BIL301 SAYISAL YÖNTEMLER
4. Hafta
Denklem Çözüm Yöntemleri

Doç. Dr. Sercan YALÇIN

DENKLEM ÇÖZÜMLERİ

Cebirsel denklem

$$f(x)=0$$

Kök,kökün bulunması

- **4.1. Grafik Yöntemleri**
- **4.2. Kapalı Yöntemler**
- **4.3. Açık Yöntemler**

Denklem kökleri mühendislikte tasarım alanında karşımıza çıkar.

- Fizik kanunlarından çıkarılan matematiksel denklemler veya modeller, bir sisteme ait bağımlı değişkenlerin tahmin edilmesinde kullanılır.

- Örnek:Bir paraşütçünün hızını bulmak için Newtonun 2. yasasını kullanalım

$$v = \frac{g\,m}{c}\left(1 - e^{-(c/m)t}\right)$$

Diğer parametreler bilinirse, paraşütçünün hızını, zamana bağlı olarak hesaplamak (v=f(t)) kolaydır

$$v = \frac{g\,m}{c}\left(1 - e^{-(c/m)t}\right)$$

- fakat
  c= ?.
- Çözüm analitik olarak mümkün değil
- Sayısal çözüm:
- f(c) =0

$$f(c) = \frac{g\,m}{c}\left(1 - e^{-(c/m)t}\right) - v$$

Bu fonksiyonu sıfır yapan kök, tekrar tekrar c'ye değerler verilerek, grafik veya diğer sayısal yöntemlerle bulunur.

Denklemlerin sayısal olarak çözümleri de diğer problem çözümleri gibi çoğunlukla yinelemeli (iteratif) yöntemlerle yapılır.

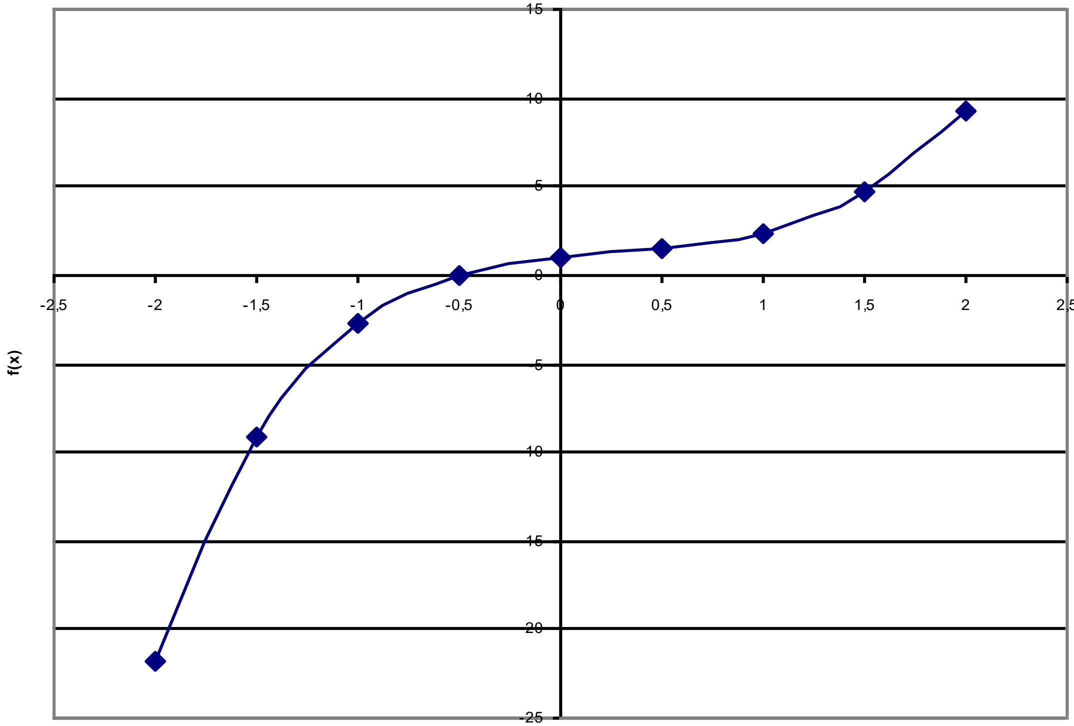
Grafik Yöntemleri

•Kökü aramaya doğru bir noktadan başlamak
çözüme ulaşmayı hızlandıracaktır

•Grafik çizimleri, kökü aramak için herhangi
bir sayısal çözüm yönteminde başlangıç tahmin değerlerinin
seçiminde bize yardımcı olur

- Örnek:$f(x)=xe^{-x}+x^3+1$ fonksiyonunun yaklaşık kökünü grafikten bulalım.

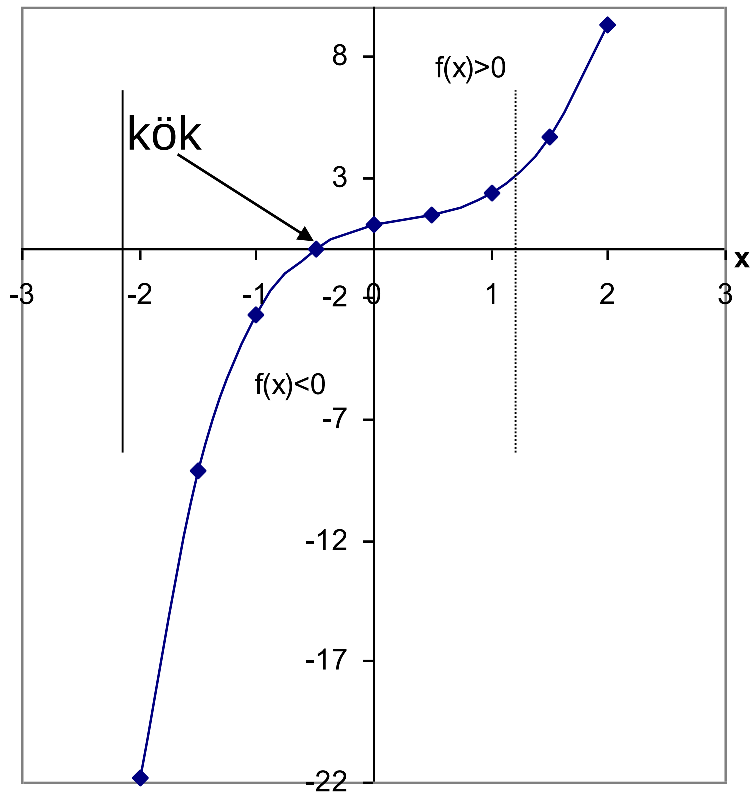
Kaba bir yaklaştırma için çizilen grafik yeterli olabilecektir.

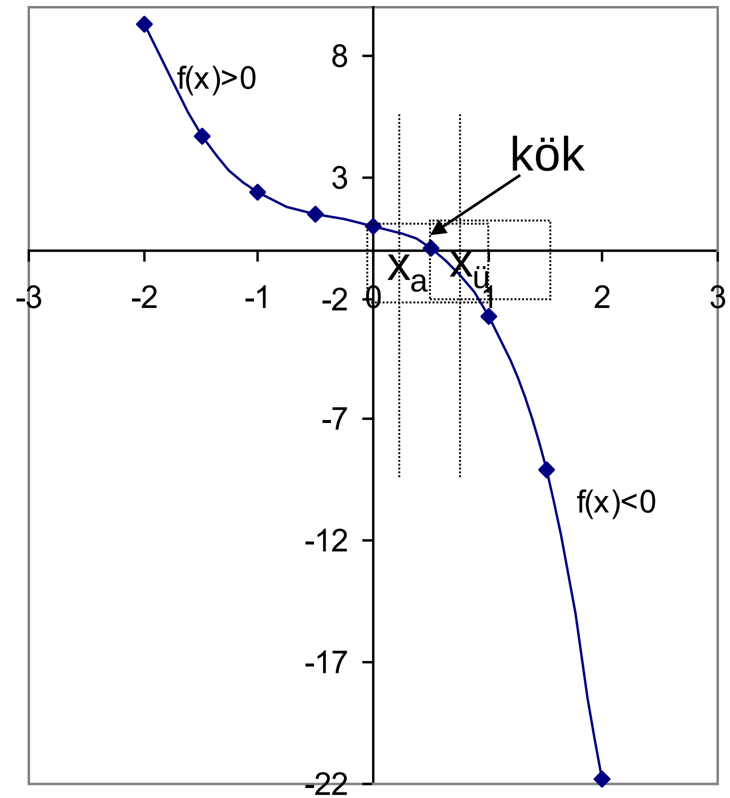


$f(x)= xe^{-x}+x^3+1$ fonksiyonunun grafiği

x	f(x)
-2	-21,7781122
-1,5	-9,097533606
-1	-2,718281828
-0,5	0,050639365
0	1
0,5	1,42826533
1	2,367879441
1,5	4,70969524
2	9,270670566

