CENG 235 ALGORİTMALARLA SAYISAL ÇÖZÜMLEME

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

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü

- Hafta 2

2. Hafta Konular

- **Hatalar**
- **Bilgisayar Aritmetiği**
- **Lineer Olmayan Denklemlerin Yaklaşık Çözüm Yöntemleri**

  --- Köke yaklaşma (Bolzano Teoremi ve Grafik ile)

Sayısal Çözümleme:

--- Mühendislik ve uygulamalı fen bilimleri alanlarında karşılaşılan problemlerin çözümleri, her zaman analitik olarak mümkün olmayabilir.

--- Karşılaşılan problemlerin lineer olmaması veya bilgisayar ile çözümü için analitik yöntemlerin uygun olmaması, uygulamacıları sayısal yöntemler kullanmaya yönlendirmektedir.

--- Bununla beraber analitik çözüme uygun problemlerin bile, işlem kolaylığı gibi elementer sebeplerden dolayı sayısal yöntemler ile çözüldüğü görülebilmektedir.

Algoritmalar: Sayısal çözüm yöntemi genellikle önceden saptanmış aritmetik ve mantıksal işlemlerden oluşur. Bu işlemlerin tümüne *çözüm algoritması* denir.

Sayısal çözümleme daha çok algoritmalar aramaya odaklıdır. Bazı problemler için uygun algoritmalar henüz bulunamamıştır, bazıları içinse birden çok algoritma vardır ve bunlar arasından seçim yapmamız gerekebilir. Bir algoritmayı diğerine tercih etmenin bir çok nedeni olabilir, bunlardan en belirgin olanları hız ve doğruluktur. Hızlılık açıkça bir avantajdır, ancak orta zorlukta problemlerde bu avantaj gelişmiş bilgisayarlar ile ortadan kalkmıştır. Daha büyük zor problemler için hız hala önemli bir faktördür ve yavaş bir algoritma pratik olmadığından seçilmeyebilir. Ancak, diğer faktörler aynı iken, daha hızlı algoritmalar seçilir.

Örnek: 2 nin karekökünü dört ondalık basamağa kadar bulunuz.

Çözüm: Sadece dört aritmetik işlemi(toplama, çıkarma, çarpma ve bölme) kullanarak bu hesabı yapan birden fazla algoritma vardır. Bunların en popüler olanı

$$x_1 = 1, x_{n+1} = \frac{1}{2}(x_n + \frac{2}{x_n})$$

şeklindedir.
Buradan birkaç hesaplama ile

$$x_2 = \frac{3}{2}, x_3 = \frac{17}{12}, x_4 = \frac{1}{2}(\frac{17}{12} + \frac{24}{17})$$

değerleri hızlıca bulunur.
Dört ondalık basamağa yuvarlanırsa:

$$x_2 = 1.5000, x_3 = 1.4167, x_4 = 1.4142$$

bulunur, ve x_4 değeri dört basamak için doğrudur.

HATALAR

--- Bir problemin çözümü için kullanılan giriş bilgileri çoğu kez bir deney, bir ölçme sonucu elde edildiklerinden veya bir tablodan alınıp, bir fonksiyondan hesaplandıklarından dolayı az yada çok hataya sebep olabilirler.

--- Ayrıca çözüm için kullanılacak algoritmada, kendi yöntemlerinden dolayı bazı kesme hatalarına sebep olabilir.

--- Bunlarından yanında bilgisayarlarda yapılacak aritmetik işlemlerden doğan yuvarlama hatalarını da ele aldığımızda

Sayısal Çözümlemede:

--- Veri hataları

--- Kesme hataları

--- Yuvarlama hataları

olmak üzere üç ayrı tür hatadan bahsedebiliriz.

1- Veri Hataları: Bu tür hatalar çözümü istenen problemlere ait matematiksel model oluşturulduğunda ortaya çıkar. Matematiksel modeli olduğunca basitleştirebilmek için ideal kabuller yapılır. Bu da veri hatalarına neden olur. Veri ölçümlerindeki hatalar, rakamları hatalı kaydetme veya matematiksel sabitlerin (π, e gibi) tam olarak temsil edilmemesi yüzünden ortaya çıkar.

--- π, e gibi sayılar sonlu ondalıklı olmadıklarından, hesaplamalarda bu tür sayıların kullanımı hatalara sebep olur.

--- Bunun yanında bir de, ondalık kısmı bulunan pek çok sayının, başka bir tabanda sonlu gösterilişi olmadığını da söylemek mümkündür.

Örneğin 0.43 sayısının ikili sayı sisteminde tam karşılığı yoktur ve

$(0.43)_{10}=(0.0110\ldots)_2$ şeklindedir.

2- Kesme Hataları:

Bu tür hatalar yaklaşık değerler veren matematiksel teknikleri kullandığımızda ortaya çıkar. Örneğin,

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + E_n(x)$$

yerine

$$e^x \cong 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

alırsak, $E_n(x)$ bize kesme hatasını verir.

Yani sonsuz terimli bir ifadeyi sayısal olarak yazarsak kesme hatası yapmış oluruz.

Bu hatalar daha çok sayısal çözümleme iterasyon metotlarına dayandığında görülür.

Bu tür hatalar kullanılan hesaplayıcının hafızasının büyüklüğüne bağlıdır.

Hatanın büyüklüğü ile hafızanın büyüklüğü ters orantılıdır.

3- Yuvarlama Hataları: Özellikle elektronik hesaplayıcılarda, bir sayının sonlu sayıdaki hane ile ifade edilmesine bağlı olarak yapılan hatalardır. Bu tür hatalar, genellikle ondalık yazımın son hanesini etkilerler.

Son hane: 0, 1, 2, 3 ve 4 ise aynı kalır.

Son hane: 5, 6, 7, 8 ve 9 ise son hanenin 1 fazlası alınır.

Örnek 1: $0.65899 \cong 0.659$

$\cong 0.6590$

Örnek 2: $0.65822 \cong 0.6582$

$\cong 0.658$

$\cong 0.66$

Örnek 3: 1.492 ve *1.066* sayıları çarpılırsa 1.590472 elde edilir.

Bu işlemi 4 basamaklı bir makinada yaparsak, 1.590472 yerine 1.590 alınır.

Burada oluşan yuvarlama hatasıdır.

(Yuvarlanacak basamaktan; <5 ise değişmez, ≥ 5 ise 1 artar.)

Buradaki yuvarlama hatası;

$$|1.590472 - 1.590| = |0.000472| = 0.472 \times 10^{-3} < \frac{1}{2} \times 10^{-3}.$$

Not: b sayı tabanı ve n virgülden sonraki basamak olmak üzere, yuvarlama hatasının üst sınırı; $\frac{1}{2} \times b^{-n}$ dir.

İterasyon:

Sayısal yöntemlerde bir önceki değeri kullanarak bir sonraki değeri elde etme işlemidir.

