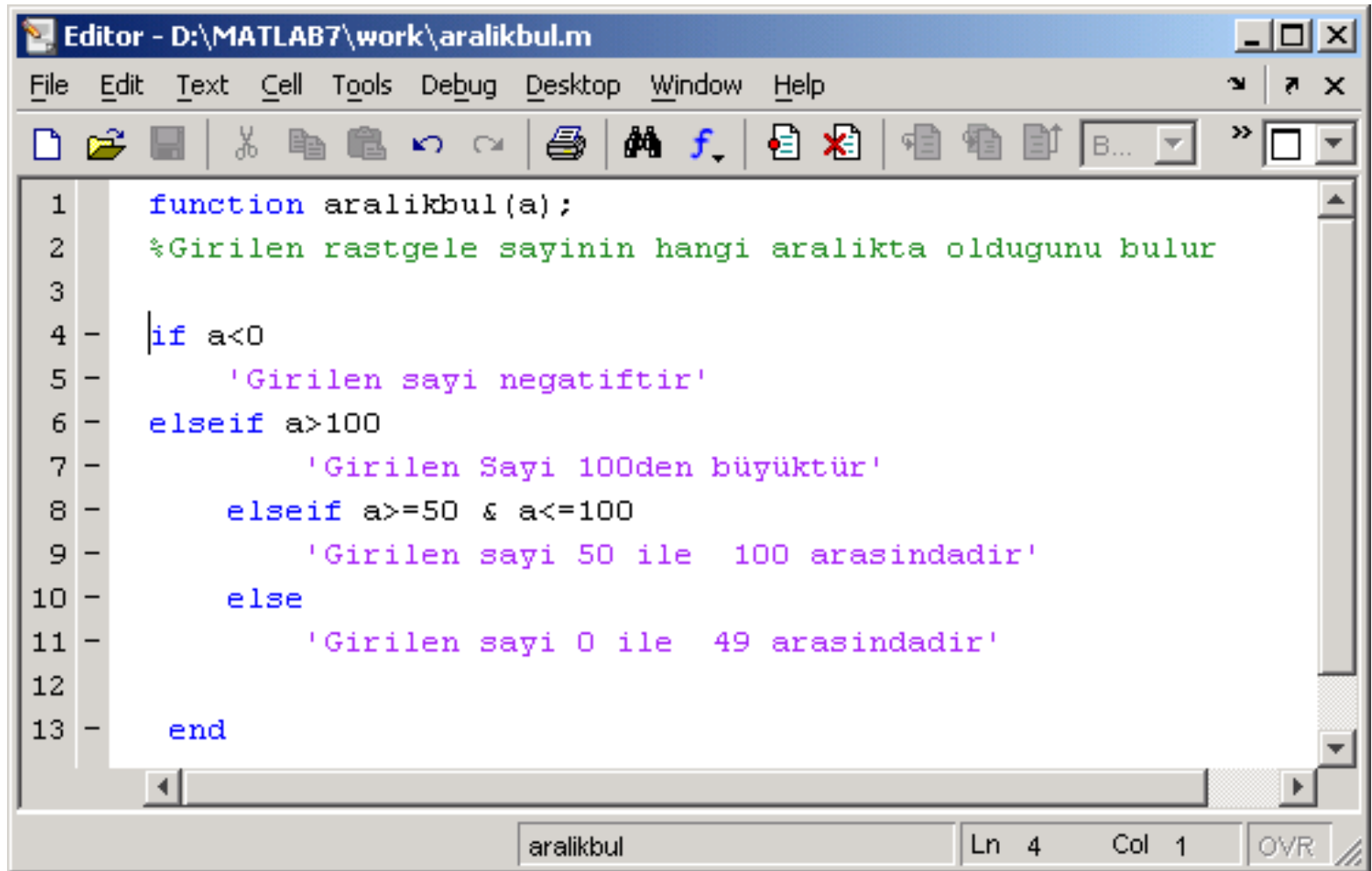




## 2.4. Komut Akışının Denetimi:

- Çok seçenekli if-elseif-else yapısı:
- **Örnek.1)** İstenen sayıda iç içe geçmiş if deyimleri ağı oluşturulabilir. Bu durumda her bir yeni if şartı bir end ile bitmelidir.
- `vize=100*rand(1)`
- `final=100*rand(1)`
- `if final>=50`
- `ort=0.4*vize+0.6*final`
- `if ort>=50`
- `disp('Gectiniz')`
- `else disp('ortalama yuzunden kaldiniz')`
- `end`
- `else disp('Final yuzunden kaldınız')`
- `end`

**Örnek.3.)** Çok seçenekli yapıya örnek olarak, girilen sayının hangi aralığa düştüğünü bulan bir fonksiyon tanımlayalım.



The image shows a MATLAB Editor window titled "Editor - D:\MATLAB7\work\aralikbul.m". The window contains a MATLAB function definition for "aralikbul(a)". The function checks the value of 'a' and returns a string indicating which range it falls into. The code is as follows:

