CENG 235 ALGORİTMALARLA SAYISAL ÇÖZÜMLEME

Prof. Dr. Tufan TURACI
tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü
- Hafta 5

5. Hafta Konular

- **Lineer Olmayan Denklemlerin Yaklaşık Çözüm Yöntemleri**

  --- Secant Yöntemi (Değişken Kesen Yöntemi)

  --- Teğet-Kiriş Yöntemi

Secant Yöntemi (Değişken Kesen Yöntemi)

Secant yöntemi, Newton Raphson yönteminin türevden kurtarılmış halidir.

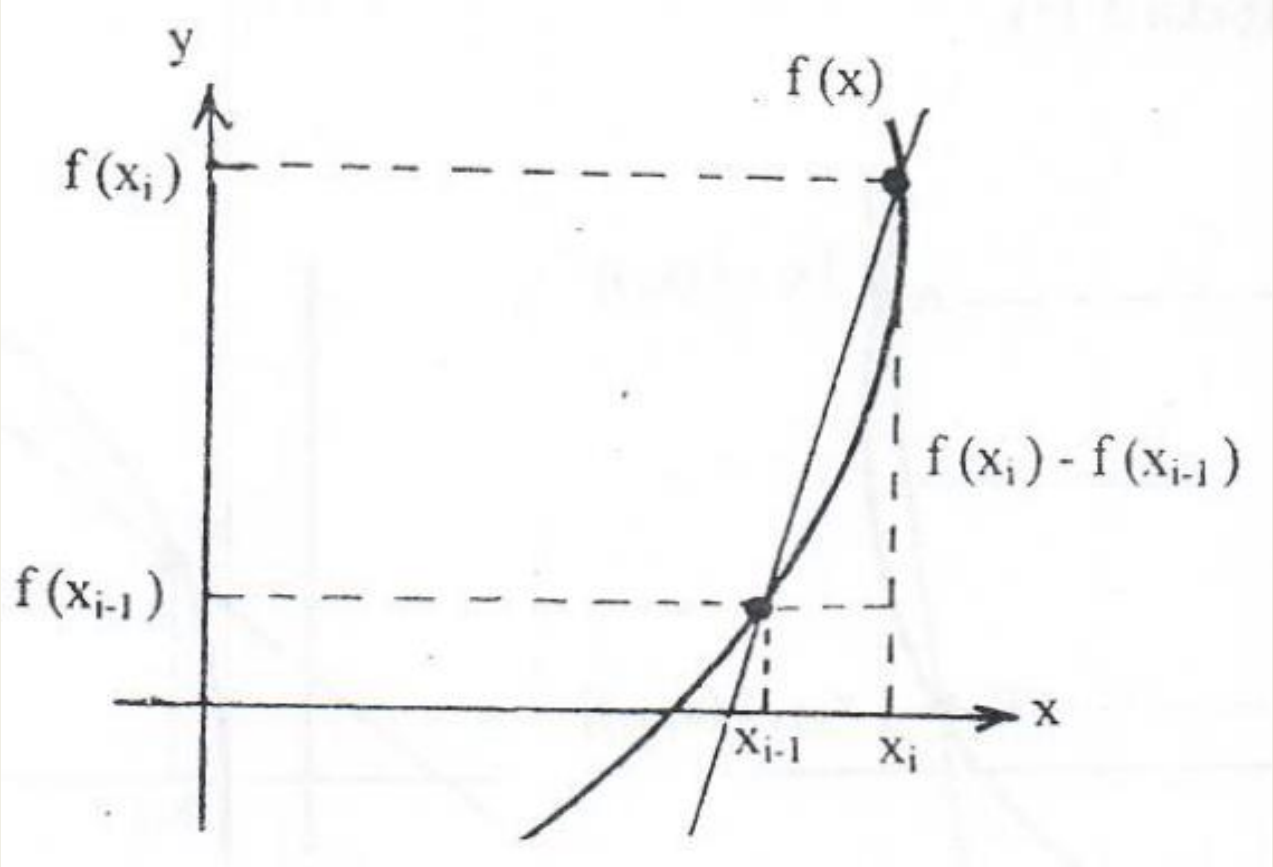
Newton-Raphson Yöntemi:

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

$f'(x_i)$ değeri sonlu fark yaklaşımı ile aşağıdaki şekilde yazılabilir:

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

$f'(x_i)$ değeri geometrik olarak aşağıdaki şekilde ifade edilir:



Böylece;

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

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

elde edilir.

