CENG 235 ALGORİTMALARLA SAYISAL ÇÖZÜMLEME

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

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü

- Hafta 1

1. Hafta Konular

- **Matematiksel Hatırlatmalar**

---Algoritmalarla Sayısal Çözümleme dersi (Nümerik Analiz, Sayısal Analiz) farklı matematiksel problemlere sayısal çözümler elde etmek için algoritmaların çalışmasını, geliştirilmesini ve analizini içerir. Sayısal Çözümleme, ***bilimsel hesaplama matematiği*** olarak adlandırılır.

---Çalıştığımız algoritmalar hızlı bilgisayarlarda kullanılmak için hedeflenir ve bu nedenle bir problemin çözümü elde edilmeden önce bir başka önemli adım devreye girer: algoritmayı bilgisayarla iletişime geçiren bir bilgisayar kodu veya programı yazılmak zorundadır.

---Matematiksel problemleri bilgisayarda sayısal olarak çözmek için ilgili algoritmaların geliştirilmelidir ve bu durum bilimsel hesaplama matematiği olarak ifade edilir.

---Algoritmalarla Sayısal Çözümleme dersinde bir çok farklı sayısal yöntemi ve algoritmaları göreceğiz.

--- C programlama dilinde verilen yöntemlerin uygulamalarını göreceğiz.

Matematiksel Hatırlatmalar

Tanım1:

Eğer f reel değişkenli reel bir fonksiyon ise bu durumda f fonksiyonunun c noktasındaki limiti (eğer mevcut ise) aşağıdaki gibi tanımlanır:

$$\lim_{x \to c} f(x) = L$$

denkleminin anlamı; her ε pozitif sayısı için x ve c arasındaki uzaklığın δ dan küçük kaldığı her durumda, $f(x)$ ve L arasındaki uzaklık ε dan küçük kalacak şekilde, ε a karşılık gelen bir δ sayısı vardır, yani

$$\text{her } 0<|x-c|<\delta \text{ için } |f(x)-L|<\varepsilon$$

Eğer bu özelliği sağlayan bir L sayısı yok ise, f nin c noktasında limiti yoktur.

Örnek: $\lim_{x \to 2} 3x+1 = 7$ old. gösteriniz.

$\forall \varepsilon > 0$, $\exists \delta > 0$ vardır ki

$|x-2| < \delta$ old. $|3x+1-7| < \varepsilon$ olsun.

$$\Downarrow$$

$$|3x-6| < \varepsilon$$

$$3.|x-2| < \varepsilon$$

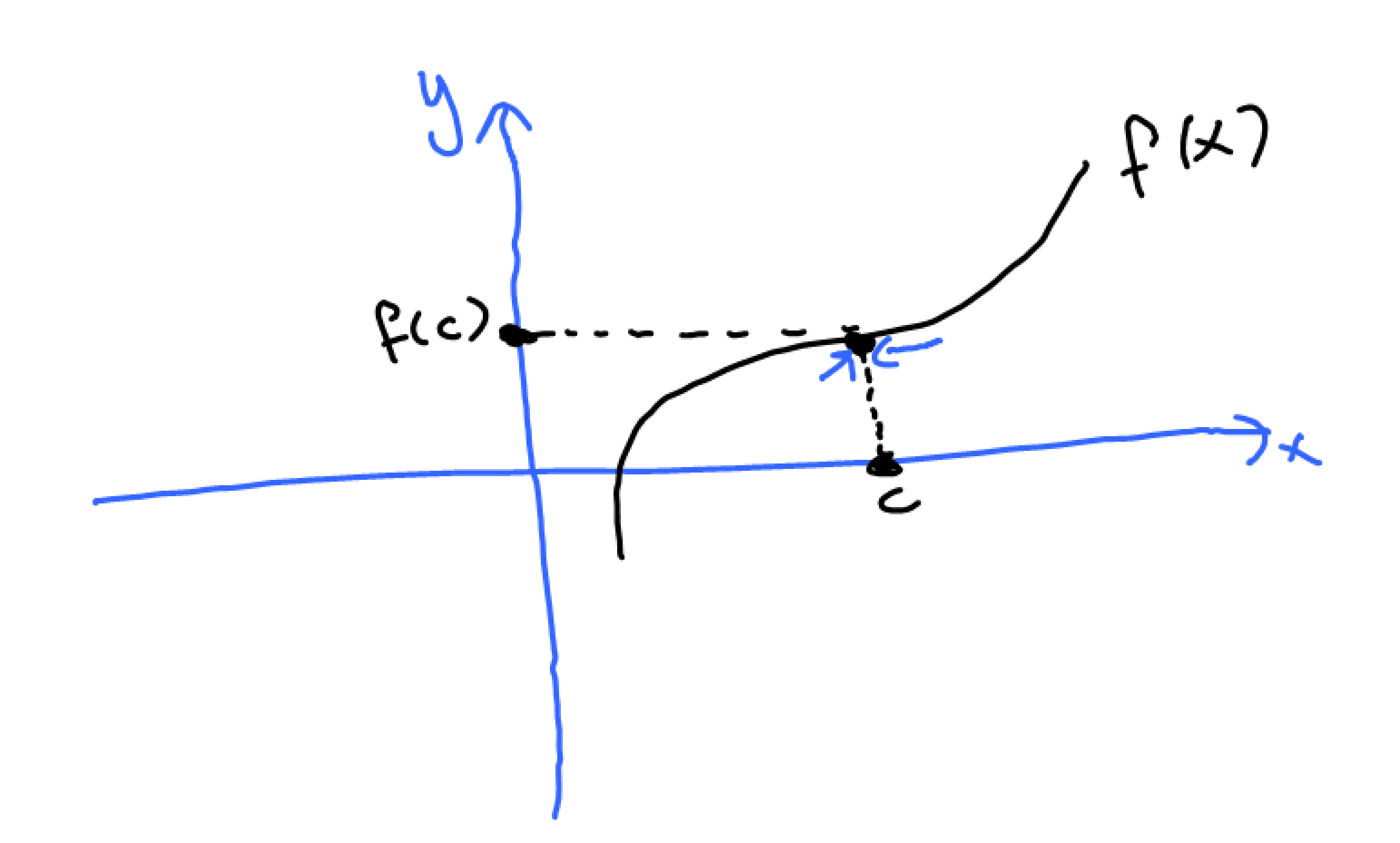
$$|x-2| < \frac{\varepsilon}{3}$$

$$|x-2| < \frac{\varepsilon}{3} = \delta$$

elde edildiğinden limit değeri doğrudur.

Tanım2:

Eğer, $\lim_{x \to c} f(x) = f(c)$ ise, f fonksiyonunu c noktasında süreklidir denir.



Tanım 3:

f fonksiyonunun c noktasındaki türevi (eğer mevcut ise) aşağıdaki gibi tanımlanır:

$$f'(c) = \lim_{x \to c} \frac{f(x) - f(c)}{x - c}$$

Bu limit herhangi bir fonksiyon ve herhangi bir c değeri için var olmak zorunda olmadığından, böyle bir fonksiyon için türev var olmayabilir. Eğer f, $f'(c)$ var oalacak şekilde bir fonksiyon ise, bu durumda f ye c de **türevlenebilirdir** denir. Eğer f fonksiyonu c de türevlenebilir ise, bu durumda c de sürekli olmak zorundadır.