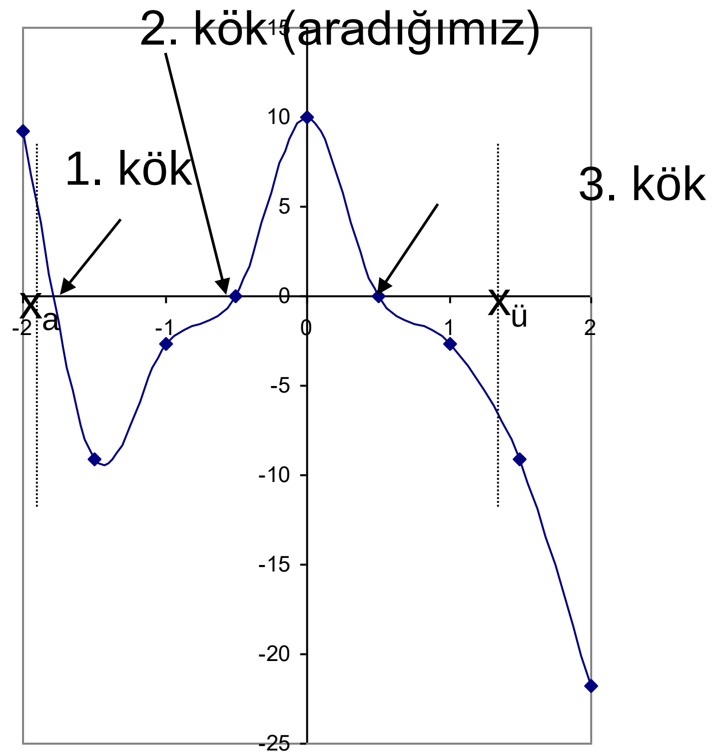
Kapalı Yöntemler



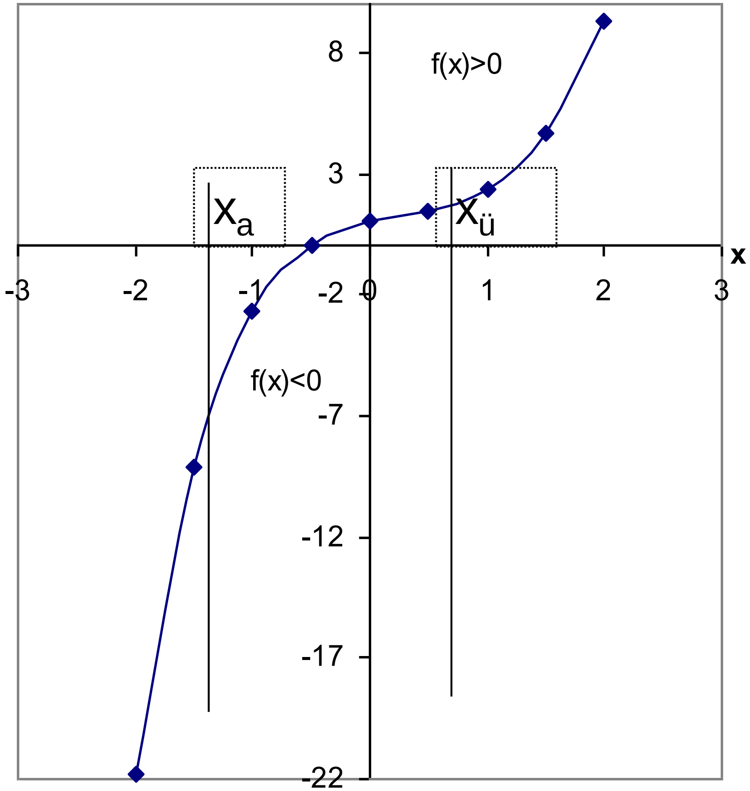


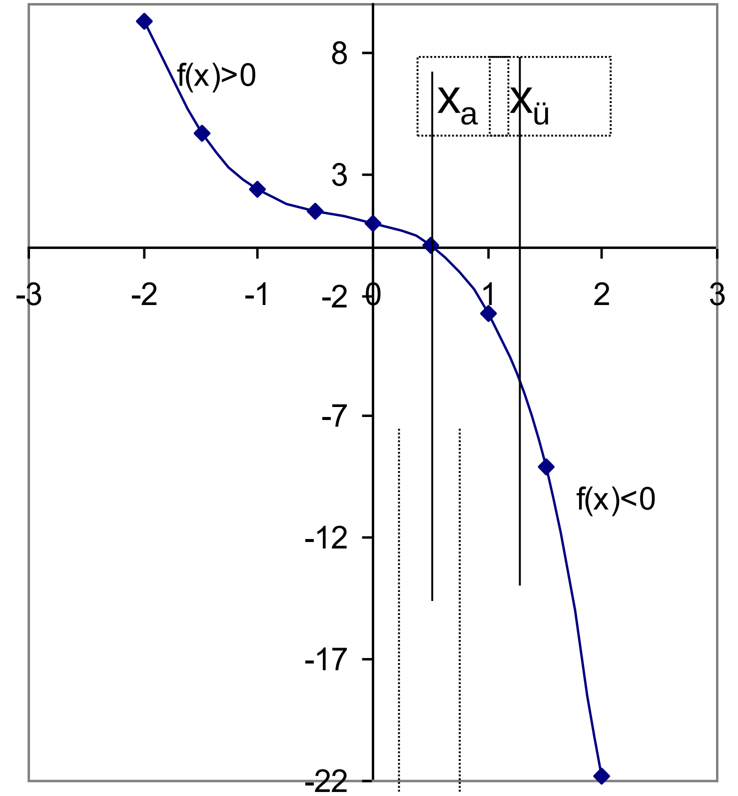
- Fonksiyonlar kök civarında işaret değiştirdikleri için, kökü sağından ve solundan kıskaca alarak bu aralığı gittikçe daraltıp köke ulaşmak mümkündür. Bunun için iki tane başlangıç değeri belirlemek gerekir.
- Kökün, bu iki değerin arasındaki kapalı bölgede olduğu bu yöntemlere kapalı yöntemler adı verilir.

Kapalı Yöntemler



■Arada başka bir kök olmaması ve kısa sürede köke yakınsaması için aralık mümkün olduğunca dar seçilmelidir.





f(x): [xa,xü]

- f(xa).f(xü)<0 x $\in$ [xa,xü]
- f(xa).f(xü)=0 $\rightarrow$ f(x_a) =0 x=x_a
  f($x_ü$) =0 x=$x_ü$
- f(xa).f(xü)>0 x $\notin$ [xa,xü]

İkiye Bölme (Bisection) Yöntemi

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

$$\boxed{x_o = \frac{x_a + x_ü}{2}}$$

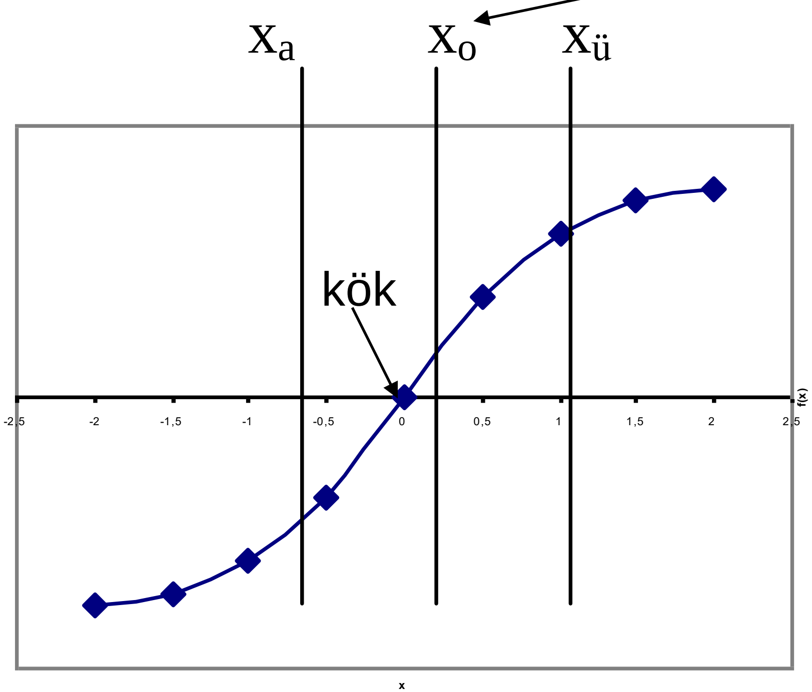
- $f(x_a).f(x_o) < 0$ x_a ile x_o farklı bölgelerde