Örnek: $x^{(k+1)} = 2x^{(k)} + \ln(x^{(k)})$

$x^{(0)} = 1$ olsun. (yerine yaz)

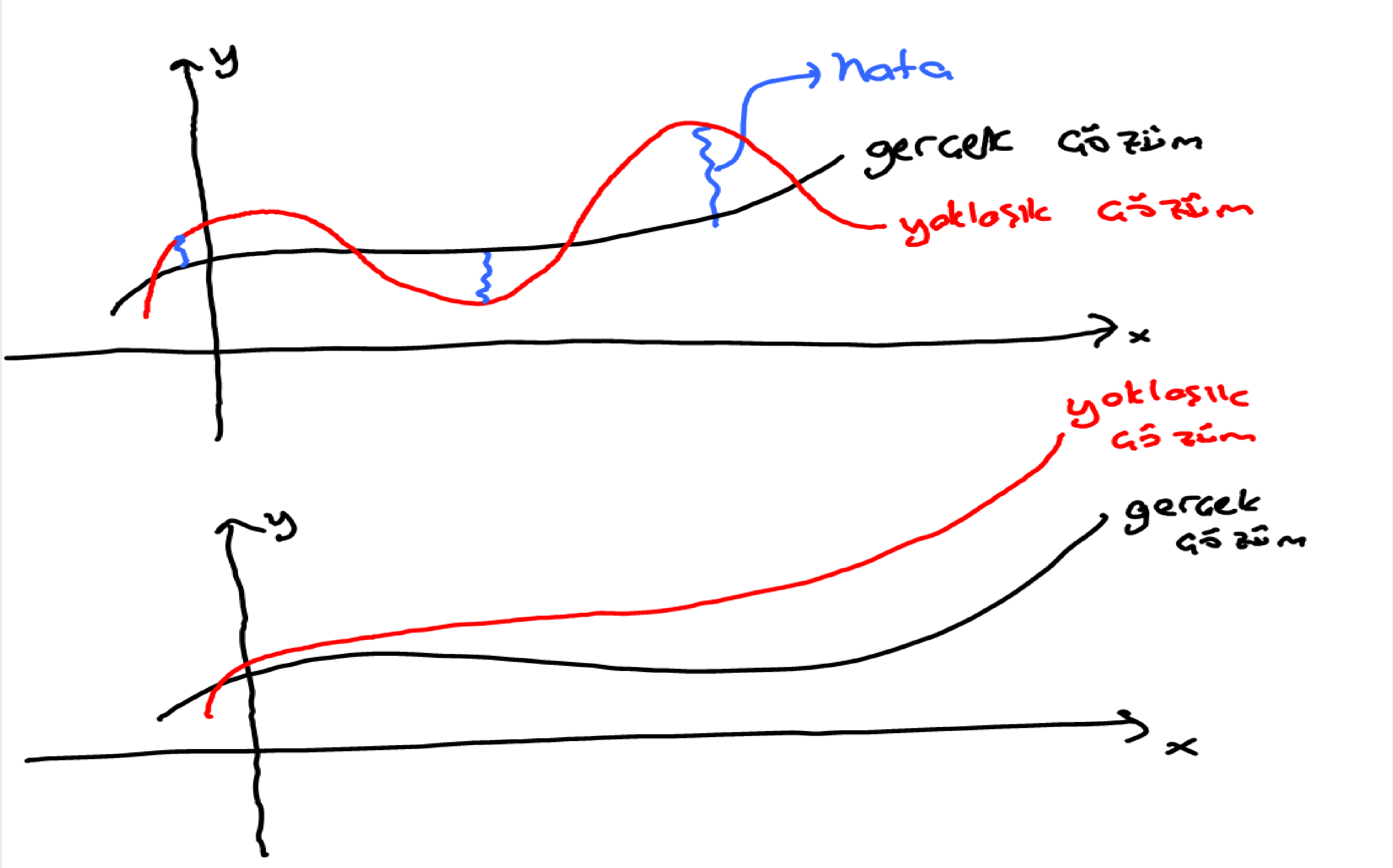
$k=0$ $x^{(1)} = \ldots\ldots$ " "

$k=1$ $x^{(2)} = \ldots\ldots$ " "

$k=2$ $x^{(3)} = \ldots\ldots$ " "

$\vdots$ $\vdots$

Kararlılık:


y
hata
gerçek çözüm
yaklaşık çözüm
x
y
yaklaşık çözüm
gerçek çözüm
x

--- Her işlemler dizisi belirli sayısal hatalar doğurduğuna göre birbiri üstüne eklenen bu hatalar son bulunan değeri bir birikmiş hata olarak etkilerler. Birikmiş hataların önemi birikim hızlarına bağlıdır.

--- Birikim hızı toplam hatanın sınırlı kalmasını sağlayacak şekilde azalıyorsa bu durumda işlemler dizisinin *kararlı* olduğunu söyleriz.

--- Hız sürekli olarak artar, böylece birikmiş hatalar sonucunun anlamsız olmasına neden olursa işlemler dizisine *kararsız* denir.

Mutlak Hata:

Bir büyüklüğün gerçek değeri ile yaklaşık değeri arasındaki farkın mutlak değeri mutlak hata olarak ifade edilir.

Yani; $\bar{x}$ gerçek değeri, x yaklaşık değeri olmak üzere mutlak hata

$$\Delta(x) = |\bar{x} - x|$$

olarak tanımlanır.

Gercek değeri $\bar{x}$ olan bir büyüklüğün

yaklaşık değeri x ise, $\bar{x} - x$ fark, x

yaklaşık sayısının hatasını gösterir.

$\Delta(x) = |\bar{x} - x|$ mutlak hata olarak ifade edilir.

x sayısının, $\bar{x}$ sayısını $\Delta(x)$ kadar bir hata ile temsil edildiğini göstermek

$\bar{x} \approx x[\Delta(x)]$ olarak ifade edilir.

Örnek:

$$x_1 \approx 269835.72 \; [0.01]$$

$$x_2 \approx 3.2 \; [0.01]$$

Her iki sayıda da mutlak hata aynı. Fakat x_1 daha büyük bir sayı old. dan, x_1'in hatası x_2'ye göre daha önemsizdir.

Buradan anlaşılıyor ki:

Duyarlılık sadece mutlak hataya değil, hesaplanan büyüklüğe de bağlıdır.

Bu yüzden Bağıl hata tanımını vereceğiz.

Bağıl Hata:

Mutlak hatanın gerçek değere bölünmesi ile elde edilen değerdir. Ancak her problem için gerçek değeri elde etme olanağı yoktur. Bu nedenle bağıl hata genel olarak mutlak hatanın yaklaşık değere bölünmesiyle elde edilir ve $\delta(x)$ ile gösterilir ve

$\delta(x) = \frac{\Delta(x)}{\bar{x}}$ → gerçek değer biliniyorsa

$\delta(x) = \frac{\Delta(x)}{x}$ → gerçek değer bilinmiyorsa

şeklinde hesaplanır.

Örnek:

$\bar{x} = \frac{1}{3}$, $x = 0.333$ ise mutlak ve bağıl hatayı bulunuz. (İşlemi sekiz ondalıklı yapınız.)