Örnek: $f(x) = 3x^2 + 1$ ise $f'(2) = ?$ (Tanımdan gösteriniz.)

$f(2)=13$

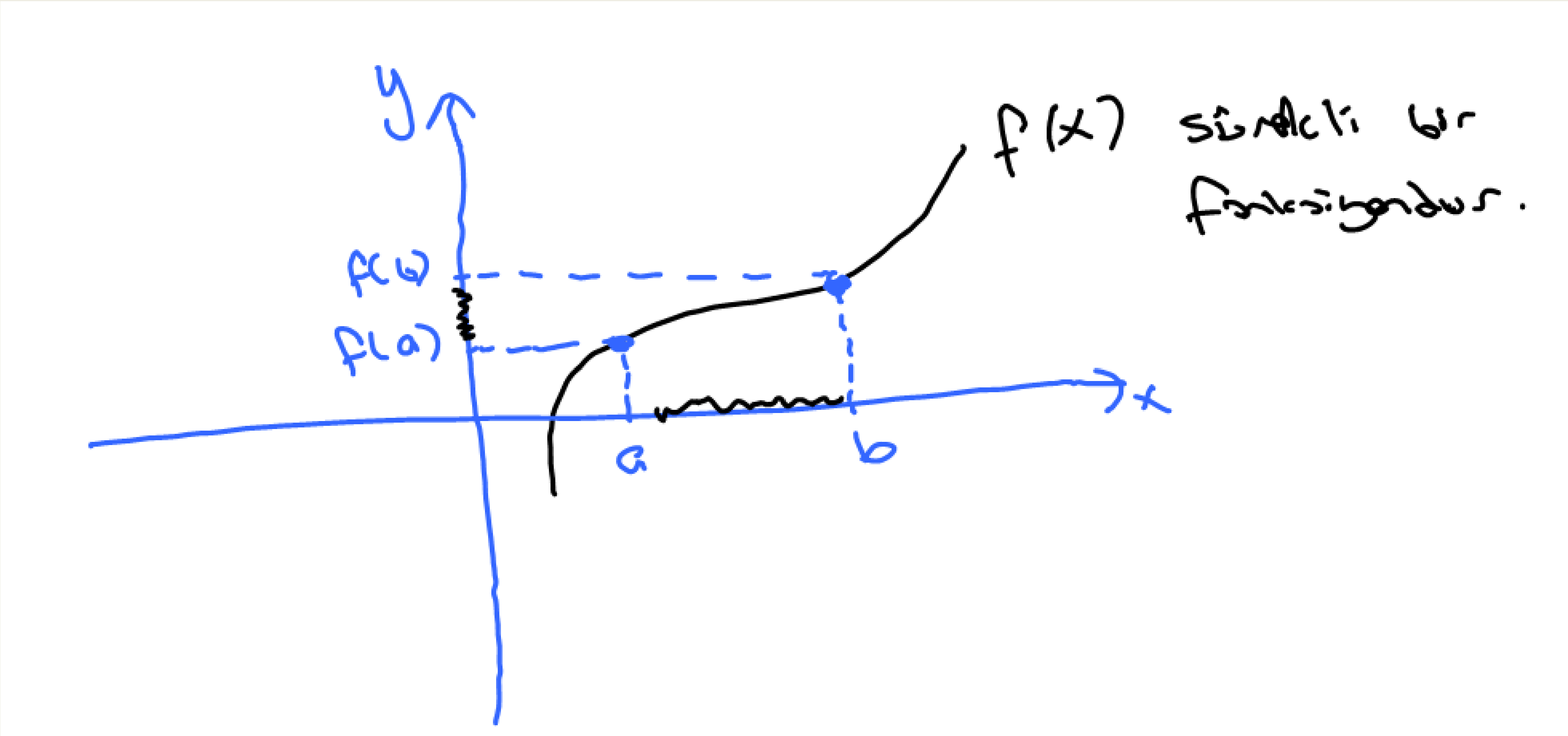
$$\lim_{x \to 2} \frac{3x^2+1-13}{x-2} = \lim_{x \to 2} \frac{3(x^2-4)}{x-2}$$

$$= \lim_{x \to 2} \frac{3.\cancel{(x-2)}.(x+2)}{\cancel{(x-2)}}$$

$$= \lim_{x \to 2} 3.(x+2) = 12$$

Teorem 1:

Bir $[a,b]$ aralığında sürekli olan bir fonksiyon $f(a)$ ile $f(b)$ arasındaki bütün değerleri alır.



Teorem 2: Lagrange Kalanlı Taylor Teoremi:

Eğer $f \in C^n[a,b]$ ve (a,b) açık aralığında f^{n+1} türevi mevcut ise, bu durumda $[a,b]$ kapalı aralığındaki herhangi x_0 ile x noktaları için

$$f(x) = \sum_{k=0}^{n} \frac{1}{k!} f^{(k)}(x_0)(x - x_0)^k + E_n(x)$$

dir. Burada hata terimi, x_0 ile x arasındaki bazı ξ için,

$$E_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x - x_0)^{n+1}$$ şeklindedir.

$x_0 = 0$ durumu Maclaurin serisidir;

$$f(x) = \sum_{k=0}^{n} \frac{1}{k!} f^{(k)}(0) x^k + E_n(x)$$ şeklindedir.

Örnek :

(a) $f(x) = e^x$ fonksiyonunu $x_0 = 1$ noktası civarında Taylor serisine açınız. (İlk 4 terimini alınız.

(b) $e^{1.01}$ değerini, (a) şıkkındaki Taylor seri açılımını kullanarak bulunuz. (e=2.7183 alıp virgülden sonra 4 ondalık duyarlılık ile işlem yapınız.)

Çözüm:

a) $f^{(0)}(x) = f(x) = e^x$ ve $f^{(1)}(x) = f^{(2)}(x) = f^{(3)}(x) = e^x$ elde edilir.

Böylece

$$f(x) \approx \sum_{k=0}^{3} \frac{1}{k!} f^{(k)}(x_0)(x-x_0)^k \quad \text{olur.}$$

$$e^x \approx \frac{1}{0!} f^{(0)}(1)(x-1)^0 + \frac{1}{1!} f^{(1)}(1)(x-1)^1 + \frac{1}{2!} f^{(2)}(1)(x-1)^2 + \frac{1}{3!} f^{(3)}(1)(x-1)^3$$

$$e^x \approx e + e.(x-1) + \frac{1}{2} e(x-1)^2 + \frac{1}{6} e(x-1)^3 \quad \text{elde edilir.}$$

b) (a)'da elde edilen eşitliği kullanırsak:

$$e^{1.01} \approx e + e.(1.01-1) + \frac{e}{2}(1.01-1)^2 + \frac{e}{6}(1.01-1)^3$$

≈ 2.7456 elde edilir.

Örnek :

(a) $f(x) = e^x$ fonksiyonunu $x_0 = 0$ noktası civarında Taylor serisine(Maclaurin seri) açınız. (Virgülden sonra 3 ondalık duyarlılık ile yuvarlama yaparak işlem yapınız.)

(b) $e^{1.01}$ değerini, (a) şıkkındaki Taylor seri açılımını kullanarak bulunuz. (e=2.7183 alıp virgülden sonra 4 ondalık duyarlılık ile işlem yapınız.)

