

# İçindekiler

- ☉ Giriş
- ☉ Matris işlemleri
- ☉ Sayı Formatları
- ☉ Temel Lineer Cebir İşlemleri
- ☉ Diziler (Arrays)
- ☉ Programı Dallandıran İfadeler (if-end , switch-case yapıları)
- ☉ Döngüler (for-end ve while-end döngüleri)
- ☉ Grafik
- ☉ Dosya yazdırma-okuma
- ☉ Fonksiyon dosyası oluşturma
- ☉ Derleme

# Kaynakça

- © Doğan, U., (2009), Temel Bilgisayar Bilimleri Ders Notları, YTÜ, Lisans Ders Notları, İstanbul.
- © İnan, A., "MATLAB Klavuzu", Papatya Yayınları, İstanbul, 2007.
- © Demirel, H., (2005), Dengeleme Hesabı, YTÜ, Lisans Ders Notları, İstanbul.
- © Ayten, U. E., “Algoritma geliştirme ve programlamaya giriş”, Temel Bilgisayar Bilimleri Ders Notları.
- © Serbes, A., “Algoritma geliştirme ve programlamaya giriş”, Temel Bilgisayar Bilimleri Ders Notları.
- © Uzunoğlu M., vd. (2002), Matlab, Türkmen Kitabevi, İstanbul.
- © <http://www.mathworks.com/matlabcentral/>
- © <http://www.mathworks.com/matlabcentral/fileexchange/>

# MATLAB (MATrix LABoratory)

<http://www.mathworks.com/matlabcentral/>

**MATLAB**, yüksek performanslı bir uygulama yazılımı ve bir programlama dilidir.

- ⌚ MATLAB'ın temelindeki yapı, boyutlandırma gerektirmeyen matrislerdir.
- ⌚ Yaptığımız tüm girdi ve çıktılar, belirteç gerektirmeksizin bir matris tanımlar.
- ⌚ İlk olarak Fortran dili ile yazılan MATLAB, daha sonra C ile yazılmıştır.

# MATLAB (MATrix LABoratory)

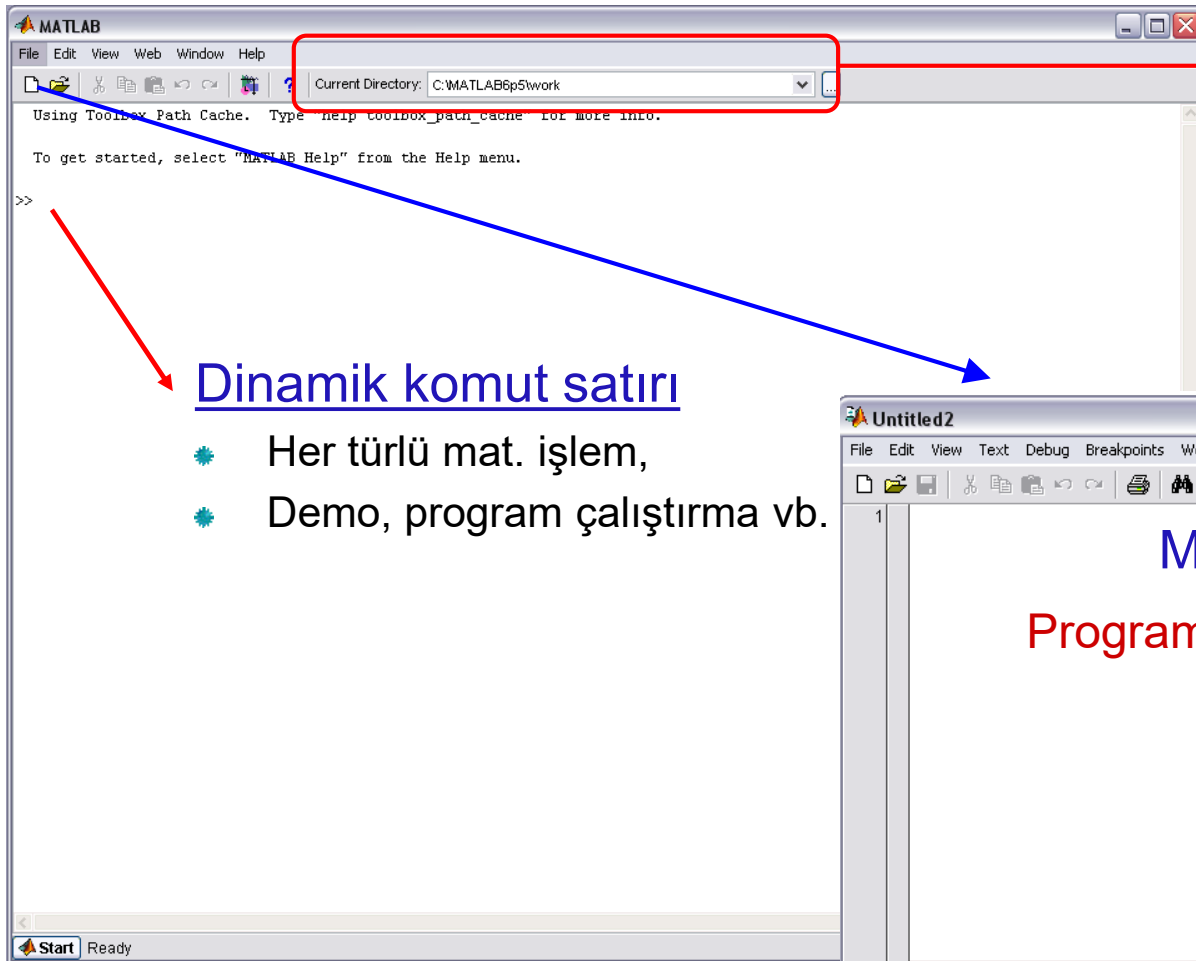
- Matlab'de hazır programlar vardır. Bu programlara **fonksiyon** adı verilir.
- Matlab fonksiyonlarının kullanımı, matematikteki  $y=f(x)$  fonksiyonunun kullanımıyla özdeştir.
- Örneğin,  $a=\sin(x)$  fonksiyonunda, **sin** fonksiyonu,  $x$  açısının (input-girdi) değerini hesaplar; kullanıcı bu değeri, örneğin, bir  $a$  değişkenine atar.  $a$  değeri **sin** fonksiyonunun bir çıktısıdır (output).

# MATLAB (MATrix LABoratory)

## Avantajları:

- ☉ Kullanım kolaylığı,
- ☉ İşletim sistemi uyumluluğu,
- ☉ Sayısal analiz işlemlerindeki kolaylıklar,
- ☉ Hazır fonksiyonlar (function files),
- ☉ Görüntüleme (visualization) kolaylığı (grafik çizim),
- ☉ GUI geliştirme kolaylığı,
- ☉ MATLAB derleyicisi (exe: executable dosya ile win32 uygulamaları),
- ☉ Toolboxes (Araç kutuları) :hazır programlar!

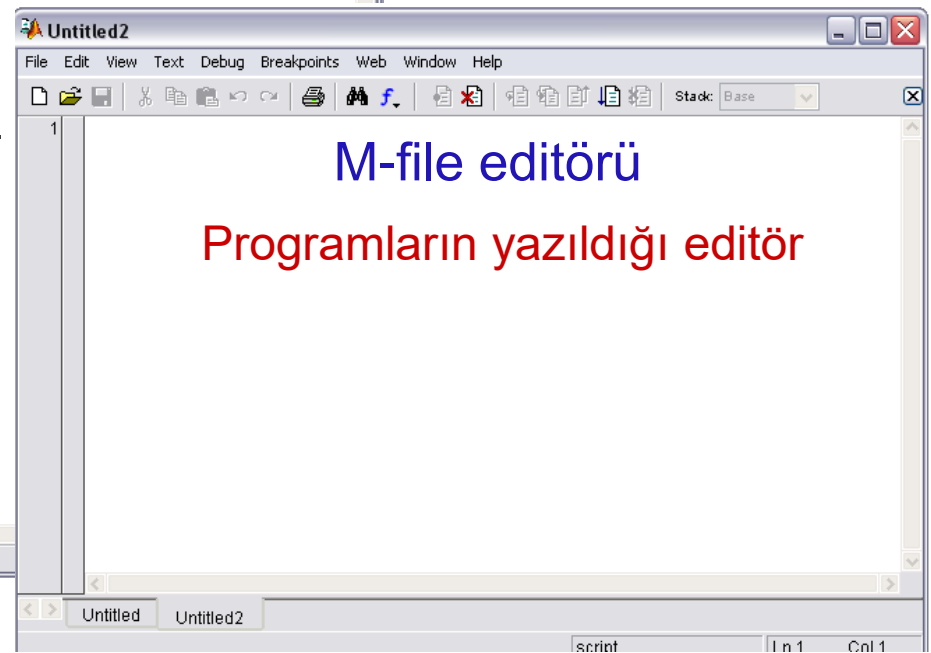
# MATLAB/Command window (komut penceresi)



Current directory  
(çalışma klasörü)

