endüstriyel matematik

prof. dr. gamze tanoğlu

ARALIK, 2022

ENDÜSTRİYEL MATEMATİK

PROF. DR. GAMZE TANOĞLU

SABAHAT DEFNE PLATTÜRK

CANBERK YURT

ISBN 978-975-6590-23-2

Aralık, 2022

Derleyen ve Düzenleyen
Utku KARA

önsöz

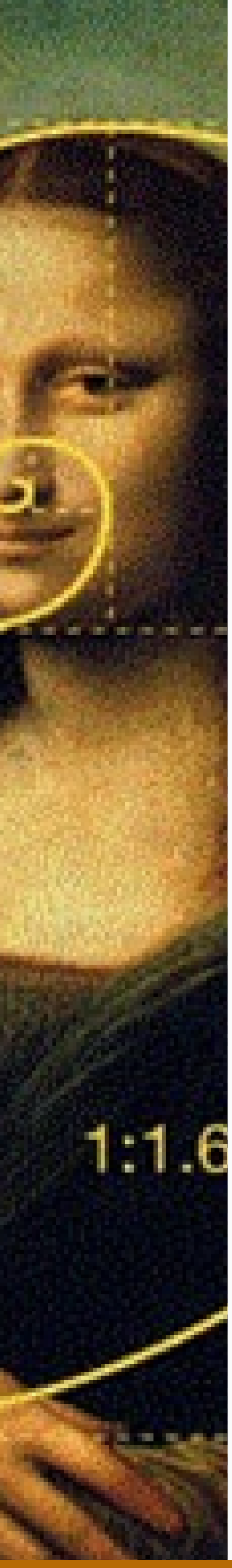
Üniversitelerin öğrencilere meslek kazandırma misyonu dijital yüzyılda yeni nesil üniversiteler tanımı ile birlikte değişerek farklı bir misyona doğru evrilmiştir. Değişen bu misyon üniversitenin ürettiği bilimi toplumun diğer paydaşları ile paylaşarak toplumu dönüştürmek biçimde ifade edilebilir.

İzmir Yüksek Teknoloji Enstitüsü, çocukları ve gençleri toplumum en değerli paydaşları görerek onların dönüşümüne ve gelişimine katkı sunmayı üniversitenin eğitim modeli olarak benimsemiştir.

Bu atölye çalışması ile lise öğrencilerinin matematik müfredatında gördüğü konuları pekiştirerek öğrenmesi benimsenmiştir. Ayrıca, öğrencilerin bilgisayar yardımı ile özellikle de algoritma mantığı kullanarak ele alınan konuları deneyimleyerek öğrenmesini hedeflemektedir. Böylece bu çalışma ile ezberden uzak kalıcı öğrenme modeli sunulmuş olup matematiğin soyut yapısından somut yapısına bir köprü oluşturulması amaçlanmaktadır.

Umuyorum bu atölye çalışması öğrencilere matematiğin günlük hayatta nerelerde kullanıldığını konusunda fikir vererek eğlenceli bir bilim olduğu konusunda da bir vizyon sunar.

Prof. Dr. Gamze TANOĞLU



İÇİNDEKİLER

projenin AMACI, KONUSU VE KAPSAMI

1.Projenin Amacı

Bu projenin amacı 11. Sınıf konularında yer alan analatik geometri, eğriler ve fonsiyonları bilgisayar ortamında başka bir bakış açısı ile anlatarak kalıcı öğrenmeyi sağlamaktır. Ele alınan konunun günlük hayattaki karşılığını gösterek öğrencilere Matematik Bilimini öğrenme konusunda motivasyon sağlamaktır.

2. Projenin Konusu ve Kapsamı

Hesaplamalı Bilim ele alınan sistemin veya fiziksel bir olayın matematiksel modelinin geliştirilmesi ve bu denklemi sayısal olarak (bilgisayar ortamında) çözmek için bir algoritmanın geliştirilmesi olarak tanımlanabilir. Ayrıca bilgisayar yazılımıyla bu algoritmanın işlerlik kazanması ve uygulanması, fiziksel olayın sayısal simülasyonu, hesaplanan sonuçların kapsamlı bir şekilde temsil edilmesi, sunulması ve hesaplanan sonuçların doğrulanması ve yorumlanması olarak özetlenebilir.

Bu atölye çalışması, doğruların ve eğrilerin matematiksel denklemleri ile ilgili kısa bir hatırlatma yapılarak başlayacaktır. Görsel olarak bu anlatım Matlab uygulamaları ile zenginleştirilecektir.

Atölye çalışmasının son kısmında eğriler ve hacim arasında ilişki kuracağız, öğrenciler bu ilişkiyi kullanarak kendilerinin yaratttığı üç boyutlu cisimleri nesnelere dönüştürerek sergileyecekler. Böylece matematiği endüstri ilişkilendirerek uygulamalı matematik hakkında öğrencilerimizin fikir sahibi olmalarını sağlayacağız.

Bu atölye çalışmaları ile öğrencilerin, matematiğin bir alanı olan hesaplamalı bilimler yardımı ile günlük hayatta matematik ne işe yarar ve nasıl kullanılır sorularına yanıt vermeye çalışacağız.

Bu proje ilehedeflenen kazanımlar:

- Tek boyut, 2 boyut, 3 boyut kavramlarını pekiştirmek ve kavratmak,
- 3d yazıcılarla nesne oluşturmak,
- Bunların günlük hayattaki karşılıklarını bulmak,
- Atölye çalışmasının sonunda bir sergi açmak

olarak belirlenmiştir.

3. Boyut Kavramı

Bu bölümde bir boyut, iki boyut, üç boyut kavramlarını MATLAB yardımı ile görselleştirerek anlatacağız.