```
1 function aralikbul(a);
2 %Girilen rastgele sayinin hangi aralikta oldugunu bulur
3
4 if a<0
5     'Girilen sayi negatiftir'
6 elseif a>100
7     'Girilen Sayi 100den büyüktür'
8 elseif a>=50 & a<=100
9     'Girilen sayi 50 ile 100 arasindadir'
10 else
11     'Girilen sayi 0 ile 49 arasindadir'
12
13 end
```

The status bar at the bottom indicates the file name "aralikbul", the current line "Ln 4", the current column "Col 1", and the view mode "OVR".

## 2.4. Komut Akışının Denetimi:

- **Switch-case yapıları** :Burada bir değişkenin durumuna göre bir önerme gurubundan uygun olanı seçilir. Başka bir deyişle bir değişken veya deyim aldığı değere bağlı olarak ilgili komutlar yürütülür.

### Yazılış Şekli

Switch *değişken değeri*

Case *değer1*

Komut Bloğu1

Case *değer2*

Komut Bloğu2

....

....

.....

.....

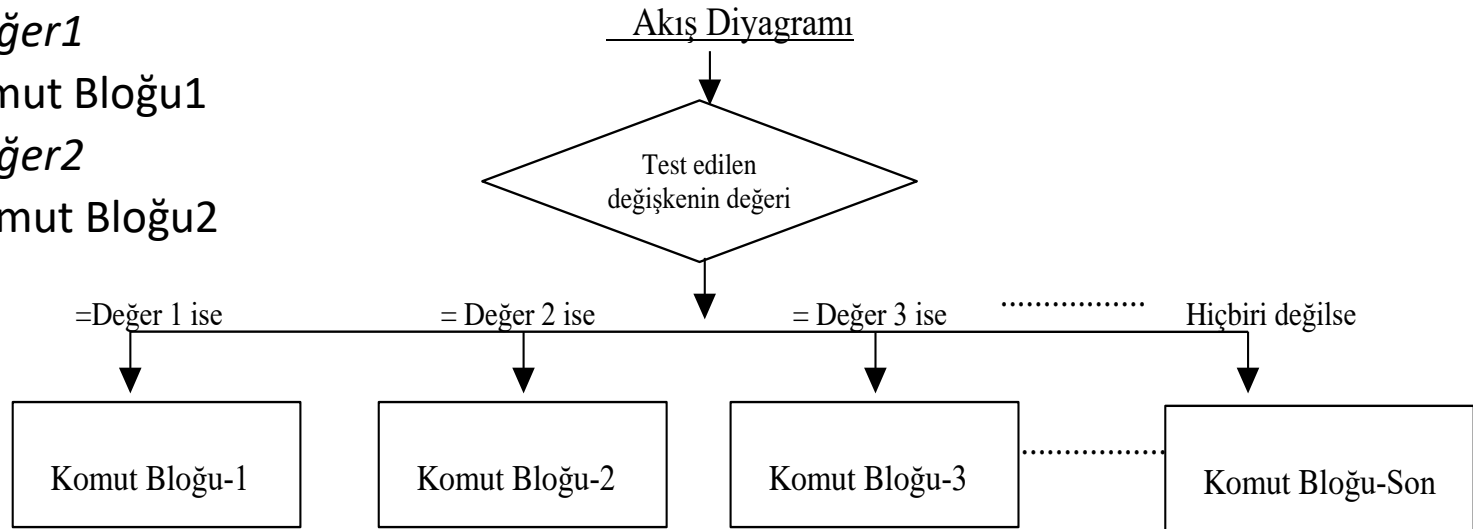
...