## Dinamik komut satırı

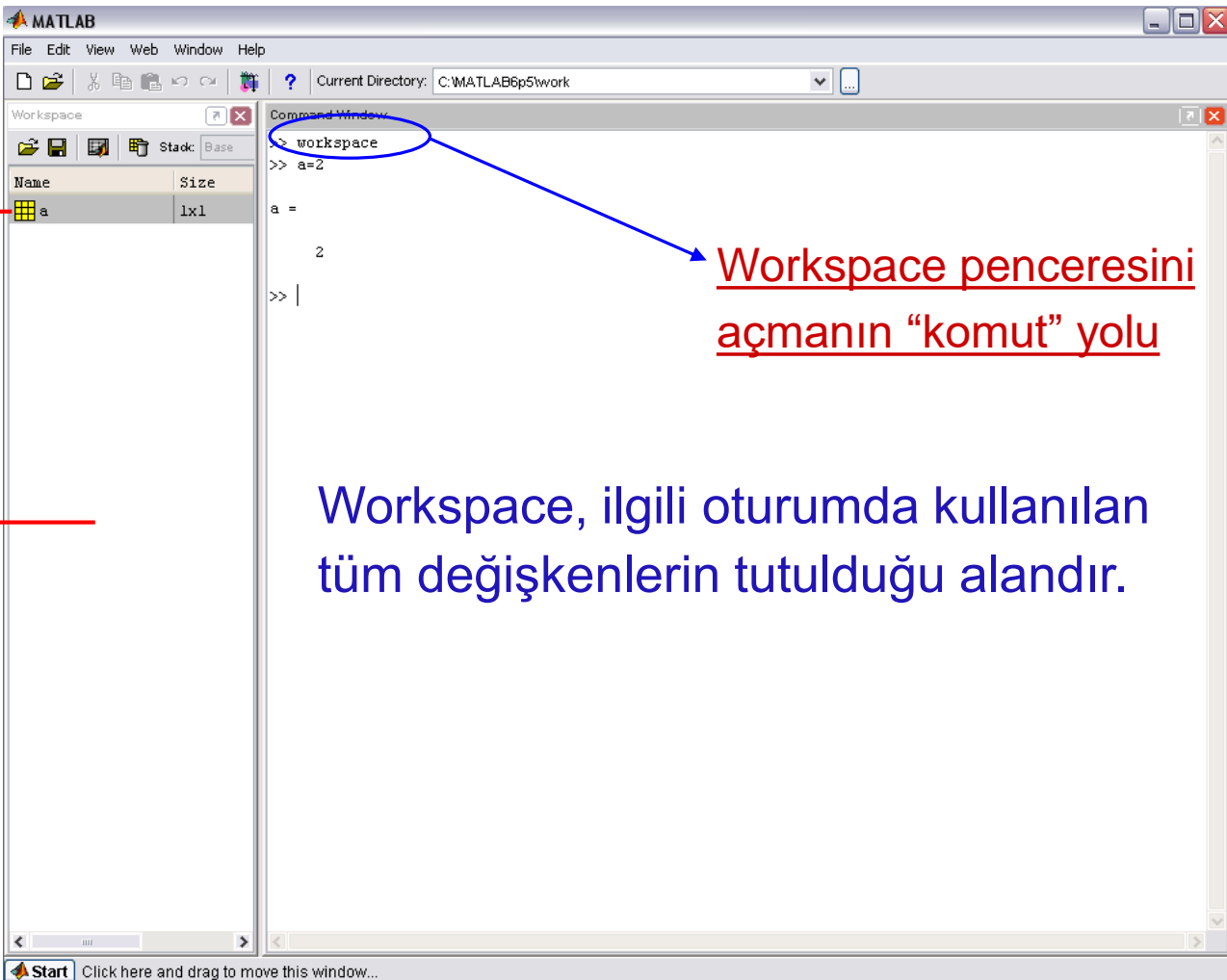
- Her türlü mat. işlem,
- Demo, program çalıştırma vb.



M-file editörü

Programların yazıldığı editör

# MATLAB/Workspace (İş alanı)



The screenshot shows the MATLAB interface with the Workspace and Command Window panes. The Workspace pane on the left displays a table with the following data:

Name	Size
a	1x1

The Command Window on the right shows the following commands and output:

```
>> workspace  
>> a=2  
  
a =  
  
2  
>> |
```

Annotations in the image include:

- A red arrow pointing from the text "Atanan değişken" to the variable 'a' in the Workspace table.
- A red arrow pointing from the text "Workspace penceresi" to the Workspace pane.
- A blue circle around the 'workspace' command in the Command Window, with a blue arrow pointing from the text "Workspace penceresini açmanın 'komut' yolu" to it.

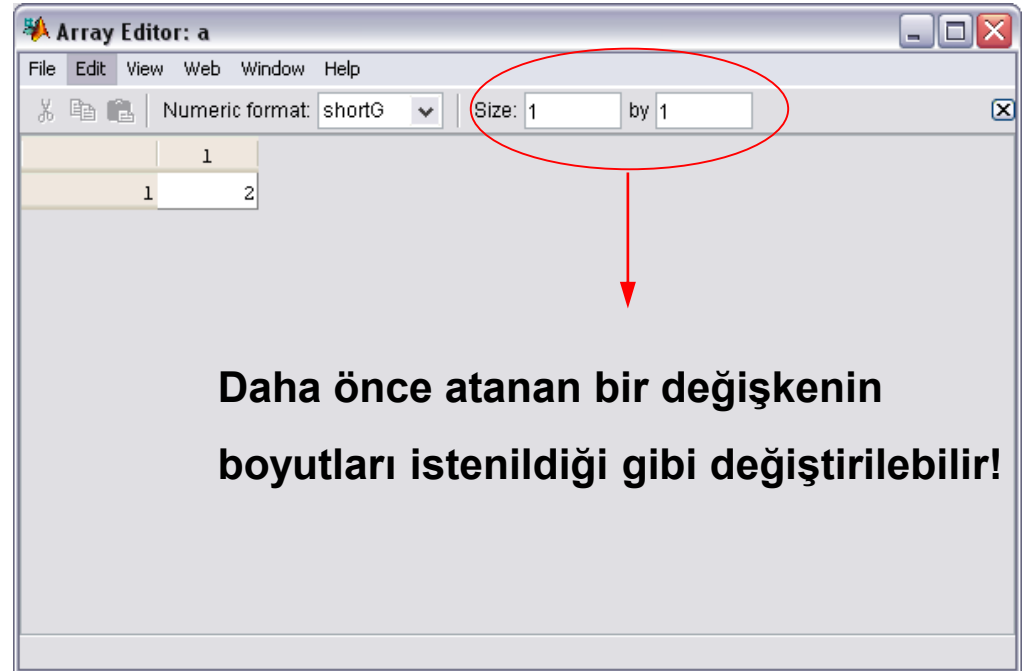
Workspace, ilgili oturumda kullanılan tüm değişkenlerin tutulduğu alandır.

# MATLAB/Array Editor (Dizi editörü)

Matris, vektör ve sayılar için excel özelliğindeki editördür.

## İki farklı biçimde görüntülenir:

- `>>open('a')`
- workspace penceresinde ilgili değişken iki kez tıklanır.





# MATLAB/Temel dosya türleri

- `*.m` MATLAB program dosyaları
- `*.fig` Grafik dosyaları ve GUI'lerin grafik parçaları
- `*.mat` Değişken ve matris dosyaları
- `*.p` pre-parsed pseudo-code dosyaları (bu dosyaların içeriği görüntülenemez ancak program olarak çağrılabilir, yani MATLAB'de çalıştırılabilir!)

# MATLAB/Diziler ve Değişkenler

- MATLAB uygulamalarının temel yapı birimi dizilerdir. Diziler satır veya sütunlar kullanılarak yapılandırılmış birimlerdir.
- Diziler, vektörler ve matrisler olmak üzere iki ana gruba ayırmak mümkündür. *Vektör*; tek boyutlu dizileri tanımlamak için kullanılır. *Matris*; iki veya daha fazla boyutlu dizileri tanımlamak için kullanılır.
- *Değişkenler*; bilgisayar hafızasında belirli bir yer kaplayan, kullanıcının belirlediği isme sahip olan dizilerdir.
- MATLAB’ de değişken tipleri “double” ve “char” dır.
- *double* – 64 bit değerli skaler veya değişkenler için kullanılır. Bu değişkenler reel, sanal veya kompleks değerler olabilir.
  - Örnek:  $deneme = 2 + i$
- *double* aynı zamanda dizileri ASCII kodlarına çevirebilir.
  - Örnek: `double('deneme')`

# MATLAB/Diziler ve Değişkenler

- **char**; tarzındaki değişkenler ise tek bir karakter veya karakter grubundan oluşan 16-bitlik bir skaler veya diziden meydana gelir.
  - **char**; tırnak içi karakterlerini (stringleri) hafızada tutmakla görevlidir.
  - **double**, karakterleri ASCII kodlarına dönüştürürken, **char** tanımlanmış her bir ASCII kodunu, karşılığı karaktere çevirir.
- 
- *d='selam'*
  - *double(d)*
  - *g=[115 108 109]*
  - *char(g)*

# MATLAB/Diziler ve Değişkenleri isimlendirmek

- **Bir değişken oluşturmak için değişkene bir isim verilir,**  
    >> var = 3.14  
    >> string = 'selam'
- **Değişken isimleri**  
    Birinci karakter mutlaka HARF olmalıdır! İlk karakter sayı olamaz!  
    İlk karakterden sonra sayı, harf, \_ ve bunların kombinasyonları  
    Büyük küçük harfe duyarlı: var ile Var birbirinden farklı  
    Değişken isimleri en fazla 63 karakterli olabilir
- **Gömülü değişkenler var. Bunları kullanmak tavsiye edilmez!!**  
    pi değişkeni 3.1515926...  
    ans en son atanan değişkeni gösterir (hesap makinesi gibi)  
    Inf ve -Inf pozitif ve negatif sonsuz sayılarıdır  
    NaN ➤ 'Not a Number'
- **TÜRKÇE karakterler yok!**  
    ç, ğ, ı, ö, ş, ü, Ç, Ğ, Ĭ, Ö, Ş, Ü kullanılmaz

# MATLAB/Temel Komutlar

- `clc` Command window'u temizler.
- `clear` İlgili oturumda atanmış tüm değişkenleri siler.
- `clear a` Yalnızca "a" değişkenini siler.
- `demo` Matlab demosunu çalıştırır.
- `date` Gün-Ay-Yıl'ı görüntüler (Örneğin, 17-Oct-2009)
- `who/whos` Çalışma alanında hangi değişkenlerin olduğunu/bu değişkenlerin yapılarını görüntüler.
- `exit` Matlab oturumundan çıkar.
- `help` Yardım menüsünü açar.
- `help f_na` f\_na fonksiyonu hakkında bilgi verir.
- `save d a` a değişkenini d dosya ismiyle **mat** uzantılı olarak kaydeder.
- `load d` a değişkenini d dosyasından geri çağırır.