Güncellenecek sınır

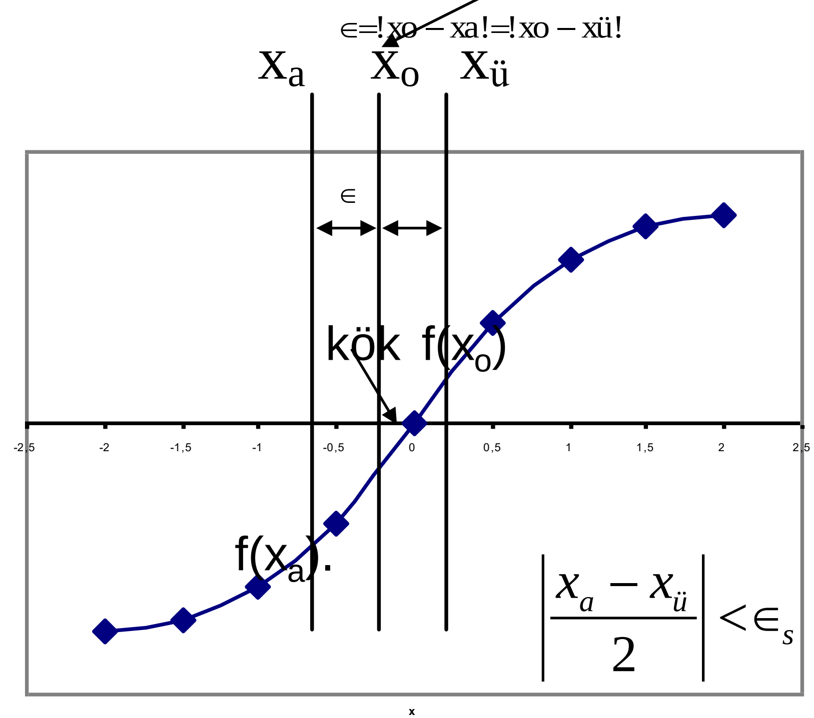
xü(yeni)=xo

- $f(x_a).f(x_o) > 0$ x_a ile x_o aynı bölgelerde

xa(yeni)=xo



Kök, x_a, x_o arasında



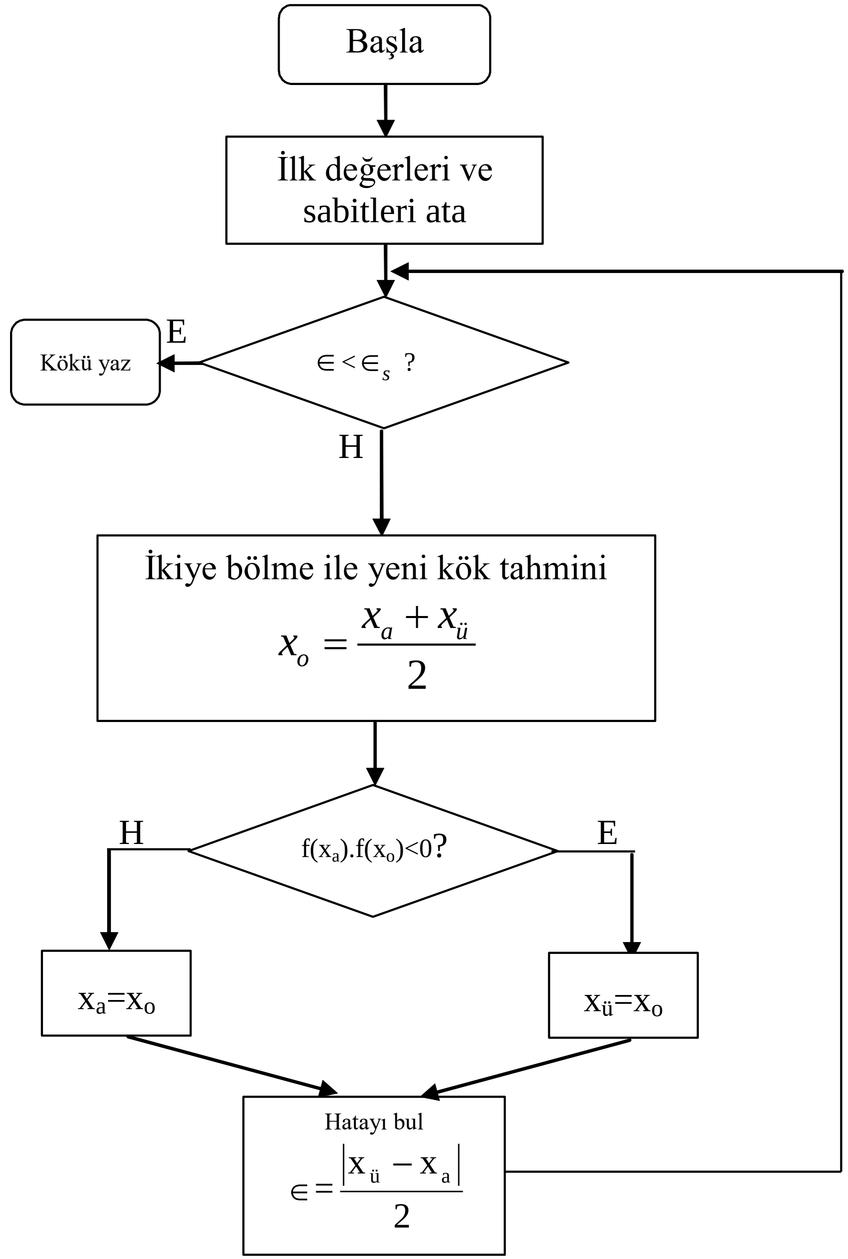
Kök, x_o, $x_ü$ arasında

Örnek: $f(x) = x.e^{-x}+x^3+1$ fonksiyonunun kökünü $\in_s$ $=1*10^{-6}$ duyarlılıkla bulalım, [-1,0], Cevap: x=-0.515438

Tablo.4.1. İkiye bölme yöntemiyle fonksiyonun kökünün yaklaşık olarak bulunması

n	x_a	$x_ü$	x_o	$f(x_a).f(x_o)$	$\in = \left\lvert \frac{x_a - x_ü}{2} \right\rvert$
1	-1.000000	0.000000	-0.500000	-	0.500000
2	-1.000000	-0.500000	-0.750000	+	0.250000
3	-0.750000	-0.500000	-0.625000	+	0.125000
4	-0.625000	-0.500000	-0.562500	+	0.062500
5	-0.562500	-0.500000	-0.531250	+	0.031250
6	-0.531250	-0.500000	-0.515625	+	0.015625
7	-0.515625	-0.500000	-0.507813	-	0.007813
.	.	.	.	.	.
.	.	.	.	.	.
19	-0.515449	-0.515442	-0.515446	+	0.000004
20	-0.515446	-0.515442	-0.515444	-	0.000002

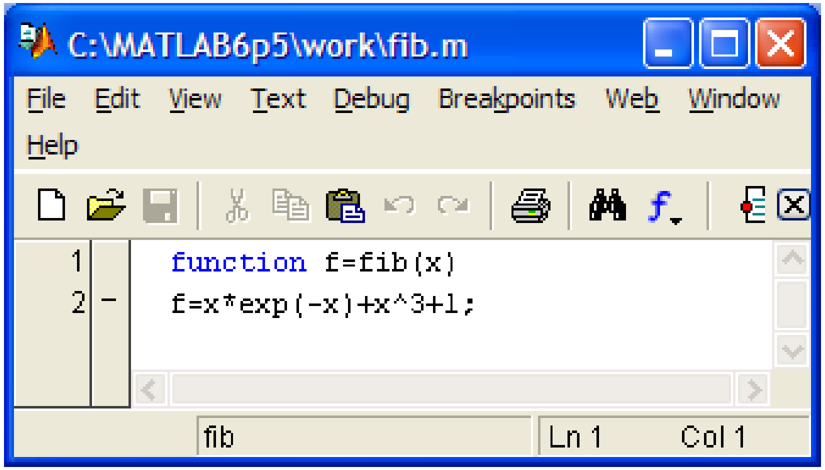
Bilgisayarda Çözüm: Programın Algoritması



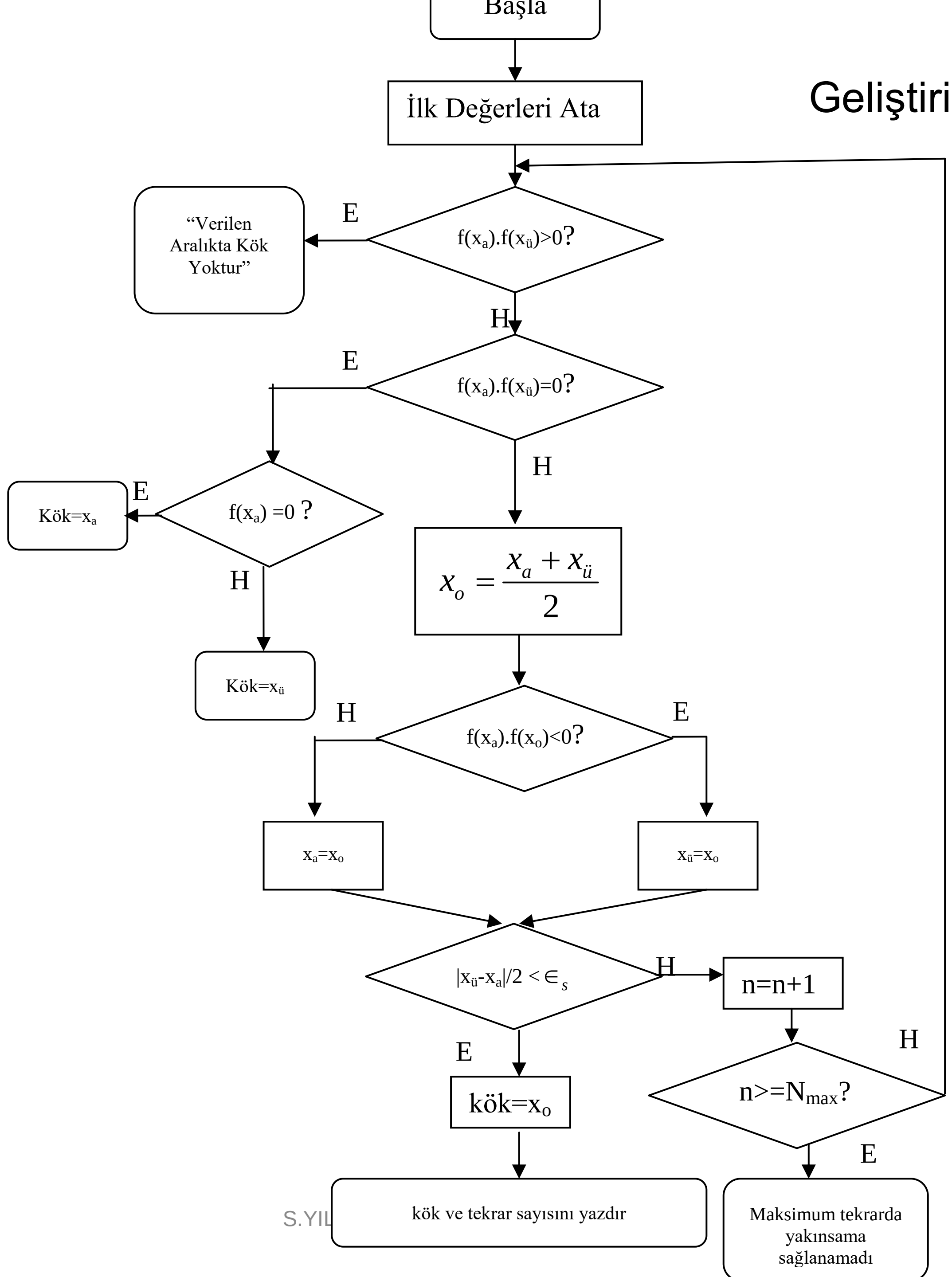
Program

```
xa=-1; xu=0; es=1e-6
while abs(xu-xa)/2>es
  xo=(xa+xu)/2
  fa=xa*exp(-xa)+xa^3+1
  fo=xo*exp(-xo)+xo^3+1
  if fa*fo<0
    xu=xo;
  else
    xa=xo
  end
end
```

Programı daha esnek hale getirebilmek için öncelikle programda kullanılacak fonksiyon başka bir .m dosyası içinde önceden tanımlanabilir.