Sonuç olarak,

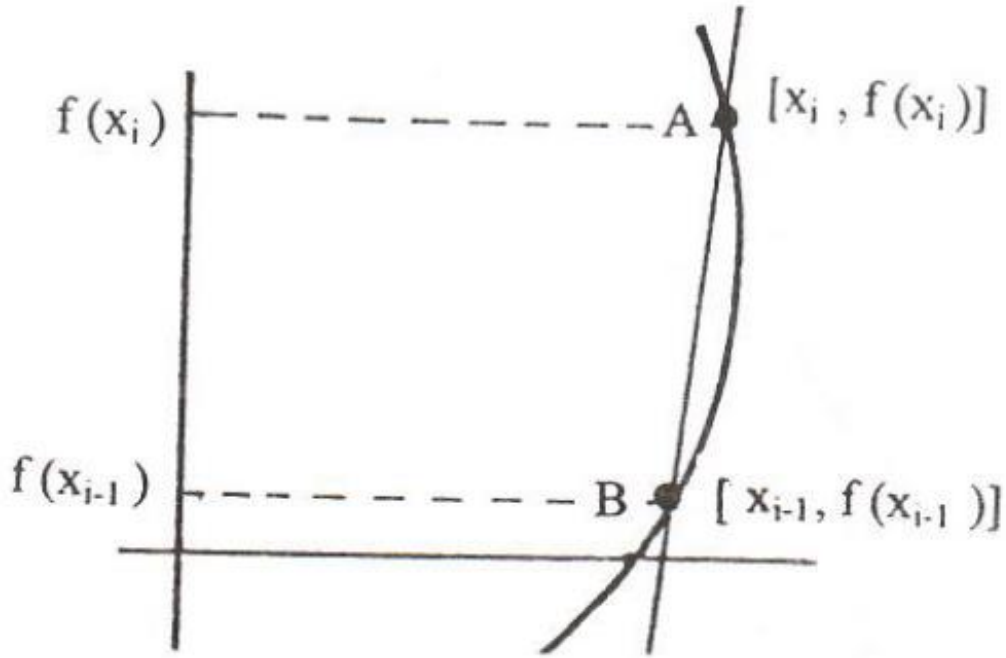
Sekant yönteminde iterasyon aşağıdaki şekilde elde edilir.

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

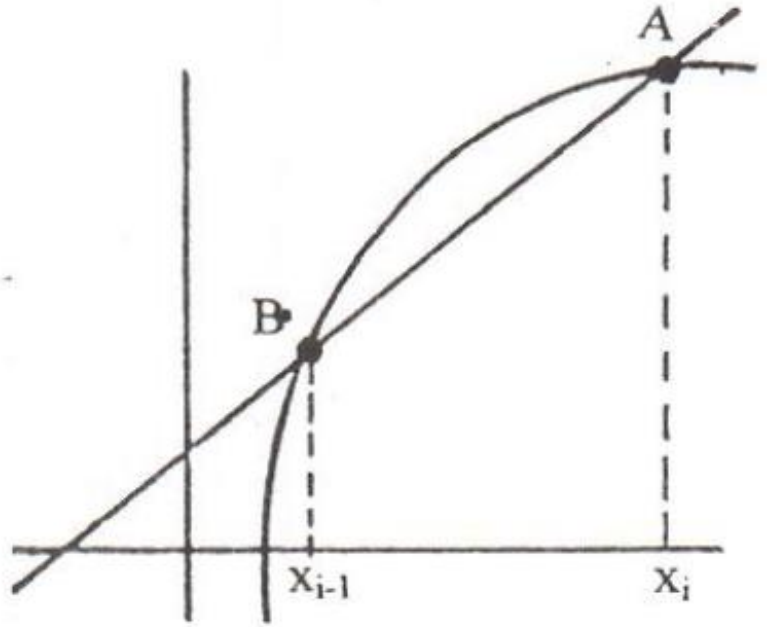
, burada $i = 1, 2, ----$ şeklindedir.

Sekant yönteminde, iterasyona baş-
lamak için iki başlangıç değeri gerekir.
- Bu iki başlangıç değeri için f(x) değerle-
rinin çarpımı pozitif olmasi gerekmez.
- Yöntem, Regula-Falsi yöntemine
benzer. Fakat, Regula-Falsi yönteminde-
gibi daima yakınsama olmaz.
Kökten uzaksamada olabilir.

Aşağıda; sekant yöntemin yakınsaması ve ıraksaması ile ilgili iki örnek verilmiştir.



Yakınsama



Iraksama

Örnek: $x^3 - 20 = 0$ denkleminin gerçek kökünü $x_0 = 2.5$, $x_1 = 3$ başlangıç değerleri için sekant yöntemi ile ilk iterasyon uygulayarak bulunuz. (4 ondalık kullanınız.)