**Save ve load komutları, **matris vb. yapıların kaydedilmesi** için çok önemlidir.**

# MATLAB/Matrislerin Girilmesi

- Matris ve vektörler [ ] köşeli parantezleri ile tanımlanır.
- Matris ve vektör girmenin 3 farklı yolu vardır:

Örneğin:

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 8 & 11 \\ 100 & 1 & 4 \end{bmatrix}$$

1.yol

```
A=[1 3 5  
7 8 11  
100 1 4]
```

2.yol

```
A=[1 3 5;7 8 11;100 1 4]
```

3.yol

```
A(1,1)=1,    A(1,2)=3,    A(1,3)=5  
A(2,1)=7,    A(2,2)=8,    A(2,3)=11  
A(3,1)=100,  A(3,2)=1,    A(3,3)=4
```

# MATLAB/Matrislerin Kaydedilmesi

- Matris ve vektörler \*.mat uzantılı olarak **save** komutuyla kaydedilir, **load** ile de istenilen yerden geri çağrılır.
- Örneğin, girilmiş bir **a** matrisini “**D:\yildiz**” klasörüne “**katsayilar.mat**” olarak kaydetmek isteyelim: Bunun için aşağıdaki komut dizisi kullanılır;

```
save D:\yildiz\katsayilar a
```

- katsayilar.mat olarak kaydedilen a matrisinin herhangi bir zamanda geri çağırılması için,

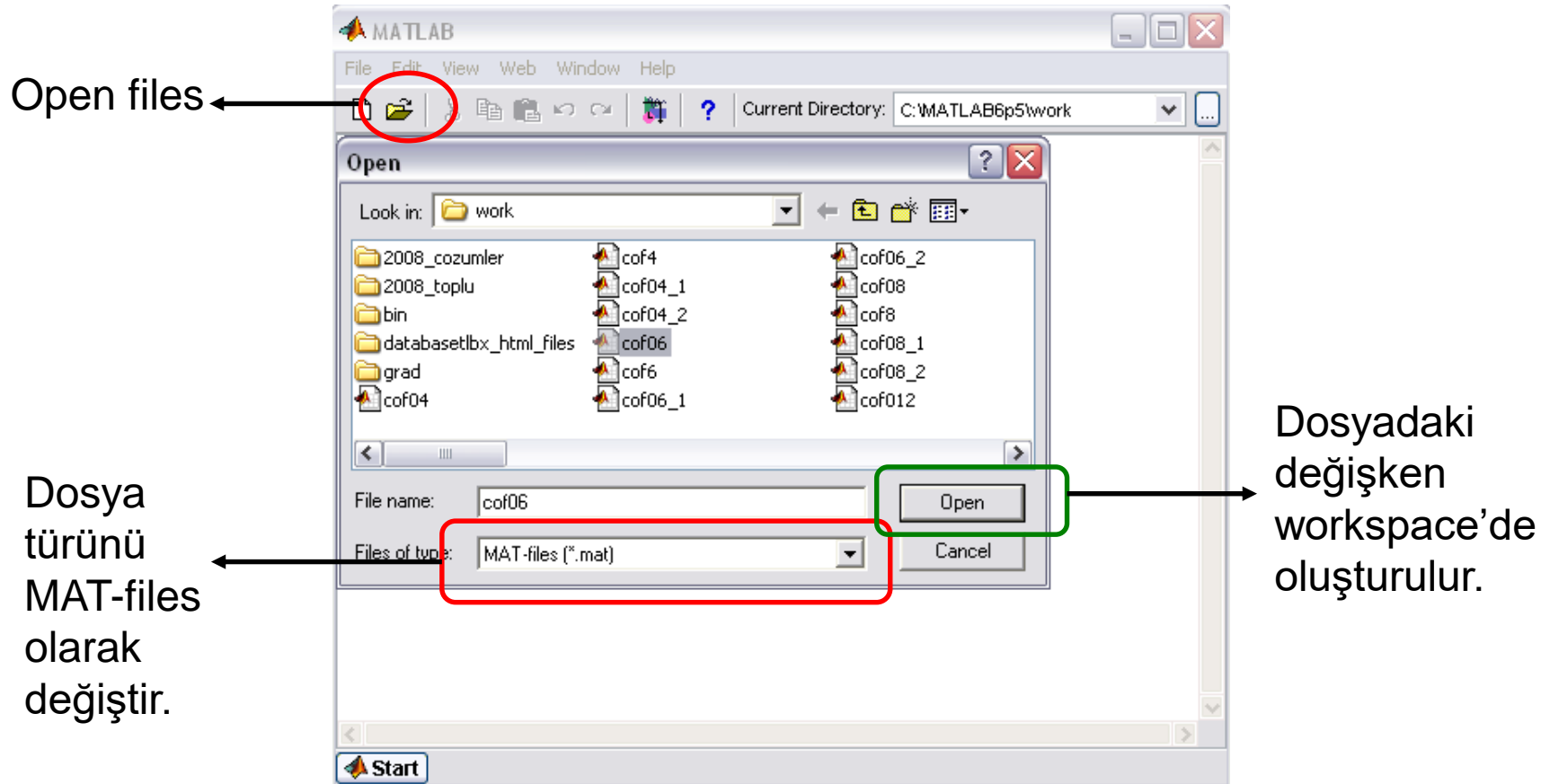
```
load D:\yildiz\katsayilar
```

komut dizisi kullanılır. Geri çağırma işleminden sonra, ilgili matris a dizisi olarak workspace’de kaydedilir (workspace’e kaydetme işleminin geçici olduğunu hatırlayınız!)

Yeni bir matrisi katsayilar.mat olarak kaydettiğimizde, önceki matrisi bir daha görme imkanı kalmaz. Yani save **overwrite (üzerine yazma)** özelliklidir.

# MATLAB/Matrislerin Kaydedilmesi

- \*.mat uzantılı dosyalar, ayrıca MATLAB'den open files kısa yolundan da geri çağrılabilir:





# MATLAB/Temel lineer cebir komutları

- `inv(a)` Bir a kare matrisinin tersini (inversini) alır.
- `a'` a matrisinin devriğini (transpozisini) alır.
- `det(a)` a matrisinin determinantını hesaplar.
- `a+b` Boyutları aynı olan a ve b matrisini toplar.
- `a-b` Boyutları aynı olan a ve b matrislerinin farkını alır.
- `a*b` Sütun sayısı m olan a matrisiyle satır sayısı m olan b matrisini çarpar.
- `a/b` b düzenli kare bir matrisse (determinantı sıfırdan farklıysa), aynı boyutlu a matrisiyle; `a*inv(b)` işlemini yapar.
- `a.*b` Boyutları aynı olan a ve b matrislerinin elemanlarını karşılıklı olarak çarpar.
- `a./b` Boyutları aynı olan a ve b matrislerinin elemanlarını karşılıklı oranlar.

# MATLAB/Temel lineer cebir komutları

- **trace(a)** Bir a matrisinin izini (köşegen elemanlarının toplamını) hesaplar.
- **diag(a)** Bir kare a matrisinin köşegen elemanlarını bir sütun vektöre atar. Ya da a bir vektör ise köşegenleri bu vektörün elemanlarından oluşan bir köşegen matris oluşturur.
- **sum(a)** a matrisinin her bir sütununun toplamını hesaplar. a bir vektör ise sonuç, vektör elemanlarının toplamı olur.
- **triu(a)** Bir matrisin üst üçgen matrisini oluşturur.
- **tril(a)** Bir matrisin alt üçgen matrisini oluşturur.
- **zeros(m,n)**  $m \times n$  boyutlu sıfır matrisi oluşturur.
- **ones(m,n)**  $m \times n$  boyutlu elemanları “1” olan matris oluşturur.
- **eye(m)**  $m \times m$  boyutlu birim matris oluşturur.

# MATLAB/Temel matris operatörleri

- `a(:)` a matrisinin sütunlarının ard arda dizilmesinden oluşan bir sütun vektör oluşturur (vec operatörü)
- `a(:,i)` a matrisinin i. sütununu alır.
- `a(j,:)` a matrisinin j. satırını alır.
- `a(:, [i j])` a matrisinin i ve j. sütununu alır.
- `a([i j], :)` a matrisinin i ve j. satırını alır.
- `e=a:b:n` a, (a+b),...,n sayılarından oluşan bir satır vektör oluşturur.
- `e=linspace(a,n,b)` a ile başlayan, n' de son bulan eleman sayısı b olan bir satır vektörü oluşturur.
- `e=logspace(a,n,b)`  $10^a$  ile başlayan,  $10^n$ ' de son bulan eleman sayısı b olan bir satır vektörü oluşturur.

Örneğin,

`e=1:1:n`, 1 ile n arasındaki tam sayılardan oluşan bir vektör.

`e=2:2:n`, 1 ile n arasındaki çift sayılardan oluşan bir vektör.

`e=1:2:n`, 1 ile n arasındaki tek sayılardan oluşan bir vektör.

`e=-10:0.1:n`, -10'dan 0.1 artımla n'ye kadar olan sayılardan oluşan bir vektör.

`e=linspace(0,10,6)`, `e=[0 2 4 6 8 10]`

`e=logspace(0,2,3)`, `e=[1 10 100]`

# MATLAB/Temel matris operatörleri

- **length(a)** a matrisinin sütun sayısını verir. a bir vektör ise sonuç, a vektörünün eleman sayısıdır.
- **[m,n]=size(a)** a matrisinin satır sayısını (m) ve sütun sayısını (n) verir.
- **max(a)** Bir a vektörünün en büyük elemanını gösterir.
- **min(a)** Bir a vektörünün en küçük elemanını gösterir.
- **[m,i]=max(a)** Bir a sütun vektörünün en büyük elemanını (m) ve bunun satır numarasını verir.
- **[m,i]=min(a)** Bir a sütun vektörünün en küçük elemanını (m) ve bunun satır numarasını verir.
- **sort(a)** Bir a vektörünün elemanlarını küçükten büyüğe sıralar.
- **a(:,i)=[]** A'nın i. sütununu siler.
- **a(i,:)=[]** A'nın i. satırını siler.

# MATLAB/Temel matris operatörleri

- `sortrows(a,i)` Bir a matrisinin elemanlarını i.sütuna göre sıralar.

Örnek:

```
a =
```

1	1000
3	10
2	5
4	1



```
>> sortrows(a,1)
```

```
ans =
```

1	1000
2	5
3	10
4	1

```
a =
```

1	1000
3	10
2	5
4	1



```
>> sortrows(a,2)
```

```
ans =
```

4	1
2	5
3	10
1	1000

# MATLAB/Uygulama-1

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 8 & 11 \\ 100 & 1 & 4 \end{bmatrix}$$

Aşağıdaki işlemleri command window'da yapınız.