Çözüm:

a) $f^{(0)}(x) = f(x) = e^x$ ve $f^{(1)}(x) = f^{(2)}(x) = f^{(3)}(x) = e^x$

$f^{(1)}(0) = f^{(2)}(0) = f^{(3)}(0) = e^0 = 1$

Böylece Maclaurin Seri açılımı:

$$e^x = \frac{1}{0!} f^{(0)}(0)(x-0)^0 + \frac{1}{1!} f^{(1)}(0)(x-0)^1 + \frac{1}{2!} f^{(2)}(0)(x-0)^2 + \frac{1}{3!} f^{(3)}(0)(x-0)^3 + \cdots$$

$$e^x = f(0) + \frac{x}{1!} f^{(1)}(0) + \frac{x^2}{2!} f^{(2)}(0) + \frac{x^3}{3!} f^{(3)}(0) + \cdots$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + E_n(x)$$ elde edilir.

b) $$e^{1.01} \approx 1 + 1.01 + \frac{1}{2}(1.01)^2 + \frac{1}{6}(1.01)^3$$

$\approx 2.6917668 \approx 2.692$ elde edilir.

Örnek :

$f(x) = cos2x$ fonksiyonunu $x_0 = 0$ noktası civarında Taylor seri (Maclaurin seri) açılımının ilk 5 teriminden faydalanarak $f(0.2)$ değerini bulunuz.

Çözüm:

$$f^{(0)}(x) = f(x) = \cos 2x \qquad f^{(0)}(0) = 1$$

$$f^{(1)}(x) = -2\sin 2x \qquad f^{(1)}(0) = 0$$

$$f^{(2)}(x) = -4\cos 2x \qquad f^{(2)}(0) = -4$$

$$f^{(3)}(x) = 8\sin 2x \qquad f^{(3)}(0) = 0$$

$$f^{(4)}(x) = 16\cos 2x \qquad f^{(4)}(0) = 16$$

$$f(x) = \cos 2x \approx f(0) + \frac{x}{1!} f^{(1)}(0) + \frac{x}{2!} f^{(2)}(0)$$

$$+ \frac{x}{3!} \cdot f^{(3)}(0) + \frac{x}{4!} f^{(4)}(0)$$

$$\approx 1 - 2x^2 + \frac{2}{3} x^4 \text{ elde edilir.}$$

$$f(0.2) \approx \cos 2(0.2) = 1 - 2.(0.2)^2 + \frac{2}{3}(0.2)^4$$

$$= 0.92107 \text{ olur.}$$

Bazı Önemli Fonksiyonların Maclaurin Seri Açılımları Formülleri:

a) $$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \qquad (-\infty < x < \infty)$$

b) $$\cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} \qquad (-\infty < x < \infty)$$

c) $$\ln(1+x) = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^k}{k!} \qquad (-\infty < x < \infty)$$

Çalışma Soruları

Soru 1:

$x_0 = 0$ noktası civarı

$f(x) = \sin 3x$ fonksiyonunun Maclaurin seri açılımının ilk 5 teriminden faydalanarak $f(0.2)$ değerini bulunuz.

Yanıt: $f(x) \cong 3x - \frac{9}{2}x^3$

$$f(0.2) = \sin\big(3(0.2)\big) = \sin(0.6) \cong 0.56400$$

Gerçek değer: *sin(0.6)=0.56464247339 (Radyan cinsinden)*

Soru 2:

$f(x)=e^{-2x}$ fonksiyonunun $x_0 = 2$ noktası civarındaki Taylor Seri açılımının ilk 4 teriminden faydalanarak $f(2.01)$ değerini bulunuz. (Virgülden sonra 4 ondalık duyarlılık ile hesaplayınız.)

Yanıt: $$f(x) \cong e^{-4} - 2e^{-4}(x-2) + 2e^{-4}(x-2)^2 - \frac{4}{3}e^{-4}(x-2)^3$$

$$f(2.01) \cong 0.0179$$

NOT 1: Taylor teoreminde, n=1(ilk iki terime açmak(kendisi ve 1. türevi)) durumu matematiksel çalışmalarda sıklıkla kullanılır ve *Ortalama-Değer Teoremi* olarak bilinir. Ortalama-Değer Teoreminin özel bir durumu *Rolle Teoremidir*.

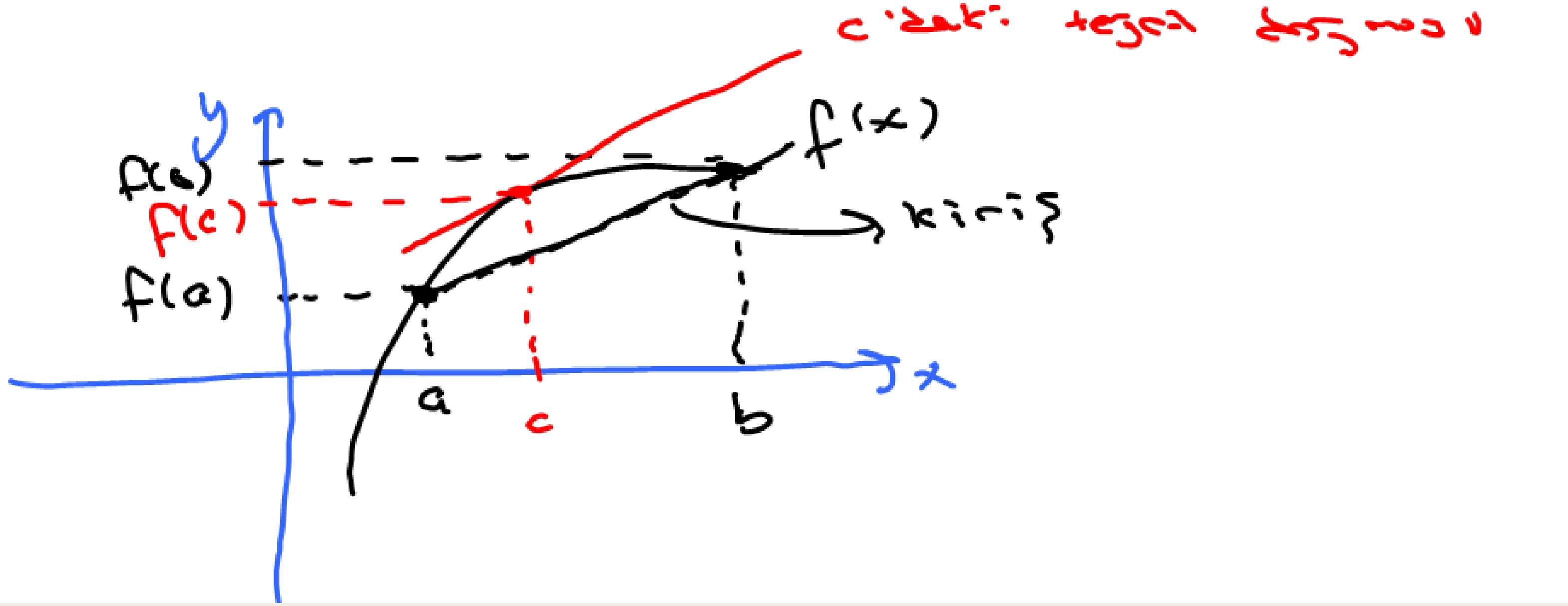
NOT 2: Sayısal Çözümlemede en çok kullanılan teoremlerden bir diğeri ise *Bolzano Teoremidir.*

Ortalama-Değer Teoremi:

Eğer $f(x)$ fonksiyonu (a,b) aralığında $f'(x)$ türevi mevcut ise, $[a,b]$ kapalı aralığında herhangi bir $c \in (a,b)$ için

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

değeri sağlanır.