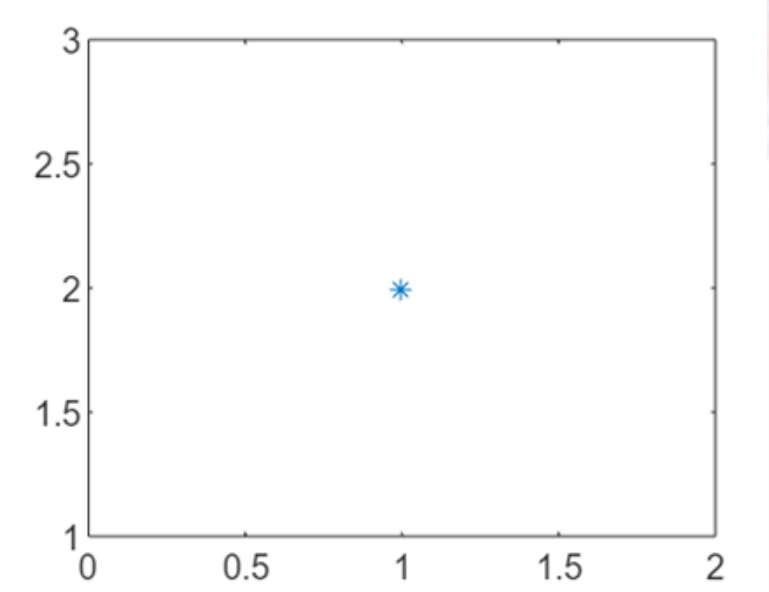
3.1. Bir Boyut

3.1.1.1Koordinat Sisteminde Bilinen Nokta Nasıl Çizilir?

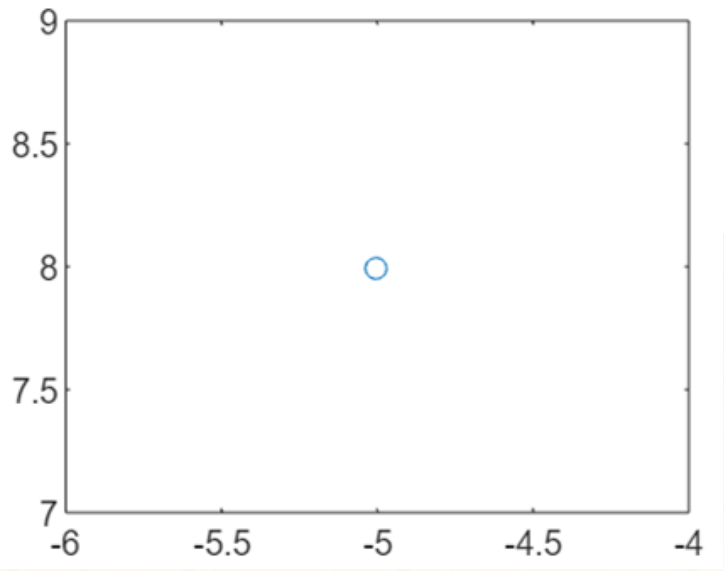
plot(a,b,’noktaya koymak istediğimiz şekil’)

Örneğin;

plot(1,2,'*') komutu koordinat sisteminde (1,2) noktasına bir * çizer.



plot(-5,8,'o') komutu koordinat sisteminde (-5,8) noktasına bir o çizer.



3.2. İki Boyut

Bu bölümde, Doğru ve Eğri denklemleri çizmek için Matlab uygulamaları yapılacaktır.

3.2.1 Koordinat Sisteminde Denklemi Bilinen Doğru Nasıl Çizilir?

Önce bazı Matlab komutlarını tanıyalım:

linspace(a,b,n) bu komut (a,b) aralığını n parçaya böler.

a:h:b bu komut (a,b) aralığını h uzunluğundaki parçalara ayırır.

- Önce doğruyu çizmek istediğimiz aralığı tanımlamalıyız.

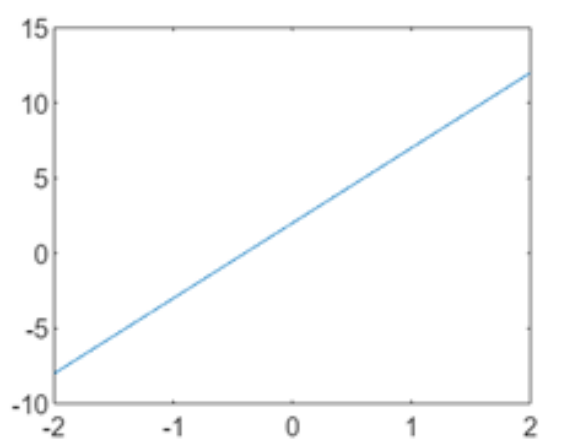
x=linspace(-2,2,50) ya da x=-2:0.1:2 gibi aralığımızı tanımlayabiliriz.

- Şimdi çizmek istediğimiz doğrunun denklemini yazalım.

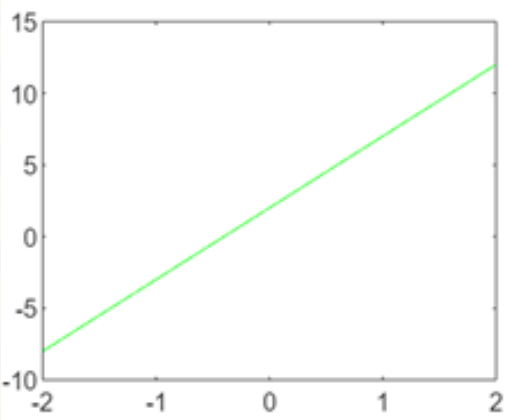
y=5.*x+2 gibi yazabiliriz. “.*” işareti burada çarpma işlemini sembolize eder.

- Aralığımız ve doğru denklemimizi tanımladığımıza göre artık doğrumuzu çizebiliriz

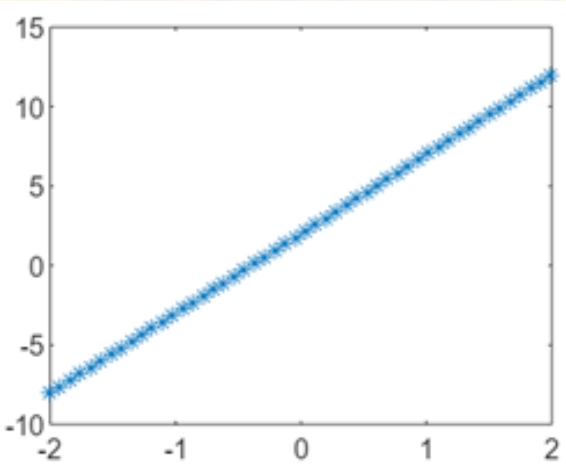
plot(x,y)
(çizgi şeklinde doğrumuzu çizer.)