Çözüm:

$\bar{x} = 0.33333333, x = 0.33300000$ biçiminde sekiz ondalıklı yazalım.

O zaman

$$\Delta(x) = |\bar{x} - x| = 0.00033333$$

$$\delta(x) = \frac{\Delta(x)}{\bar{x}} = \frac{0.00033333}{0.33333333} = 0.00099999 \cong 0.001$$

olarak bulunur.

Örnek:

$\bar{x} = 23.496$ ve $x = 23.494$ ise mutlak ve bağıl hatayı bulunuz.

Çözüm:

Mutlak ve bağıl hata

$$\Delta(x) = |23.496 - 23.494| = 0.002$$

$$\delta(x) = \frac{\Delta(x)}{\bar{x}} = \frac{0.002}{23.496} = 0.00008512$$

olarak bulunur.

Örnek:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots$$

serisinde ilk beş terimi alarak sin(0.5) değerine yaklaşınız. Yaklaşımdaki mutlak ve bağıl hatayı bulunuz. **(**Virgülden sonra altı ondalıklı yapınız.)

$$\bar{x} = \sin(0.5) = 0.479426$$

$$\mathrm{x} = \sin(0.5) = 0.5 - 0.020833 + 0.000260 - 0.000002 + 0.000000 = 0.479425$$

$$\Delta(x) = |\bar{x} - x| = 0.000001$$ *Mutlak hata*

$$\delta(x) = \frac{\Delta(x)}{\bar{x}} = \frac{0.000001}{0.479426} = 0.000002$$ *Bağıl hata*

Örnek:

$A \approx 298\,[0.5]$ ise A ve B'nin bağıl

$B \approx 12\,[0.5]$ hatalarını ve hata yüzdelerini bulunuz.

Çözüm:

$$\delta(A) = \frac{0.5}{298} = 0.00167$$

$$\delta(B) = \frac{0.5}{12} = 0.0416$$

Her iki sayının hata yüzdesi bağıl hataların 100 ile çarpılması ile ifade edilir.

A sayısının yüzde hatası = 0.167

B sayısının yüzde hatası = 4.16 olur.

Toplama İşleminde Hatalar

$$A_1 \approx a_1 \, [\Delta(a_1)]$$

$$A_2 \approx a_2 \, [\Delta(a_2)]$$

$$\vdots$$

$$A_n \approx a_n \, [\Delta(a_n)]$$

ise:

$$\Delta(a_1 + a_2 + \dots + a_n) = \Delta(a_1) + \Delta(a_2) + \dots + \Delta(a_n)$$

$$\delta(a_1 + a_2 + \dots + a_n) = \frac{\Delta(a_1) + \Delta(a_2) + \dots + \Delta(a_n)}{a_1 + a_2 + \dots + a_n}$$

şeklindedir.

Özel olarak:

$$\Delta(a_1 + a_2) = \Delta(a_1) + \Delta(a_2)$$

$$\delta(a_1 + a_2) = \frac{\Delta(a_1) + \Delta(a_2)}{a_1 + a_2}$$

olur.

Çıkarma İşleminde Hatalar

$$A_1 \approx a_1 \, [\Delta(a_1)]$$
$$A_2 \approx a_2 \, [\Delta(a_2)]$$ olmak üzere

$$\Delta(a_1 - a_2) = \Delta(a_1) + \Delta(a_2)$$ (Mutlak hata)

$$\delta(a_1 - a_2) = \frac{\Delta(a_1) + \Delta(a_2)}{|a_1 - a_2|}$$ (Bağıl hata)

Çarpma İşleminde Hatalar

$$A_1 \approx a_1 [\Delta(a_1)]$$
$$A_2 \approx a_2 [\Delta(a_2)]$$ olmak üzere

$$\Delta(a_1 \cdot a_2) = a_2 \cdot \Delta(a_1) + a_1 \cdot \Delta(a_2)$$ (Mutlak hata)

$$\delta(a_1 \cdot a_2) = \delta(a_1) + \delta(a_2)$$ (Bağıl Hata)

Örnek:

$$x_1 \approx 26.83\,[0.1]$$
$$x_2 \approx 18.75\,[0.3]$$
$$x_3 \approx 12.61\,[0.2] \quad \text{ise}$$

$x_1 + x_2 - x_3$ değerinin mutlak ve bağıl hatalarını bulunuz.

$$\delta(x_1) = \frac{0.1}{26.83} = 0.00372$$

$$\delta(x_2) = \frac{0.3}{18.75} = 0.016$$

$$\delta(x_3) = \frac{0.2}{12.61} = 0.0158$$

$$\Delta(x_1 + x_2 - x_3) = \Delta(x_1) + \Delta(x_2) + \Delta(x_3)$$

$$= 0.1 + 0.3 + 0.2$$

$$= 0.6$$ olur. (mutlak hata)

$$x_1 + x_2 - x_3 = 26.83 + 18.75 - 12.61$$

$$= 32.97$$

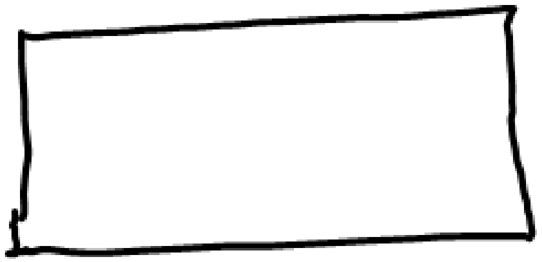
$$\delta(x_1 + x_2 - x_3) = \frac{\Delta(x_1 + x_2 - x_3)}{x_1 + x_2 - x_3}$$

$$= \frac{0.6}{32.97} = 0.01819$$

elde edilir.

(Bağıl hata)

Örnek: Bir dikdörtgenin kenarları $\Delta(x) = 0.01$ ve $\Delta(y) = 0.01$ mutlak hata ile $x = 4.02$ ve $y = 4.96$ olarak ölçülmüş ise bu dikdörtgenin alanında yapılan mutlak ve bağıl hatayı bulunuz.



$\bar{Y} \approx y[\Delta(y)]$

$\text{Alan} = \bar{X} \cdot \bar{Y}$

$\bar{X} \approx x[\Delta(x)]$

$$\bar{X} \approx 4.02\ [0.01]$$

$$\bar{Y} \approx 4.96\ [0.01]$$

$$\bar{X} . \bar{Y} \approx x . y\ [\Delta(x.y)]$$

$$\Delta(x.y) = |y| . \Delta(x) + |x| . \Delta(y)$$

$$= 4.96 . (0.01) + 4.02 . (0.01)$$

$$= 0.0898$$ (mutlak hata)

$$\delta(x.y) = \frac{\Delta(x.y)}{|x.y|} = \frac{0.0898}{19.9392}$$

$$= 0.0045 \text{ (Bağıl hata)}$$

veya

$$\delta(x) = \frac{0.01}{4.02} = 0.0025$$

$$\delta(y) = \frac{0.01}{4.96} = 0.002$$

$$\delta(x.y) = \delta(x) + \delta(y) = 0.0045 \text{ (Bağıl hata)}$$

BİLGİSAYAR ARİTMETİĞİ

IEEE 754-2008 standardına göre 64 bit bir bilgisayar sisteminde reel sayıların ikili sistemde gösterimi şu şekildedir.

s (İşaret biti):1 bit	c (Üstel kısım) : 11 bit	f (Kesir kısmı): 52 bit

Bu sayının genel ifadesi $(-1)^s \times \ 2^{c-1023} \times (1+f)$ şeklindedir. Üstel kısım c'nin alabileceği en büyük değer 2047'dir. Sıfıra yakın sayılarda adil gösterim için üstel kısım c-1023 olarak alınır. 52 bit kesir (fraction) kısm en az 16-17 ondalık basamağa denk gelmektedir.