- 1) A matrisini giriniz.
- 2) A matrisinin determinantını hesaplayınız.
- 3) A matrisinin tersini bulunuz. Çıkan sonucu bir B matrisine atayınız.
- 4) A\*B işlemini yapınız. Elde edilen sonucu irdeleyiniz.
- 5) A matrisinin 1. sütununu a1, 3. sütununu a3 vektörlerine atayınız.
- 6) Köşegenleri A matrisinin köşegenlerinden oluşan bir C köşegen matrisi oluşturunuz.
- 7) a1'in devriği ile a3 vektörünü çarpınız.
- 8) a1 ile a3 vektör elemanlarını karşılıklı çarpınız.
- 9) A'nın 3. satırını, diğer satır elemanlarını girmeden, [5 6 7] olarak değiştiriniz.
- 10) A'nın 1 ve 2. satırlarını siliniz.

# MATLAB/Uygulama-1:Çözüm

```
① >> A=[1 3 5;7 8 11;100 1 4];
② >> det(A)
ans =
    -728
③ >> B=inv(A)
B =

    -0.0288    0.0096    0.0096
   -1.4725    0.6813   -0.0330
    1.0893   -0.4107    0.0179
>> A*B
ans =
```

Birim matris

```
    1.0000         0    0.0000
         0    1.0000    0.0000
         0    0.0000    1.0000
```

```
>>a1=A(:,1);a3=A(:,3);
>>C=diag(diag(A));
⑥ >>a1'*a3
⑦ ans=
    482
```

```
⑧ >> a1.*a3
ans=
     5
    77
   400
⑨ >>A(3,:)= [5 6 7]
A =

     1     3     5
     7     8    11
     5     6     7
⑩ >> A([1 2],:)=[]
A =

     5     6     7
>>
```

# MATLAB/Uygulama-2

$$\mathbf{B} = \begin{bmatrix} 10 & 5 & 5 \\ 70 & 8 & 7 \\ 10 & 1 & 3 \end{bmatrix}$$

Aşağıdaki işlemleri command window'da yapınız.

- 1) B matrisini giriniz.
- 2) B matrisini mevcut çalışma klasörünüze **katsayılar** ismiyle kaydediniz.
- 3) Dosyanın kaydedilip kaydedilmediğini kontrol ediniz. (Open Files penceresinden)
- 4) MATLAB oturumundaki tüm değişkenleri siliniz (clear)
- 5) Command window'da yazılmış tüm ifadeleri temizleyiniz. (clc)
- 6)  $\mathbf{B} \times 2$  işlemini yapınız.
- 7) B matrisini geri çağırınız.
- 8) B matrisinin üst ve alt üçgen matrislerini oluşturunuz.
- 9)  $\mathbf{C} = [\mathbf{B} \text{ zeros}(3,2)]$  işlemini yapınız.



# MATLAB/Uygulama-2:Çözüm

① >> B=[10 5 5;70 8 7;10 1 3];

② >> save katsayilar B

④ >> clear

⑤ >> clc

⑥ >> B\*2  
??? Undefined function or variable 'B'. **Neden?**

⑦ >> load katsayilar

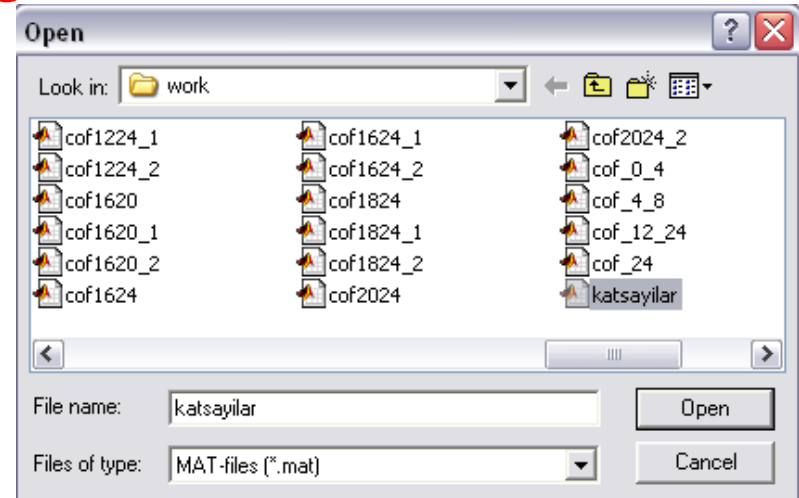
⑧ >> triu(B)

```
ans =  
    10     5     5  
     0     8     7  
     0     0     3
```

>>tril(B)

```
ans =  
    10     0     0  
    70     8     0  
    10     1     3
```

③



⑨

>> C=[B zeros(3,2)]

C =

10	5	5	0	0
70	8	7	0	0
10	1	3	0	0

# MATLAB/Değişkene Değer Atamak

- `input` fonksiyonu klavyeden giriş yapmayı sağlayan bir fonksiyondur.
- `degisken=input('bir deger giriniz: ')` şeklinde kullanılır.

```
>> a=input('bir deęer giriniz: ')
bir deęer giriniz: 5

a =

    5
```

- Eğer bir karakter değişken olarak atanmak istiyorsa;
- `degisken=input('bir deęer giriniz:', 's' )` şeklinde olmalıdır.

# MATLAB/Sonuçların Görüntülenmesi

- Matlab programında bir değişiklik yapılmazsa, ondalıklı sayıları virgülden sonra 4 basamak olarak gösterir.
- **format short** → Virgülden sonra 4 basamaklı gösterim
- **format long** → Virgülden sonra 14 basamaklı gösterim
- **format short e** → 5 basamaklı üstel gösterim
- **format long e** → Virgülden sonra 15 basamak ve üstel gösterim
- **format short g** → Yuvarlatılmış en uygun gösterim (toplam 5 basamak)
- **format long g** → Yuvarlatılmış en uygun gösterim (toplam 15 basamak)
- **format rat** → İki sayının oranı şeklinde gösterim

# MATLAB/Sonuçların Görüntülenmesi

```
>> format short  
>> a=10.123000123123123123;  
>> a
```

a =

10.1230

```
>> format long  
>> a=10.123000123123123123;  
>> a
```

a =

10.123000123123123

```
>> format short e  
>> a=10.123000123123123123;  
>> a
```

a =

1.0123e+001

```
>> format long e  
>> a=10.123000123123123123;  
>> a
```

a =

1.012300012312312e+001

```
>> format short g  
>> a=10.123000123123123123;  
>> a
```

a =

10.123

```
>> format long g  
>> a=10.123000123123123123;  
>> a
```

a =

10.1230001231231

# MATLAB/Sonuçların Görüntülenmesi

```
>> format rat
>> a=10.123000123123123123;
>> a

a =

    1893/187
```

- Diğer bir görüntüleme fonksiyonu “**disp**” fonksiyonudur. disp(x) şeklindedir. Bir dizi veya metni görüntüler. Dizinin değişken adı ekrana yazılmaz. X bir karakter ise metin olarak görüntülenir.

```
disp(' ')
disp('  A-Deg  B-Deg  C-Deg')
disp('  =====  =====  =====')
disp(rand(4,3))
```

**Ekranda görülen;**

A-Deg	B-Deg	C-Deg
=====	=====	=====
0.1389	0.2722	0.4451
0.2028	0.1988	0.9318
0.1987	0.0153	0.4660
0.6038	0.7468	0.4186

# MATLAB/Sonuçların Görüntülenmesi

- Hesaplanan sonuçları görüntülemenin diğer bir yolu **fprintf** fonksiyonudur.
- **fprintf(format, değer)** şeklinde kullanılır.
- **format** – sunulacak bilginin şeklini, bu bilgiye ilişkin açıklamaları ve bu açıklamaların nasıl bir şekilde sunulacağını belirleyen ifadelerden oluşur.
- **değer** – sunulacak bilgiyi içerir.

```
>> x=123.2;  
>> fprintf('Islem sonucu %f bulunmustur. \n', x)  
Islem sonucu 123.200000 bulunmustur.
```

```
>> x=123.2;  
>> fprintf('Islem sonucu = %f \n', x)  
Islem sonucu = 123.200000
```

- **'%f'** – gösterilecek değişken gösterimine ait açıklamanın hangi şekilde gösterileceğini ifade eder.
- **'\n'** – komut istem ifadesinin (>>) kaç satır boşluktan sonra yazılacağını gösterir.

# MATLAB/Sonuçların Görüntülenmesi

```
>> x=123.2;  
>> fprintf('Islem sonucu = %5.1f \n', x)  
Islem sonucu = 123.2
```

- ‘%’ ile ‘f’ işaretleri arasına ondalık sayılar yazarak, gösterilecek değerin biçimini değiştirmek mümkündür.

**fprintf('Islem sonucu = %5.1f \n', x)**

Alan genişliği

Ondalık kısım

```
>> x=123.2;  
>> fprintf('Islem sonucu = %15.1f \n', x)  
Islem sonucu =      123.2
```

```
>> x=123.2;  
>> fprintf('Islem sonucu = %25.1f \n', x)  
Islem sonucu =                123.2
```

# MATLAB/Sonuçların Görüntülenmesi

## fprintf için biçimler

<i>Belirteç</i>	<i>Tanımı</i>
<code>%e</code>	Küçük harf ile üstel gösterim
<code>%E</code>	Büyük harf ile üstel gösterim
<code>%f</code>	Kayan noktalı gösterim, aksi belirtilmedikçe virgülden sonra 6 basamak gösterir.
<code>%s</code>	Karakter dizilerini gösterir.
<code>%d</code>	Virgüllü sayıları daima 10'un kuvvetiyle gösterir.