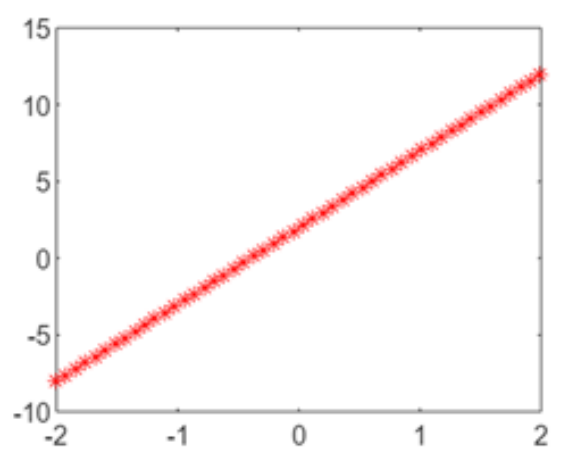
plot(x,y,’g’)
(yeşil renkli çizer.)



plot(x,y,’*’)
(doğru üzerinde * çizer.)



plot(x,y,’r*’)
(kırmızı renkli * çizer.)



3.2.2 Herhangi Bir Fonksiyonun Grafiği Nasıl Çizilir?

- Yine fonksiyon değişkenimiz olan x in aralığını tanımlayarak başlayalım.

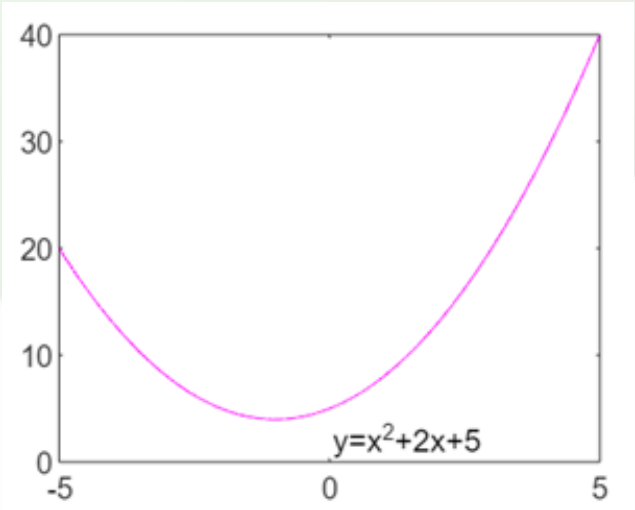
Herhangi bir (a,b) aralığını n parçaya bölmek için linspace(a,b,n) yazabiliriz.

- Fonksiyonumuzu tanımlayalım.

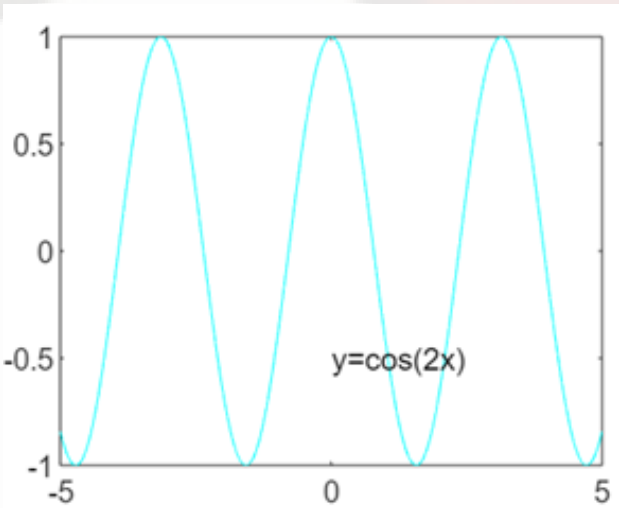
y=x.^2+2.*x+5 gibi 2. Dereceden fonksiyonumuzu tanımlayabiliriz. Bu fonksiyon cos(x), sin(x), exp(x) gibi de tanımlanabilir.

- plot(..) komutunu kullanarak fonksiyonumuzu çizdirelim.

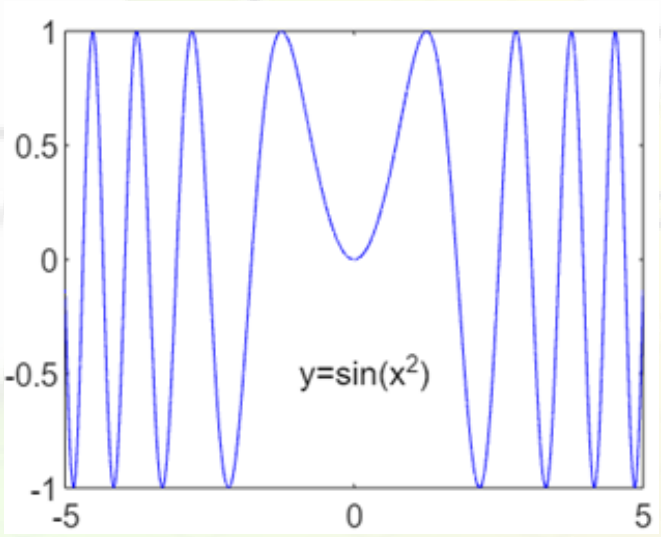
```
x=linspace(-5,5,100);
y=x.^2+2.*x+5;
plot(x,y,'m')
(renk kodu 'm')
```