Örnek: Aşağıdaki makine sayısını ondalık sayıya çeviriniz.

0 10000000011 1011100100010000000000000000000000000000000000000000

Çözüm:

$$\text{Üstel Kısım (c)} = 1 \times 2^{10} + 1 \times 2^{1} + 1 \times 2^{0} = 1027$$

$$\text{Kesir Kısım (f)} = 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + 1 \times 2^{-8} + 1 \times 2^{-12}$$

$$= \frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{4096} = 0{,}722900390625$$

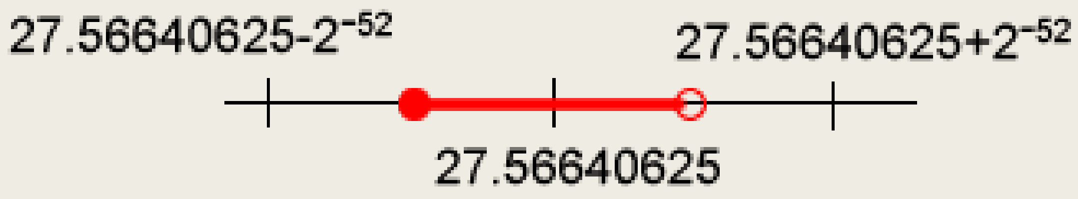
$$\text{Ondalık Sayı} = (-1)^{0} \times 2^{1027-1023} \times (1 + 0{,}722900390625) = 27{,}56640625$$

Bu makine sayısının bir fazlası ve bir eksiği aşağıdaki şekildedir:

0 10000000011 1011100100010000000000.........001 (Bir fazlası)

0 10000000011 1011100100001111111111.........111 (Bir eksiği)

Bu sayının bir fazlasının ve bir eksiğinin olduğu aralık aşağıdaki şekildedir:



Bu aralıkta sonsuz sayıda reel sayı vardır.

Dolayısıyla 64 bitlik bir sistemde yukarıdaki bu üç makine sayısının temsil edemediği reel sayılar mevcuttur.

0 10000000011 1011100100010000………000

şeklinde verilen makine sayısı kırmızı aralıktaki bütün sayıların temsilen kullanılır.

[27.5664062499999982236431605997495353221893310546875,

27.5664062500000017763568394002504646778106689453125**].**

Örnek:

(a) 64 bitlik bir sistemde en küçük pozitif sayı:

$$0\ 00000000001\ \ 0000000000000000\ldots\ldots.\ 000$$

dır. Onluk sistemdeki karşılığı:

$$(-1)^0 \times\ 2^{1-1023} \times (1+0) \approx 0{,}22 \times 10^{-307}$$

dir.

(b) 64 bitlik bir sistemde en büyük pozitif sayı:

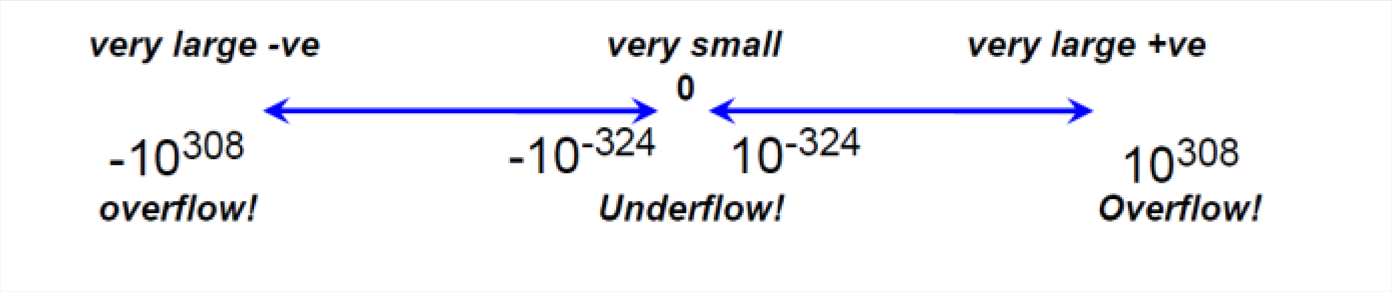
$$0\ 11111111111\ \ 1111111111\ldots\ldots.\ 111$$

dır. Onluk sistemdeki karşılığı:

$$(-1)^0 \times\ 2^{2047-1023} \times (2-2^{-52}) \approx\ 0.17977\ \times\ 10^{309}$$

dir.

Bu sayıdan daha büyük bir sayının oluşması durumunda bilgisayar overflow hatası verir.



Sıfır sayısını temsil için ise s=0 yada s=1, c=0, f=0 değerleri alınır.

Bu sayıdan daha küçük pozitif bir sayının oluşması durumunda bilgisayar overflow hatası verir.

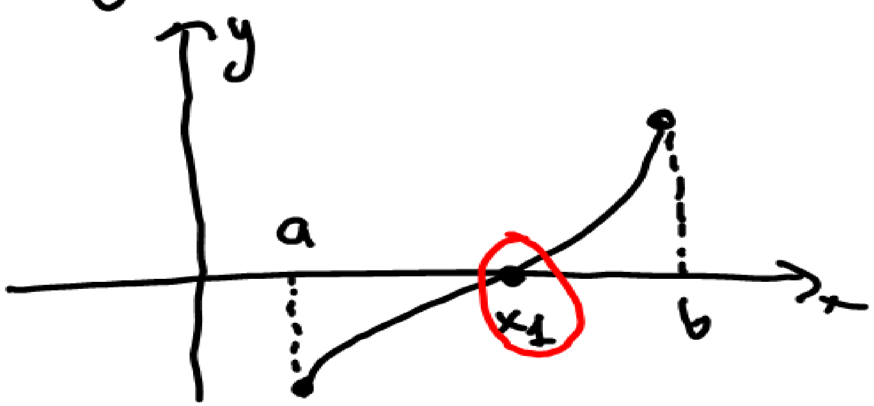
Lineer Olmayan Denklemlerin Yaklaşık Çözüm Yöntemleri