$f(2.5) = -4.375$

$f(3) = 7$

$\Big\} \; f(2.5) \cdot f(3) < 0$

$[2.5, 3]$ arasında en az bir kök var.

Sekant yöntemi, iterasyon:

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

1. iterasyon (i = 1 için)

$$x_2 = \frac{x_0 \cdot f(x_1) - x_1 \cdot f(x_0)}{f(x_1) - f(x_0)}$$

2. iterasyon (i = 2 için)

$$x_3 = \frac{x_1 \cdot f(x_2) - x_2 \cdot f(x_1)}{f(x_2) - f(x_1)}$$

1. iterasyon

$$x_2 = \frac{(2.5).f(3) - 3.f(2.5)}{f(3) - f(2.5)}$$

$$x_2 = \frac{(2.5).(7) - 3.(-4.375)}{[7 - (-4.375)]}$$

$$x_2 = 2.6923$$

$f(2.6923) = 19.5151$

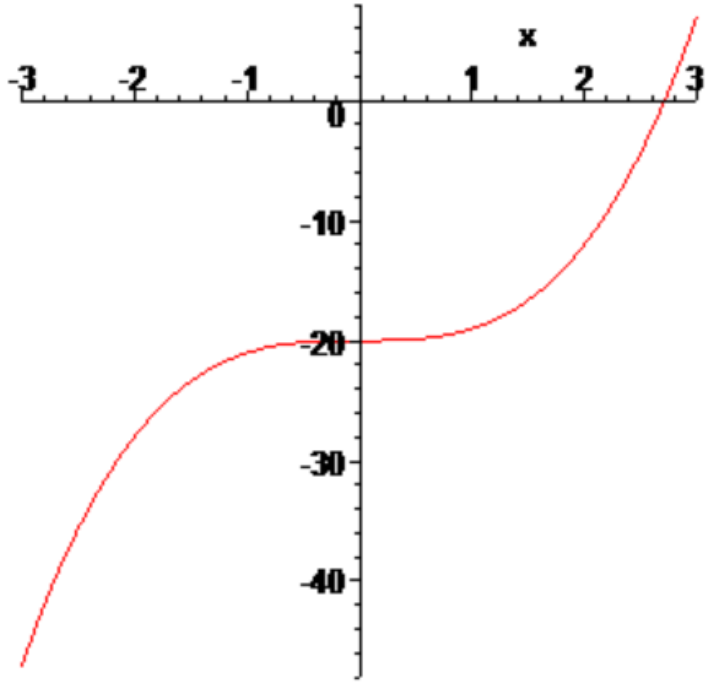
2. iterasyon

$$x_3 = \frac{3.f(2.6923) - (2.6923).(f(3))}{f(2.6923) - f(3)}$$

$$x_3 = 2.7122$$

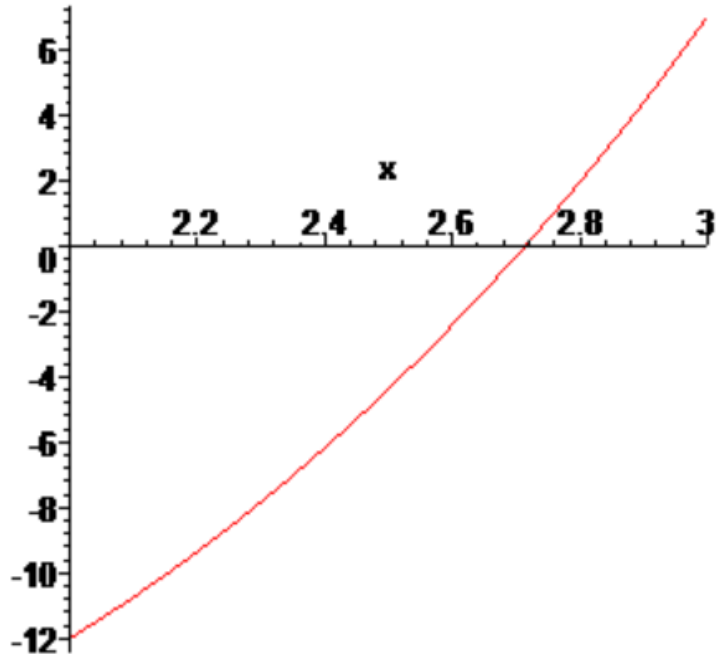
elde edilir.

```
plot(x*x*x-20,x=-3..3);
```