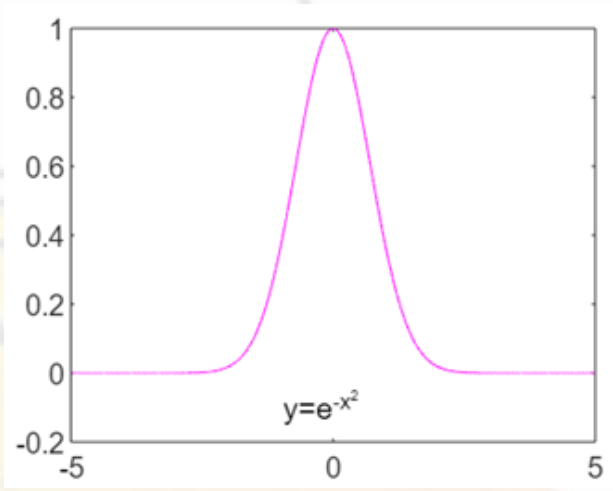

```
x=linspace(-5,5,100);
y=cos(2.*x);
plot(x,y,'c')
(renk kodu 'c')
```



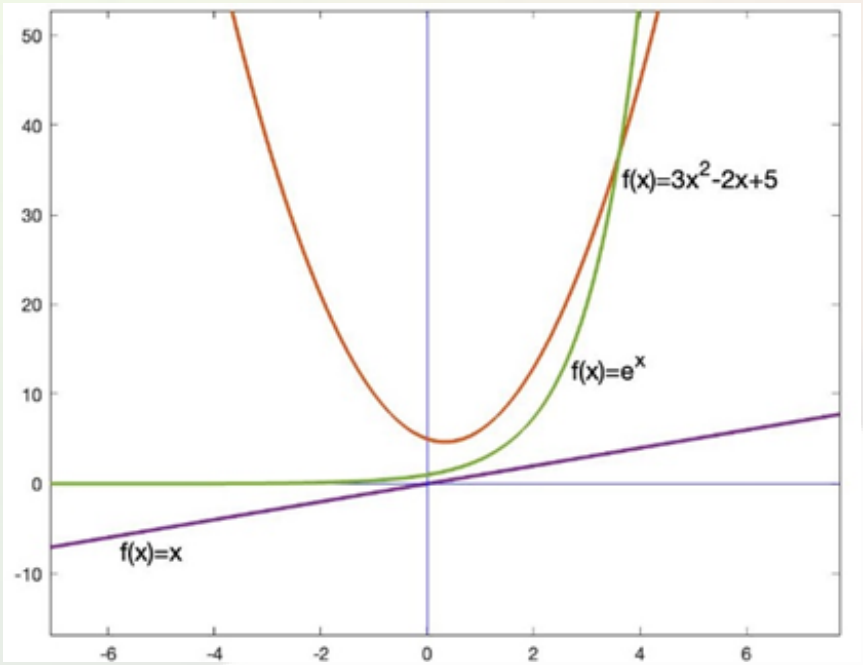
```
x=linspace(-5,5,100);
y=sin(x.^2);
plot(x,y,'b')
(renk kodu 'b')
```




```
x=linspace(-5,5,100);
y=exp(-x.^2);
plot(x,y,'m')
(renk kodu 'm')
```



Eğri nedir?



Eğri, doğrusal olmayan noktalar kümesidir. İkinci dereceden eğrilere parabol denir.

Genel Parabol Denklemi

$$f(x)=ax^2+bx+c$$

Parabol denklemi ayrıca $y=a(x-r)^2+k$ şeklinde de yazılabilir.

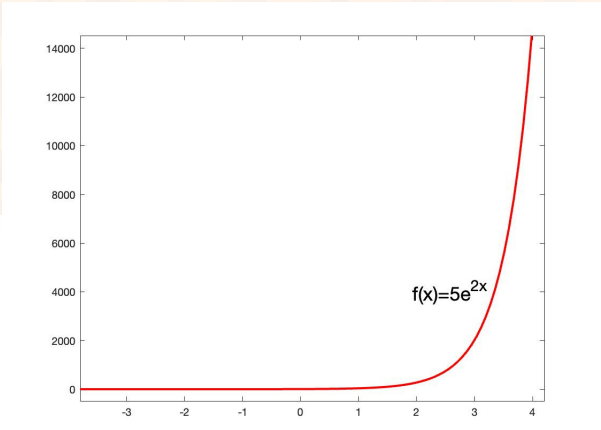
- a>0 ken kollar yukarı bakar.
- a<0 ken kollar aşağı bakar.
- |a| değeri büyüdükçe kollar y eksenine yaklaşır.

Tepe noktası: Parabolün en büyük veya en küçük değerini aldığı noktadır. T(r,k) şeklinde gösterilir.

$r=(-b)/2a$, $k=f(r)$ şeklinde bulunur.

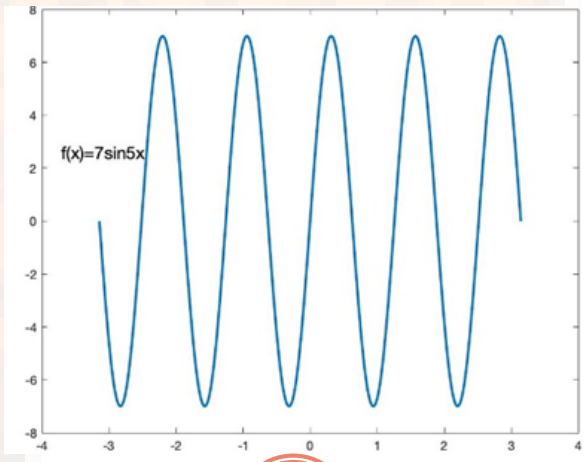
Üstel Eğri

$f(x)=ae^{bx}$



Trigonometrik Eğri

$f(x)=a\sin(bx)$



4. Döngüler

4.1. for Döngüsü

for döngüsü, döngüyü kaç kere çalıştıracağımızı bildiğimiz durumlarda kullanılır.

Basit bir örnek ile for döngüsünü ve iterasyon mantığını anlayalım.

```
x=0;
for i=1:10
    x=x+1;
    disp(x)

end
```

bu döngü i=1 ile başlayıp her seferinde içerisine yazdığımız işlemi yaparak (yani x'e 1 ekleyerek) ve i'yi 1 arttırarak çalışmaya devam eder. i=10 olduğunda döngü son kez çalışıp durur.

disp(..) komutu içerisinde yazan sayıyı, yada sayı kümesini ekrana yazdırmaya yarar.