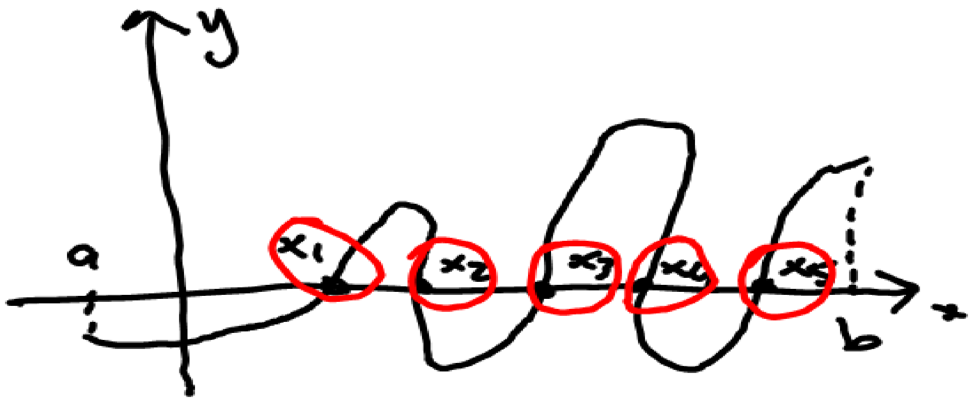
$ax^2+bx+c=0$ cebirsel denklem, lineer olmayan denklem

$e^x+x=2$ transandantal (cebirsel olmayan), lineer olmayan denklem

$f(x)=0$ lineer olmayan denklem

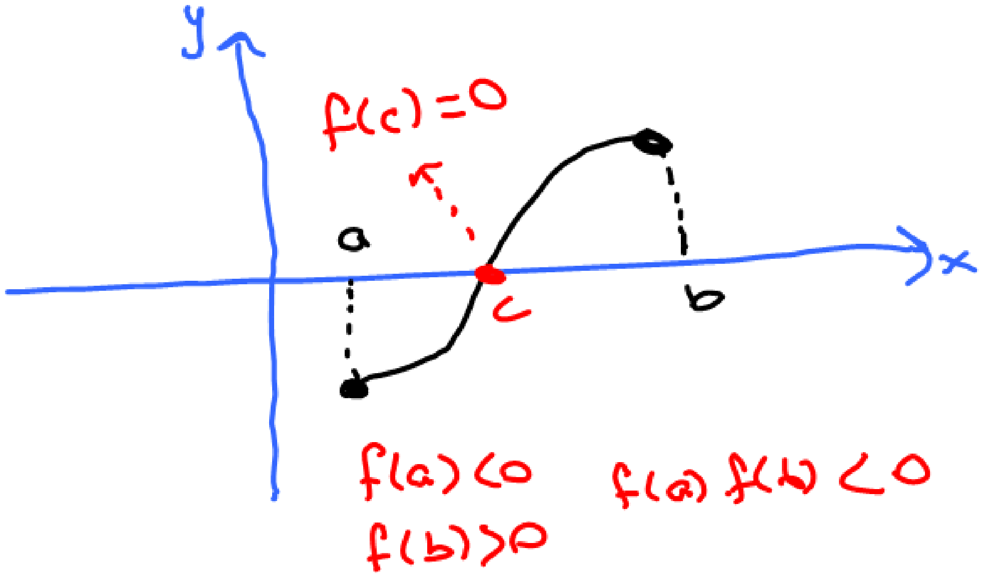
$y=f(x)$'in x eksenini kestiği nokta:

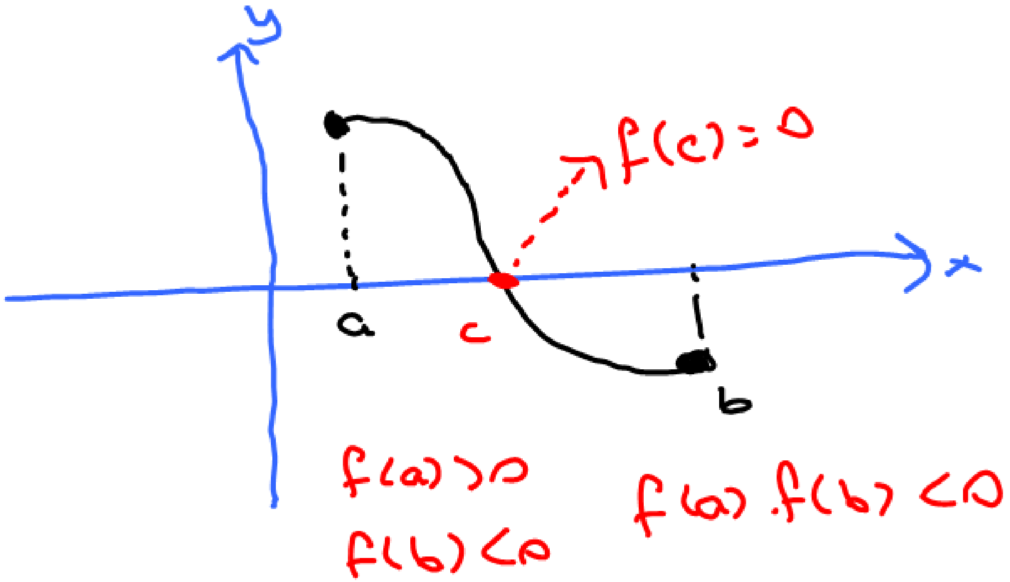




Bolzano Teoremi:

$f(x)$ 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:

$x^3+x^2-1=0$ 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.

$\downarrow$

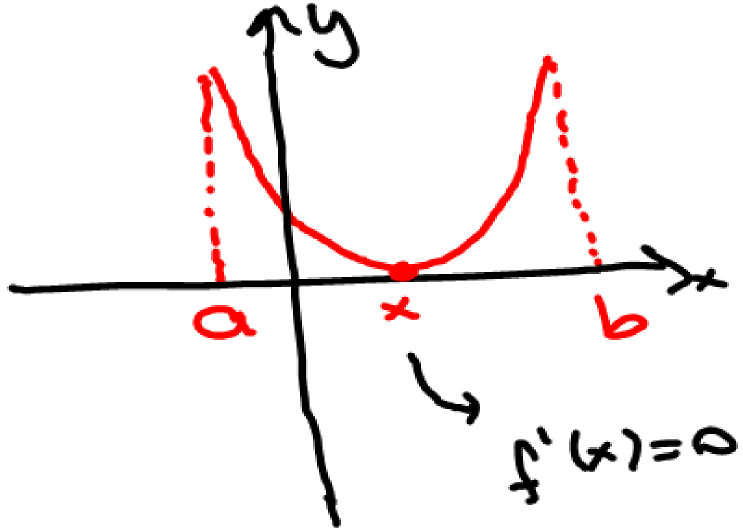
denklemin kökü

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

Ek olarak:

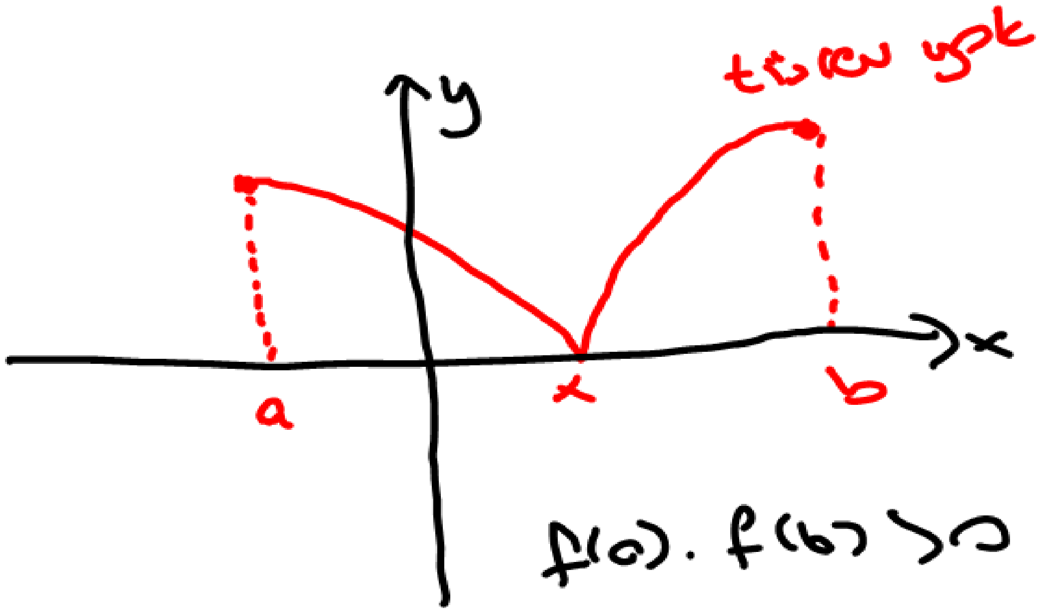
--- $f(x)$ fonksiyonu, $[a,b]$ aralığında monoton ise bu aralıkta tek kökü vardır.

--- $a<b$ olmak üzere $f'(x)\geq 0$ (veya $f'(x)\leq 0$) ve bu aralıkta $f(x)$ sürekli ise bu durumda $f(a).f(b)<0$ olması halinde $[a,b]$ aralığında bir tek kök vardır. Aksi durumda kök yoktur.



$f(a)f(b)>0$

monoton değil



monoton değil

*) $f(x)$, $[a,b]$ aralığında hiç işaret değiştirmiyorsa, belli bir $c \in [a,b]$ noktasında $f(c)=0$ oluyorsa ya $f'(c)=0$'dır ya da $f'(c)$ yoktur.

Örnek: $f(x) = \frac{1}{3}x^3 - \frac{5}{2}x^2 + 6x + 10$ fonksiyonunun

x eksenini kestiği noktaları araştırınız.

Çözüm: $f(x)$, $\mathbb{R}$ de süreklidir.

$$f'(x) = x^2 - 5x + 6$$

$$(x-2)(x-3) = 0$$

$$x = 2 \quad x = 3$$

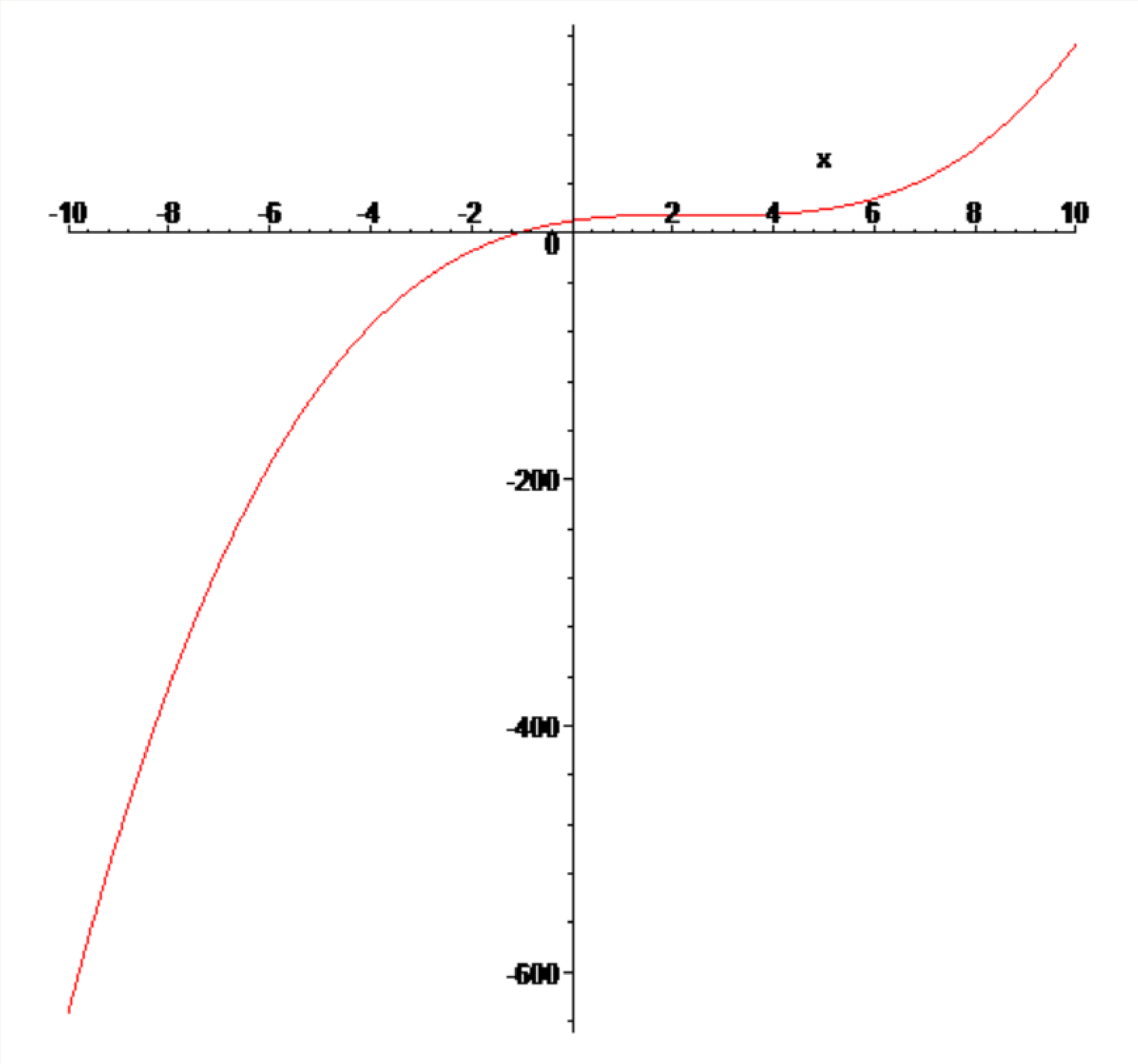
x	$-\infty$		2		3		$+\infty$
$f'(x)$		$+$	0	$-$	0	$+$	
$f(x)$		$\nearrow$	$\frac{44}{3}$	$\searrow$	$\frac{17}{2}$	$\nearrow$	

$(-\infty, 2) \Rightarrow$ bir tek kök var.