[-3,3] aralığındaki grafik

```
plot(x*x*x-20,x=2..3);
```



[2,3] aralığındaki grafik

Algoritma:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<math.h>
float F(float x)
{return pow(x,3)-20;}
int main()
{setlocale(LC_ALL, "Turkish");
float x0=2.5,x1=3,x2,x; int i=0;
printf("Yönteme başladığımız noktalar: x1=%.4f ve x2=%.4f\n",x0,x1);
do
{ x=x1;
x2=((x0*F(x1))-(x1*F(x0)))/(F(x1)-F(x0));
x0=x1;
x1=x2;
i++;
printf("%d. adımdaki yaklaşık kök= %.4f\n",i,x2);
}while (i<2);
printf("f(%.4f) = %.4f\n",x2,F(x2));
getch ();
return 0;
}
```

Ekran Çıktısı:

```
Yönteme başladığımız noktalar: x1=2,5000 ve x2=3,0000
1. adımdaki yaklaşık kök= 2,6923
2. adımdaki yaklaşık kök= 2,7122
f(2,7122) = -0,0482

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Aynı soruyu $\varepsilon = 10^{-4}$ hata ve 4 ondalık ile hesapladığımızda yaklaşık kök 4. adımda $x_4 = 2.7144$ elde edilir.

Ekran Çıktısı:

```
Yönteme başladığımız noktalar: x1=2,5000 ve x2=3,0000
1. adımda yaklaşık değer= 2,6923
2. adımda yaklaşık değer= 2,7122
3. adımda yaklaşık değer= 2,7144
4. adımda yaklaşık değer= 2,7144
yaklaşık kök 4. adımda hesaplanmıştır:
yaklaşık kök =2,7144
f(2,7144) = -0,0000

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Algoritma:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<math.h>
#define hata 0.0001
float F(float x)
{return pow(x,3)-20;}
int main()
{setlocale(LC_ALL, "Turkish");
float x0=2.5,x1=3,x2,x; int i=0;
printf("Yönteme başladığımız noktalar: x1=%.4f ve
x2=%.4f\n",x0,x1);
do
{ x=x1;
x2=((x0*F(x1))-(x1*F(x0)))/(F(x1)-F(x0));
x0=x1;
x1=x2;
i++;
printf("%d. adımda yaklaşık değer= %.4f\n",i,x2);
}while (fabs(x2-x)>hata);
printf("yaklaşık kök %d. adımda hesaplanmıştır:\n",i);
printf("yaklaşık kök =%.4f\n",x2);
printf("f(%.4f) = %.4f\n",x2,F(x2));
getch ();
return 0;
}
```

Örnek! $e^{-x} - x = 0$ denkleminin kökünü

$x_0 = 0$, $x_1 = 1$ başlangıç değerleri ile

Sekant yöntemi ile $\varepsilon = 10^{-3}$ hata ile

4 adımda ile çözünüz.

$f(0) = 1$ $\Rightarrow f(0).f(1) < 0$

$f(1) = -0.6321$ $[0,1]$ aralığında kök vardır.

1. iterasyon ($i = 1$ için)

$$x_2 = \frac{x_0 \cdot f(x_1) - x_1 \cdot f(x_0)}{f(x_1) - f(x_0)}$$

$$x_2 = \frac{0 \cdot f(1) - 1 \cdot f(0)}{f(1) - f(0)}$$

$$x_2 = 0.6127$$

2. iterasyon $f(0.6127) = -0.0708$

$$x_3 = \frac{x_1 . f(x_2) - x_2 . f(x_1)}{f(x_2) - f(x_1)}$$

$$x_3 = \frac{1 . f(0.6127) - (0.6127) . f(1)}{f(0.6127) - f(1)}$$

$$x_3 = 0.5638$$

3. iterasyon $f(0.5638) = 0.0052$

$$x_4 = \frac{x_2 \cdot f(x_3) - x_3 \cdot f(x_2)}{f(x_3) - f(x_2)}$$

$$x_4 = \frac{(0.6127) \cdot f(0.5638) - (0.5638) \cdot (f(0.6127))}{f(0.5638) - f(0.6127)}$$

$$x_4 = 0.5672$$

4. iterasyon $f(0.5672) = 0.0000$

$$x_5 = \frac{x_3 \cdot f(x_4) - x_4 \cdot f(x_3)}{f(x_4) - f(x_3)}$$

$$x_5 = \frac{(0.5638) \cdot f(0.5672) - (0.5672) \cdot f(0.5638)}{f(0.5672) - f(0.5638)}$$

$$x_5 = 0.5671$$

$|x_5 - x_4| = 0.0001 < 0.001$

Ard. dan yaklaşık kök hesaplanmıştır.