4.2. While Döngüsü

while döngüsü döngüyü kaç kere çalıştıracağımızı bilmediğimiz durumlarda kullanılır. Örneğin değişkenimizin belli bir sayıdan küçük ya da büyük kalmasını istediğimizde while kullanabiliriz.

Basit bir örnek ile while döngüsünü ve iterasyon mantığını anlayalım.

```
i=1;
x=0;
while i<=10
 x=2*x+1;
     fprintf(' %d ',x)
i=i+1;
end
```

PROGRAM ÇIKTISI: 1 3 7 15 31 63 127 255 511 1023

bu döngü i=1 ile başlayıp her seferinde içerisine yazdığımız işlemi yaparak ve i'yi 1 arttırarak çalışmaya devam eder. i=10 olduğunda döngü son kez çalışıp durur.

4.2.1 fprintf Komutu

Bu komut içerisinde tırnak içinde yazılan metni ekrana yazdırır. fprintf kullanarak sayı yazdırmak için fprintf('%d',sayı) yazılabilir.

Örnek 1: Girilen sayıya limit'ten küçük olduğu sürece x sayısını ekleyen döngüyü yazalım.

Bu örnekte girilen sayı,limit ve x değerlerini kullanıcıdan almak için Matlab'ın input komutunu kullanacağız. Bu komut aşağıdaki gibi kullanılır.

```
sayi=input('sayı giriniz: \n');
x=input('eklenecek sayıyı giriniz: \n');
limit=input('limit giriniz: \n');
```

Kullanıcıdan değerleri aldığımıza göre döngümüzü yazabiliriz.

```
while sayi<limit-x
   sayi=sayi+x;
end
fprintf('%d',sayi)
```

Örnek 2 : Fibonacci Dizisinin ilk 20 Elemanını Birlikte Bulalım

FİBONACCİ DİZİSİ NEDİR: Fibonacci dizisi, 0 ve 1 ile başlayan ve her sayının kendisinden önce gelen iki sayının toplanması ile elde edildiği bir sayı dizisidir. İtalyan matematikçi Leonardo Fibonacci'den adını alır.

Bu dizinin elemanlarını veren Matlab kodunu yazmak için önce sıfırlarla dolu bir vektör tanımlayalım. Bunun için zeros(satır,sütun) Matlab komutunu kullanabiliriz.

x=zeros(1,20) % 20 eleman bulmak istediğimiz için vektörün içine 20 tane sıfır doldurduk.

Ardından ilk iki elemanımızı vektör içindeki ilk iki eleman olarak yerleştirelim.

```
x(1)=0;
x(2)=1;
```

Şimdi sıra iterasyon yaparak diğer elemanları bulmaya geldi. Yukarıda da bahsedildiği gibi her sayı, kendinden önce gelen iki sayının toplamına eşit olmalı. Döngümüzü yazalım,

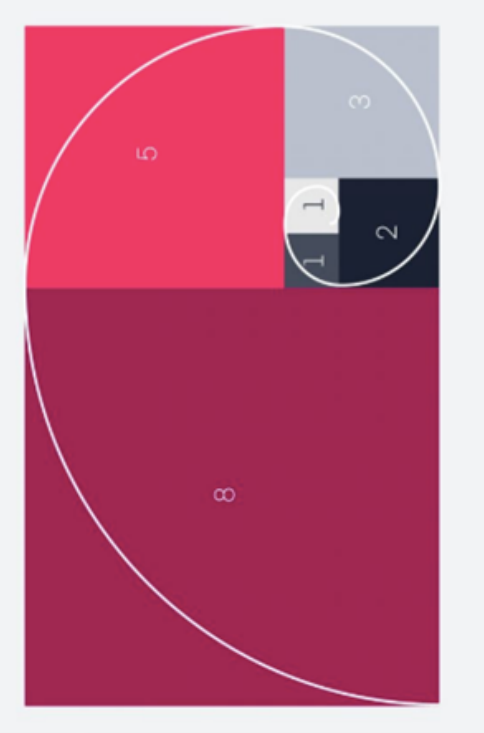
```
for i=3:20
    x(i)=x(i-1)+x(i-2);

end
```

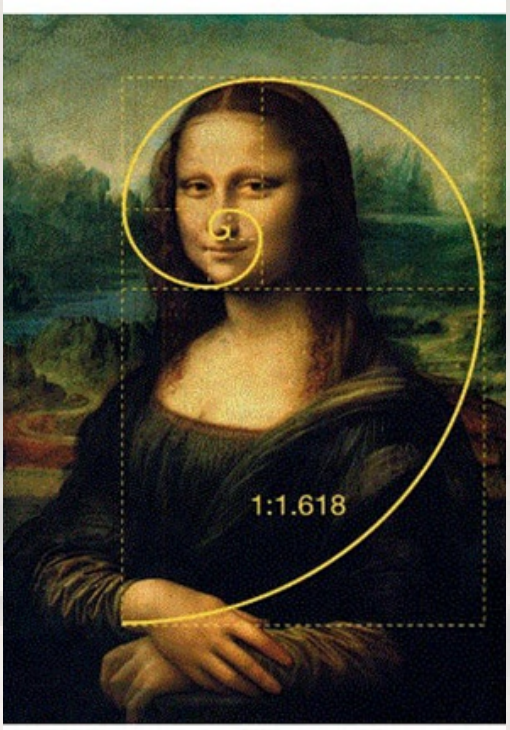
bu döngüde i'nin ilk değeri 3 çünkü x vektörünün ilk iki elemanını zaten daha önce tanımlamıştık.

Ve Fibonacci Dizisinin ilk 20 elemanı:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181

Örnek 3 : Altın Oran





Altın oran, π gibi irrasyonel bir sayıdır ve ondalık sistemde yazılışı; 1,61803398874989 4...'tür.(noktadan sonraki ilk 15 basamak.) Bu oranın kısaca gösterimi:

φ= (1+√5)/2 dir.