Geliştirilmiş algoritma



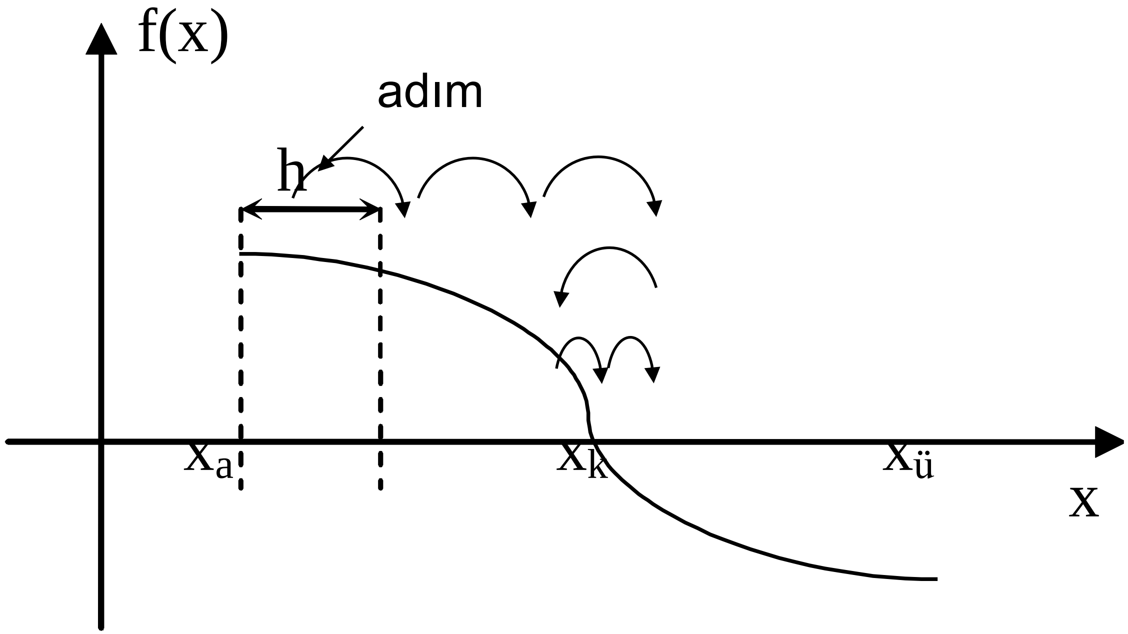
C:\MATLAB6p5\work\ikiyebolme.m*

File Edit View Text Debug Breakpoints Web Window Help

```matlab
xa=-1; xu=0; epsilon=1e-6; Nmax=100; n=0;

while n<Nmax

    if fib(xa)*fib(xu)>0
        disp('Verilen aralikta kok yoktur')
        n=Nmax;

    elseif fib(xa)*fib(xu)==0
           if fib(xa)==0
              kok=xa;
           else kok=xu;
           end

    else
         xo=(xa+xu)/2;

         if fib(xa)*fib(xo)<0
             xu=xo;
         else
             xa=xo;
         end
   end

   if abs(xu-xa)/2 <epsilon
       kok=xo;
       disp('Kok=')
       disp(kok)
       disp('Tekrar Sayisi=')
       disp(n)
       n=Nmax;
   else
        n=n+1;
    end

end
```

script   Ln 1   Col 38

Adım Küçülterek Köke Yaklaşma Yöntemi



f(x).f(x+h) >0 ⟶ $\boxed{\text{x(yeni)=x+h}}$

f(x).f(x+h)<0 ⟶ h(yeni)=h/10

Örnek: Herhangi bir f(x) fonksiyonunun kökü 5.42 olsun.