Algoritma:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<math.h>
#define hata 0.001
float F(float x)
{return exp(-x)-x;}
int main()
{setlocale(LC_ALL, "Turkish");
float x0=0,x1=1,x2,x; int i=0;
printf("Yönteme başladığımız noktalar: x0=%.4f ve
x1=%.4f\n",x0,x1);
do
{ x=x1;
x2=((x0*F(x1))-(x1*F(x0)))/(F(x1)-F(x0));
x0=x1;
x1=x2;
i++;
printf("%d. adımda yaklaşık değer= %.4f\n",i,x2);
}while (fabs(x2-x)>hata);
printf("yaklaşık kök %d. adımda hesaplanmıştır:\n",i);
printf("yaklaşık kök =%.4f\n",x2);
printf("f(%.4f) = %.4f\n",x2,F(x2));
getch ();
return 0;
}
```

Ekran Çıktısı:

```
Yönteme başladığımız noktalar: x0=0,0000 ve x1=1,0000
1. adımda yaklaşık değer= 0,6127
2. adımda yaklaşık değer= 0,5638
3. adımda yaklaşık değer= 0,5672
4. adımda yaklaşık değer= 0,5671
yaklaşık kök 4. adımda hesaplanmıştır:
yaklaşık kök =0,5671
f(0,5671) = -0,0000

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Çalışma Sorusu : $\ln x - \cos x = 0$ denkleminin kökünü

$x_0 = 2.5$, $x_1 = 3$ başlangıç değerleri ile

Sekant yöntemi ile $\varepsilon = 10^{-3}$ hata ve

4 rakamla hesaplayınız.

Yanıt: $x_k = 1.30030$ (6. iterasyonda hesaplanmıştır)

Teğet-Kiriş Yöntemi

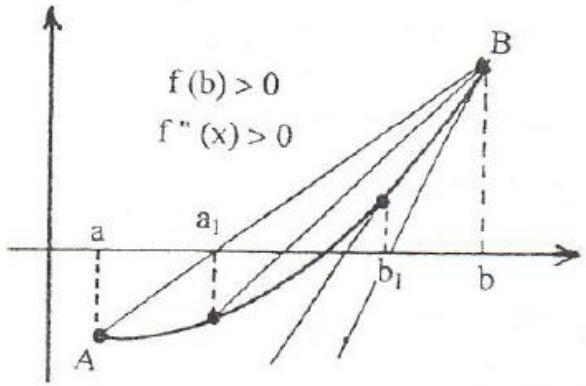
- Newton-Raphson yöntemi ile Regula-Falsi yönteminin kullanıldığı bir yöntemdir.
- Teğet x-eksenini grafiğin konveks tarafından keser, kiriş de konkav tarafından keser
- Böylece kökün bulunduğu aralık bir iterasyon sonucunda her iki yanlarda daraltılmış olur.

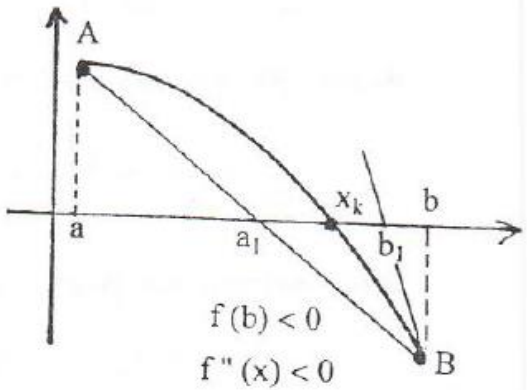
* $[a,b]$ aralığı her adımdan sonra $[a_k, b_k]$ şeklinde değişir.
* Adım sayısı arttıkça $|b_k - a_k| \to 0$ olur.

$x_k \in [a_0, b_0]$ olsun.

- Eğer $f(b).f''(x) > 0$ ise teğet B noktasından çizilir.

- Aşağıdaki şekilde bu durum gösterilmiştir.





- Bu durumda

$$b_{k+1} = b_k - \frac{f(b_k)}{f'(b_k)} \rightarrow \text{Newton - Raphson}$$

$$a_{k+1} = \frac{a_k.f(b_k) - b_k.f(a_k)}{f(b_k) - f(a_k)} \rightarrow \text{Regula-Falsi}$$