Fibonacci dizisinin ardışık her 2 elemanının oranı aslında altın orana yaklaşır. Programlayarak görelim.

```
phi=(1+sqrt(5))/2; % altın oran değerini tanımlayalım.
for i=1:19
altin_oran(i)=x(i+1)/x(i);
% büyük olan elemanın küçük olan elemana oranını alıyoruz.
hata(i)=abs(altin_oran(i)-phi);

% bulduğumuz değer ile altın oranın farkının mutlak değerini alıyoruz.
end
disp(hata) % bu satır hata değerlerini ekrana yazdırır.
```

5. Matematiksel Modelleme

Hesaplamalı Bilim ele alınan sistemin veya fiziksel bir olayın matematiksel modelinin geliştirilmesi ve bu denklemi sayısal olarak (bilgisayar ortamında) çözmek için bir algoritmanın geliştirilmesi olarak tanımlanabilir.

Bilgisayar yazılımıyla bu algoritmanın işlerlik kazanması ve uygulanması, fiziksel olayın sayısal simülasyonu, hesaplanan sonuçların kapsamlı bir şekilde temsil edilmesi, sunulması ve hesaplanan sonuçların doğrulanması ve yorumlanması olarak özetlenebilir.

Matematiksel modelleme en genel anlamıyla matematik veya matematik dışındaki bir olayı, olguyu, olaylar arasındaki ilişkileri matematiksel olarak ifade etmeye çalışma, bu olaylar ve olgular içerisinde matematiksel örüntüler ortaya çıkarma sürecidir.

Matematiksel Model, bir sistemi açıklamaya, farklı bileşenlerin etkilerini incelemeye ve bir davranış hakkında öngörüde bulunmak için yardımcı olabilir. Bu modelleri denklemler yardımı ile ifade edeceğiz. Bu bölümde Matematiksel Model nedir? Sorusuna yanıt aranacak olup, cep telefonumuzun irrasyonel sayısının yaklaşık değerinin hesaplanmasının altında yatan matematiksel sistemi anlamaya çalışacağız. Kendi algoritmamızı yazarak bu hesaplamayı yapacağız. Aşağıda yaklaşık değerleri verilen irrasyonel sayılarının yaklaşık değerlerini yazacağız algoritma ile hesaplayacağız.

$\sqrt{1} = 1$	$\sqrt{11} = 3.3166$	$\sqrt{21} = 4.5825$
$\sqrt{2} = 1.4142$	$\sqrt{12} = 3.4641$	$\sqrt{22} = 4.6904$
$\sqrt{3} = 1.732$	$\sqrt{13} = 3.6055$	$\sqrt{23} = 4.7958$
$\sqrt{4} = 2$	$\sqrt{14} = 3.7416$	$\sqrt{24} = 4.8989$
$\sqrt{5} = 2.236$	$\sqrt{15} = 3.8729$	$\sqrt{25} = 5$
$\sqrt{6} = 2.4494$	$\sqrt{16} = 4$	$\sqrt{26} = 5.099$
$\sqrt{7} = 2.6457$	$\sqrt{17} = 4.1231$	$\sqrt{27} = 5.1961$
$\sqrt{8} = 2.8284$	$\sqrt{18} = 4.2426$	$\sqrt{28} = 5.2915$
$\sqrt{9} = 3$	$\sqrt{19} = 4.3588$	$\sqrt{29} = 5.3851$
$\sqrt{10} = 3.1622$	$\sqrt{20} = 4.4721$	$\sqrt{30} = 5.4772$

5.1 Kirişler Yöntemi (Secant Method)

Kirişler yöntemi, denklemin kökünün yaklaşık değerlerinin tahmin yolu ile belirlendiği, ilk 2 değerin bu şekilde girildiği ve bu değerler kullanılarak çizilen kirişler yardımıyla yaklaşık kökün bulunduğu bir sayısal kök bulma yöntemidir.

Yöntemin Türetilişi:

$(x_0, y_0), (x_1, y_1)$ *noktalarından geçen doğrunun eğimi* $\Rightarrow m = \frac{y_1 - y_0}{x_1 - x_0}$

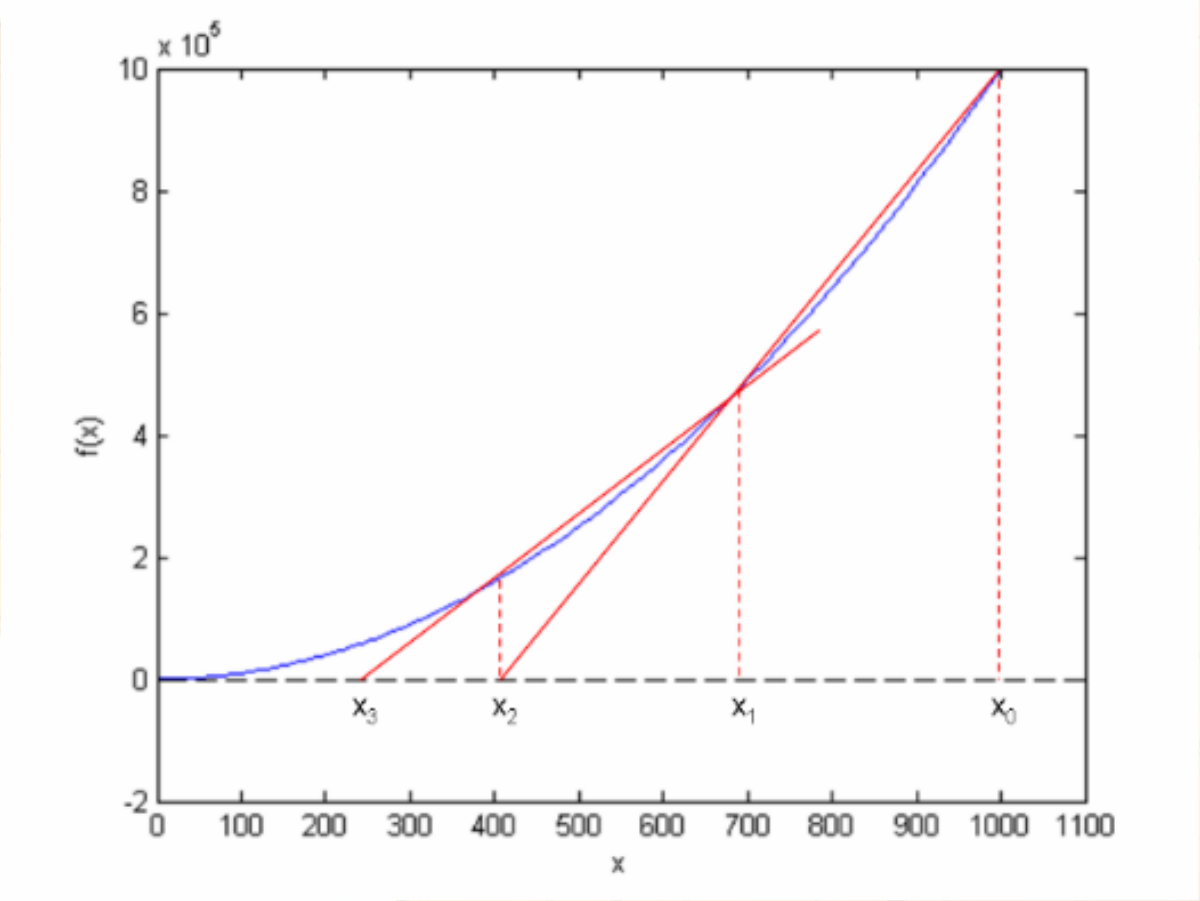
$(x_1, y_1), (x_2, y_2)$ *noktalarından geçen doğrunun eğimi* $\Rightarrow m = \frac{y_2 - y_1}{x_2 - x_1}$