[4 6] aralığında kökü aramaya başlarsak; xa=4, h=1

x= 4

5 h=0.1

5,1

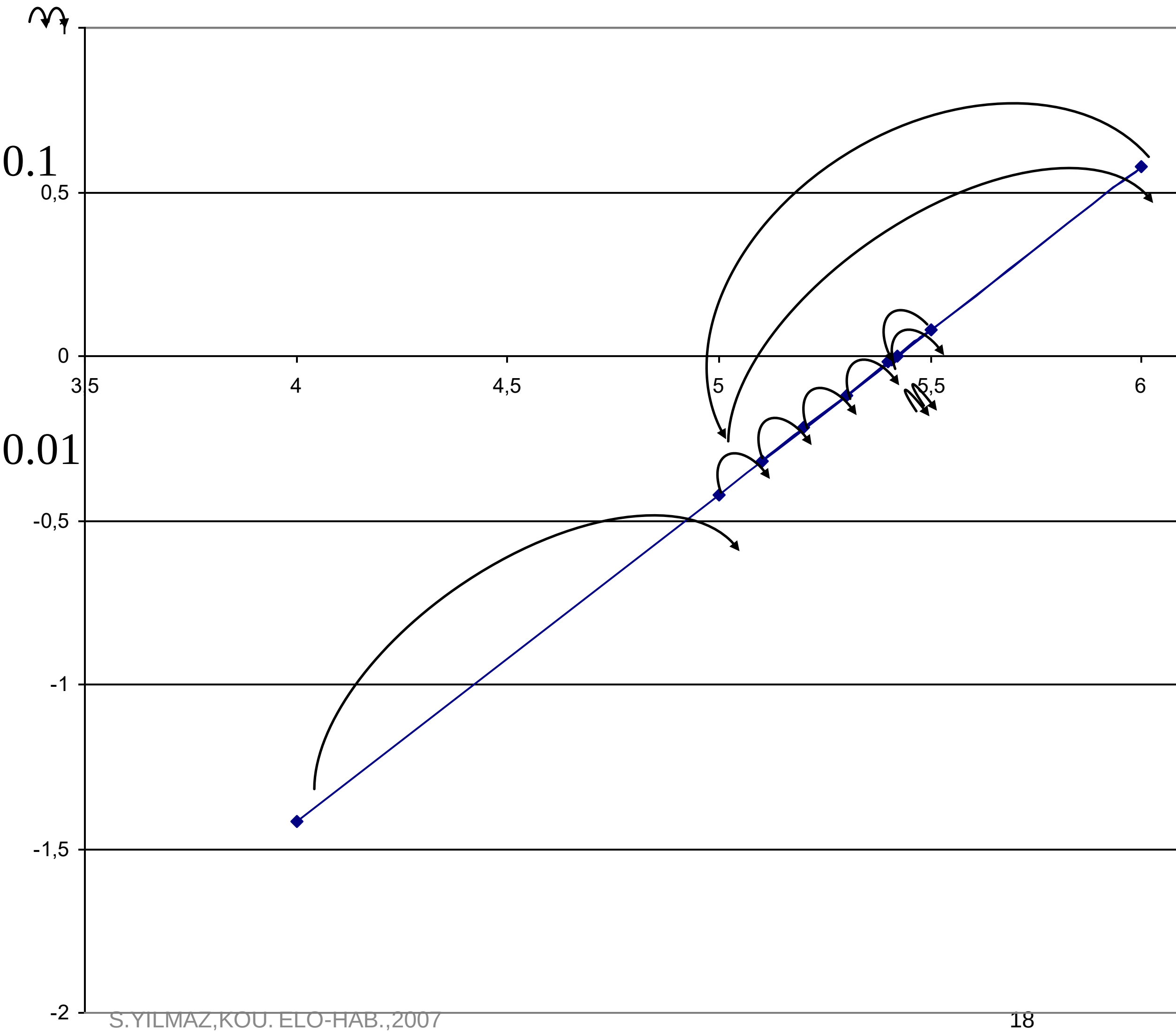
5,2

5,3

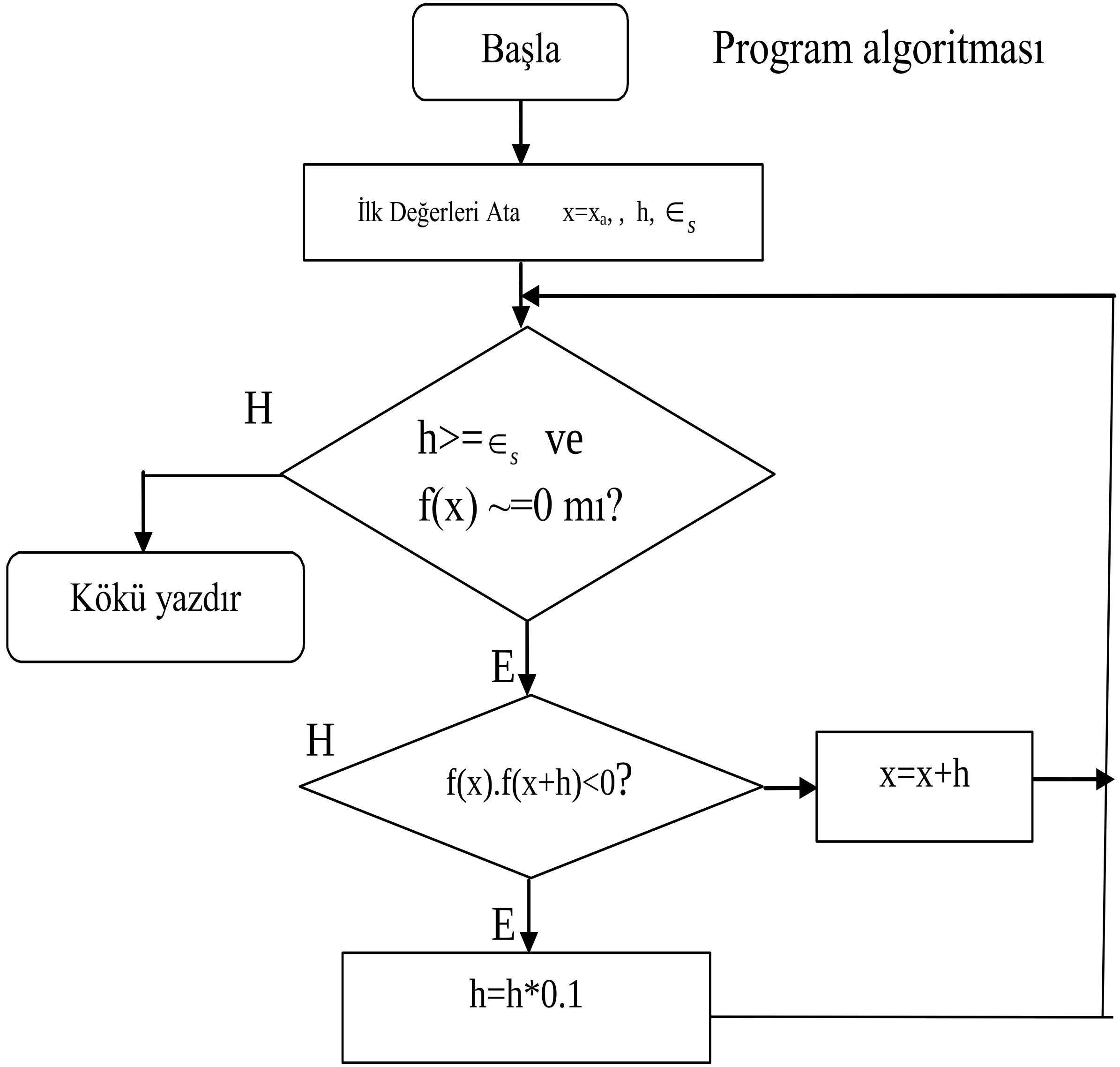
5,4 h=0.01

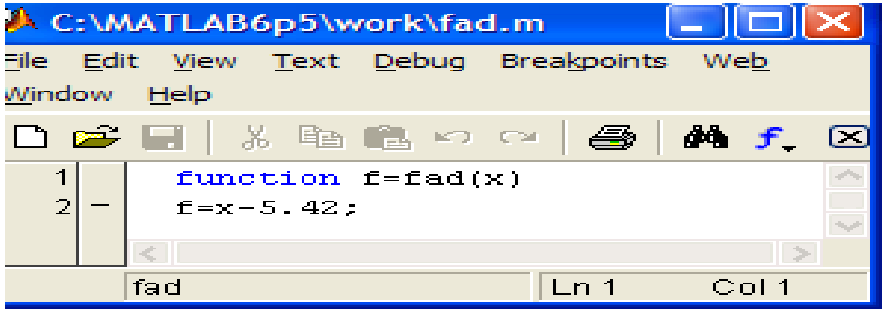
5,41

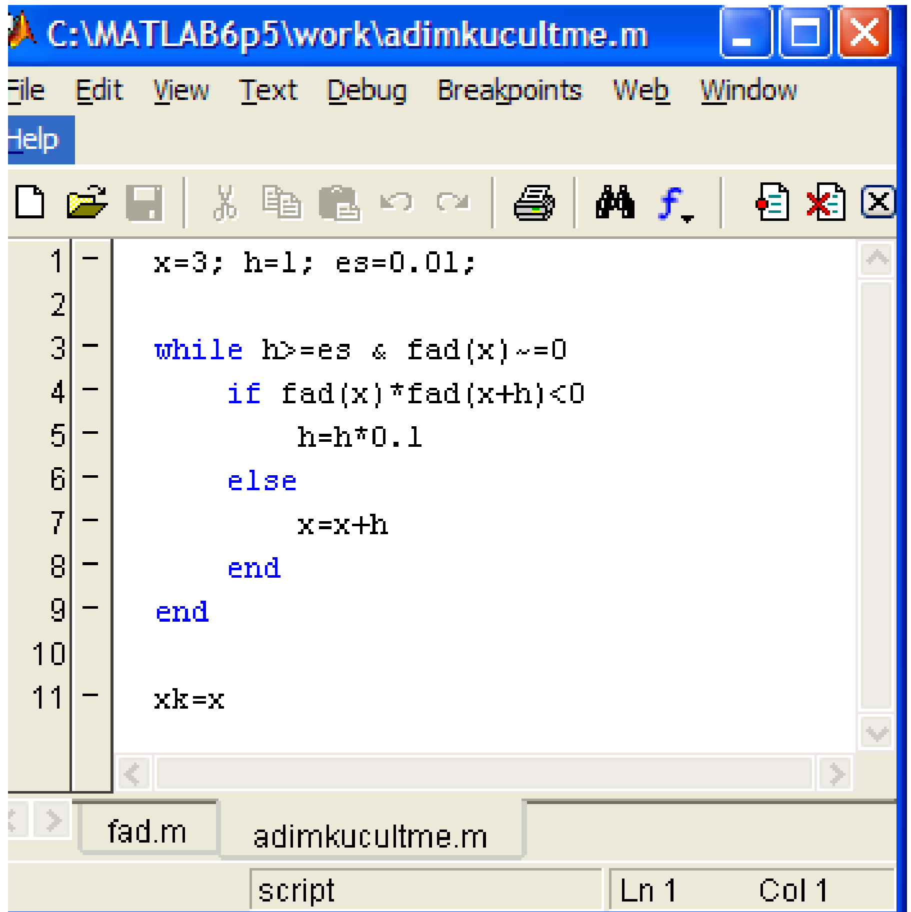
5,42



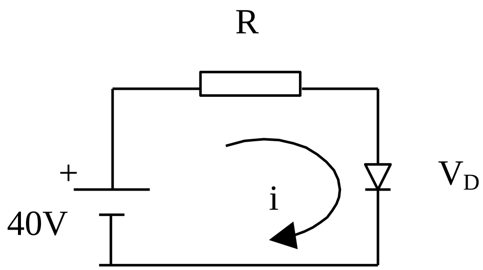
Program algoritması




C:\MATLAB6p5\work\fad.m
File Edit View Text Debug Breakpoints Web
Window Help
function f=fad(x)
f=x-5.42;
fad
Ln 1 Col 1

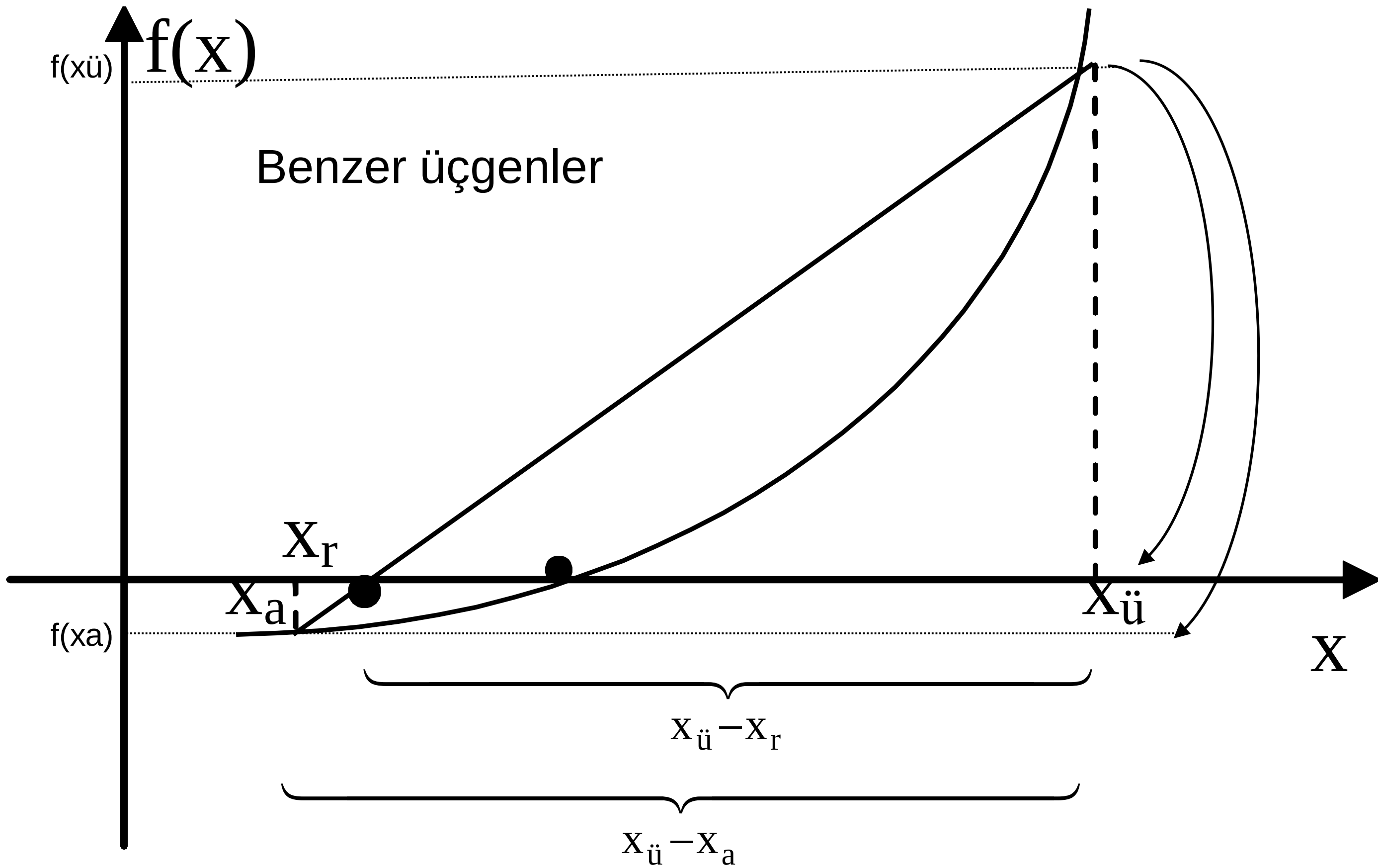

C:\MATLAB6p5\work\adimkucultme.m
File Edit View Text Debug Breakpoints Web Window
Help
x=3; h=1; es=0.01;
while h>=es & fad(x)~=0
if fad(x)*fad(x+h)<0
h=h*0.1
else
x=x+h
end
end
xk=x
fad.m
adimkucultme.m
script
Ln 1 Col 1

Ödev: a) R=10 Ω ve diyot gerilimi V_D= ln(150 i_D+1) olarak kabul edelim. Şekilde devrede i_D akımını i_D = [3 , 4] aralığında adım küçültme yöntemiyle $\in = 0.01$ duyarlılıkla hesaplayın. İlk adım büyüklüğümüz h=0,1 olarak başlasın. ($h_{yeni} = \dfrac{h_{eski}}{10}$)



b) Problemi bilgisayarda çözmek için bir algoritma hazırlayın ve bildiğiniz bir programlama dilinde yazın.