k=0 için $[a_1, b_1]$ → yeni aralık

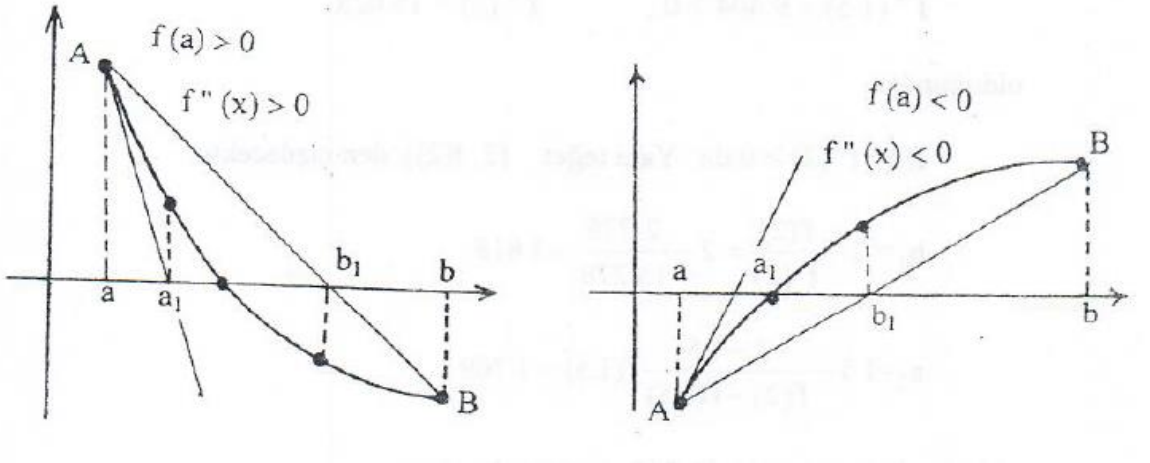
k=1 için $[a_2, b_2]$ → yeni aralık

⋮ ⋮

$x_k \in [a_0, b_0]$ olsun.

- Eğer $f(a) \cdot f''(x) > 0$ ise teğet A noktasından çizilir.

- Aşağıdaki şekilde bu durum gösterilmiştir.



- Bu durumda

$$a_{k+1} = a_k - \frac{f(a_k)}{f'(a_k)}$$ → Newton - Raphson

$$b_{k+1} = \frac{b_k \cdot f(a_k) - a_k \cdot f(b_k)}{f(a_k) - f(b_k)}$$ → Regula-Falsi

k=0 için $[a_1, b_1]$ → yeni aralık

k=1 için $[a_2, b_2]$ → yeni aralık

⋮ ⋮

Örnek 1: $2e^x - \ln x + x - 13.31 = 0$ denkleminin $[1.5, 2]$ kapalı aralığındaki kökünü Teğet-kiriş yöntemi ile $\varepsilon = 10^{-4}$ hata ve 4 ondalık kullanarak hesaplayınız.

$f(1.5) = -2.2527$

$f(2) = 2.7750$

$f''(1.5) = 9.4078$

$f''(2) = 15.0281$

$f(2) \cdot f''(2) > 0$ old. dan teğet (2, f(2)) nok.dan çizilir.

$a_0 = 1.5$ ve $b_0 = 2$

$f'(1.5) = 9.2967$

$f'(2) = 15.2781$

$$b_1 = b_0 - \frac{f(b_0)}{f'(b_0)} = 2 - \frac{f(2)}{f'(2)}$$

$$= 2 - \frac{2.7750}{15.2781} = 1.8184$$

$\star b_1$ için Newton-Raphson uygulandı.

★ a_1 için Regula-Falsi uygulandı.

$$a_1 = \frac{a_0 f(b_0) - b_0 f(a_0)}{f(b_0) - f(a_0)}$$

$$= \frac{(1.5) f(2) - (2)(f(1.5))}{f(2) - f(1.5)}$$

$$= \frac{(1.5)(2.7750) - 2(-3.2521)}{(2.7750) - (-3.2521)}$$

$$= \frac{4.1625 + 6.5042}{6.0271} = 1.7698$$

Yeni aralık : $[1.7698 , 1.8184]$

$1.8184 - 1.7698 > 10^{-4}$ devam ediyor...

2. iterasyon

$a_1 = 1.7698$ $\quad b_1 = 1.8184$

$f(1.7698) = -0.3718$ $\quad f(1.8184) = 0.2340$

$f''(1.7698) = 12,0585$ $\quad f''(1.8184) = 12.6260$

$f'(1.7698) = 12.1742$ $\quad f'(1.8184) = 12.7737$

$f(1.8184) \cdot f''(1.8184) > 0$ olduğundan

teğet $(1.8184, f(1.8184))$ noktasından çizilir.

b_2 için newton-raphson uygulanır.

$$b_2 = b_1 - \frac{f(b_1)}{f'(b_1)} = 1.8184 - \frac{f(1.8184)}{f'(1.8184)}$$

$$= 1.8184 - \frac{0.2340}{12.7737}$$

$$= 1.8000$$

a_2 için Regula-Falsi uygulanır.

$$a_2 = \frac{a_1 \cdot f(b_1) - b_1 \cdot f(a_1)}{f(b_1) - f(a_1)}$$

$$= \frac{(1.7698) \cdot f(1.8184) - (1.8184) \cdot f(1.7698)}{f(1.8184) - f(1.7698)}$$

$$= 1.7996$$

Yeni aralık : $[\overset{a_2}{1.7696}, \overset{b_2}{1.8000}]$

$1.8000 - 1.7696 > 10^{-4}$ devam ediriz.