Eğimleri birbirine eşitleyip ortaya çıkan denklemin kökü olan x_2'yi bulmak için $y_2 = 0$ koyalım.

$$\frac{y_1 - y_0}{x_1 - x_0} = \frac{0 - y_1}{x_2 - x_1} \quad \Rightarrow \quad x_2 = x_1 - y_1 \cdot \frac{x_1 - x_0}{y_1 - y_0}$$

x_2'yi önceki iki x değerini kullanarak elde edebileceğimiz formülü bulduğumuza göre, bu formülü kullanarak Matlab kodumuzu yazıp iterasyon yapmaya başlayabiliriz.

Kirişler Yönteminin iterasyon mantığı aşağıdaki grafik ile geometrik olarak gözlemlenebilir. Grafikteki kırmızı doğrular iterasyon yapılırken çizilen kirişlerdir.



Algoritma Nasıl Çalışıyor?

Grafikteki x(0) ve x(1), tahmin değerleri, kullanılarak ilk kiriş çizilir ve x(2) noktası bulunur. Daha sonra x(1) ve x(2) kullanılarak 2. kiriş çizilir ve x(3) noktası bulunur. Denklemin köküne yeterince yaklaşıncaya kadar bu işlem devam eder.

5.2 KİRİŞLER YÖNTEMİ ALGORİTMASI

Input

- f fonsiyonu
- x(1), x(2) başlangıç değerleri
- n: maksimum iterasyon sayısı
- $\varepsilon = 10^{-16}$ hata toleransı

for i=,...n **do**

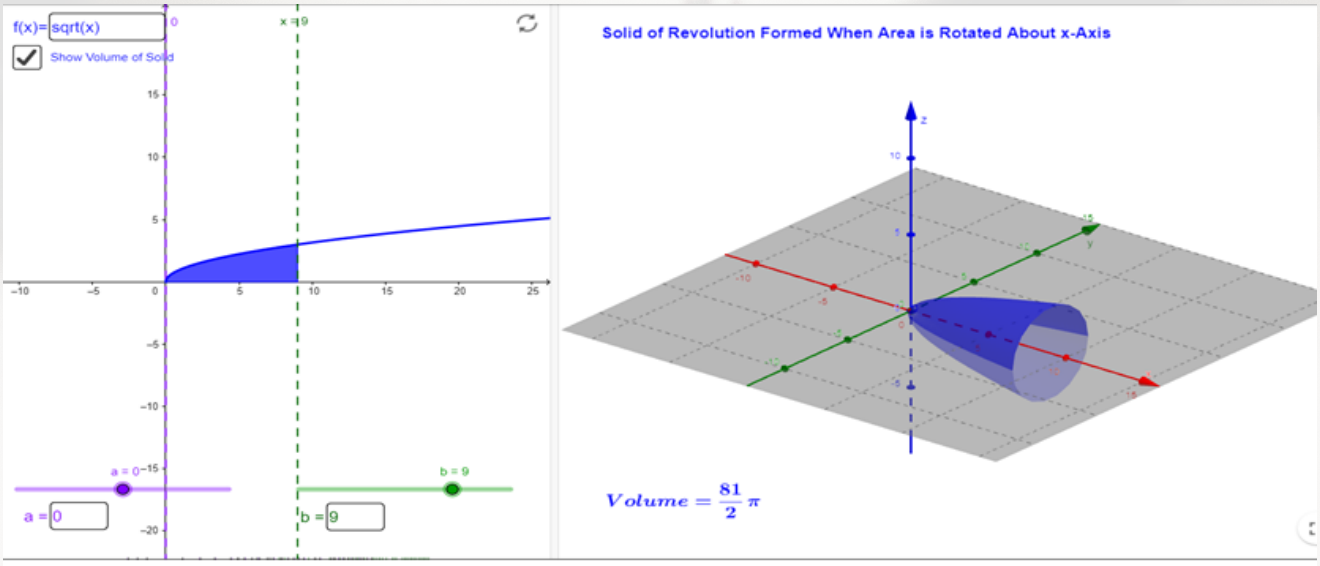
$$x_{i+1} = x_i - f(x_i) \cdot \frac{x_i - x_{i-1}}{f(x_i) - y_{i-1}}$$