$(2, 3) \Rightarrow$ reel kök yok (monoton azalan)

$(3, +\infty) \Rightarrow$ reel kök yok (monoton artan)

$f(x) = \frac{1}{3}x^3 - \frac{5}{2}x^2 + 6x + 10$ fonksiyonunun grafiği



İstenilen *h* uzunluktaki aralıklarda Bolzano Teoriminin C kodu:

```
#include<stdio.h>
#include<locale.h>
#include<math.h>
#include<conio.h>
float F(float x)
//{return exp(x)-3*x;}
//{return pow(x,2)-4*x+4;}
{return (float(1.0/3)*pow(x,3))-(float(5.0/2)*pow(x,2))+6*x+10;}

int main()
{setlocale(LC_ALL, "Turkish");
float a,b,i,h=0.5; int s=0;
printf("a değerini giriniz: ");
scanf("%f",&a);
printf("b değerini giriniz: ");
scanf("%f",&b);
```

```
for (i=a;i<=b;i=i+h)
{if (F(i)==0) { printf ("%.5f degeri fonksiyonun gercek kokudur.\n",i);
        s++;
                continue;
        }

if ((F(i)*F(i+h))<0) {printf("Bolzano teoreminden %.5f ile %.5f arasında en az bir bir kök vardır...\n",i,i+h);
            s++;}
}


if (s!=0) printf("%.5f ile %.5f arasında en az %d tane kök vardır...\n",a,b,s);
     else
      printf(" %.5f ile %.5f arasında  kök yoktur...\n",a,b);
getch();
return 0;
}
```

Ekran Çıktısı:

h=0.5 için:

```
a değerini giriniz: -10
b değerini giriniz: 10
Bolzano teoreminden -1,50000 ile -1,00000 arasında en az bir bir kök vardır...
-10,00000 ile 10,00000 arasında en az 1 tane kök vardır...

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

h=0.25 için:

```
a değerini giriniz: -10
b değerini giriniz: 10
Bolzano teoreminden -1,25000 ile -1,00000 arasında en az bir bir kök vardır...
-10,00000 ile 10,00000 arasında en az 1 tane kök vardır...

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

Grafik Yöntemi ile Aralıkların Belirlenmesi

$f(x)=0$ lineer olmayan bir denklem

$f(x) = f_1(x) - f_2(x) = 0 \Rightarrow f_1(x) = f_2(x)$

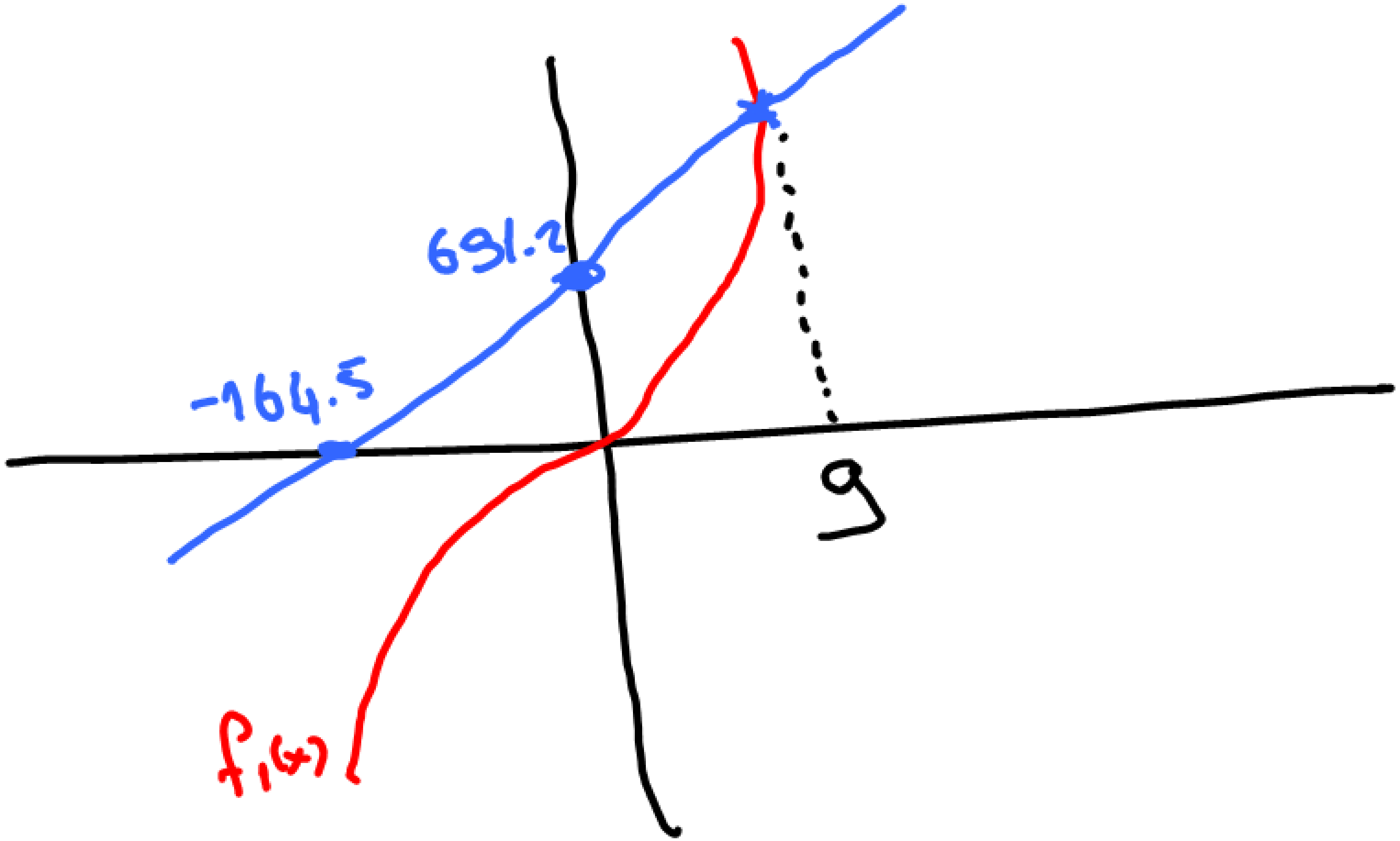
$y=f_1(x)$ ve $y=f_2(x)$ in kesim noktasının

apsisi istenilen köklerdir.

Örnek! $x^3 - 4.2x - 691.2 = 0$ denkleminin köklerinin bulundukları aralıkları araştırınız.

$f_1(x) = x^3$ ve $f_2(x) = 4.2x + 691.2$ olsun.