otherwise

{yani deyim hiçbir duruma (Case'e) denk düşmezse}

Komut Bloğu-Son

end



## 2.4. Komut Akışının Denetimi:

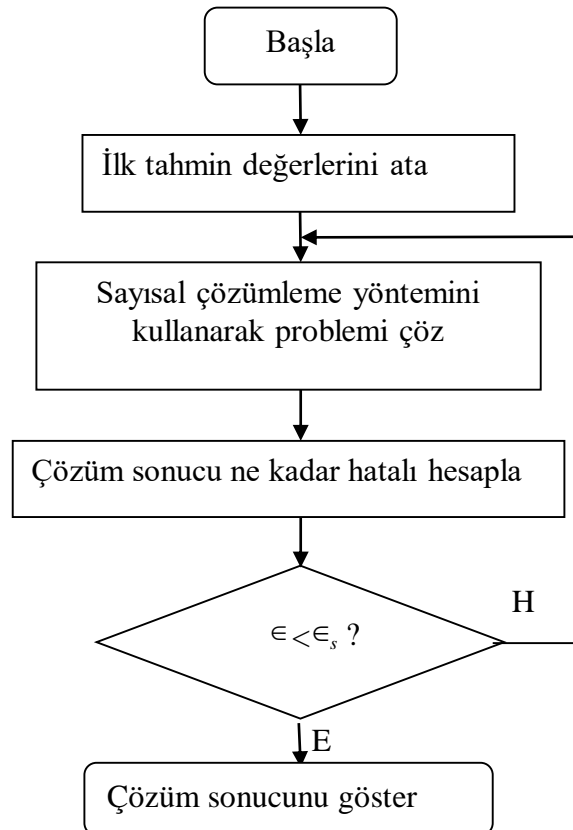
- Switch-case yapıları :

**Örnek.1)** 1 ile 10 arasında rastgele üretilen bir tamsayının elemanı olduğu kümenin üretilme olasılığının %'de kaç olduğunu ekranda gösteren program.  
1. Küme elemanları{1,2}, 2.Küme elemanları{3,4,5}, Geri kalanların bulunduğu kümenin elemanları{6,7,8,9,10}

```
a=rand
x=ceil(10*a)
switch x
case{1,2}
    disp('Olasılık=%20');
case{3,4,5}
    disp('Olasılık=%30');
otherwise
    disp('Olasılık=%50');
end
```