Yer Değiştirme (Regula Falsi) Yöntemi



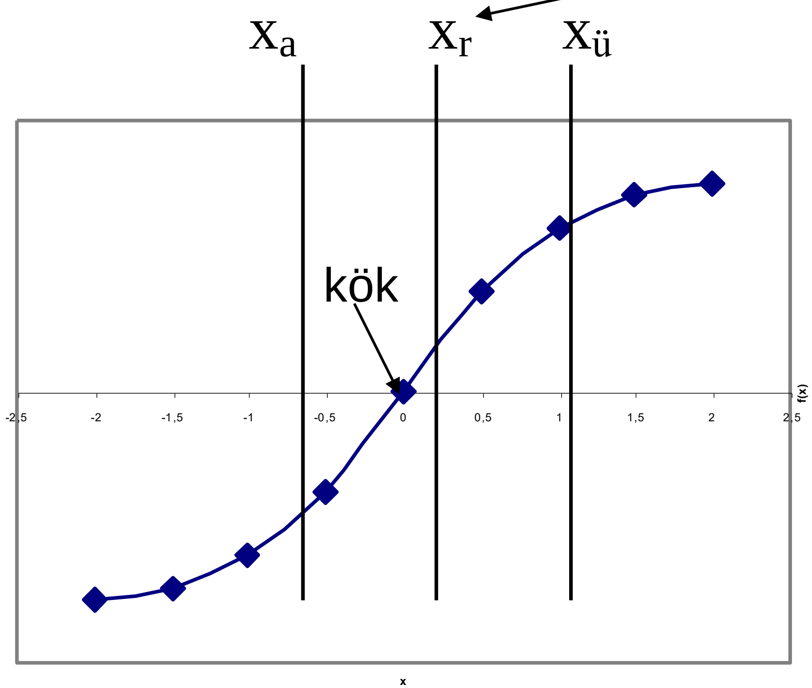
$$\frac{f(x_{ü})}{f(x_{ü})+(-f(x_a))}=\frac{x_{ü}-x_r}{x_{ü}-x_a}$$

$$\boxed{x_r = x_{ü} - \frac{f(x_{ü})\left(x_a - x_{ü}\right)}{f(x_a) - f(x_{ü})}}$$

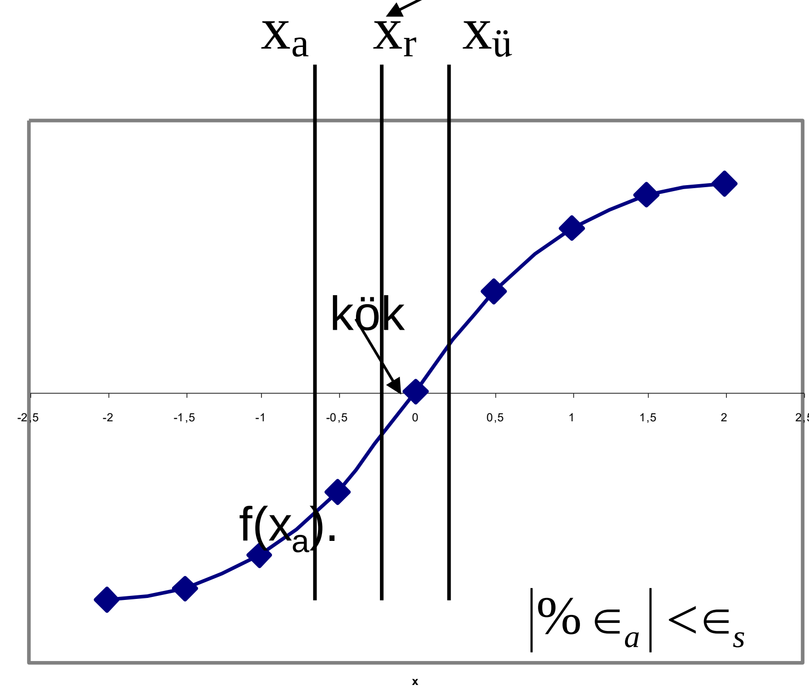
$$x_r = x_{ü} - \frac{f(x_{ü})(x_a - x_{ü})}{f(x_a) - f(x_{ü})}$$

Güncellenecek sınır

- $f(x_a).f(x_r) < 0$ x_a ile x_r farklı bölgelerde xü(yeni)=xr
- $f(x_a).f(x_r) > 0$ x_a ile x_r aynı bölgelerde xa(yeni)=xr



Kök, x_a, x_r arasında



Kök, x_r, $x_{ü}$ arasında

Örnek:

Kütlesi m=68.1kg olan bir paraşütçünün, t=10 s serbest düştükten sonra 40m/s hıza sahip olabilmesi için gerekli direnç katsayısını yer değiştirme yöntemiyle iki iterasyon adımı için belirleyin.(, xa=12, xü=16)

$$f(c) = \frac{g\,m}{c}\left(1 - e^{-(c/m)t}\right) - v$$

Çözüm: Burada kök x=c direncidir,

• ilk iterasyon:

xa=12 $\longrightarrow$ f(xa)=6.0699

xü=16 $\longrightarrow$ f(xü)=-2.2688

$$x_r = 16 - \frac{-2.2688(12-16)}{6.0669-(-2.2688)} = 14.9113$$

f(xr)=-0.25413

•İkinci iterasyon: f(xa)*f(xr)= -1.5426 <0

xr , xü ile aynı bölgede olduğu için bir sonraki iterasyonun üst sınırı olacaktır.

xü=14.9113 $\longrightarrow$ f(xü)= -0.2543

xa=12 $\longrightarrow$ f(xa)=6.0699

$$x_r = 14.9113 - \frac{-0.2543(12-14.9113)}{6.0669-(-0.2543)} = 14.7942$$

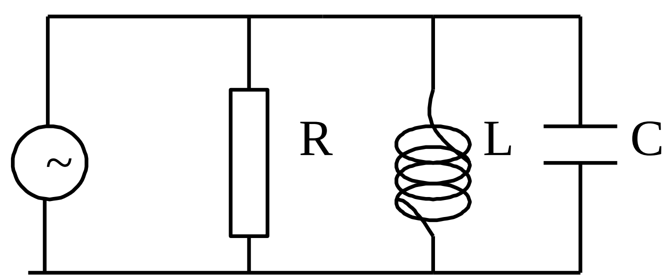
Ödev: a) Şekildeki elektrik devresinde Kirschoff yasaları kullanılarak sistemin empedansı $\frac{1}{Z}=\sqrt{\frac{1}{R^2}+\left(wC-\frac{1}{wL}\right)^2}$ şeklinde ifade edilebilir. Burada Z=empedans(Ω) ve w=açısal frekanstır. [w_a= 50 ve $w_ü$= 300] ilk tahminlerinden başlayarak yer değiştirme (regula falsi) yöntemiyle 100 Ω empedans veren açısal frekansı ilk 3 adım için bulun.

Hesaplamalarda virgülden sonra 5 basamağı dikkate alın

R=225Ω, C=0.6*10^{-6} F ve L=0.5 H.

Regula Falsi Formülü $x_r = x_ü - \frac{f(x_ü)(x_a - x_ü)}{f(x_a) - f(x_ü)}$

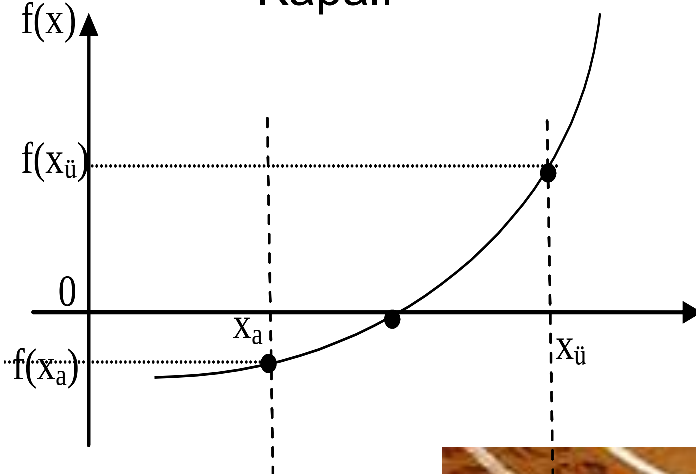
b) soruyu mutlak yüzde yaklaşım hatası $|\%e_a| \leq 10^{-3}$ duyarlılıkla bulan program algoritmasını oluşturun ve programı yazın.



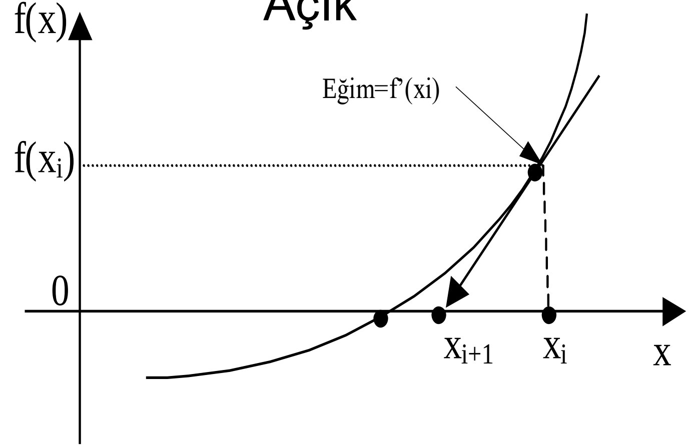
Açık Yöntemler

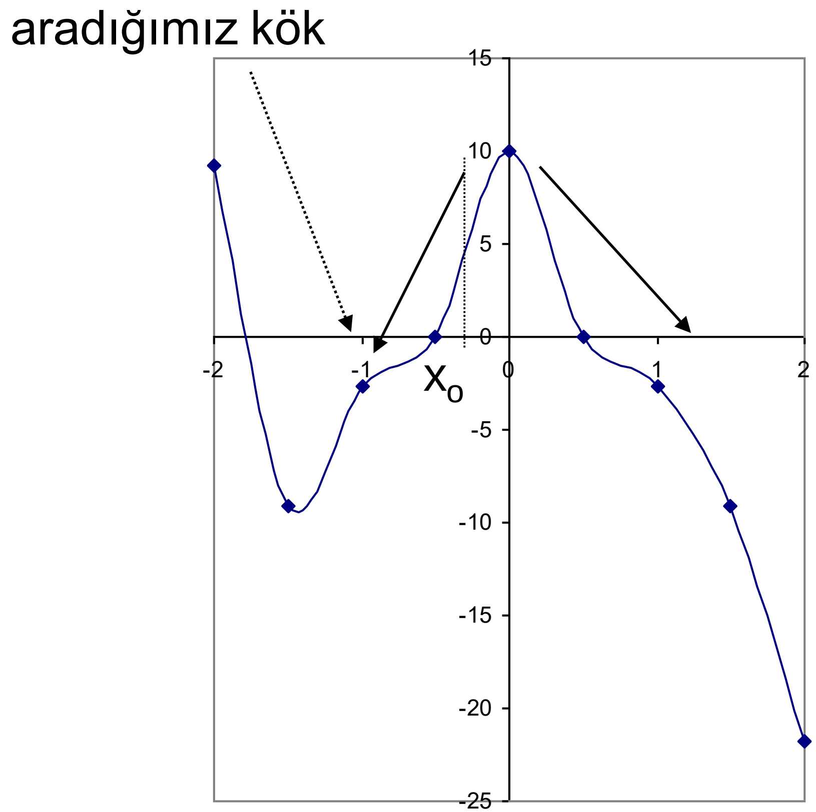
- Kökü iki başlangıç değeri arasında kıskaca alma (f(xa).f(xü) <0) sorgulaması yok

Kapalı



Açık





Açık yöntemler hızlıdır fakat bazen başlangıç noktası uygun seçilmediğinde ıraksayabilirler.