# MATLAB/Sonuçların Görüntülenmesi

Bir sonuç `fprintf` fonksiyonu ile yazıldıktan sonra bir diğeri yazılacaksa bunun komut satırının neresine yazılacağı da bu fonksiyonda belirtilebilir. Bunun için kullanılan karakterler:

<i>Karakter</i>	<i>Tanımı</i>
<code>\b</code>	backspace
<code>\n</code>	Yeni satıra gönderir
<code>\r</code>	Satır başına döndürür
<code>\t</code>	Yatay Tab

# MATLAB/Sonuçların Görüntülenmesi

```
>> a=100.25;  
>> b=255.23;  
>> fprintf('a değeri = %5.2f\r ve b değeri = %5.3f \n', a,b)  
a değeri = 100.25  
ve b değeri = 255.230
```

```
>> a=100.25;  
>> b=255.23;  
>> fprintf('a değeri = %5.2f\b ve b değeri = %5.3f \n', a,b)  
a değeri = 100.2 ve b değeri = 255.230
```

```
>> a=100.25;  
>> b=255.23;  
>> fprintf('a değeri = %5.2ft ve b değeri = %5.3f \n', a,b)  
a değeri = 100.25        ve b değeri = 255.230
```

Matris elemanlarının istenilen hanede gösterilmesi için, **printmatrix** fonksiyonu oldukça kullanışlıdır (File-exchange sayfasından download edilebilir!)

- Görüntülenecek açıklamalarda tek tırnak('), yüzde (%) ve ters bölme (\) işaretleri kullanılmak isteniyorsa, *arada boşluk bırakmadan ikişer adet* kullanılmalıdır.

# MATLAB/Sonuçların Görüntülenmesi

- Hesaplanan sonuçları string olarak bir değişkene atamak için `sprintf` komutu kullanılır.
- `değişken=sprintf(format, değer)` şeklinde kullanılır.

```
>> a=100.25;  
>> sonuc=sprintf('a degeri = %5.2f',a)  
  
sonuc =  
  
a degeri = 100.25
```

```
>> a=100.25;  
>> sonuc=fprintf('a degeri = %5.2f',a)  
a degeri = 100.25  
sonuc =
```

17

Gösterilen ifadenin kaç karakterden oluştuğunu ifade eder

# MATLAB/Sonuçların Görüntülenmesi

- Değişkendeki karakter dizisini sayısal bir değere dönüşümü için `sscanf` komutu kullanılır.
- `değişken=sscanf(karakter dizisi, biçim)` şeklinde kullanılır.

```
>> a='1.728';  
>> b=sscanf(a,'%f')
```

```
b =
```

```
1.7280
```

# MATLAB/Diziler (Arrays)

- **Sayılar** : Sayı dizileri (numeric array)
- **Karakterler** : Karakter dizileri (character array)

Örnek:

```
c=1999 (numeric array)
d='Yildiz Teknik Universitesi' (character array)
f=[1999 2000] (numeric, matrix)
g=[d 'Insaat Fakultesi'] (character, matrix)
```

**Not: Numeric ile character dizileri bir matriste bir arada bulunamaz! Yani, bir matris hem sayı hem de bir kelimeyi aynı anda içeremez!**

- **Hücreler** : Hücre dizileri (cell array)
- **Yapılar** : Yapı dizileri (structure array)

# MATLAB/Diziler (Karakter Dizileri)

- Karakterler dizisi veya yalnızca karakter/sözcük (string), **iki tek tırnak arasındaki ifade edilen**, gerçekte ASCII kod tablosunda sayısal kodlarla belirtilen ilk 127 karakterden oluşan karakter dizileridir (char array). Karakter uzunluğu, dizideki karakter sayısıdır. Her bir karakter bellekte 1 byte (8 bit) yer kaplar.

```
>> ders='bilgisayar bilimleri'
```

```
ders =
```

```
bilgisayar bilimleri
```

- Burada verilen ders değişkeninde her bir harf ASCII kod tablosundaki sayısal bir koda sahiptir.

```
>> kod=double(ders)
```

```
kod =
```

```
98 105 108 103 105 115 97 121 97 114 32 98 105 108 105 109 108 101 114 105
```

# MATLAB/Diziler (Karakter Dizileri)

- Tam tersi olarak ASCII karşılığı verilen bir dizinin karakter karşılığını bulmak için `char` fonksiyonu kullanılır.

```
>> char(kod)

ans =

bilgisayar bilimleri
```

- Çok Boyutlu Dizi Gösterimleri

```
>> x='1001';
>> a=[ders,' ders kodu ',x]

a =

bilgisayar bilimleri ders kodu 1001
```

# MATLAB/Diziler (Karakter Dizileri)

- Karakter dizilerinin karşılaştırılması
  - `strcmp`: İki dizinin aynı olup olmadığını belirler.
  - `strcmpi`: iki dizinin harf durumu (büyük harf-küçük harf) dikkate alınmadan aynı olup olmadığını belirler.
  - `strncmp`: iki dizinin ilk n karakterinin aynı olup olmadığını belirler.
  - `strncmpi`: ilk dizinin harf durumu (büyük harf-küçük harf) dikkate alınmadan ilk n karakterinin aynı olup olmadığını belirler.



# MATLAB/Diziler (Karakter Dizileri)

```
>> a='matematik';  
>> b='mathematik';  
>> c='MaTematiK';  
>> e='matematik';  
>> x=strcmp(a,b)
```

x =

0

```
>> x=strcmp(a,e)
```

x =

1

```
>> x=strcmp(a,c)
```

x =

0

```
>> a='matematik';  
>> b='mathematik';  
>> c='MaTematiK';  
>> e='matematik';  
>> x=strcmpi(a,b)
```

x =

0

```
>> x=strcmpi(a,c)
```

x =

1

```
>> a='matematik';  
>> b='mathematik';  
>> c='MaTematiK';  
>> e='matematik';  
>> x=strncmp(a,b,4)
```

x =

0

```
>> x=strncmp(a,b,2)
```

x =

1

```
>> x=strncmp(a,c,1)
```

x =

0

```
>> a='matematik';  
>> b='mathematik';  
>> c='MaTematiK';  
>> e='matematik';  
>> x=strncmpi(a,c,1)
```

x =

1

# MATLAB/Diziler (Karakter Dizileri)

- **Dizilerin tek tek karşılaştırılması:** Karakter dizileri aynı boyutlu veya bir elemandan oluşması koşuluyla karakter eşitlik karşılaştırılması için ilişkisel operatörler (>, >=, <, <=, ==, ~=) kullanılabilir.

```
>> x='matlab';  
>> y='matema';  
>> x==y  
ans =  
     1     1     1     0     0     0
```

- **Büyük-Küçük Harf Dönüşümleri:**
  - **upper:** Dizideki tüm harfleri büyük harfe dönüştürür.
  - **lower:** Dizideki tüm harfleri küçük harfe dönüştürür.

```
>> upper('matLab')  
  
ans =  
  
MATLAB
```

```
>> lower('MATIAB')  
  
ans =  
  
matlab
```

# MATLAB/Diziler (Karakter Dizileri)

- **isletter** fonksiyonu karakter dizilerini oluşturan elemanların bir harf mi yoksa bir rakam mı veya bir boşluk mu olduğunu sorgular.
- **isspace** fonksiyonu karakter dizilerini oluşturan elemanların boşluk karakteri olup olmadığını denetler.
- **ischar** fonksiyonu dizinin bir karakter dizisi olup olmadığını sorgular.

```
>> ders='TBB GN 16';ders1=20;  
>> isletter(ders)  
ans =  
  
    1    1    1    0    1    1    0    0    0  
  
>> isspace(ders)  
ans =  
  
    0    0    0    1    0    0    1    0    0  
  
>> ischar(ders)  
ans =  
  
    1  
>> ischar(ders1)  
ans =  
  
    0
```

# MATLAB/Diziler (Hücre Dizileri)

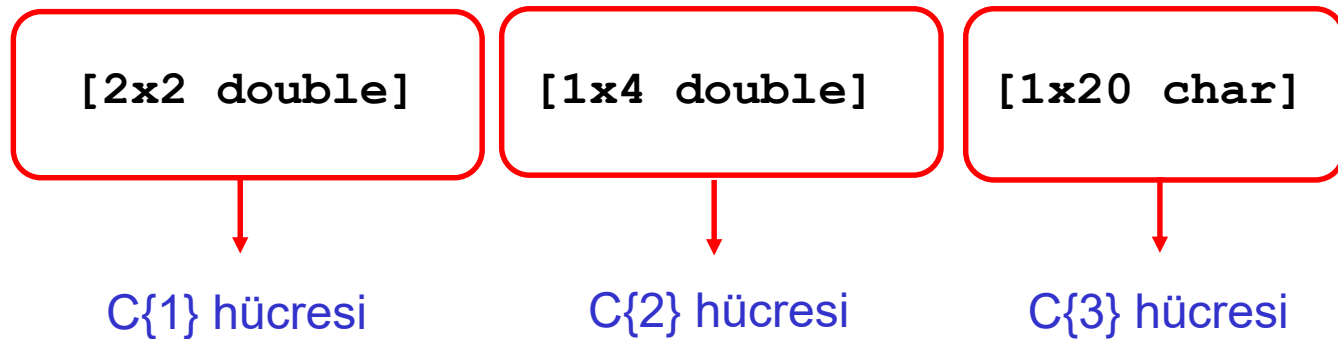
- **Hücre Dizileri (Cell arrays) { } ile tanımlanır.** Böylesi bir dizi, farklı matrisleri aynı isim altında tutmak ve işlemek için kullanılmaktadır.

Örneğin,

```
C{1}=[1 2;3 5],C{2}=[4 4 4 4];C{3}=[('yildiz teknik'),(' inaat')];
```

girildiğinde, C bir hücre dizisi olur. Bu hücre geri çağırıldığında,

C =



Her bir hücre ayrı ayrı işlenir.