if $|x_i - x_{i-1}| < \varepsilon$ **then stop**

end

6. Endüstriyel Matematik

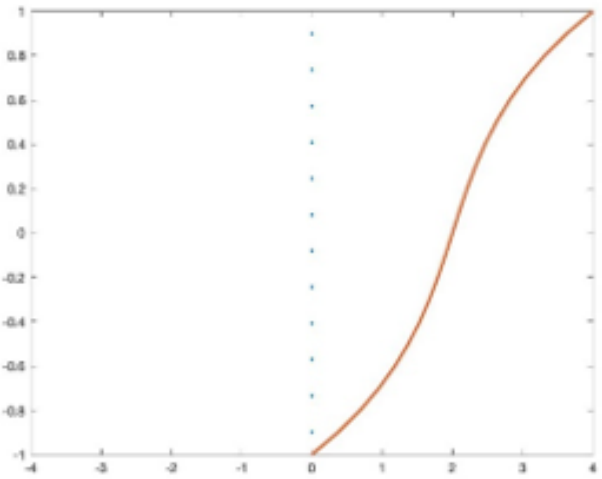
Atölye çalışmasının son kısmında eğriler ve hacim arasında ilişki kuracağız, öğrenciler bu ilişkiyi kullanarak kendilerinin 3d yazıcı ile yarattığı üç boyutlu cisimleri nesnelere dönüştürerek sergileyecekler. Böylece matematiği endüstri ile ilişkilendirerek uygulamalı matematik hakkında öğrencilerimizin fikir sahibi olmalarını sağlayacağız.

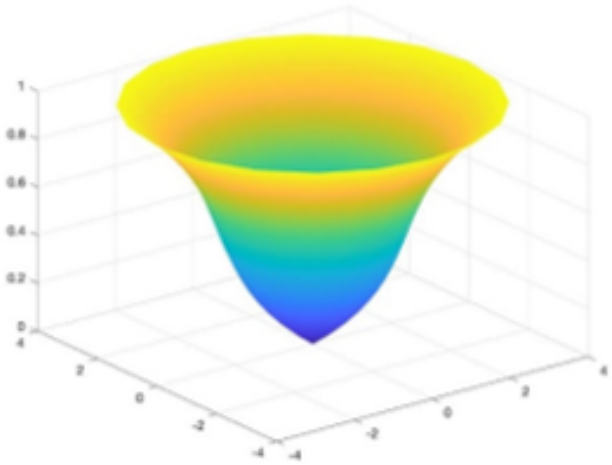


6.1 Matlab Uygulama: Cylinder Metodu

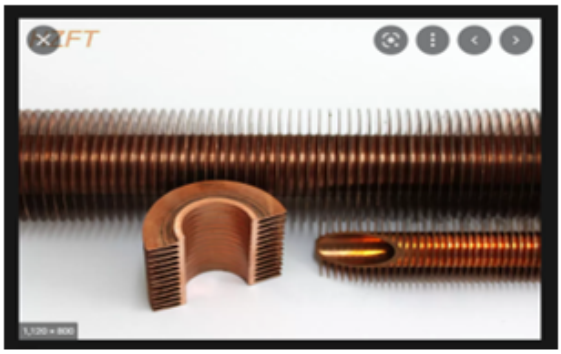
Herhangi bir eğriyi istediğimiz eksen üzerinde çevirerek 3 boyutlu bir cisim elde etmek için için Matlab'ın cylinder komutunu kullanabiliriz.

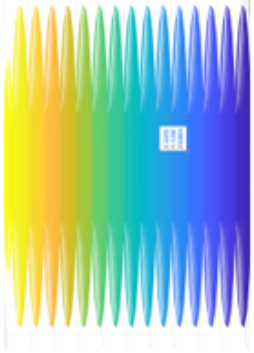
```
x = -1:0.1:1;
y = x.^3 +x+2;
plot(x,y)
figure
[X,Y,Z] = cylinder(y);
surf(X,Y,Z)
shading interp
```

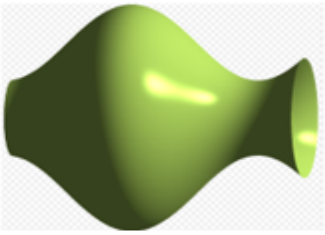




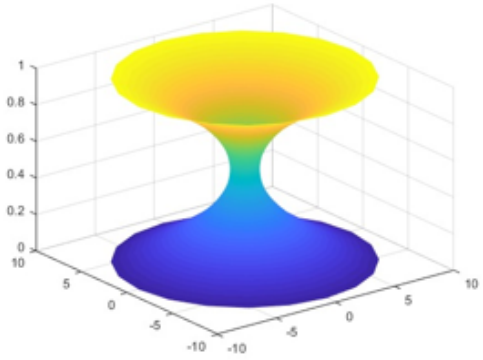
6.2 3D Baskı: Cisim Üretmek

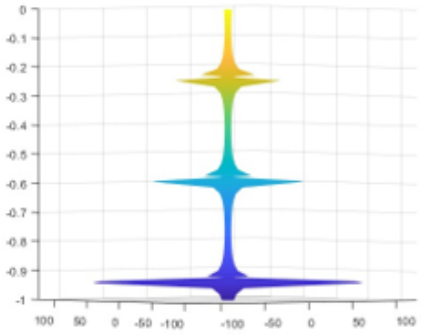


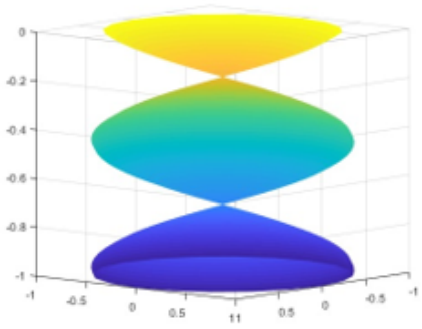


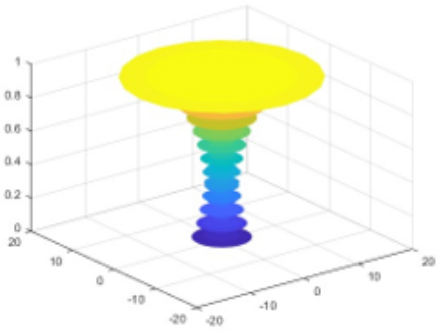


Önceki çalışmalardan örnekler:









3D BASKI ÖRNEĞİ