Örnek :

$f:[1,3] \to R$ ve $f(x)=x^2$ fonk. nun ortalama değeri nedir?

Çözüm:

$f'(x)=2x$

$f'(c)=2c$

Teoremden

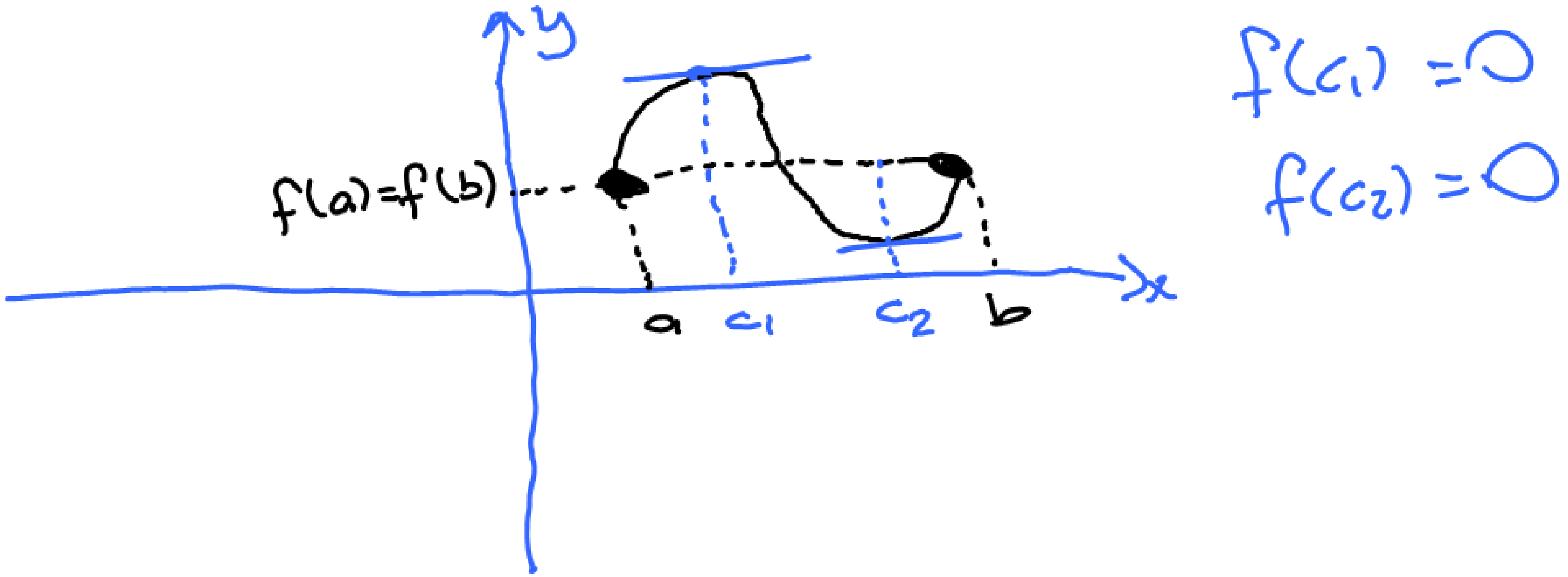
$$2c=\frac{f(3)-f(1)}{3-1}$$

$$2c=\frac{9-1}{2} \Rightarrow 4c=8$$

$$\boxed{c=2}$$

Rolle Teoremi:

Eğer $f(x)$ fonksiyonu (a,b) aralığında $f'(x)$ türevi mevcut ve $f(a)=f(b)$ ise, $[a,b]$ kapalı aralığında bazı c değerleri için $f'(c)=0$ 'dır.



Örnek :

$f:[0,4]\to\mathbb{R}$ ve $f(x)=4x-x^2$ olsun. $[0,4]$ kapalı aralığında Rolle teoremi uygulanabilir mi? Eğer uygulanırsa uygun bir $c\in\mathbb{R}$ bulunuz.

Çözüm:

$f(x)=4x-x^2 \to$ süreklidir. (polinom fonk.)

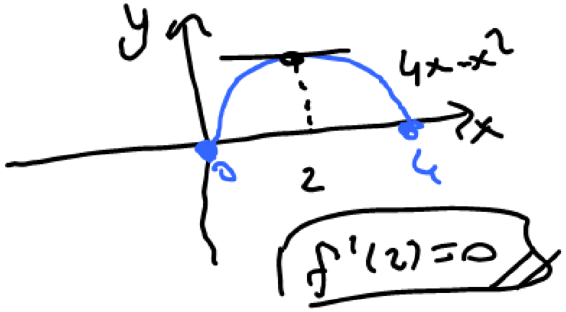
$f(0)=0$, $f(4)=0$ $\Big\rangle$ $f(0)=f(4)$ $\Big\rangle$ Teorem uygulanabilir.