# MATLAB/Diziler (Hücre Dizileri)

- **`C=cell(n)`**  $n \times n$  hücreden oluşan boş bir hücreyi **C**'ye atar. Örneğin  $n=2$  için

```
>> C=cell(2)
```

```
C =
```

```
    []    []  
    []    []
```

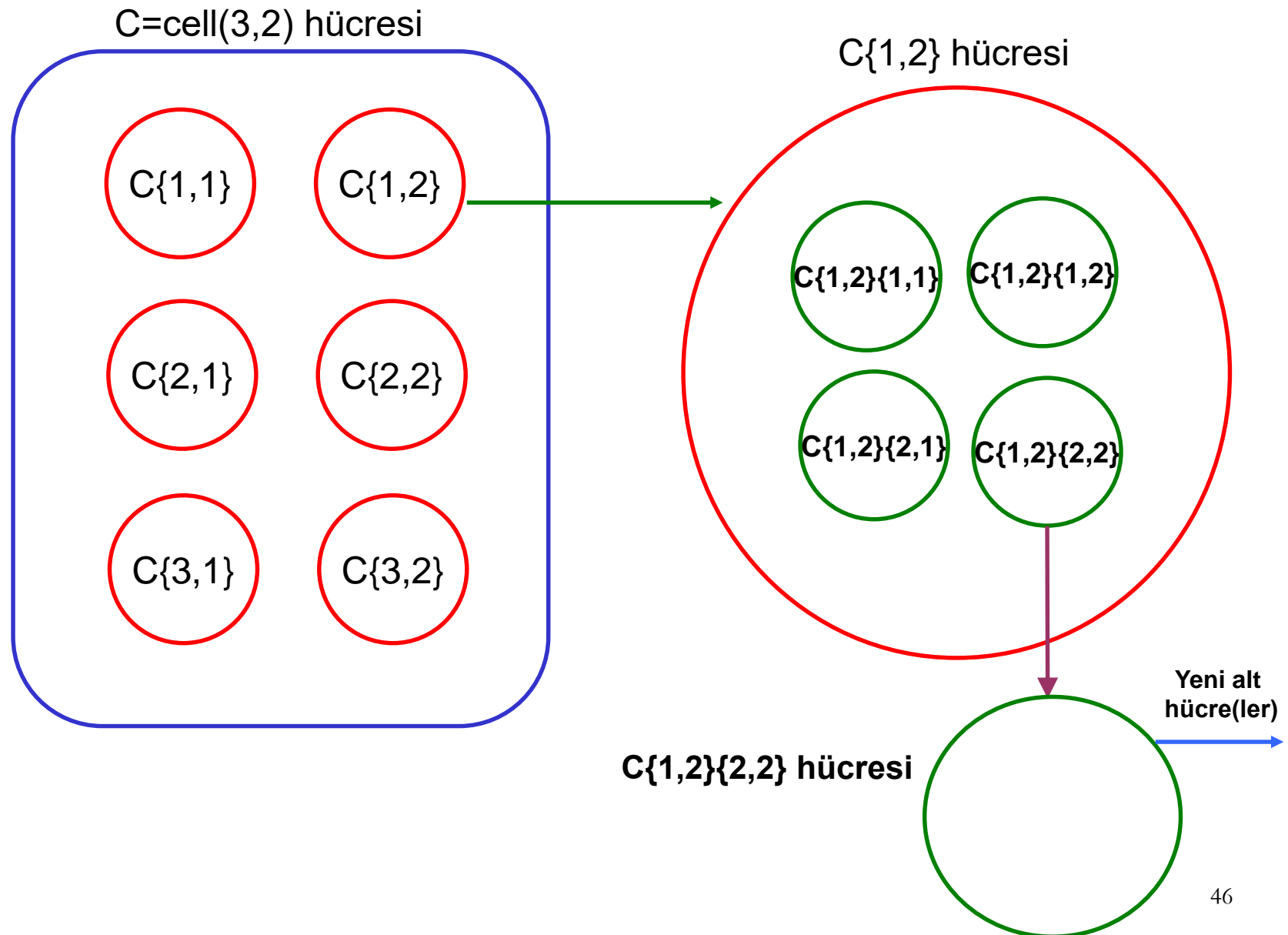
**hücresi oluşturulur.**

- **Bir hücrenin içine istenilen sayıda yeni hücreler eklemek mümkündür; Örneğin, `C{1}{1}=[2 3]` ile **C** aşağıdaki biçimde değişir;**

```
C =
```

```
 {1x1 cell}    []  
          []    []
```

# MATLAB/Diziler (Hücre Dizileri)



# MATLAB/Diziler (Yapı Dizileri)

- Yapı dizileri (Structure arrays), veri tabanları için oldukça kullanışlı bir dizi türüdür.

A yapı dizisi çağrıldığında,

```
A.name='Bahattin';  
A.sname='Erdogan';  
A.univ='YTU';  
A.city='Istanbul';  
A.email='berdogan@yildiz.edu.tr';  
A.year=2012;
```

ile A, bir structure array olur.

```
>>A  
  
A =  
      name: 'Bahattin'  
     sname: 'Erdogan'  
      univ: 'YTU'  
      city: 'Istanbul'  
   email: 'berdogan@yildiz.edu.tr'  
     year: 2012
```

Hücre ve yapı dizileri, mat uzantılı dosyalar olarak, daha önce açıklanan save komutuyla kaydedilip, load komutuyla geri çağrılabilir.

# MATLAB/Diziler arasında dönüşümler

- **num2str(a)** Bir a sayısını bir karaktere atama (From **numeric** to (2) **string**)

```
>> a=10;  
>> sonuc=num2str(a)  
sonuc =  
10  
  
>> ischar(sonuc)  
ans =  
  
1
```

- **str2num(a)** Karakter olan bir a sayısını sayı değerine atama

```
>> deger=str2num(sonuc)  
deger =  
10  
  
>> isnumeric(deger)  
  
ans =  
  
1
```



# MATLAB/Diziler arasında dönüşümler

- **mat2str(a)** Bir a matrisini bir karakter dizisine atama

```
>> sonuc=mat2str(rand(2))
sonuc =

[0.162611735194631 0.498364051982143;0.118997681558377 0.959743958516081]
>> ischar(sonuc)
ans =
    1

>> isnumeric(sonuc)
ans =

    0
```

- **int2str(a)** Bir a tam sayısını bir karaktere atama

```
>> a=100.254;
>> sonuc=int2str(a)

sonuc =

100
```

# MATLAB/Diziler arasında dönüşümler

- **char (a)** Bir a hücrelerini bir karakter dizisine atama

```
>> sonuc{1,1}='3';  
>> sonuc{1,2}='4';  
>> sonuc{2,1}='5';  
>> sonuc{2,2}=['7' '5';'7' '8'];  
>> sonuc
```

sonuc =

```
'3' '4'  
'5' [2x2 char]
```

```
>> arama=char(sonuc)
```

arama =

```
3  
5  
4  
75  
78
```

# MATLAB/Diziler arasında dönüşümler

- **cellstr(a)** Bir a karakterini bir hücre dizisine atama

```
>> deneme=cellstr(arama)
```

```
deneme =
```

```
'3'
```

```
'5'
```

```
'4'
```

```
'75'
```

```
'78'
```

- **num2cell(a)** Bir a sayısını bir hücre dizisine atama

```
>> a=2;
```

```
>> deneme=num2cell(a)
```

```
deneme =
```

```
[2]
```

# MATLAB/Diziler arasında dönüşümler

- **Örnek:** Bir işlem sonucunda  $a=10.234$  elde edilsin. “Elde edilen sonuc=10.234” karakterini görüntülemek için,

```
[ 'Elde edilen sonuc=' num2str(a) ]
```

Her iki ifade karakter olmalıdır!



yapısı düşünülebilir.

- Bunun daha gelişmiş biçimi, fprintf ile sağlanır:
- `fprintf('Elde edilen sonuc= %6.3f \n',a)`

# MATLAB/Trigonometrik Fonksiyonlar

- **sin(x)**            **x açısının sinüsünü verir.**
- **asin(x)**            **sinüsü x olan açıyı verir.**
- **cos(x)**            **x açısının kosinüsünü verir.**
- **acos(x)**            **kosinüsü x olan açıyı verir.**
- **tan(x)**            **x açısının tanjantını verir.**
- **atan(x)**            **tanjantı x olan açıyı verir.**
- **cot(x)**            **x açısının kotanjantını verir.**
- **acot(x)**            **kotanjantı x olan açıyı verir.**
- **sec(x)**            **x açısının sekantını verir.**
- **asec(x)**            **sekantı x olan açıyı verir.**
- **csc(x)**            **x açısının kosekantını verir.**
- **acsc(x)**            **kosekantı x olan açıyı verir.**

**MATLAB' da trigonometrik fonksiyonlar için tanımlı açı değerleri radyan cinsinden olmalıdır.**

# MATLAB/Uygulama-3

## Aşağıdaki işlemleri command window'da yapınız.

1. fprintf fonksiyonunu kullanarak, a=10.45623 sayısını 3 haneye kadar yazdırınız.
2. ['sayinin degeri=' a] ifadesini, a virgülden sonra 2 hane olacak biçimde yazdırınız.
3. Yukarıdaki ifadeyi bir b değişkenine atayınız (sprintf ile)
4. b'nin bir karakter dizisi olup olmadığını denetleyiniz.
5. a değerini önünde 5 karakter boşluk kalacak biçimde 2 haneye kadar yazdırınız.
6. a değişkenini msgbox(a,'sonuc') ifadesiyle bir GUI'ye yazdırınız.
7. b değişkenini msgbox(b,'sonuc') ifadesiyle bir GUI'ye yazdırınız.
8. a'nın karakökünü c değerine atayınız. b ve ['sayinin karakoku', c] ifadesi alt alta olacak biçimde (c, virgülden sonra 5 hane gösterilecek) msgbox içinde yazdırınız.

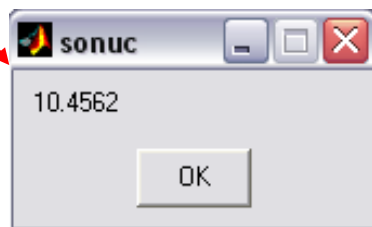
**msgbox(mesaj,başlık)**

mesaj karakter dizisi, karakter matrisi veya hücre dizisi şeklinde olmalıdır.

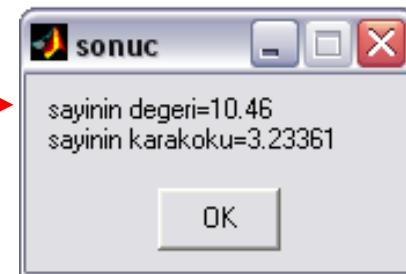
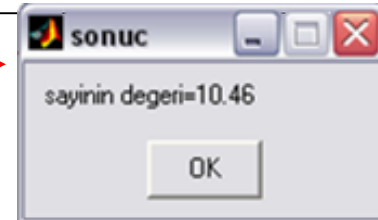
# MATLAB/Uygulama-3:Çözüm

```
1 >> a=10.45623;fprintf('%1.3f',a)
10.456
>>
2 >>fprintf('sayinin
degeri=%1.2f',a)
sayinin degeri=10.46
3 >>
>>b=sprintf('sayinin degeri=
%1.2f',a)
b =
4 sayinin degeri=10.46

>>ischar(b)
ans=
1
5 >>fprintf('%10.2f',a)
10.46
6 >>msgbox(num2str(a),'sonuc')
```



```
7 >> msgbox(b,'sonuc')
>>
8 >>c=sqrt(a)
c =
3.2336
>>b1=sprintf('sayinin
karakoku=%1.5f',c);
>>g=char(b,b1);
>>msgbox(g,'sonuc')
```



**Not: b ve b1 karakter dizilerini alt alta yazdırmanın bir diğer yolu, bunları bir hücre dizisi altında düşündür;**

```
G=cell(2,1);G{1}=b;G{2}=b1;
msgbox(G,'sonuc')
```

**benzer sonucu üretir.**

# MATLAB/Uygulama-4

Aşağıdaki işlemleri command window'da yapınız.