# Sayısal Çözümleme

## Problem çözme adımları



## İkiye Bölme (Bisection) Yöntemi

[xa,xü] aralığındaki köke yaklaşmak için aralığın orta noktasını bulalım

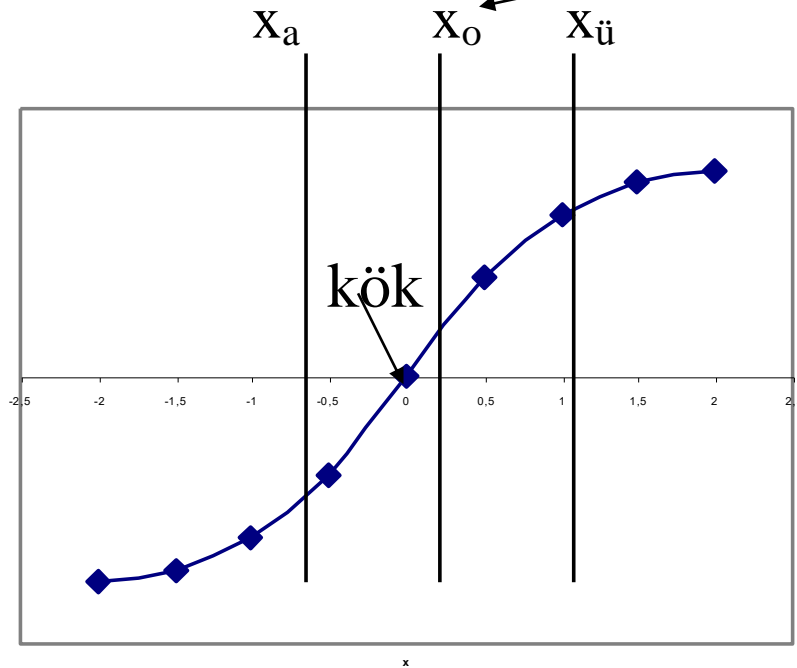
$$x_o = \frac{x_a + x_{\ddot{u}}}{2}$$

- $f(x_a) \cdot f(x_o) < 0$   $x_a$  ile  $x_o$  farklı bölgelerde
- $f(x_a) \cdot f(x_o) > 0$   $x_a$  ile  $x_o$  aynı bölgelerde