$f_1(x)$ ve $f_2(x)$'in grafikleri:

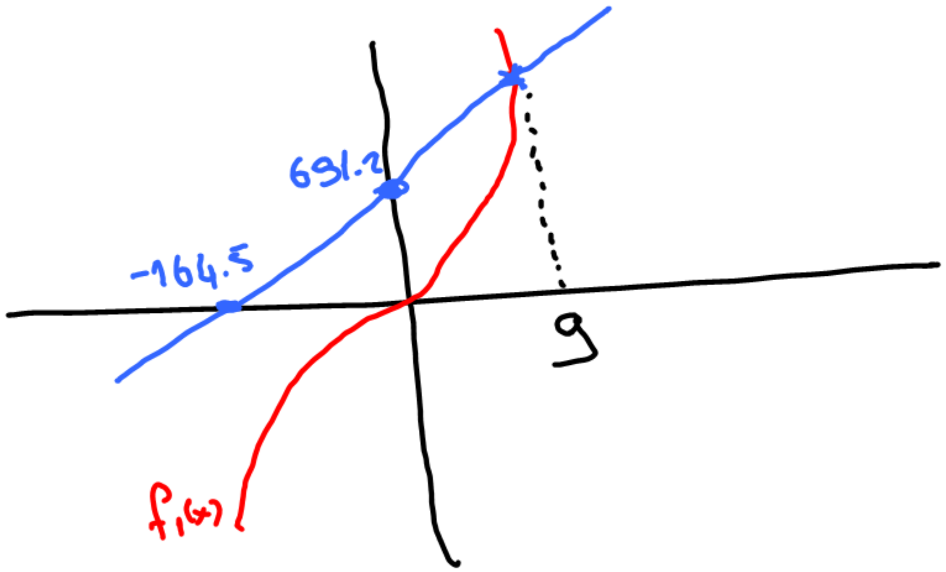


$$f'(x) = 3x^2 - 4.2 \Rightarrow x^2 = \frac{4.2}{3}$$

$$x = \mp 1.18$$

x	$-\infty$		-1.18		1.18		$+\infty$
$f'(x)$		$+$	0	$-$	0	$+$	
$f(x)$	$-\infty$	↗	-687.8	↘	-694.5	↗	$+\infty$

$f_1(x)$ ve $f_2(x)$ 'in grafikleri:



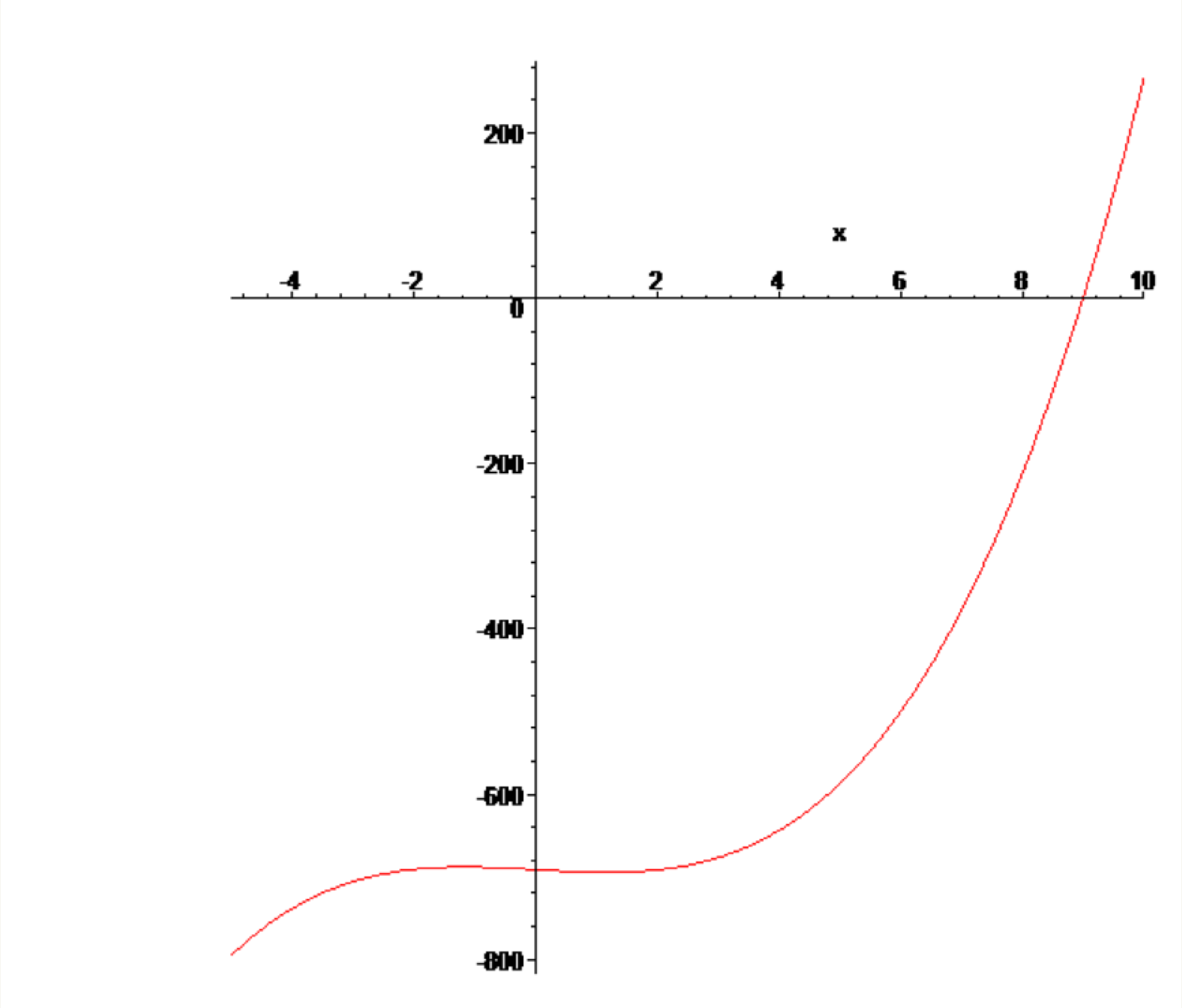
x	$-\infty$		-1.18		1.18		$+\infty$
$f'(x)$		$+$	0	$-$	0	$+$	
$f(x)$	$-\infty$	$\nearrow$	-687.8	$\searrow$	-694.5	$\nearrow$	$+\infty$

$(-\infty, -1.18) \Rightarrow$ kök yok

$(-1.18, +1.18) \Rightarrow$ kök yok

$(+1.18, +\infty) \Rightarrow$ bir kök var.

$f(x) = x^3 - 4.2x - 691.2$ fonksiyonun grafiği:



Ödev 1: $x - \sin x - 1 = 0$ denkleminin köklerinin bulunduğu aralıkları grafik yöntemi ile bulunuz.

Yanıt: $(1, \pi)$ aralığında kök vardır.

Ödev 2: $x - e^x - x^3 = 0$ denkleminin köklerinin bulunduğu aralıkları grafik yöntemi ile bulunuz.

Yanıt: $(-2, -1)$ aralığında kök vardır.

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.