

CENG 235 ALGORİTMALARLA SAYISAL ÇÖZÜMLEME

Prof. Dr. Tufan TURACI

tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü
- Hafta 4

4. Hafta Konular

- **Lineer Olmayan Denklemlerin Yaklaşık Çözüm Yöntemleri**
 - **Bisection (Yarılama) Yöntemi**
 - **Regula-Falsi Yöntemi**

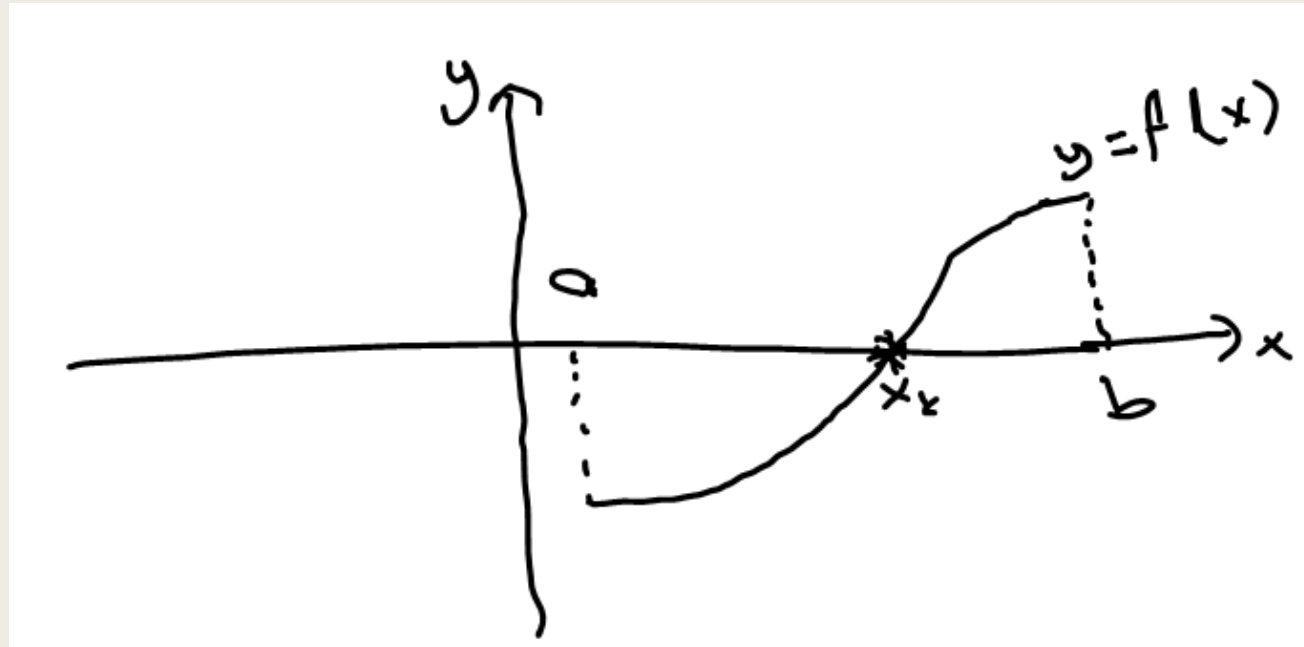
Bisection (Yarılama) Yöntemi

— f , bir aralığın uç noktalarında zıt işaretlere sahip sürekli bir fonksiyon olsun.

($[a, b]$ de f sürekli ve $f(a) \cdot f(b) < 0$ olsun.)

--- Bu durumda f fonksiyonu $[a, b]$ de bir sifıra sahiptir.

($\exists r \in (a, b)$ vardır öyle ki $f(r) = 0$ dır.)



Yorulanma yöntemi; aralığın ikiye
bölünerek, kâkin bulunduğu aralığın küçül-
tülmesiyle köke yaklaşıma dağınmaktadır.