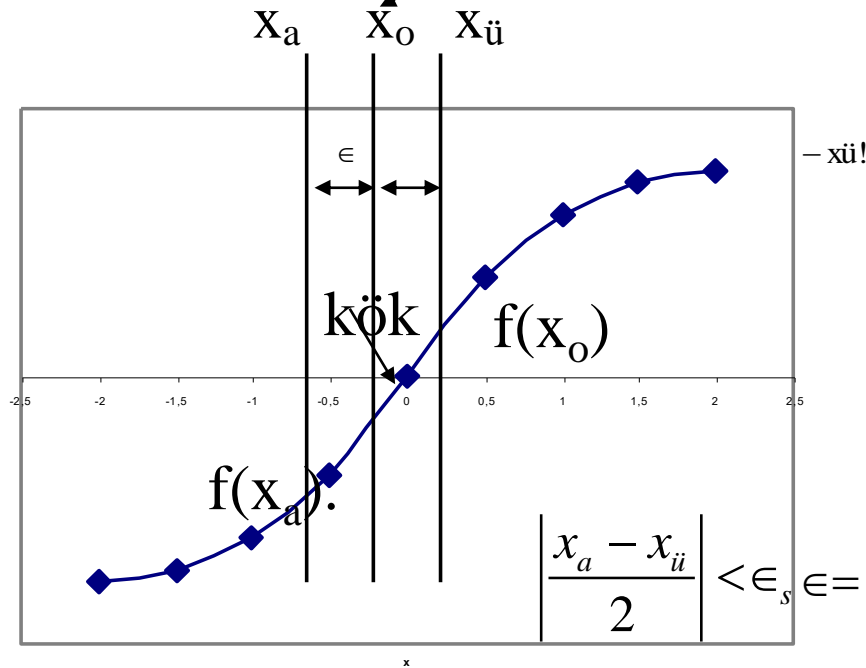
Güncellenecek sınır

$$x_{\ddot{u}}(\text{yeni})=x_o$$

$$x_a(y_{eni}) = x_o$$



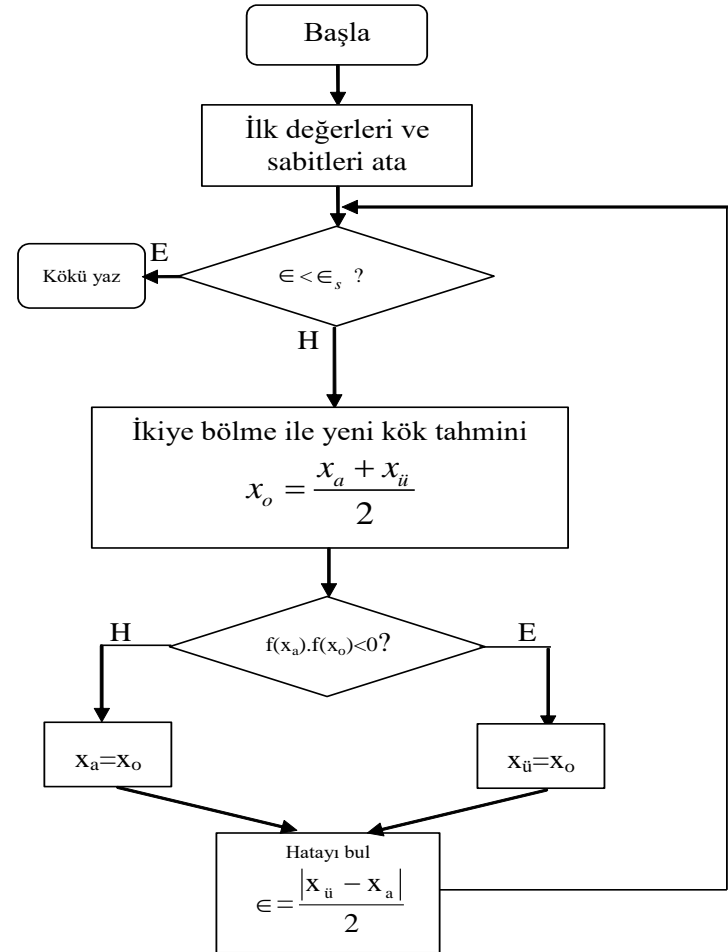
## Kök, $x_a$ , $x_o$ arasında



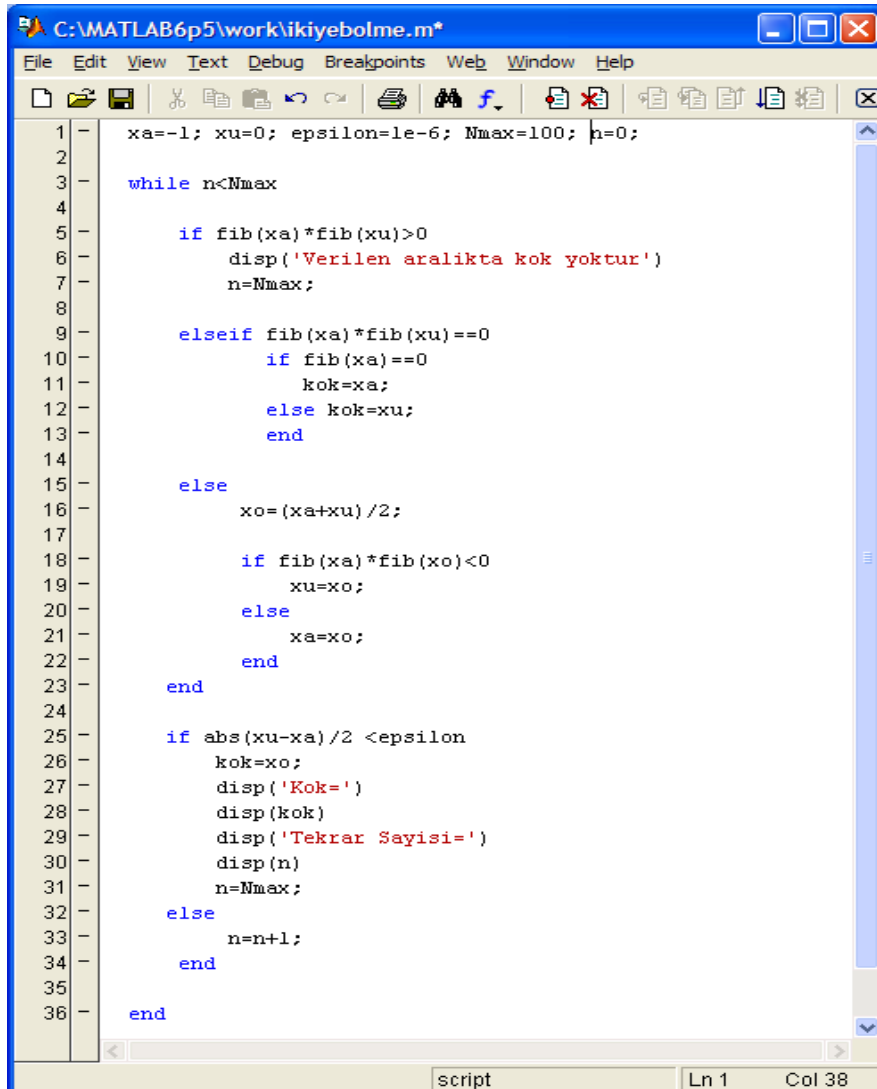
Kök,  $x_o$ ,  $x_{\ddot{u}}$  arasında

**Örnek:**  $f(x) = x.e^{-x} + x^3 + 1$  fonksiyonunun kökünü  $1 \cdot 10^{-6}$  duyarlılıkla  $[-1,0]$  aralığında bulalım,

Bilgisayarda çözüm:  
Akış şeması



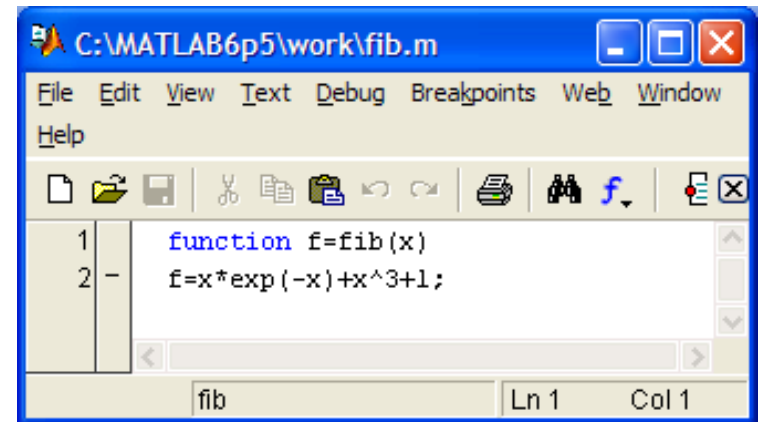
# Matlab programı



A screenshot of the MATLAB script editor window. The title bar shows the file path 'C:\MATLAB6p5\work\ikiyebolme.m'. The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. The toolbar contains icons for file operations and editing. The script is as follows:

```
1 xa=-1; xu=0; epsilon=1e-6; Nmax=100; n=0;
2
3 while n<Nmax
4
5     if fib(xa)*fib(xu)>0
6         disp('Verilen aralikta kok yoktur')
7         n=Nmax;
8
9     elseif fib(xa)*fib(xu)==0
10        if fib(xa)==0
11            kok=xa;
12        else kok=xu;
13        end
14
15    else
16        xo=(xa+xu)/2;
17
18        if fib(xa)*fib(xo)<0
19            xu=xo;
20        else
21            xa=xo;
22        end
23    end
24
25    if abs(xu-xa)/2 <epsilon
26        kok=xo;
27        disp('Kok=')
28        disp(kok)
29        disp('Tekrar Sayisi=')
30        disp(n)
31        n=Nmax;
32    else
33        n=n+1;
34    end
35
36 end
```

The status bar at the bottom indicates 'script', 'Ln 1', and 'Col 38'.