Basit Sabit Noktalı İterasyon:

- Bütün açık yöntemler kökün bulunması için bir formül kullanırlar.

- $f(x)=0 \longrightarrow x = g(x)$

$f(x)=x^2-2x+3=0 \longrightarrow x=\underbrace{\frac{x^2+3}{2}}_{g(x)}$

veya

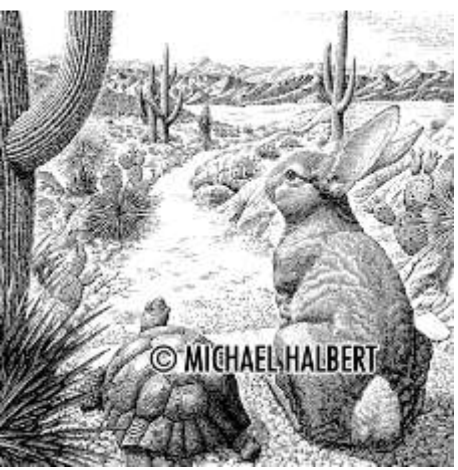
$f(x)=\sin x=0 \longrightarrow x=\underbrace{\sin x+x}_{g(x)}$

- $x_{i+1}=g(x_i)$

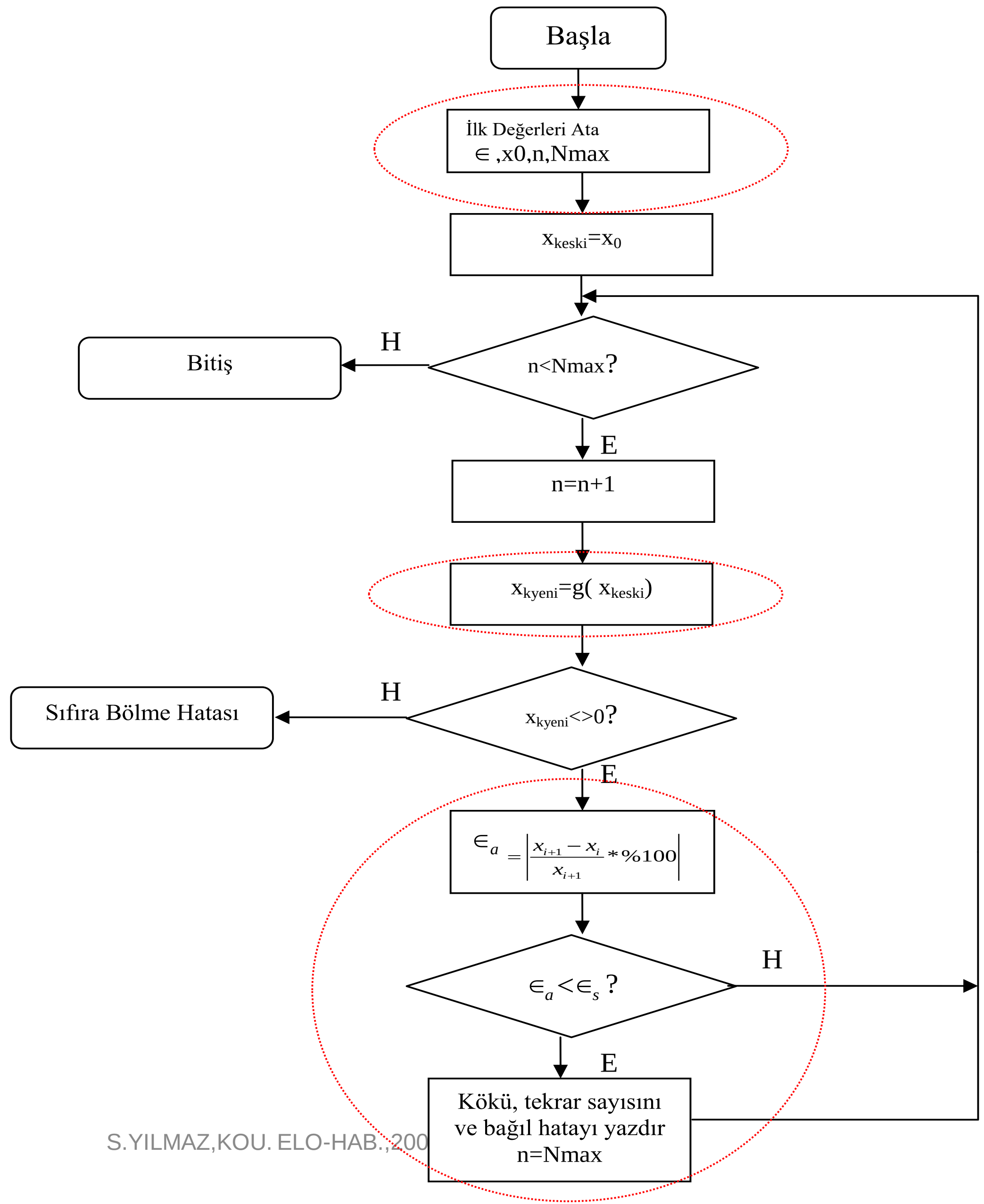
Örnek: Basit sabit noktalı iterasyon kullanarak $f(x)=e^{-x}-x$ fonksiyonunun kökünün yerini yüzde yaklaşım hatası $\% \in_a$ 1.2'nin altına düşene kadar hesaplayınız. Her adım için % yaklaşım hatasını mutlak değer olarak bulunuz. ($x_0=0$)

Çözüm: $x_{i+1} = e^{-x_i}$, İlk tahmin olarak $x_0=0$ ile başlayarak tablodaki değerler bulunabilir.

i	x_i	$\in_a = \left\| \frac{x_{i+1}-x_i}{x_{i+1}} \right\| * \%100$
0	0	
1	1.000000	100
2	0.367879	171,8285
3	0.692201	46,85373
4	0.500473	38,30936
5	0.606244	17,44694
6	0.545396	11,15666
7	0.579612	5,903259
8	0.560115	3,480892
9	0.571143	1,930865
10	0.564879	1,10891



Sabit noktalı iterasyon için algoritma

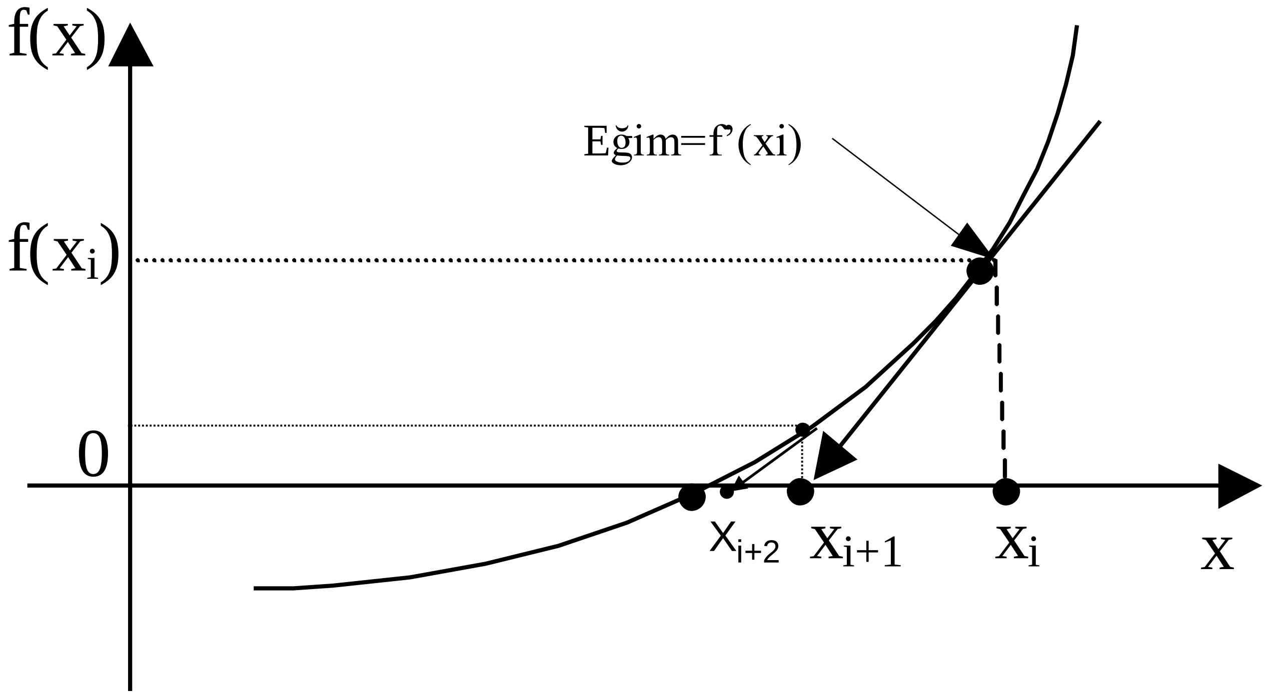


```
x0=0; es=1.2; n=0; Nmax=100;
xkeski=x0;
while (n<Nmax)
   n=n+1;
   xkyeni=g(xkeski)
   if xkyeni~=0
      ea=abs((xkyeni-xkeski)/xkyeni)*100
      if ea<es
         disp('Kök='); disp(xkyeni);
         disp('Tekrar Sayisi='); disp(n);
         disp('Yüzde bagil Hata=');disp(ea);
         n=Nmax;
      end
   else disp('Sifira bolme hatasi');
   end
   xkeski=xkyeni;
end
```

g.m dosyası

```
function [xkyeni] = g(xkeski)
xkyeni=1.0*exp(-xkeski);
```

Newton-Raphson Yöntemi



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

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

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

- Çözüm: Fonksiyonun birinci türevi

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

fonksiyon ve türevi denklemde yerine konulursa

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

$x_0=0$

i	x_i	$(\%)\in_a$
0	0	
1	0.500000000	100
2	0.566311003	11,70929095
3	0.567143165	0,146728736
4	0.567143290	2,20403E-05