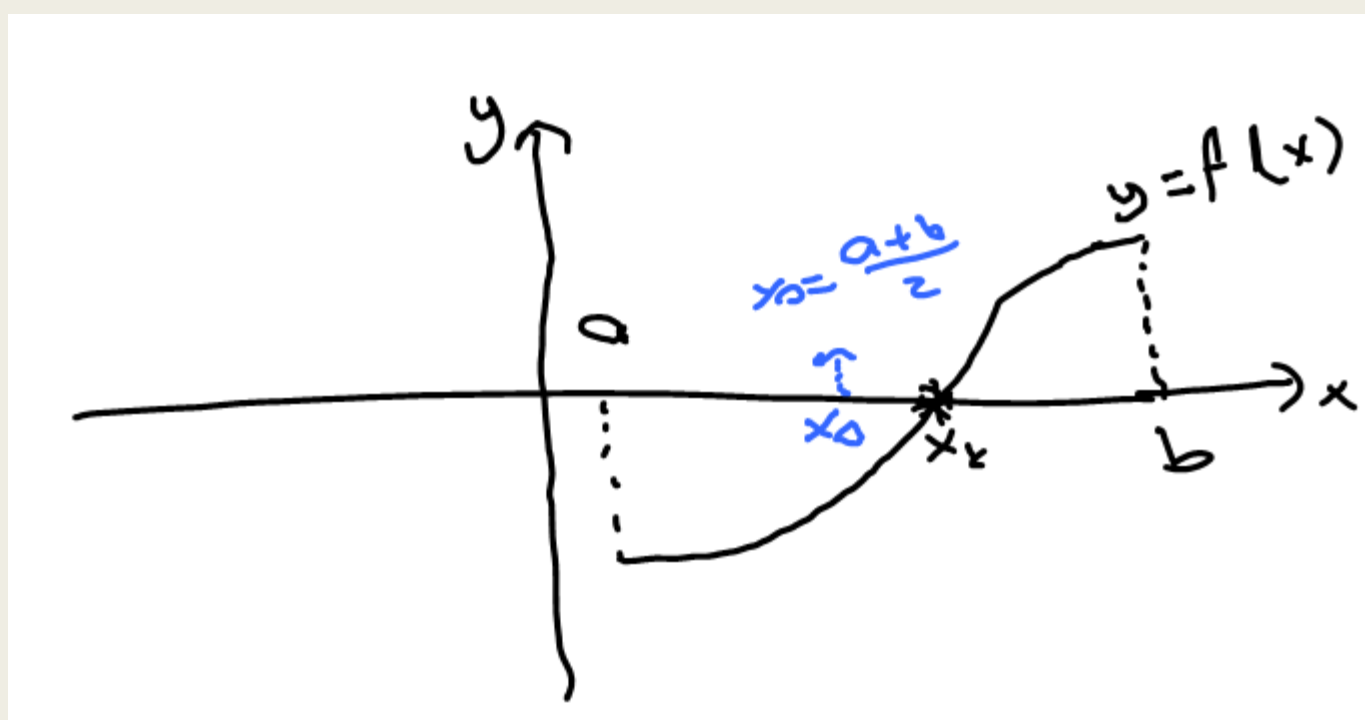
$x_k \rightarrow$ kök a nok $üze$ $x_k \in (a, b)$ olsun.

$x_0 = \frac{a+b}{2}$ alalım.

$f(x_0) \cdot f(a) < 0$ ise aralık $[a, x_0]$ olur.

$f(x_0) \cdot f(b) < 0$ ise aralık $[x_0, b]$ olur.

* Köke yaklaşıma kadar iterasyon
bu şekilde devam eder.



Örnekte $x_k \in (a, b)$ aralığı 1. iterasyon
dan sonra $x_k \in (x_0, b)$ olarak
güncellenmiştir.

Örnek $\sin x + \cos x - x = 0$ denkleminin $[0, 2]$ kapalı aralığındaki kökünü $\epsilon = 0.002$ hata ile yarılama yöntemi kullanarak 4 ondalık ile bulunuz.