3. iterasyon

$a_2 = 1.7696$ $\quad b_2 = 1.8000$

$f(1.7696) = -0.0035$ $\quad f(1.8000) = 0.0321$

$f''(1.7696) = 12.4033$ $\quad f''(1.8000) = 12.4085$

$f'(1.7696) = 12.5388$ $\quad f'(1.8000) = 12.5443$

$f(1.8000) \cdot f''(1.8000) > 0$ olduğundan

teğet $(1.8000, f(1.8000))$ noktasından çizilir.

b_3 için Newton-raphson

$$b_3 = b_2 - \frac{f(b_2)}{f'(b_2)} = 1.7999$$

a_3 için Regula-Falsi

$$a_3 = \frac{a_2 \cdot f(b_2) - b_2 \cdot f(a_2)}{f(b_2) - f(a_2)} = 1.7999$$

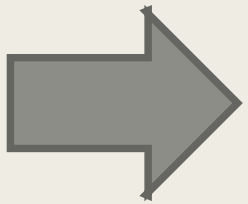
$|b_3 - a_3| = 0.0000 < 10^{-4}$ olduğundan

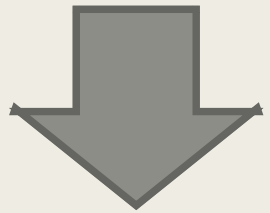
$x_k = 1.7999$ elde edilir.

$f(1.7999) = 0.0000$ 'dır.

Algoritma:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<math.h>
#define hata 0.0001
float F(float x)
{return 2*exp(x)-log(x)+x-13.31;}
float FT(float x)
{return 2*exp(x)-(1/x)+1;}
float FTT(float x)
{return 2*exp(x)+(1/pow(x,2));}
int main()
{setlocale(LC_ALL, "Turkish");
float a1,b1,a0,b0; int i=0;
printf("a değerini giriniz: ");
scanf("%f", &a0);
printf("b değerini giriniz: ");
scanf("%f", &b0);
if ((F(a0) * F(b0) >= 0))
        { printf("%.4f ve %.4f arasında kök yoktur...",a0,b0);
          return 0;}
do
{printf("\n");
printf ("f(%.4f)= %.4f\n",a0,F(a0));
printf ("f(%.4f)= %.4f\n",b0,F(b0));
printf ("f "(%.4f)= %.4f\n",a0,FTT(a0));
printf ("f "(%.4f)= %.4f\n",b0,FTT(b0));
printf ("f '(%.4f)= %.4f\n",a0,FT(a0));
printf ("f '(%.4f)= %.4f\n",b0,FT(b0));
```





```
if ((F(a0)*FTT(a0))>0) { a1=a0-(F(a0)/FT(a0));
                         b1=((b0*F(a0))-(a0*F(b0)))/(F(a0)-F(b0));
                         printf("Teğet (%.4f,f(%.4f)) noktasından çizilmiştir...\n",a0,a0);}

if ((F(b0)*FTT(b0))>0) {b1=b0-(F(b0)/FT(b0));
                         a1=((a0*F(b0))-(b0*F(a0)))/(F(b0)-F(a0));
                         printf("Teğet (%.4f,f(%.4f)) noktasından çizilmiştir...\n",b0,b0); }
a0=a1;
b0=b1;
i++;
printf("%d. adımda yeni aralık= (%.4f , %.4f)\n",i,a1,b1);
printf("\n");
}while (fabs(a1-b1)>hata);

printf("yaklaşık kök %d. adımda hesaplanmıştır:\n",i);
printf("yaklaşık kök =%.4f\n",a1);
printf("f(%.4f) = %.4f\n",a1,F(a1));
getch ();
return 0;
}
```

Ekran Çıktısı:

```
a değerini giriniz: 1,5
b değerini giriniz: 2

f(1,5000)= -3,2521
f(2,0000)= 2,7750
f''(1,5000)= 9,4078
f''(2,0000)= 15,0281
f'(1,5000)= 9,2967
f'(2,0000)= 15,2781
Teğet (2,0000,f(2,0000)) noktasından çizilmiştir...
1. adımda yeni aralık= (1,7698 , 1,8184)