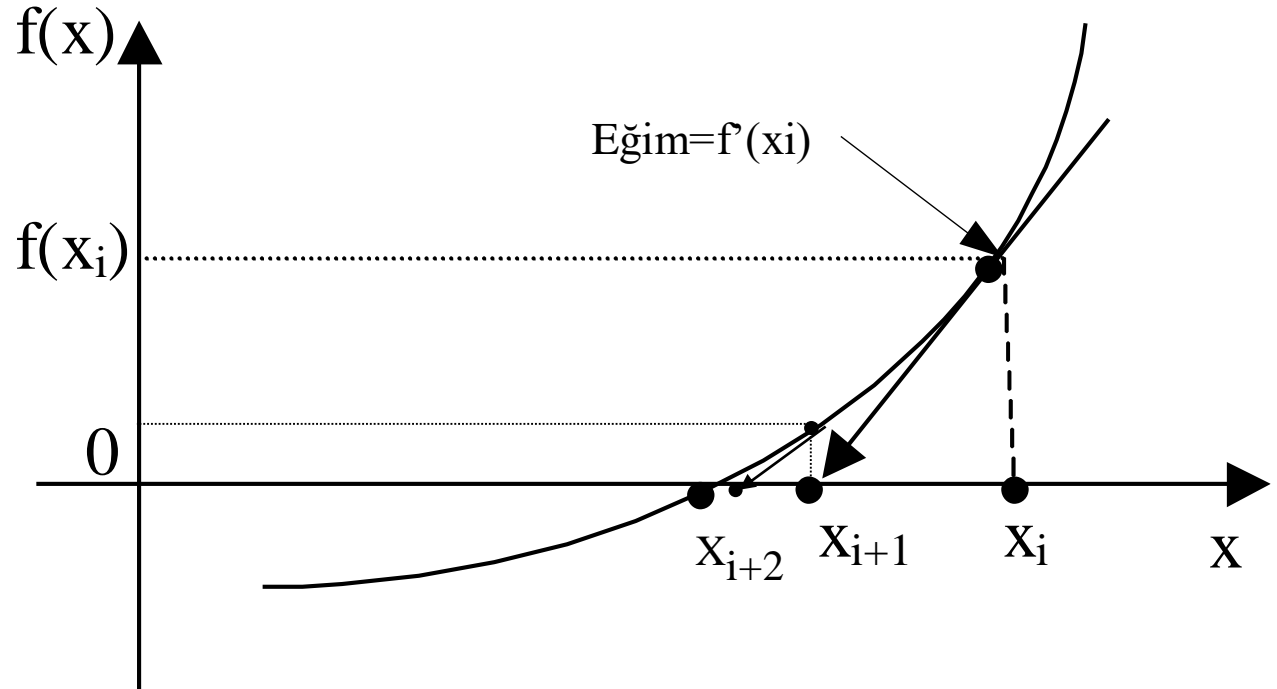
A screenshot of the MATLAB function editor window. The title bar shows the file path 'C:\MATLAB6p5\work\fib.m'. The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. The toolbar contains icons for file operations and editing. The function definition is as follows:

```
1 function f=fib(x)
2 f=x*exp(-x)+x^3+1;
```

The status bar at the bottom indicates 'fib', 'Ln 1', and 'Col 1'.