$$f(0) = 1$$

$$f(0) \cdot f(2) < 0$$

$$f(2) = -1.5068$$

$$x_0 = \frac{0+2}{2} = 1 \quad f(1) = 0.3818$$

$$f(0) \cdot f(1) > 0 \rightarrow \text{kök yok}$$

$$f(1) \cdot f(2) < 0 \rightarrow \text{kök var}$$

$$\text{Yeni aralık } x_k \in [1, 2]$$

2. iterasyon

Aralık: $[1, 2]$

$$x_1 = \frac{1+2}{2} = 1.5 \quad f(1.5) = -0.4318$$

$$\begin{aligned} f(1) &> 0 \\ f(2) &< 0 \end{aligned}$$

$$f(1.5) < 0$$

Yeni aralık $[1, 1.5]$

$|x_1 - x_0| > \epsilon \rightarrow$ iterasyon devam

3. iterasyon

Aralık: $[1, 1.5]$

$$f(1) > 0$$

$$f(1.5) < 0$$

$$x_2 = \frac{1 + 1.5}{2} = 1.25$$

$$f(1.25) = 0.0143 > 0$$

$$f(1.25) \cdot f(1.5) < 0 \quad \text{old. den}$$

yeni aralık $[1.25, 1.5]$

$$|x_2 - x_1| > \varepsilon \rightarrow \text{devam}$$

4. iteration

Aralık $[1.25, 1.5]$

$$f(1.25) > 0$$

$$f(1.5) < 0$$

$$x_3 = \frac{1.25 + 1.5}{2} = 1.375$$

$$f(1.375) = -0.1996 < 0$$

Yeni aralık $[1.25, 1.375]$
 $|x_3 - x_2| > \epsilon \rightarrow \text{devam}$

5. iterasyon

Aralık: $[1.25, 1.375]$

$$f(1.25) > 0$$

$$f(1.375) < 0$$

$$x_4 = \frac{1.25 + 1.375}{2} = 1.3125$$

$$f(1.3125) = -0.0902 < 0$$

✓
yeni aralık: $[1.25, 1.3125]$

$$|x_4 - x_3| > \varepsilon \rightarrow \text{devam}$$

G. iterasyon

Aralık: $[1.25, 1.3125]$

$$f(1.25) > 0 \quad x_5 = \frac{1.25 + 1.3125}{2} = 1.2813$$

$$f(1.3125) < 0 \quad f(1.2813) = -0.0374 < 0$$

↙
yeni aralık: $[1.25, 1.2813]$

$$|x_5 - x_4| > \varepsilon \rightarrow \text{devam}$$

7. iteration

Aralık: $[1.25, 1.2813]$

$$f(1.25) > 0$$

$$x_6 = \frac{1.25 + 1.2813}{2} = 1.2656$$

$$f(1.2813) < 0$$

$$\text{ } \quad \quad \quad f(x_6) < 0$$

Yeni aralık: $[1.25, 1.2656]$

$$|x_6 - x_5| > \varepsilon \rightarrow \text{devam.}$$

8. iterasyon

Aralık: $[1.25, 1.2656]$

$$f(1.25) > 0$$

$$x_2 = \frac{1.25 + 1.2656}{2} = 1.2578$$

$$f(1.2656) < 0$$

$$f(x_2) > 0$$

_____ ↙
yeni aralık $[1.2578, 1.2656]$

$$|x_2 - x_1| > \varepsilon \rightarrow \text{devam.}$$

9. iterasyon

Aralık: $[1.2578, 1.2656]$

$$f(1.2578) > 0$$

$$x_8 = \frac{1.2578 + 1.2656}{2}$$

$$f(1.2656) < 0$$

$$x_8 = 1.2617$$

$$f(x_8) < 0$$

└
yeni aralık: $[1.2578, 1.2617]$

$$|x_8 - x_7| > \epsilon \rightarrow \text{devam}$$

10. iterasyon:

Aralık: $[1.2578, 1.2617]$

$$f(1.2578) > 0 \quad x_9 = \frac{1.2578 + 1.2617}{2}$$

$$f(1.2617) < 0 \quad x_9 = 1.2598$$

$$|x_9 - x_8| = |1.2598 - 1.2617| \\ = 0.0019 < \varepsilon \quad \text{old. den}$$

Yalıtımsızlık $x_9 = 1.2598$ olarak

bulundu. ($\varepsilon = 0.002$ hata ile)

C kodu:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<math.h>
#define hata 0.002

float F(float x)
{ return sin(x)+cos(x)-x; }

int main()
{ float t;
  setlocale(LC_ALL, "Turkish");
  float a,b,x,x0,x1; int i=0;
  printf("a değerini giriniz: ");
  scanf("%f", &a);
  printf("b değerini giriniz: ");
  scanf("%f", &b);
```





```
if ((F(a) * F(b) >= 0)) { printf("%.4f ve %.4f arasında kök yoktur...",a,b);  
    return 0;}  
x0=(a + b) / 2;  
do { i++;  
    printf("%d. adımda yaklaşık kök= %.4f\t ",i,x0);  
    x1=x0;  
    if (F(x0) * F(a) < 0)  
        b = x0;  
    else  
        a = x0;  
    printf("yeni aralık= [%.4f , %.4f]\n",a,b);  
    x0=(a + b) / 2;  
} while (fabs(x0-x1)>hata);  
  
printf("%d. adımda döngüden çıkıldı. %d. adımdaki yaklaşık kök= %.4f\n",i+1,i+1,x0);  
printf("%.3f hata ile yaklaşık kök =%.4f\n",hata,x0);  
printf("f(%.4f)= %.3f",x0,F(x0));  
getch ();  
return 0;  
}
```

Ekran Çıktısı:

```
a değerini giriniz: 0
b değerini giriniz: 2
1. adımda yaklaşık kök= 1,0000   yeni aralık= [1,0000 , 2,0000]
2. adımda yaklaşık kök= 1,5000   yeni aralık= [1,0000 , 1,5000]
3. adımda yaklaşık kök= 1,2500   yeni aralık= [1,2500 , 1,5000]
4. adımda yaklaşık kök= 1,3750   yeni aralık= [1,2500 , 1,3750]
5. adımda yaklaşık kök= 1,3125   yeni aralık= [1,2500 , 1,3125]
6. adımda yaklaşık kök= 1,2813   yeni aralık= [1,2500 , 1,2813]
7. adımda yaklaşık kök= 1,2656   yeni aralık= [1,2500 , 1,2656]
8. adımda yaklaşık kök= 1,2578   yeni aralık= [1,2578 , 1,2656]
9. adımda yaklaşık kök= 1,2617   yeni aralık= [1,2578 , 1,2617]
10. adımda döngüden çıkıldı. 10. adımdaki yaklaşık kök= 1,2598
0,002 hata ile yaklaşık kök =1,2598
f(1,2598)= -0,002
-----
Process exited with return value 0
Press any key to continue . . .
```

Aynı denklemi: $\epsilon = 10^{-5}$ hata ve 5 ondalık
kullanarak çözümsü sayısal denklemin
kökü 1.25874 elde edilecektir.

```
a değerini giriniz: 0
b değerini giriniz: 2
1. adımda yaklaşık kök= 1,00000 yeni aralık= [1,00000 , 2,00000]
2. adımda yaklaşık kök= 1,50000 yeni aralık= [1,00000 , 1,50000]
3. adımda yaklaşık kök= 1,25000 yeni aralık= [1,25000 , 1,50000]
4. adımda yaklaşık kök= 1,37500 yeni aralık= [1,25000 , 1,37500]
5. adımda yaklaşık kök= 1,31250 yeni aralık= [1,25000 , 1,31250]
6. adımda yaklaşık kök= 1,28125 yeni aralık= [1,25000 , 1,28125]
7. adımda yaklaşık kök= 1,26563 yeni aralık= [1,25000 , 1,26563]
8. adımda yaklaşık kök= 1,25781 yeni aralık= [1,25781 , 1,26563]
9. adımda yaklaşık kök= 1,26172 yeni aralık= [1,25781 , 1,26172]
10. adımda yaklaşık kök= 1,25977 yeni aralık= [1,25781 , 1,25977]
11. adımda yaklaşık kök= 1,25879 yeni aralık= [1,25781 , 1,25879]
12. adımda yaklaşık kök= 1,25830 yeni aralık= [1,25830 , 1,25879]
13. adımda yaklaşık kök= 1,25854 yeni aralık= [1,25854 , 1,25879]
14. adımda yaklaşık kök= 1,25867 yeni aralık= [1,25867 , 1,25879]
15. adımda yaklaşık kök= 1,25873 yeni aralık= [1,25873 , 1,25879]
16. adımda yaklaşık kök= 1,25876 yeni aralık= [1,25873 , 1,25876]
17. adımda yaklaşık kök= 1,25874 yeni aralık= [1,25873 , 1,25874]
18. adımda döngüden çıkıldı. 18. adımdaki yaklaşık kök= 1,25874
0,00001 hata ile yaklaşık kök =1,25874
f(1,25874)= -0,00001
-----
Process exited with return value 0
Press any key to continue . . .
```

Örnek 1

$$x \cdot \sin x - 1 = 0$$

denklemi

nin $[0, 2]$ kapalı aralığındaki kökünü
 $\epsilon = 0.01$ hata ile yarılama yöntemi
kullanarak 6 ondalık ile buluyoruz.

$$\begin{array}{l} f(0) < 0 \\ f(2) > 0 \end{array} \Rightarrow \begin{array}{l} f(0) \cdot f(2) < 0 \\ x_k \in [0, 2] \text{ dir.} \end{array}$$

1. iterasyon $x_0 = \frac{0+2}{2} = 1 \Rightarrow f(1) < 0$

$f(1) \cdot f(2) < 0$ old. den $x_k \in [1, 2]$
eide edilir.

\vdots

İterasyonların devamı sonucunda **0,01** hata ile yaklaşık kök **1,117188** olarak bulunur.

Programın çıktısı aşağıdaki şekildedir:

```
a değerini giriniz: 0
b değerini giriniz: 2
1. adımda yaklaşık kök= 1,000000      yeni aralık= [1,000000 , 2,000000]
2. adımda yaklaşık kök= 1,500000      yeni aralık= [1,000000 , 1,500000]
3. adımda yaklaşık kök= 1,250000      yeni aralık= [1,000000 , 1,250000]
4. adımda yaklaşık kök= 1,125000      yeni aralık= [1,000000 , 1,125000]
5. adımda yaklaşık kök= 1,062500      yeni aralık= [1,062500 , 1,125000]
6. adımda yaklaşık kök= 1,093750      yeni aralık= [1,093750 , 1,125000]
7. adımda yaklaşık kök= 1,109375      yeni aralık= [1,109375 , 1,125000]
8. adımda döngüden çıkıldı. 8. adımdaki yaklaşık kök= 1,117188
0,01 hata ile yaklaşık kök =1,117188
f(1,117188)= 0,004208
-----
Process exited with return value 0
Press any key to continue . . .
```


Örnek 1 $x^3 + 2x^2 + 6x + 3 = 0$ denkleminin $[-1, 0]$ kapalı aralığındaki kökünü $\epsilon = 10^{-6}$ hata ile yarılama yöntemi ile kullanarak 6 ondalık ile buluyoruz.

$$\begin{aligned} f(0) &= 3 \\ f(-1) &= -2 \end{aligned} \Rightarrow \begin{aligned} &f(0) \cdot f(-1) < 0 \\ &[-1, 0] \text{ kök var.} \end{aligned}$$

1. iterasyon $x_0 = \frac{0 + (-1)}{2} = -0.5$

$$f(-0.5) > 0$$

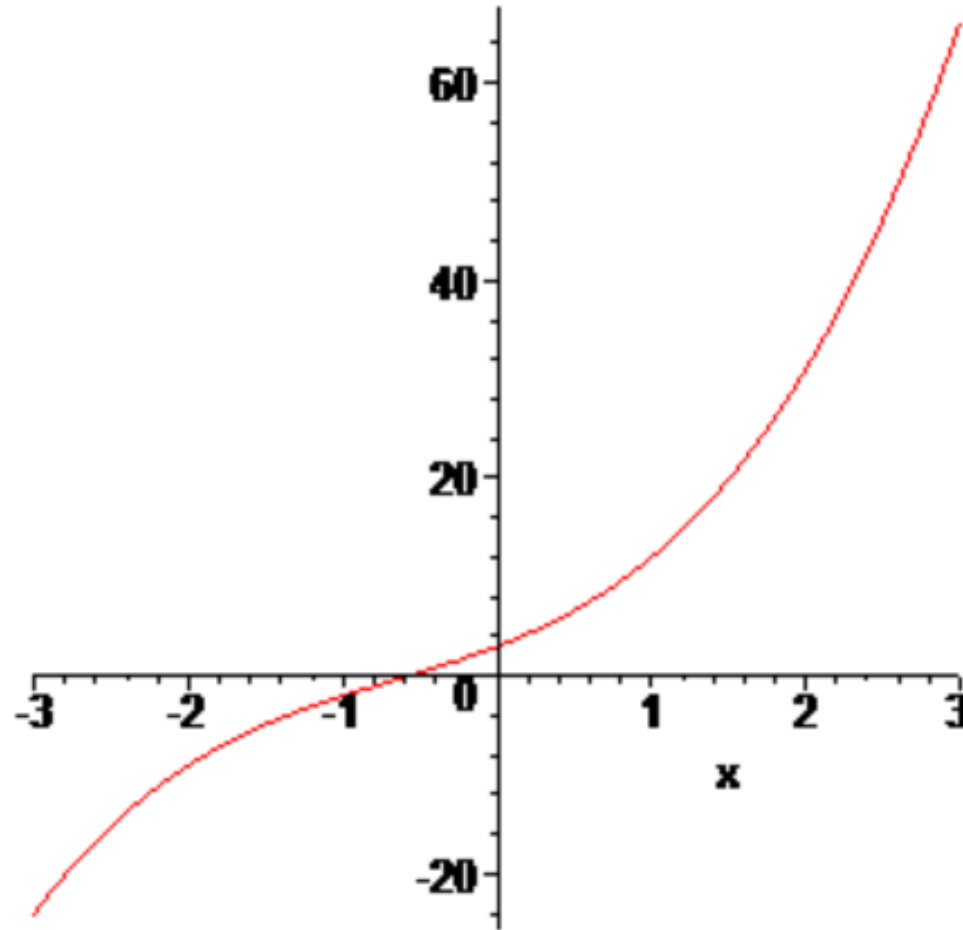
$$f(-1) \cdot f(-0.5) < 0 \text{ yeni aralık: } [-1, -0.5]$$

\vdots

İterasyonların devamı sonucunda **0,000001** hata ile yaklaşık kök **-0,579507** olarak bulunur.

Grafik:

```
plot(x*x*x+2*x*x+6*x+3,x=-3..3);
```



Programın çıktısı aşağıdaki şekildedir:

```
a değerini giriniz: -1
b değerini giriniz: 0
1. adımda yaklaşık kök= -0,500000      yeni aralık= [-1,000000, -0,500000]
2. adımda yaklaşık kök= -0,750000      yeni aralık= [-0,750000, -0,500000]
3. adımda yaklaşık kök= -0,625000      yeni aralık= [-0,625000, -0,500000]
4. adımda yaklaşık kök= -0,562500      yeni aralık= [-0,625000, -0,562500]
5. adımda yaklaşık kök= -0,593750      yeni aralık= [-0,593750, -0,562500]
6. adımda yaklaşık kök= -0,578125      yeni aralık= [-0,593750, -0,578125]
7. adımda yaklaşık kök= -0,585938      yeni aralık= [-0,585938, -0,578125]
8. adımda yaklaşık kök= -0,582031      yeni aralık= [-0,582031, -0,578125]
9. adımda yaklaşık kök= -0,580078      yeni aralık= [-0,580078, -0,578125]
10. adımda yaklaşık kök= -0,579102     yeni aralık= [-0,580078, -0,579102]
11. adımda yaklaşık kök= -0,579590     yeni aralık= [-0,579590, -0,579102]
12. adımda yaklaşık kök= -0,579346     yeni aralık= [-0,579590, -0,579346]
13. adımda yaklaşık kök= -0,579468     yeni aralık= [-0,579590, -0,579468]
14. adımda yaklaşık kök= -0,579529     yeni aralık= [-0,579529, -0,579468]
15. adımda yaklaşık kök= -0,579498     yeni aralık= [-0,579529, -0,579498]
16. adımda yaklaşık kök= -0,579514     yeni aralık= [-0,579514, -0,579498]
17. adımda yaklaşık kök= -0,579506     yeni aralık= [-0,579514, -0,579506]
18. adımda yaklaşık kök= -0,579510     yeni aralık= [-0,579510, -0,579506]
19. adımda yaklaşık kök= -0,579508     yeni aralık= [-0,579508, -0,579506]
0,000001 hata ile yaklaşık kök =-0,579507
f(-0,579507)= 0,000000
-----
Process exited with return value 0
Press any key to continue . . .
```

Çalışma Sorusu

$e^x - x - 2 = 0$ denkleminin $[1, 1.8]$ kapalı aralığındaki kökünü $\epsilon = 10^{-6}$ hata ile yarılama yöntemi ile kullanarak 6 ondalık ile bulunuz.

Yanıt: 1,146194

Çalışma Sorusu

$2^x - 5x + 2 = 0$ denkleminin $[0, 1]$ kapalı aralığındaki kökünü $\epsilon = 0.1$ hata ile yarılama yöntemi ile kullanarak 4 ondalık ile bulunuz.

Yanıt: 0.6875

Regula-Falsi Yöntemi

- Regula-Falsi yöntemi köke kirişler çizerek yaklaşır.
- Temel olarak yarılama yöntemine bağlı olsa da Regula-Falsi yöntemi daha hızlı yakınsar.
- Bu metodun uygulanabilmesi için Bisection (Yarılama) yönteminde olduğu gibi zıt işaretli fonksiyon değerine sahip iki başlangıç noktası gerekmektedir.
- Yani $f(a).f(b)<0$ olmalıdır.

İlk kök yaklaşımı:

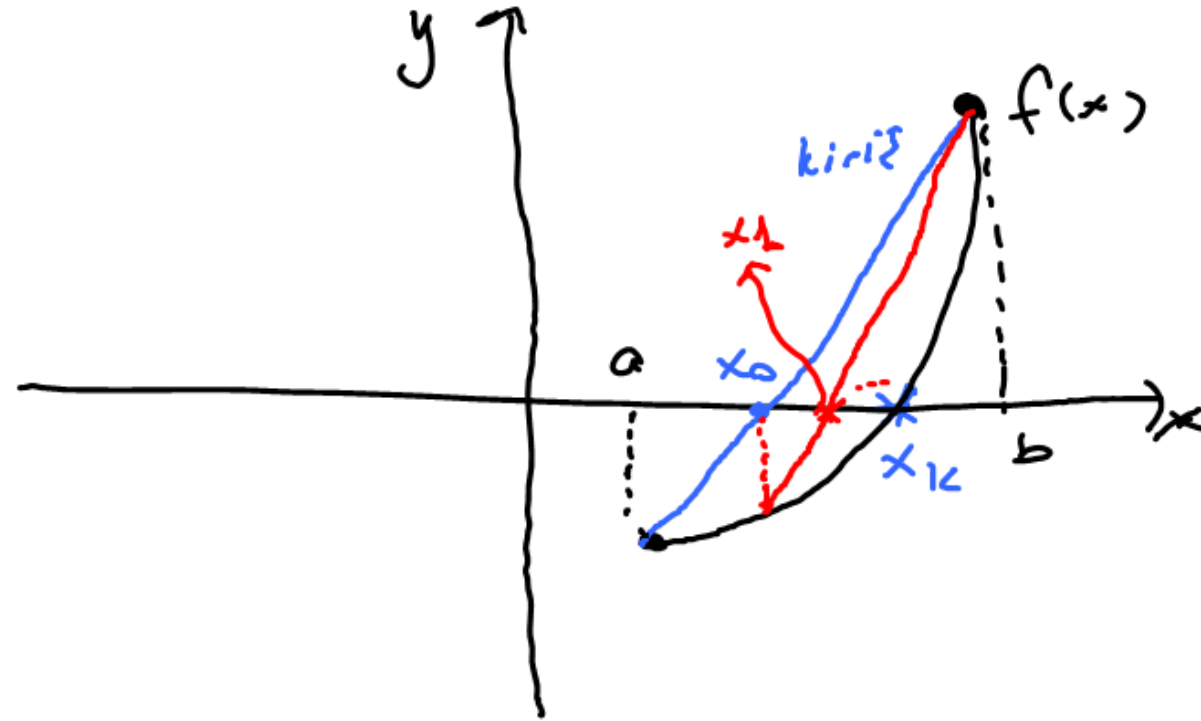
$$x_0 = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$

---- Yeni iterasyon için kökün bulunduğu aralıklar belirlenir ve yeni a ve b değerleriyle ikinci kök yaklaşımı elde edilir. Bu şekilde, belirlenen hata değerinden daha küçük kalana kadar, iterasyona devam edilir.

--- Yarılama yöntemine göre daha hızlı yakınsar.

--- İlk iterasyon kökün bulunduğu $[a,b]$ kapalı aralığındaki a ve b değerleri kullanılarak başlar.

Geometrik olarak aşağıdaki şekilde iterasyon devam eder:



$$x_k \in [a, b]$$

$$f(a) \cdot f(x_0) < 0 \quad \text{ise} \quad x_k \in [a, x_0] \quad (b = x_0 \text{ olur})$$

$$f(b) \cdot f(x_0) < 0 \quad \text{ise } x_c \in [x_0, b] \quad \left(\begin{array}{l} a = x_0 \\ \text{or} \end{array} \right)$$

AB) kirisinin eğimi:

$$m = \frac{f(b) - f(a)}{b - a}$$

Bir noktası ve eğimi belli bir
doğrunun den: $m = \frac{f(b) - f(a)}{b - a}$, $(a, f(a))$

$$y - f(a) = \frac{f(b) - f(a)}{b - a} \cdot (x - a)$$

$y = 0$ için x_0 elde edilir.

Yani: $x_0 = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$

Örnek: $x + \ln x - 5 = 0$ denkleminin

$[3.2, 4]$ kapalı aralığındaki kökünü

$\epsilon = 10^{-4}$ hata ile Regula-Falsi yöntemi

kuullanarak 4 ondalık ile bulunur.

$$f(3.2) = -0.6368 < 0$$

$$f(4) = 0.3863 > 0$$

$$x_0 = \frac{(3.2) \cdot f(4) - 4 \cdot f(3.2)}{f(4) - f(3.2)} = 3.6980$$

$$f(3.6980) = 0.0057 > 0$$

Yeni aralık: $[3.2, 3.6980]$ olur.

$$x_1 = \frac{(3.2) \cdot f(3.6980) - (3.6980) \cdot f(3.2)}{f(3.6980) - f(3.2)}$$

$$x_1 = 3.6935$$

$$|x_1 - x_0| = |3.6935 - 3.6980| = 0.0045 < 10^{-4}$$

$$f(3.6935) = 0.0001 > 0$$

Yeni aralık: $[3.2, 3.6935]$ elde edilir.

$$x_2 = \frac{(3.2) \cdot f(3.6935) - (3.6935) \cdot f(3.2)}{f(3.6935) - f(3.2)}$$

$$x_2 = 3.6939$$

$|x_1 - x_2| = 0.0001$ ok. iterasyon biter.

$\epsilon = 10^{-4}$ hata ile yaklaşık kök $x_k = 3.6939$

C kodu:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<math.h>
#define hata 0.0001
```

```
float F(float x)
{ return x+log(x)-5; }
```

```
int main()
{ setlocale(LC_ALL, "Turkish");
  float a,b,x,x0,x1; int i=0;
  printf("a değerini giriniz: ");
  scanf("%f", &a);
  printf("b değerini giriniz: ");
  scanf("%f", &b);
```





```
if ((F(a) * F(b) >= 0)) { printf("%.4f ve %.4f arasında kök yoktur...",a,b);  
    return 0;}  
x0=(a*F(b)-b*F(a)) / (F(b)-F(a));  
do { i++;  
    printf("%d. adımda yaklaşık kök= %.4f\t ",i,x0);  
    x1=x0;  
    if (F(x0) * F(a) < 0)  
        b = x0;  
    else  
        a = x0;  
    printf("yeni aralık= [%.4f,%.4f]\n",a,b);  
    x0=(a*F(b)-b*F(a)) / (F(b)-F(a));  
} while (fabs(x0-x1)>hata);  
printf("yaklaşık kök %d. adımda hesaplanmıştır:\n",++i);  
printf("yaklaşık kök =%.4f\n",x0);  
printf("f(%.4f) = %.4f\n",x0,F(x0));  
getch ();  
return 0;  
}
```