f(1,7698)= -0,3718
f(1,8184)= 0,2340
f''(1,7698)= 12,0585
f''(1,8184)= 12,6260
f'(1,7698)= 12,1742
f'(1,8184)= 12,7737
Teğet (1,8184,f(1,8184)) noktasından çizilmiştir...
2. adımda yeni aralık= (1,7996 , 1,8000)
```

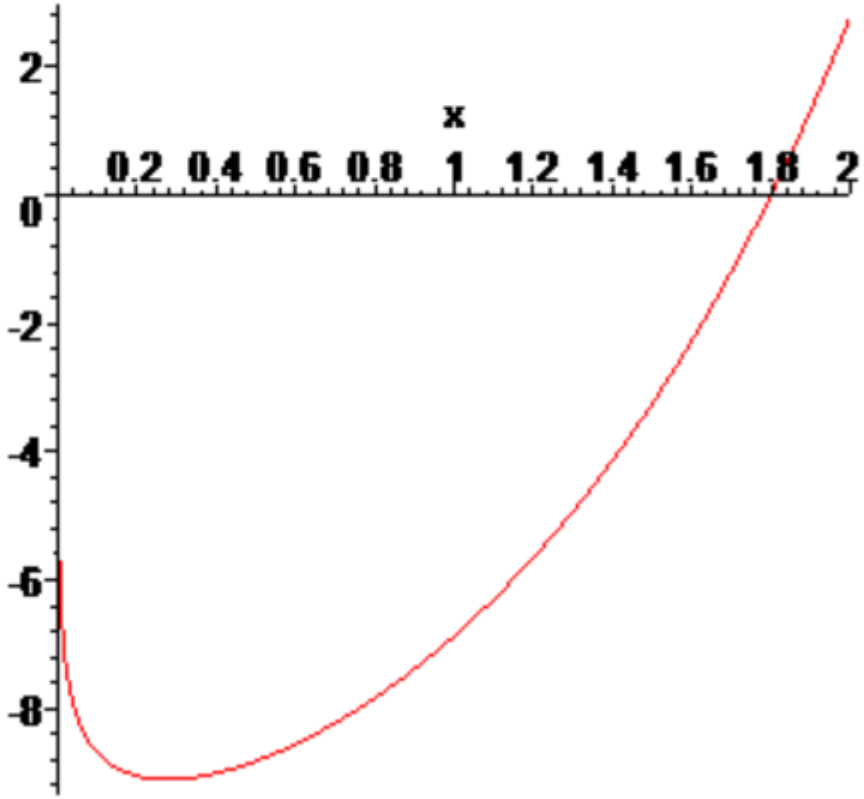
```
f(1,7996)= -0,0035
f(1,8000)= 0,0021
f''(1,7996)= 12,4033
f''(1,8000)= 12,4085
f'(1,7996)= 12,5388
f'(1,8000)= 12,5443
Teğet (1,8000,f(1,8000)) noktasından çizilmiştir...
3. adımda yeni aralık= (1,7999 , 1,7999)

yaklaşık kök 3. adımda hesaplanmıştır:
yaklaşık kök =1,7999
f(1,7999) = -0,0000

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

[0,2] kapalı aralığında fonksiyonun grafiği aşağıda verilmiştir:

```
plot(2*exp(x)-log(x)+x-13.31,x=0..2);
```



Örnek2: $x^3 + 2x \cdot e^{-x} + x^2 - 5x - 5.115 = 0$ denk-

lemin [2,3] aralığındaki kökünü Taget-kiriş

yöntemi ile $\varepsilon = 10^{-5}$ hata ve 5 ondalık

kullanarak hesaplayınız.

Yanıt:

```
yaklaşık kök 4. adımda hesaplanmıştır:
yaklaşık kök =2,49989
f(2,49989) = 0,00000

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

$[a_0, b_0] = [2, 3]$

$[a_1, b_1] = [2.36656, 2.59638]$

$[a_2, b_2] = [2.49280, 2.50458]$

$[a_3, b_3] = [2.49987, 2.49990]$

$[a_4, b_4] = [2.49985, 2.49989]$

Kaynaklar

- Numerical Analysis, Richard L. Burden, Brooks/Cole Cengage Learning, Boston., 2009.
- Numerical Methods for Mathematics, Science, and Engineering, 2nd Edition, John H. Mathews, Prentice Hall International Edition, 1992.
- Nümerik Analiz, (Numerical Analysis, D. Kincaid, W. Cheney, 3rd ed.(2002)), Nuri Özalp, Elif Demirci, Gazi Kitabevi Yayınları, 2012.
- Sayısal Analiz ve Mühendislik Uygulamaları, İrfan Karagöz, Nobel Yayıncılık, 2011.
- Sayısal Çözümleme, Recep Tapramaz, Literatür yayıncılık, 2002.
- Bilgisayar Uygulamalı Sayısal Analiz Yöntemleri, Eyüp Sabri Türker, Engin Can, II. Baskı, Değişim Yayınları.