1. Sonraki işlemlerde kullanılacak bir a sayı değerini, inputdlg fonksiyonu ile girdiren komutu yazınız.
2. a değerinin bir sayı olup olmadığını irdelleyiniz.
3.  $a*2$  işlemini yapınız. Bu işlemin neden sonuç vermediğini irdelleyiniz.
4. a değerini, gerekli ise, sayı dizisine dönüştürünüz.

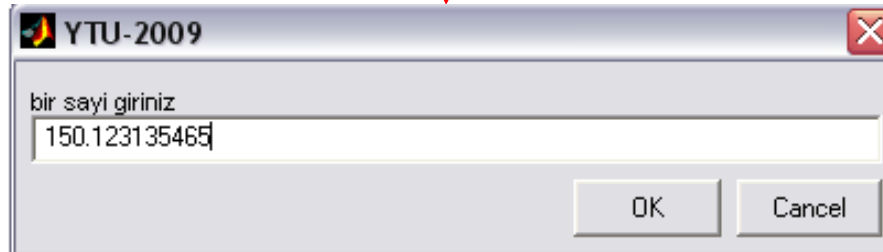
`inputdlg('ifade','başlık')`



# MATLAB/Uygulama-4:Çözüm

①

```
>> a=inputdlg('Bir sayi giriniz','YTU-2012')
```



```
a =
```

```
'150.123135465'
```

②

```
>> isnumeric(a)
```

```
ans =
```

```
0
```

inputdlg ile karakter hücre dizisi oluşturulur.  
Bu nedenle, girilen verinin sayı yapılması  
gerekir.

③

```
>> a*2
```

```
>> ?? Error using ==> *
```

```
Function '*' is not defined for values of class 'cell'.
```

④

```
>> a=str2num(char(a))
```

```
>>
```

```
a =
```

```
150.1231
```

# MATLAB/Programı Dallandıran İfadeler

- **Dal yapıları**, program kodlarından istenilenleri seçen ve onları işleten, istenilen kodları ise değerlendirme dışı bırakabilen MATLAB ifadeleridir.
- `if`
- `Switch, case`
- `try/catch`

yapıları ile oluşturulur. Bu bölümde try/catch yapısına değinilmeyecektir.

# MATLAB/if,end yapısı

- **if** (eğer) yapısı bir koşulun gerçekleşmesi durumunda bir işlemi yaptırmak için sıklıkla kullanılır. Bu ifade,

```
if koşul
    işlem
end
```

```
if koşul
    işlem
else
    işlem
end
```

```
if koşul
    işlem
elseif
    işlem
else
    işlem
end
```

biçimlerindedir.

**Örnek:** Girilen bir sayının negatif olması durumunda, sayıyı doğal logaritmasıyla değiştiren bir kod düşünelim:

```
a=input(' bir sayı giriniz= ');
if a<0
    a=log(a);
else
    a=a;
end
a
```

**“Diğer durumda”**  
anlamındadır:  
Burada,  $a > 0$   
koşulunu temsil eder.

**Else yapısı kullanılmasaydı**

```
a=input(' bir sayı giriniz= ');
if a<0
    a=log(a);
end
if a>0
    a=a;
end
a
```

# MATLAB/switch,case yapısı

- **switch** (değiştir) if yapısına benzer. Burada daha çok sözel olarak belirtilen durumlara göre yönlendirme işlemi yapılır. Bu yapının kullanımı **case** ile aşağıdaki gibidir;

```
switch durum
    case durum1
        işlem1
    case durum2
        işlem2
    otherwise
        işlem3
end
```

→ Kullanımı kişiye bağlıdır.

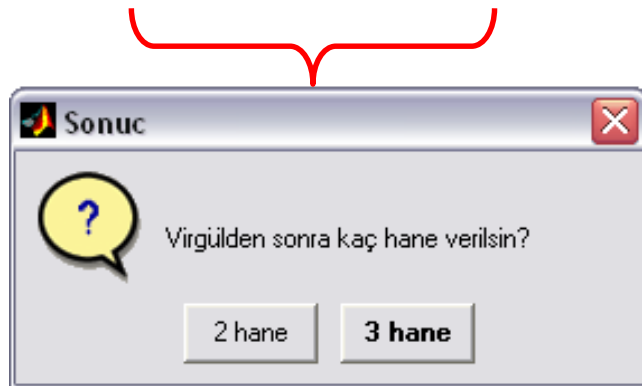
**Örnek:** gun degiskeninin, is gunu olup olmadığına karar vermek için aşağıdaki kodlar düşünülür;

```
clear,clc
gun=input('hangi gun=', 's');
switch lower(gun)
    case {'pazartesi', 'sali', 'carsamba', 'persembe', 'cuma'}
        disp('iş günü')
    case {'cumartesi', 'pazar'}
        disp('TATİL!')
end
```

# MATLAB/switch,case yapısı

- Kullanıcı tarafından girilen bir  $a=10.2424542$  değişkeninin virgülden sonra 2' mi 3 hane mi yazdırılacağını sorgulayan bir questdlg GUI'sini düşünelim:

```
a=10.2424542;  
buton=questdlg('Virgülden sonra kaç hane verilsin?','Sonuc','2 hane','3 hane','3 hane');  
switch buton  
    case {'2 hane'}  
        fprintf('%1.2f',a)  
    case {'3 hane'}  
        fprintf('%1.3f',a), end
```



“2 hane” düğmesinin tıklanması durumunda,

10.24

sonucu görüntülenir.

`butonadı = questdlg(soru, başlık, btn1, btn2, seçenek);`

# MATLAB/for,end döngüsü

- **for,end** döngüsü bir işlemin birden daha fazla sayıda yaptırılmasında kullanılır. (Örneğin, kök bulma problemlerinde kullanılan iterasyon çözümleri). Kullanımı,

```
for i=1:n      (i→Tam sayı (integer))
    işlem
end
```

biçimindedir.

**Örnek:** 1'den N'ye kadar olan sayıların toplamını yapan bir program düşünelim.

```
clear,clc
N=input('bir sayı giriniz=');
say=0; %sayaç
for i=1:N
    say=say+i; %birikimli (kümülatif toplam)
end
say
```

# MATLAB/while,end döngüsü

- while,end döngüsü, belirli bir durumun gerçekleşmesi durumunda bir işlemin birden daha fazla sayıda yaptırılmasında kullanılır.

```
done=0;  
while done==0  
    işlem  
end
```

1. Buradaki, while,end döngüsü, done değişkeni ancak ve ancak 0 olduğu zaman çalışacaktır.
2. Bir önceki satırda, done değişkeni 0 olarak atanmış olduğu için while,end döngüsü çalışır.


**Örnek:** 1'den N'ye kadar olan sayıların toplamını while,end döngüsü ile yapan bir program düşünelim.

```
clear,clc  
N=input('bir sayı giriniz=');  
say=0; i=0;done=0;  
while done==0  
    i=i+1; %bir önceki örnekte for,end döngüsündeki "i" ye karşılık gelir.  
    if i==N  
        done=1;  
    end  
    say=say+i;  
end  
say
```

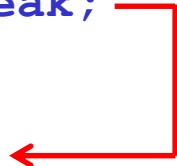
i, son sayıya (N'ye) ulaştığında, done değişkenine 0'dan farklı bir sayı atanır. Böylece, while'ın olduğu satıra gelindiğinde, done "0" olmadığı için while, end döngüsü çalışmaz (döngü sonlanır). Program, bu döngünün end satırının hemen altındaki satırdan işleme devam eder (burada, say değişkeni command window'da yazdırılır.).

# MATLAB/return ve break

```
for i=1:n  
    işlem  
    if koşul  
        break;  
    end  
end  
işlem
```



```
done=0;  
while done==0  
    işlem  
    if koşul  
        break;  
    end  
end  
işlem
```




- **break** komutu döngüleri sonlandırır.




# MATLAB/return ve break

```
for i=1:n
    işlem
    if koşul
        return;
    end
end
program biter ←
```



```
done=0;
while done==0
    işlem
    if koşul
        return;
    end
end
program biter ←
```



- **return** komutu programı sonlandırır.