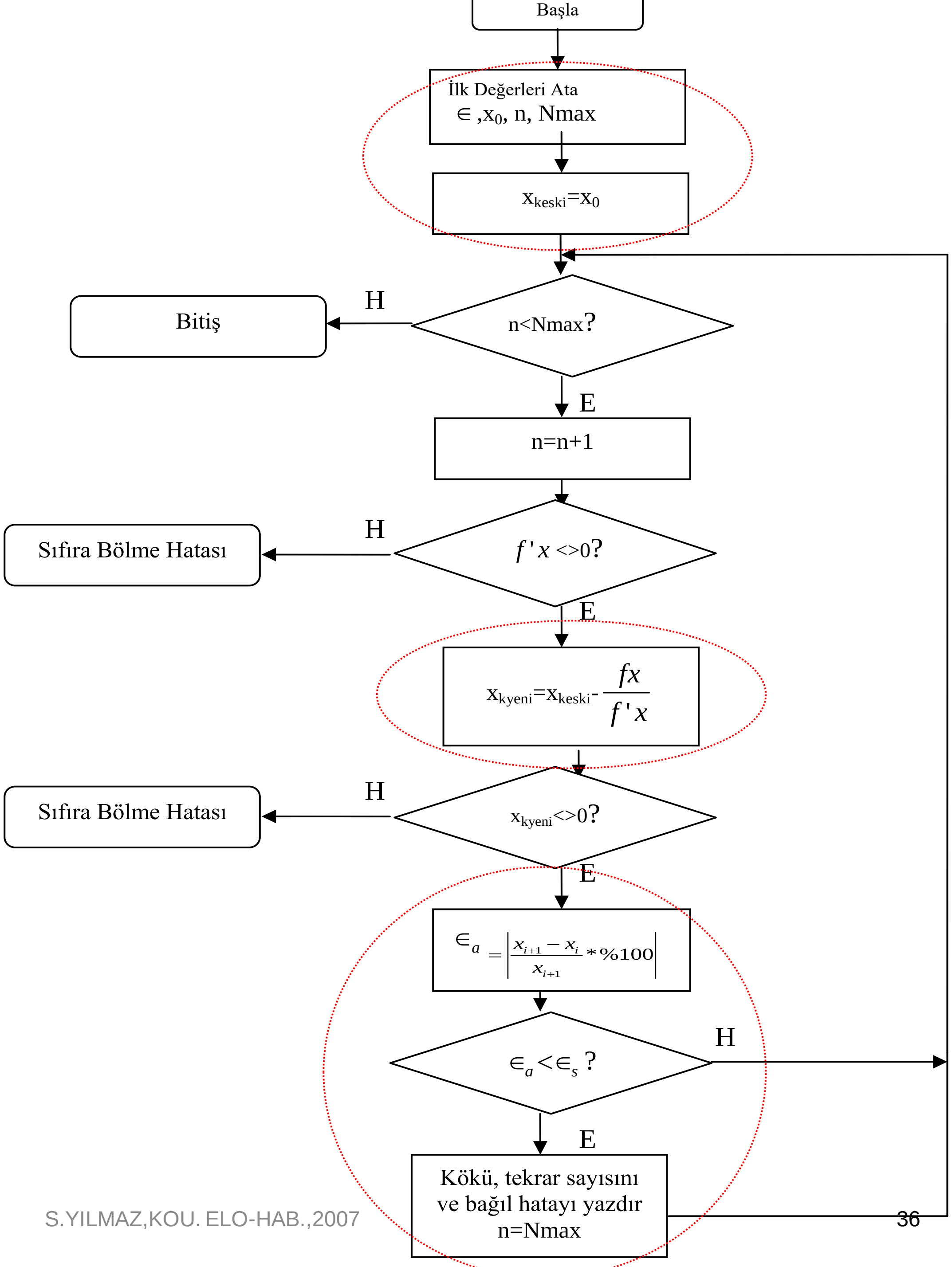
- f.m dosyasının içeriği:

```
function [fx] = f(x)
fx=1.0*exp(-x)-x;
```

fturev.m dosyasının içeriği:

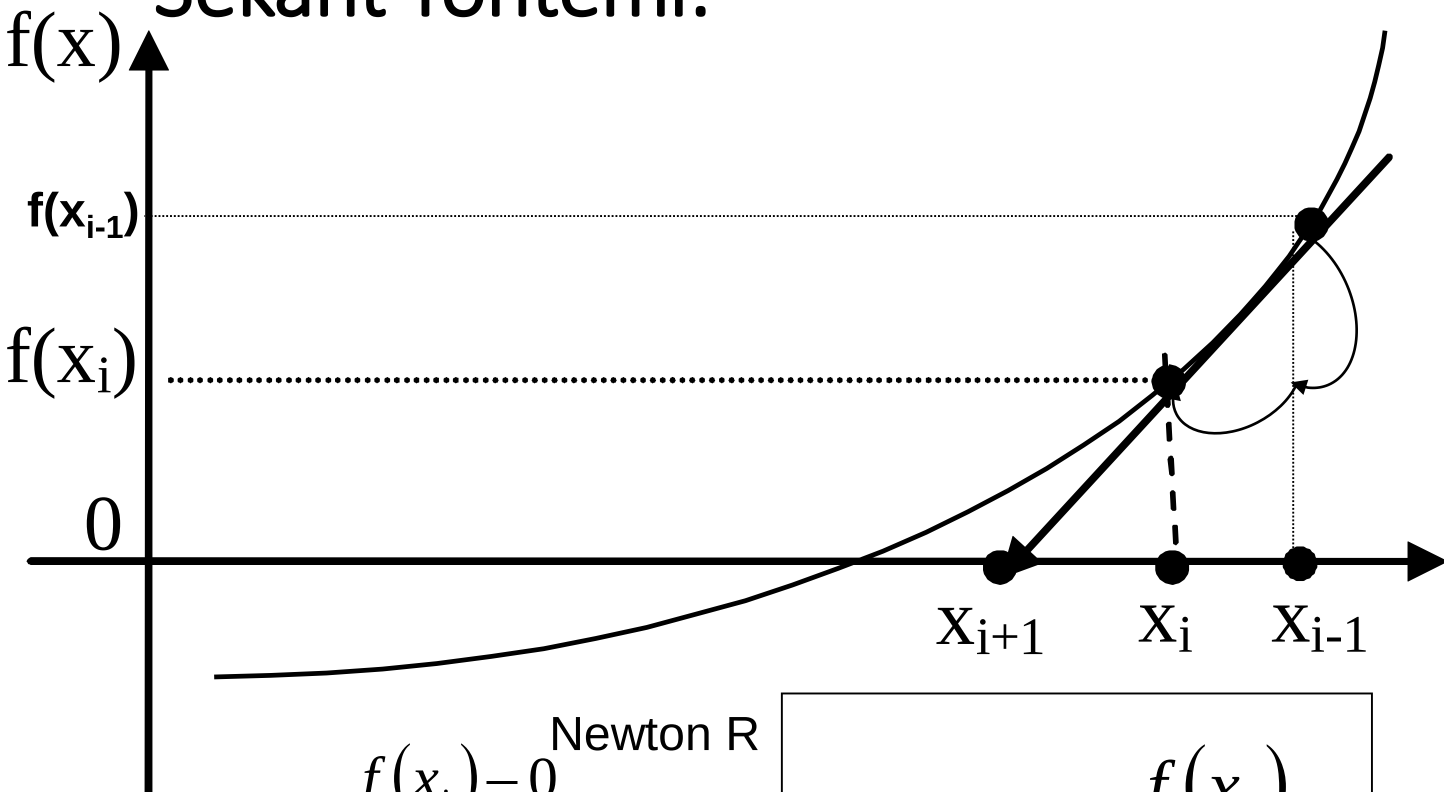
```
function [fturevx] = fturev(x)
fturevx=-1.0*exp(-x)-1;
```

```
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-f(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
```



Başla
İlk Değerleri Ata
∈ ,x0, n, Nmax
xkeski=x0
n<Nmax?
H
Bitiş
E
n=n+1
f'x <>0?
H
Sıfıra Bölme Hatası
E
xkyeni=xkeski- fx / f'x
xkyeni<>0?
H
Sıfıra Bölme Hatası
E
∈a = |(xi+1 − xi)/xi+1 *%100|
∈a<∈s ?
H
E
Kökü, tekrar sayısını
ve bağıl hatayı yazdır
n=Nmax

Sekant Yöntemi:



Newton R

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

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

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

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

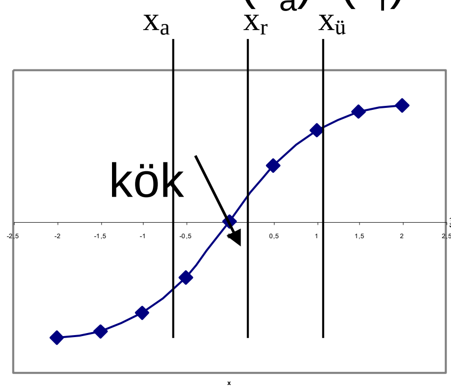
İkisinde de iki ilk tahmin değeri var

Regula Falsi

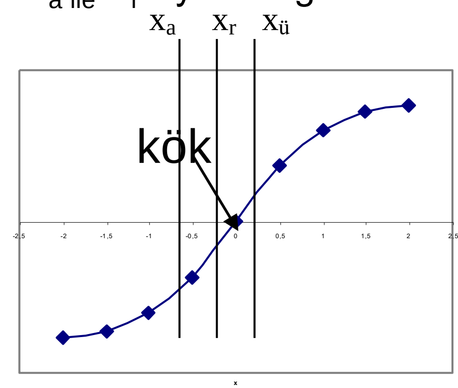
$$\boxed{x_r = x_{ü} - \frac{f(x_{ü})\left(x_a - x_{ü}\right)}{f(x_a) - f(x_{ü})}}$$

Güncellenecek sınır

- $f(x_a).f(x_r) < 0$ x_a ile x_r farklı bölgelerde xü(yeni)=xr
- $f(x_a).f(x_r) > 0$ x_a ile x_r aynı bölgelerde xa(yeni)=xr



Kök, x_a, x_r arasında



Kök, x_r, $x_ü$ arasında

Sekant

x_{i+1} x_i x_{i-1}

$$\boxed{x_{i+1} = x_i - \frac{f(x_i)\left(x_{i-1} - x_i\right)}{f(x_{i-1}) - f(x_i)}}$$