Ekran Çıktısı:

```
a değerini giriniz: 3,2
b değerini giriniz: 4
1. adımda yaklaşık kök= 3,6980   yeni aralık= [3,2000,3,6980]
2. adımda yaklaşık kök= 3,6935   yeni aralık= [3,2000,3,6935]
yaklaşık kök 3. adımda hesaplanmıştır:
yaklaşık kök =3,6934
f(3,6934) = 0,0000

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

Örnek: $1-x - e^{-2x} = 0$ denkleminin
[0.5, 1] kapalı aralığındaki kökünü
 $\epsilon = 10^{-3}$ hata ile Regula-Falsi yöntemi
kullanarak 3 ondalık ile bulunuz.

$$f(0.5) = 0.132 > 0 \quad x_k \in [0.5, 1]$$

$$f(1) = -0.135 < 0$$

$$x_0 = \frac{(0.5) \cdot f(1) - 1 \cdot f(0.5)}{f(1) - f(0.5)} = 0.747$$

$$f(0.747) = 0.029 > 0$$

$$\text{Yeni aralık: } [0.747, 1]$$

$$x_1 = \frac{(0.747) \cdot f(1) - 1 \cdot f(0.747)}{f(1) - f(0.747)}$$

$$x_1 = 0.791$$

$$|x_1 - x_0| > \epsilon = 10^{-3} \text{ iterasyon devam.}$$

$$f(0.791) = 0.003 > 0$$

$$\text{Yeni aralık: } [0.791, 1]$$

$$x_2 = \frac{(0.791) \cdot f(1) - 1 \cdot f(0.791)}{f(1) - f(0.791)}$$

$$x_2 = 0.796$$

$|x_2 - x_1| > \epsilon = 10^{-3}$ iterasyon devam eder.

$$f(0.796) > 0$$

Yeni Aralık: $[0.796, 1]$

$$x_3 = \frac{(0.796) \cdot f(1) - 1 \cdot f(0.796)}{f(1) - f(0.796)}$$

$$x_3 = 0.797$$

$$|x_3 - x_2| = 0.001 \quad \text{iterasyon biter.}$$

$$\epsilon = 10^{-3} \text{ hata ile } x_k = 0.797 \text{ dur.}$$

$$f(0.797) = 0.000$$

↓
yaklaşık kök.

İterasyonların devamı sonucunda **0,001** hata ile yaklaşık kök **0,797** olarak bulunur.

Programın çıktısı aşağıdaki şekildedir:

```
a değerini giriniz: 0,5
b değerini giriniz: 1
1. adımda yaklaşık kök= 0,747      yeni aralık= [0,747,1,000]
2. adımda yaklaşık kök= 0,791      yeni aralık= [0,791,1,000]
3. adımda yaklaşık kök= 0,796      yeni aralık= [0,796,1,000]
yaklaşık kök 4. adımda hesaplanmıştır:
yaklaşık kök =0,797
f(0,797) = 0,000

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

Çalışma
Sorusu

$x^3 - 5x^2 - 2x + 10 = 0$ denkleminin
[1,3] kapalı aralığındaki kökünü

$\epsilon = 0.1$ hata ile Regula-falsi yöntemini
kullanarak 4 ondalık ile bulunuz.

Yanıt: 1.4127

Çalışma
Sorusu

$e^{-x} - 2 = 0$ denkleminin
[-1,0] kapalı aralığındaki kökünü

$\epsilon = 0.1$ hata ile Regula-Falsi yöntemini
kullanarak 5 ondalık ile bulunuz.

Yanıt: -0.67669

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 Yyıncılık, 2011.
- Sayısal Çözümleme, Recep Tapramaz, Literatür yayıncılık, 2002.
- Bilgisayar Uygulamalı Sayısal Analiz Yöntemleri, Eyüp Sabri Türker, Engin Can, II. Baskı, Değişim Yayınları.