$f'(x)=4-2x$

$f'(c)=4-2c=0$

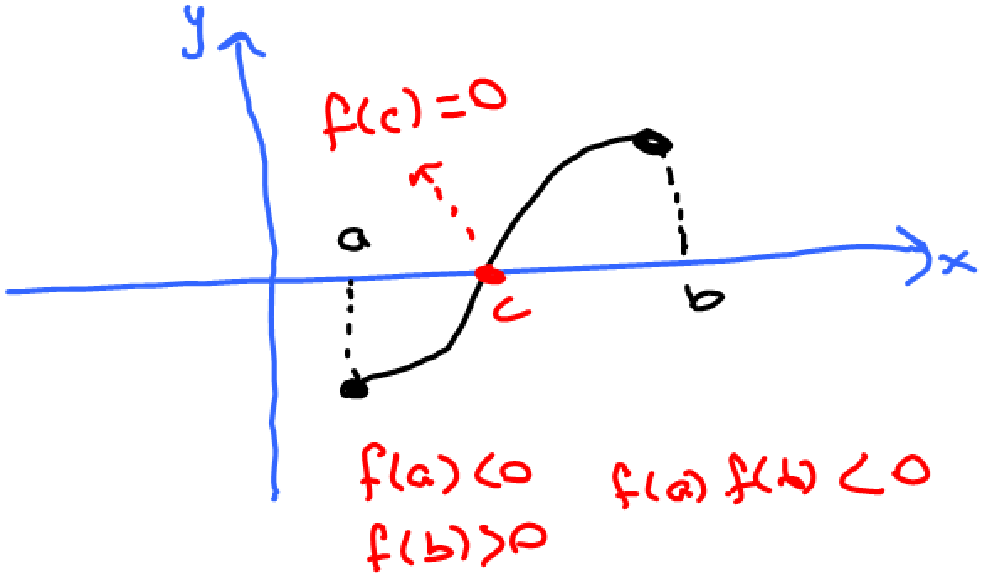
$\boxed{c=2}$

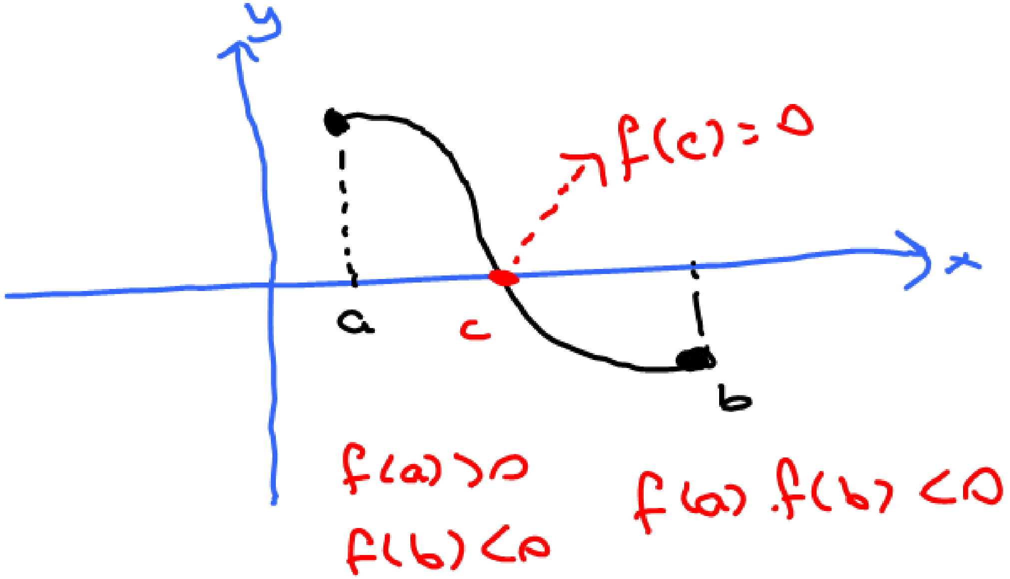
$2\in[0,4]$



Bolzano Teoremi:

$f(x)$ fonksiyonu, $[a,b]$ kapalı aralığında sürekli ve $f(a).f(b)<0$ ise $\exists c \in (a,b)$ değeri için $f(c)=0$ 'dır.





Örnek 1:

$x^3+x^2-1=0$ denkleminin, $[0,1]$ kapalı aralığında bir kökünün olduğunu gösteriniz.

Çözüm:

$f(x)=x^3+x^2-1$ olmak üzere $f(x)$ fonksiyonu $[0,1]$'de süreklidir.

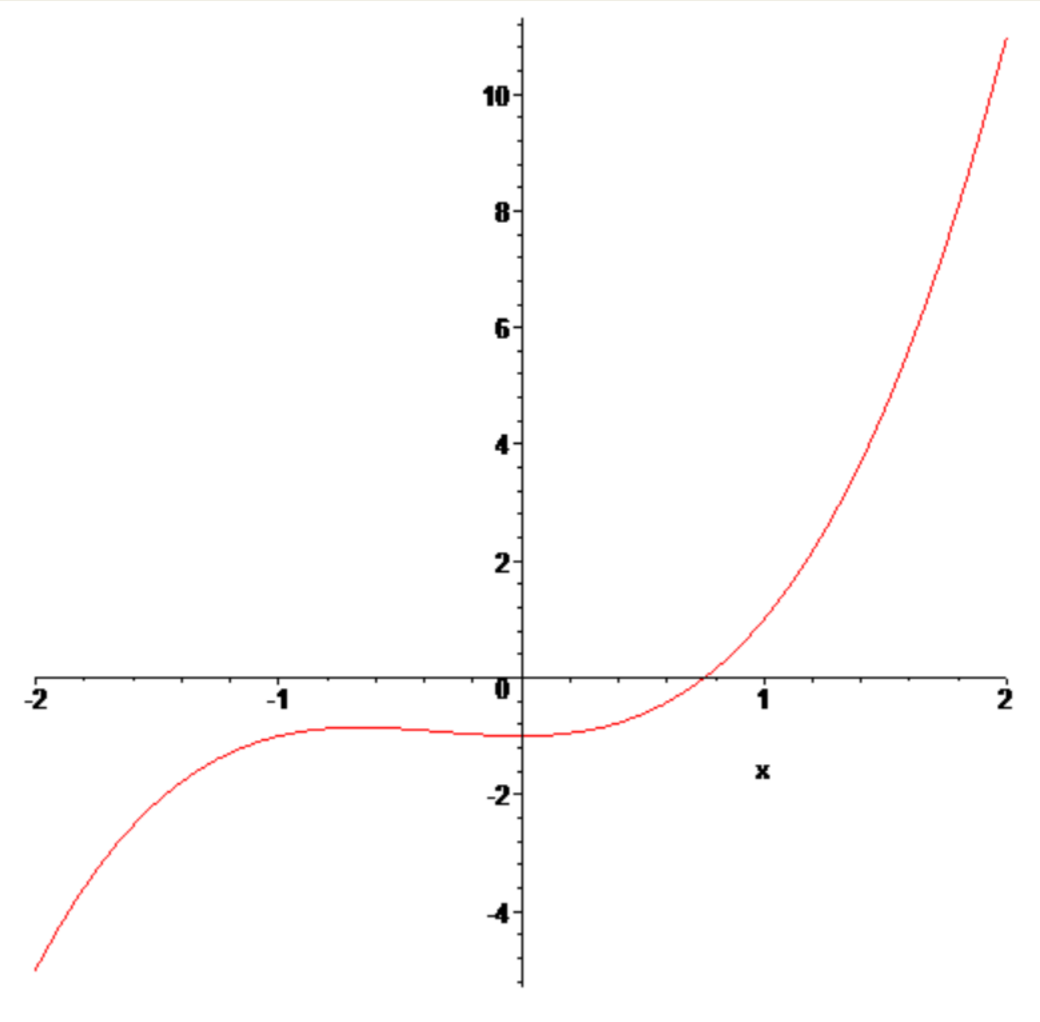
$f(0)=-1$ Böylece; $f(0)\cdot f(1)<0$ old. dan

$f(1)=1$ $\exists c\in[0,1]$ için $f(c)=0$ 'dır.

denklemin kökü

Not: Bolzano teoremi yardımıyla $f(x)=0$ denkleminin, $[a,b]$ kapalı aralığında bir kökünün olup-olmadığı araştırılır.

Örnekte verilen $f(x)=x$^3+x^2-1 fonksiyonun grafiği:

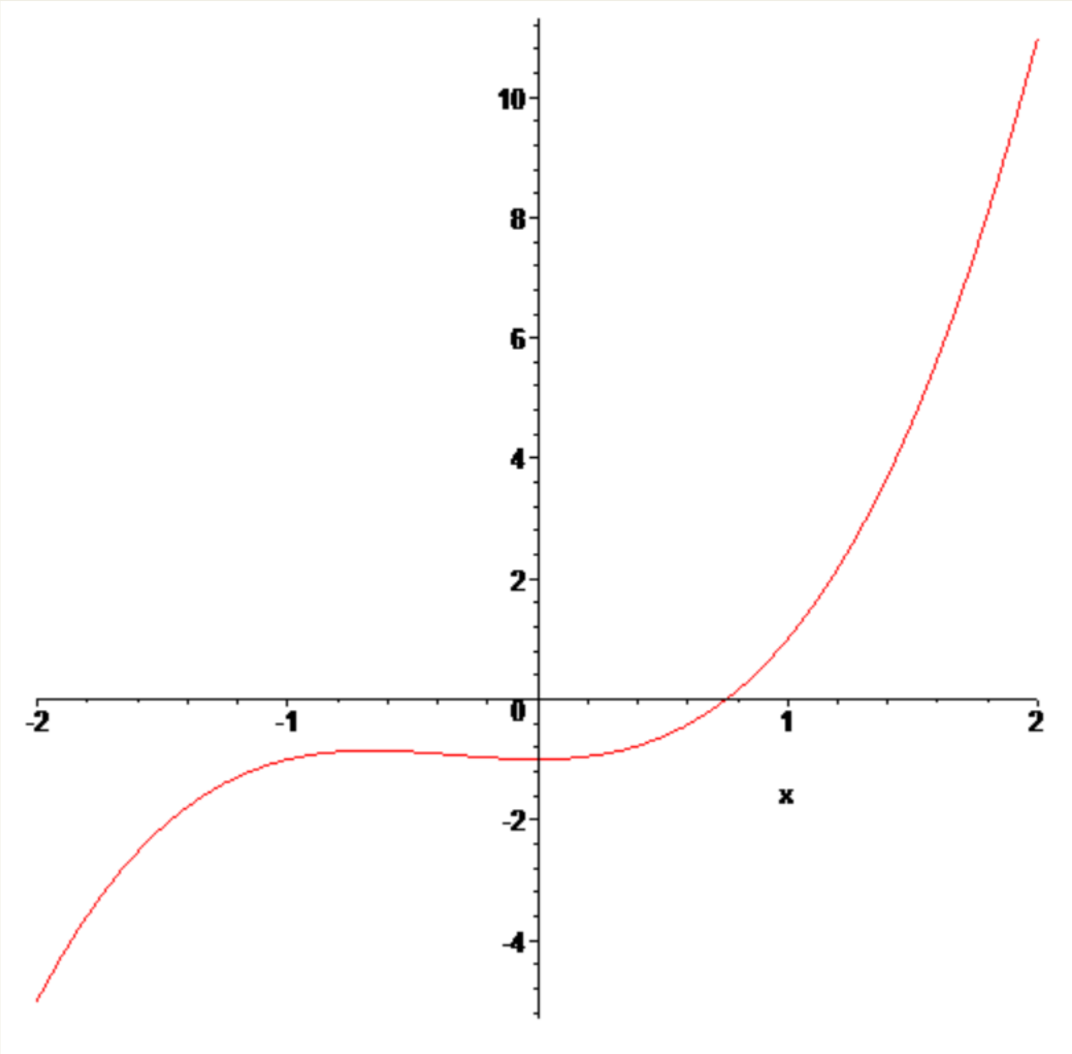


Bolzano teoreminin C kodu verilen örnek için aşağıdaki gibidir:

```
#include<stdio.h>
#include<locale.h>
#include<math.h>
#include<conio.h>
float F(float x)
{return pow(x,3)+pow(x,2)-1;}

int main()
{setlocale(LC_ALL, "Turkish");
float a,b;
printf("a değerini giriniz: ");
scanf("%f",&a);
printf("b değerini giriniz: ");
scanf("%f",&b);
if ((F(a)*F(b))<0) printf("Bolzano teoreminden %.2f ile %.2f arasında en az bir bir kök vardır...",a,b);
                   else
                   printf("Bolzano teoreminin şartları sağlanmamıştır, kökün varlığı için kesin bir şey söylenemez...");
getch();
return 0;
}
```

Fonksiyonun Grafiği:



Programın Çıktıları:

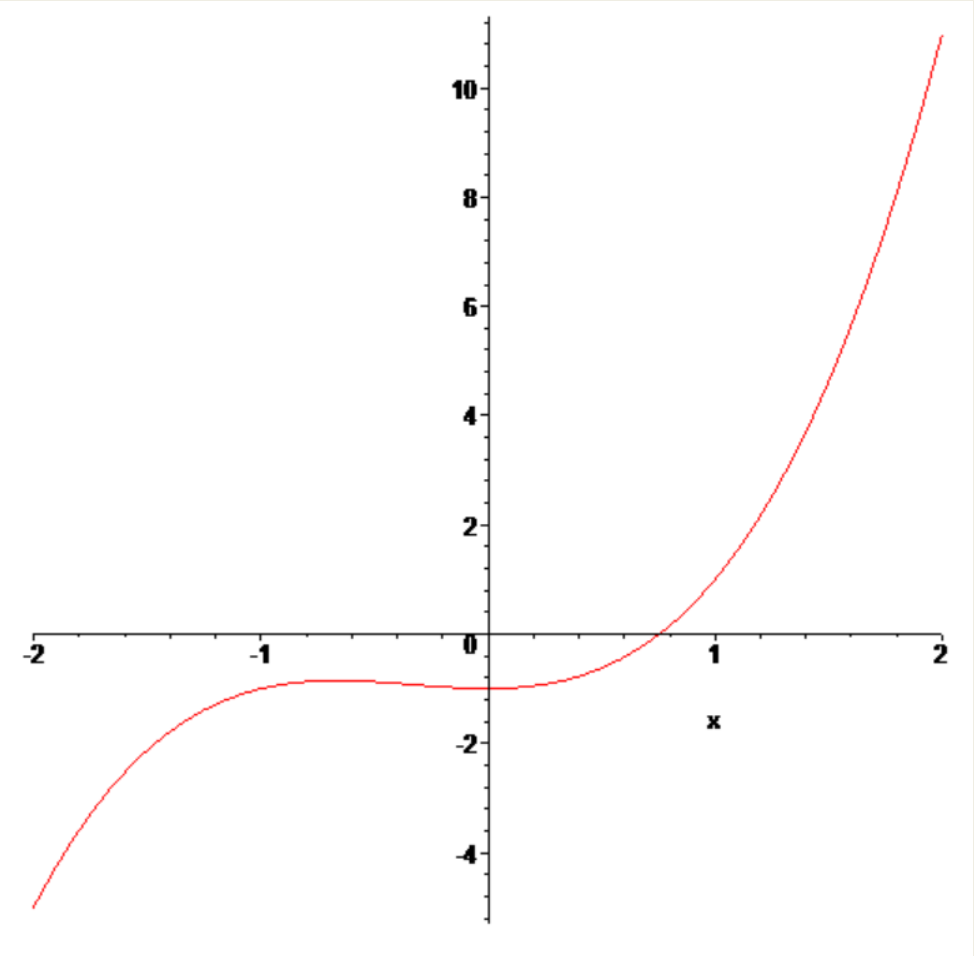
```
a değerini giriniz: -2
b değerini giriniz: 2
Bolzano teoreminden -2,00 ile 2,00 arasında en az bir bir kök vardır...
--------------------------------
Process exited after 33.73 seconds with return value 0
Press any key to continue . . .
```

```
a değerini giriniz: 0
b değerini giriniz: 2
Bolzano teoreminden 0,00 ile 2,00 arasında en az bir bir kök vardır...
--------------------------------
Process exited after 4.599 seconds with return value 0
Press any key to continue . . .
```

```
a değerini giriniz: 0
b değerini giriniz: 1
Bolzano teoreminden 0,00 ile 1,00 arasında en az bir bir kök vardır...
--------------------------------
Process exited after 3.414 seconds with return value 0
Press any key to continue . . .
```

```
a değerini giriniz: 0,5
b değerini giriniz: 1
Bolzano teoreminden 0,50 ile 1,00 arasında en az bir bir kök vardır...
--------------------------------
Process exited after 7.334 seconds with return value 0
Press any key to continue . . .
```

Fonksiyonun Grafiği:



Programın Çıktıları:

```
a değerini giriniz: -2
b değerini giriniz: 0,5
Bolzano teoreminin şartları sağlanmamıştır, kökün varlığı için kesin bir şey söylenemez...
--------------------------------
Process exited after 7.497 seconds with return value 0
```

```
a değerini giriniz: 0
b değerini giriniz: 0,5
Bolzano teoreminin şartları sağlanmamıştır, kökün varlığı için kesin bir şey söylenemez...
--------------------------------
Process exited after 7.703 seconds with return value 0
Press any key to continue . . .
```

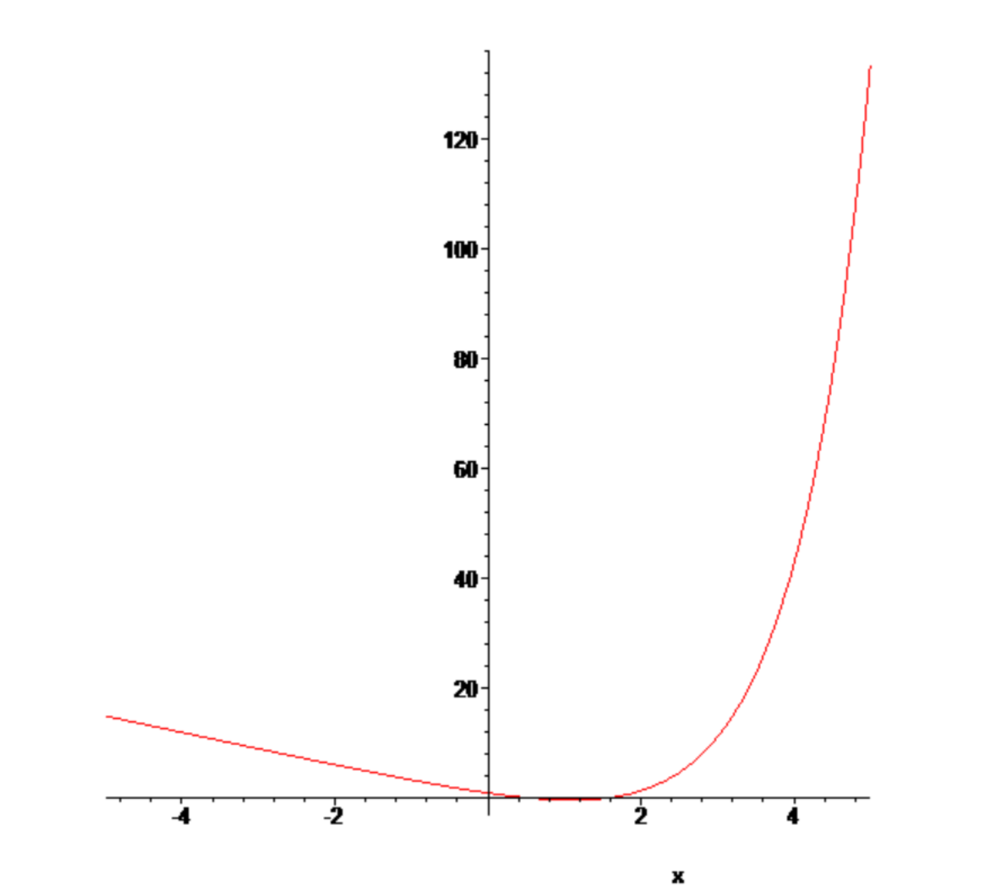
```
a değerini giriniz: 1
b değerini giriniz: 2
Bolzano teoreminin şartları sağlanmamıştır, kökün varlığı için kesin bir şey söylenemez...
--------------------------------
Process exited after 3.377 seconds with return value 0
Press any key to continue . . .
```

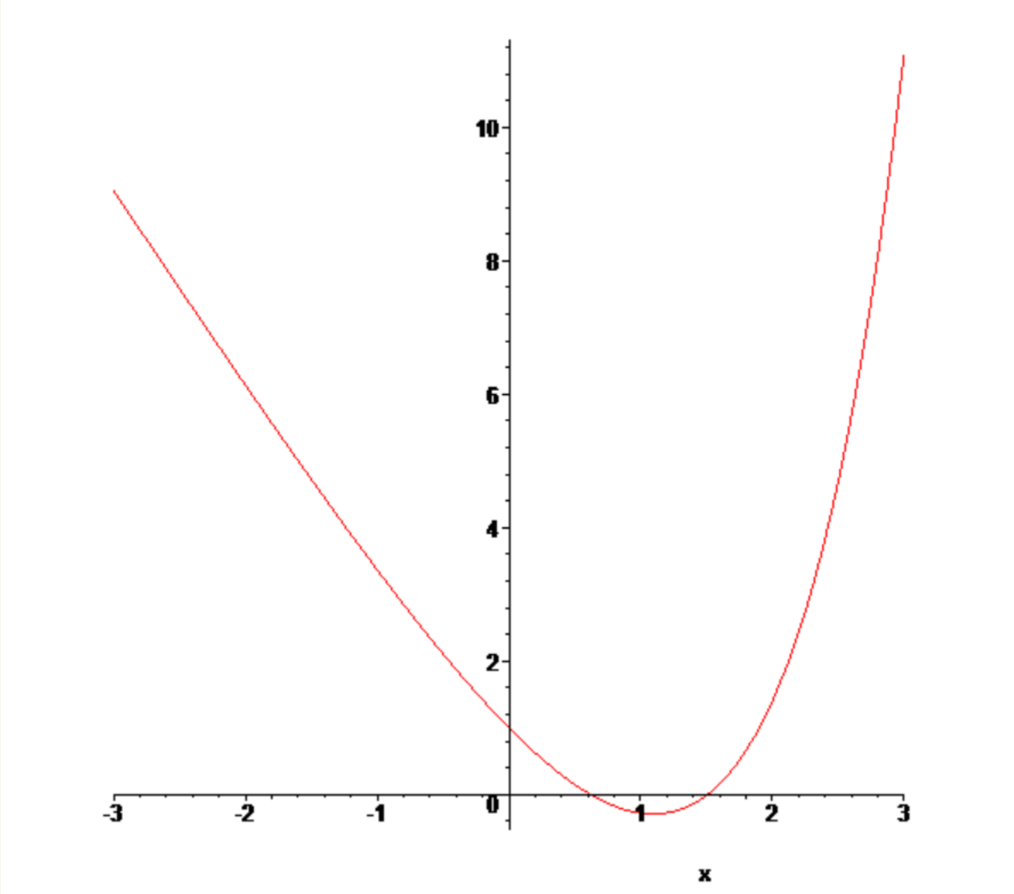
Örnek 2: $f(x) = e\^x - 3x$ fonksiyonunun, girilen aralıkta kökünün olup olmadığını araştırınız.