## 4.3.2. Newton-Raphson Yöntemi



$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

**Örnek:** Newton-Raphson yöntemini kullanarak,  $f(x)=e^{-x}-x$  fonksiyonunun kökünü  $x_0=0$  ilk tahminini yaparak bulun. (Yüzde bağıl yaklaşma hatası  $3 \cdot 10^{-5}$ 'in altına düşene kadar iterasyona devam edin)

**Çözüm:** Fonksiyonun birinci türevi

$$f'(x) = -e^{-x} - 1$$

fonksiyon ve türevi denklemde yerine konulursa

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$x_{i+1} = x_i - \frac{e^{-x_i} - x_i}{-e^{-x_i} - 1}$$

```

es=3e-5; n=0; Nmax=100;
xkeski=0;
while (n<Nmax)
    n=n+1;
    if fturev(xkeski)==0
        disp('Sifira bolme hatasi');
    else
        xkyeni=xkeski-fkendi(xkeski)/fturev(xkeski)
        if xkyeni~=0
            ea=abs((xkyeni-xkeski)/xkyeni)*100
            if ea<es
                disp('Kök='); disp(xkyeni);
                disp('Tekrar Sayisi='); disp(n);
                disp('Yüzde bagil Hata=');disp(ea);
                n=Nmax;
            end
        else disp('Sifira bolme hatasi');
        end
        xkeski=xkyeni;
    end
end
end

```

```

function f=fturev(x)
f=-exp(-x)-1;

```

```

function f=fkendi(x)
f=exp(-x)-x;

```

# Cramer Yöntemi

$$x_k = \frac{\det A_k}{\det A}$$

$$[Ak] = \begin{matrix} & \begin{matrix} 1 & \dots & k \end{matrix} \\ \begin{bmatrix} a_{11} & \dots & b_1 & \dots & a_{1n} \\ a_{21} & \dots & b_2 & \dots & a_{2n} \\ \vdots & & & & \\ a_{n1} & \dots & b_n & \dots & a_{nn} \end{bmatrix} \end{matrix}$$

Örnek: Aşağıda verilen denklem takımını Cramer kuralıyla çözün.

$$3x_1 + 4x_2 - 5x_3 = -47$$

$$-2x_1 - 5x_2 + 7x_3 = 56$$

$$-7x_1 + 2x_2 - 3x_3 = 15$$

*Çözüm:*

$$\begin{bmatrix} 3 & 4 & -5 \\ -2 & -5 & 7 \\ -7 & 2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -47 \\ 56 \\ 15 \end{bmatrix}$$

$$|A| = 3(15 - 14) - 4(6 + 49) - 5(-4 - 35) = 3 - 220 + 195 = -22$$

$$E = \begin{bmatrix} -47 \\ 56 \\ 15 \end{bmatrix}$$

$$x_1 = -\frac{1}{22} \begin{vmatrix} -47 & 4 & -5 \\ 56 & -5 & 7 \\ 15 & 2 & -3 \end{vmatrix} = -\frac{1}{22} [-47(15-14) - 4(-168-105) - 5(112+75)] = -\frac{1}{22} (110) = -5$$

Benzer biçimde  $x_2$  ve  $x_3$  elemanları da bulunur.

$$x_2 = -\frac{1}{22} \begin{vmatrix} 3 & -47 & -5 \\ -2 & 56 & 7 \\ -7 & 15 & -3 \end{vmatrix} = -\frac{1}{22} [3(-168-105) + 47(6+49) - 5(-30+392)] = -\frac{1}{22} (-44) = 2$$

$$x_3 = -\frac{1}{22} \begin{vmatrix} 3 & 4 & -47 \\ -2 & -5 & 56 \\ -7 & 2 & 15 \end{vmatrix} = -\frac{1}{22} [3(-75-112) - 4(-30+392) - 47(-4-35)] = -\frac{1}{22} (-176) = 8 \text{ bulunur.}$$

$$x_1 = -5, x_2 = 2 \text{ ve } x_3 = 8$$

## MATLAB ÇÖZÜMÜ

```
function xk=x(k)
A=[3 4 -5
   -2 -5 7
   -7 2 -3]
E=[-47
   56
   15]
Ak=A;
for i=1:size(A,1)
    Ak(i,k)=E(i);
end
xk=det(Ak)/det(A);
```