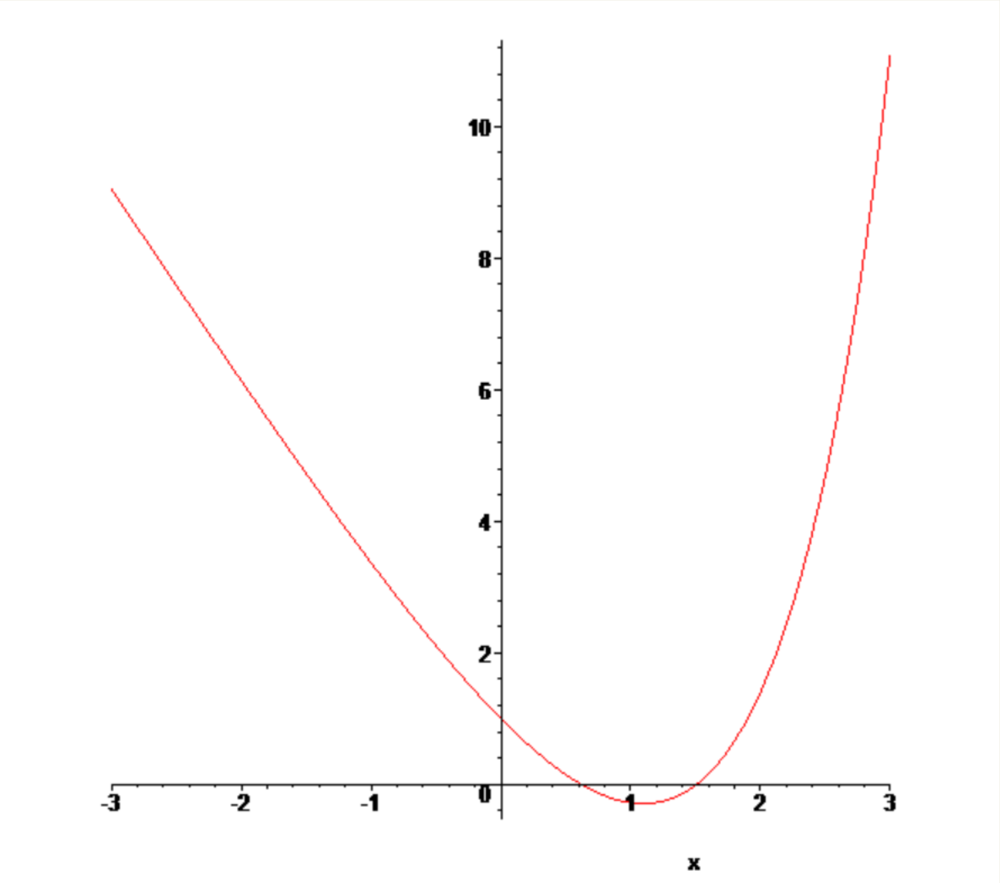
```
#include<stdio.h>
#include<locale.h>
#include<math.h>
#include<conio.h>
float F(float x)
{return exp(x)-3*x;}

int main()
{setlocale(LC_ALL, "Turkish");
float a,b;
printf("a değerini giriniz: ");
scanf("%f",&a);
printf("b değerini giriniz: ");
scanf("%f",&b);
if ((F(a)*F(b))<0) printf("Bolzano teoreminden %.2f ile %.2f arasında en az bir bir kök vardır...",a,b);
                   else
                   printf("Bolzano teoreminin şartları sağlanmamıştır, kökün varlığı için kesin bir şey söylenemez...");
getch();
return 0;
}
```

$f(x)=e^x-3x$ fonksiyonunun grafiği:








```
a değerini giriniz: 0
b değerini giriniz: 1
Bolzano teoreminden 0,00 ile 1,00 arasında en az bir bir kök vardır...
--------------------------------
Process exited after 3.247 seconds with return value 0
Press any key to continue . . .
```

```
a değerini giriniz: 1
b değerini giriniz: 2
Bolzano teoreminden 1,00 ile 2,00 arasında en az bir bir kök vardır...
--------------------------------
Process exited after 2.776 seconds with return value 0
Press any key to continue . . .
```

```
a değerini giriniz: 2
b değerini giriniz: 3
Bolzano teoreminin şartları sağlanmamıştır, kökün varlığı için kesin bir şey söylenemez...
--------------------------------
Process exited after 4.732 seconds with return value 0
Press any key to continue . . .
```

```
a değerini giriniz: 0
b değerini giriniz: 3
Bolzano teoreminin şartları sağlanmamıştır, kökün varlığı için kesin bir şey söylenemez...
--------------------------------
Process exited after 3.393 seconds with return value 0
Press any key to continue . . .
```

Yakınsaklık (Yakınsama):

Bir dizinin bir noktaya yakınsaklığının hızı farklılıklar gösterebilir. Bazı durumlarda çok yavaş olabileceği gibi bazen çok hızlı yakınsayabilir.

Örnek! $x_n = \left(1+\frac{1}{n}\right)^n$. Biliyoruz ki $\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n = e$ 'dir.

Aşağıda verilen C kodunda e sayısına çok yavaş yakınsadığı görülebilir.

$e = 2.718281828459045$ → virgülden sonraki 15 terim!

```c
#include <stdio.h>
#include <math.h>
#include <conio.h>
int main(){
int n;
double x0,x1;
for (n=1;n<=1000000;n++){
x1=(1.0+1.0/double(n));
x0=pow(x1,double(n));
printf( "%d %.10f \n",n ,x0);
}
getch();
return 0;
}
```

```
1 2.0000000000
2 2.2500000000
3 2.3703703704
4 2.4414062500
5 2.4883200000
6 2.5216263717
7 2.5464996970
8 2.5657845140
9 2.5811747917
10 2.5937424601
11 2.6041990119
12 2.6130352902
13 2.6206008879
14 2.6271515563
15 2.6328787177
16 2.6379284974
17 2.6424143752
18 2.6464258211
19 2.6500343266
20 2.6532977051
```

```
9980 2.7181456545
9981 2.7181456681
9982 2.7181456818
9983 2.7181456954
9984 2.7181457091
9985 2.7181457227
9986 2.7181457363
9987 2.7181457499
9988 2.7181457636
9989 2.7181457772
9990 2.7181457908
9991 2.7181458044
9992 2.7181458180
9993 2.7181458316
9994 2.7181458452
9995 2.7181458588
9996 2.7181458725
9997 2.7181458860
9998 2.7181458996
9999 2.7181459132
10000 2.7181459268
```

```
999980 2.7182804693
999981 2.7182804691
999982 2.7182804696
999983 2.7182804694
999984 2.7182804693
999985 2.7182804691
999986 2.7182804696
999987 2.7182804695
999988 2.7182804693
999989 2.7182804697
999990 2.7182804695
999991 2.7182804694
999992 2.7182804692
999993 2.7182804696
999994 2.7182804694
999995 2.7182804692
999996 2.7182804696
999997 2.7182804694
999998 2.7182804696
999999 2.7182804694
1000000 2.7182804692
```

Çok yavaş yakınsıyor…

Örnek:

$x_1 = 2$

$$x_{n+1} = \frac{1}{2}x_n + \frac{1}{x_n}, \quad (n \geq 1)$$

dizisi $\sqrt{2}$'ye yakınsar.

$$\sqrt{2} = 1.414213562373095$$

```c
#include <stdio.h>
#include <math.h>
#include <conio.h>
main(){
int n;
double x0=2.0,x1;
for (n=1;n<=10;n++){
printf( "%d %.15f \n",n ,x0);
x1=x0/2.0+1.0/x0;
x0=x1;
}
getch();
return 0;
}
```

```
1 2.000000000000000
2 1.500000000000000
3 1.416666666666667
4 1.414215686274510
5 1.414213562374690
6 1.414213562373095
7 1.414213562373095
8 1.414213562373095
9 1.414213562373095
10 1.414213562373095

--------------------------------
Process exited after 2.845 seconds with return value 0
Press any key to continue . . .
```

6. Adımda sonuca ulaştı! (Çok hızlı yakınsıyor…)

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.