## MATLAB/Uygulama-5

- Klavyeden girilen iki sayının toplamının bulunması ve ekrana virgülden sonra 3 hane olarak “toplam işleminin sonucu ... bulunmuştur” şeklinde yazdırılması.

```
clear
clc
% iki sayinin toplamının bulunmasi.
A=input('birinci sayiyi giriniz= ');
B=input('ikinci sayiyi giriniz= ');
toplam=A+B;
fprintf('toplam isleminin sonucu %1.3f bulunmustur \n',toplam)
```

## MATLAB/Uygulama-6

- Klavyeden girilen N sayısına göre faktöriyel değerini hesaplanması ve virgülden sonra 3 hane olarak “işlem sonucu ... bulunmuştur” şeklinde yazdırılması.

```
clear,clc
N=input('bir sayi giriniz=');
faktoriyel=1; %sayac
for i=1:N
    faktoriyel=faktoriyel*i;
end
faktoriyel
fprintf('işlem sonucu %1.3f bulunmuştur \n',faktoriyel)
```

# MATLAB/Uygulama-7

- Klavyeden girilen N sayısına göre;
  - 1' den N' e kadar tamsayıların toplamı
  - 1' den N' e kadar tek tamsayıların toplamı
  - 1' den N' e kadar çift tamsayıların toplamını veren programı yazınız.

```
clear
clc
% 1' den N' e kadar tamsayıların toplamı (T1)
% 1' den N' e kadar tek tamsayıların toplamı (T2)
% 1' den N' e kadar çift tamsayıların toplamını veren programı yazınız.(T3)

N=input('N sınır değerini giriniz= ');
T1=0;T2=0;T3=0;
for i=1:N
    T1=T1+i;
end
for j=1:2:N
    T2=T2+j;
end
for k=2:2:N
    T3=T3+k;
end
fprintf('1 den %d e kadar tamsayıların toplamı= %d \n',N,T1)
fprintf('1 den %d e kadar tek tamsayıların toplamı= %d \n',N,T2)
fprintf('1 den %d e kadar çift tamsayıların toplamı= %d \n',N,T3)
```

## MATLAB/Uygulama-8

- Bir metin içinde kaç tane “a” harfi olduğunu bulan programın yazılması.
- **a=metin icerisinde kac tane a var**

```
clear
clc
a='metin icerisinde kac tane a var';
s=0;
for i=1:length(a)
    if a(i)=='a'
        s=s+1;
    end
end
s
```

# MATLAB/Uygulama-9

- input fonksiyonu ve for end döngüsü kullanılarak A matrisi elemanlarını oluşturunuz.

```
clear
clc
m=input('A matrisinin satir sayisini giriniz= ');
n=input('A matrisinin sutun sayisini giriniz= ');
for i=1:m
    for j=1:n
        fprintf('A matrisinin %d,%d.ci elemanini giriniz:',i,j)
        A(i,j)=input("");
    end
end
A
```

# MATLAB/Uygulama-10

1 ve 2. noktanın X ve Y koordinatlarının girilmesinden sonra, (1-2) açıklıklık açısının kaçınıcı bölgeye düştüğünü belirleyen, ilgili bölgeyi bir msgbox kutusunda yazdıran bir program yazınız.

## Çözüm

```
clear
clc
X1=input('X1=');Y1=input('Y1=');
X2=input('X2=');Y2=input('Y2=');
DX=X2-X1;DY=Y2-Y1;
if (DX>0) & (DY>0)
    a='Aci 1. bolgede';
end

if (DX<0) & (DY>0)
    a='Aci 2.bolgede';
end

if (DX<0) & (DY<0)
    a='Aci 3.bolgede';
end

if (DX>0) & (DY<0)
    a='Aci 4.bolgede';
end

msgbox(a,'Bolge?')
```

# MATLAB/Uygulama-11

1 ve 2. noktanın X ve Y koordinatlarının girilmesinden sonra, (1-2) açıklık açısını hesaplayan bir program yazınız.

Çözüm

```
clear
clc
X1=input('X1=');Y1=input('Y1=');
X2=input('X2=');Y2=input('Y2=');
DX=X2-X1;DY=Y2-Y1;

if (DX~=0) & (DY~=0),a=atan(DY/DX);a=a*200/pi;
    if (DX>0) & (DY>0),a=a;end
    if (DX<0) & (DY>0),a=a+200;end
    if (DX<0) & (DY<0),a=a+200;end
    if (DX>0) & (DY<0),a=a+400;end
end

if (DX==0) & (DY>0),a=100;end
if (DX==0) & (DY<0),a=300;end
if (DX>=0) & (DY==0),a=0;end
if (DX<0) & (DY==0),a=200;end
a
%veya output a, aşağıdaki biçimde yazdırılabilir.
fprintf('(1-2) aciklik acisi= %1.5f grad \n',a)
```

Dış koşul



# MATLAB/Uygulama-12

1 ve 2. noktanın X ve Y koordinatlarının girilmesinden sonra, (1-2) açıklıklık açısını ve (1-2) kenar uzunluğunu hesaplayan bir program yazınız.

## Çözüm

```
clear
clc
X1=input('X1=');Y1=input('Y1=');
X2=input('X2=');Y2=input('Y2=');
DX=X2-X1;DY=Y2-Y1;
```

```
if (DX~=0) & (DY~=0) ,a=atan(DY/DX) ;a=a*200/pi;
    if (DX>0) & (DY>0) ,a=a;end
    if (DX<0) & (DY>0) ,a=a+200;end
    if (DX<0) & (DY<0) ,a=a+200;end
    if (DX>0) & (DY<0) ,a=a+400;end
end
```

```
if (DX==0) & (DY>0) ,a=100;end
if (DX==0) & (DY<0) ,a=300;end
if (DX>=0) & (DY==0) ,a=0;end
if (DX<0) & (DY==0) ,a=200;end
```

```
S=sqrt(DX^2+DY^2) ;%kenar
```

```
fprintf('(1-2) aciklik acisi= %1.5f grad \n' ,a)
fprintf('(1-2) kenar uzunlugu= %1.3f m' ,S)
```

Sonuçları, ayrıca sprintf fonksiyonu ile bir msgbox'a alt alta yazdırınız.

# MATLAB/Uygulama-13

- Kullanıcıyı,1'i seçmesi durumunda YTÜ web sayfasına, 2'yi seçmesi durumunda istediğiniz bir web sayfasına yönlendiren bir menü programı yazınız.

## Çözüm

```
clear
clc
disp('[1]...YTU web sayfası')
disp('[2]...Bahattin Erdogan web sayfası')
a=input('<Selection>=');

while (a>2) | (a<=0)
    a=input('lütfen seçim numarasını doğru giriniz');
end

if a==1
    web www.yildiz.edu.tr -browser
end
if a==2
    web www.yildiz.edu.tr/~berdogan -browser
end
```

# MATLAB/Uygulama-14

n sayıda ölçünün tek tek düzeltilmesi ve standart sapması girildiğinde ilgili ölçünün kaba hatalı olup olmadığını belirleyen, kaba hatalı ölçüleri (varsa) yazdıran bir program oluşturunuz.

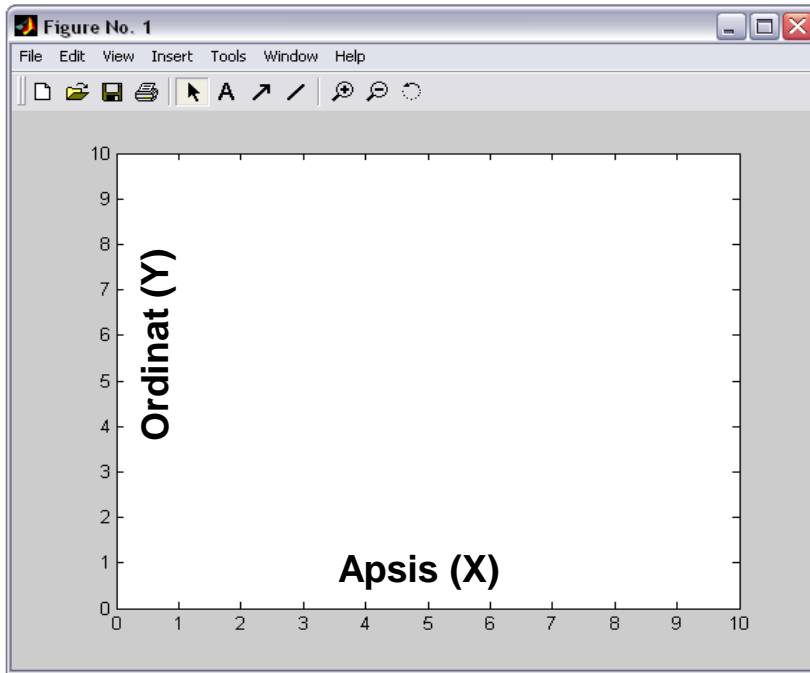
## Çözüm

```
clear,clc
n=input('olcu sayisi=');say=0;
for i=1:n
    v=input('duzeltme=');
    s=input('standart sapma=');
    if abs(v)>=(3*s)
        disp('-----')
        disp([int2str(i) '. olcu kaba hatali'])
        disp('-----')
        say=say+1;KH(say,1)=i;
    end
    if abs(v)<(3*s)
        disp('-----')
        disp([int2str(i) '. olcu normal'])
        disp('-----')
    end
end
if say==0
    disp('Kaba hatali olcu yok')
end
if say>0
    disp('Kaba hatali olan olculer')
    KH
end
```

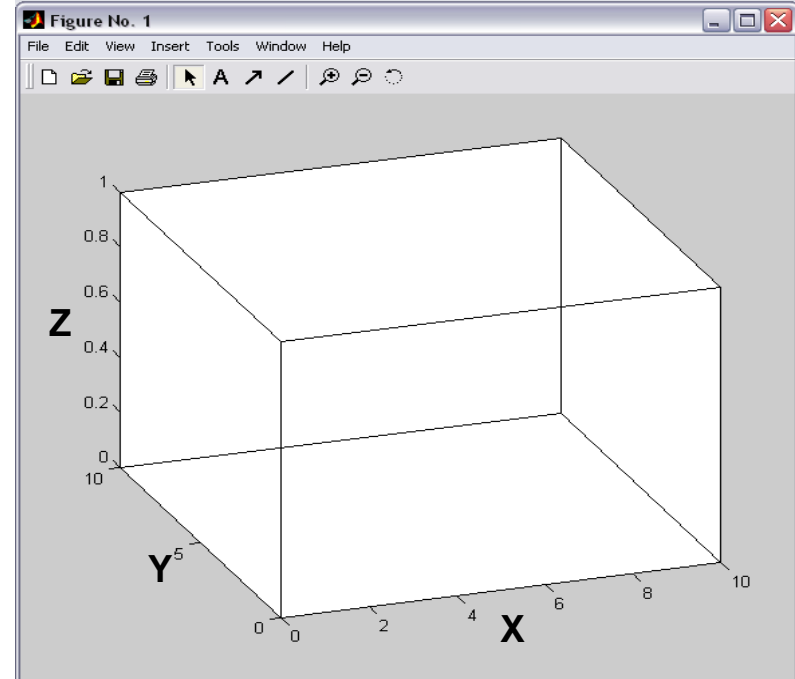
# MATLAB/Grafik

- Matlab'de grafikler “figure” penceresinde çizdirilir.
- İki ve üç boyutlu çizim yanı sıra, kutupsal koordinat sisteminde de çizim olanağı bulunur (bak., polar).

## İki Boyutlu Koordinat Sistemi



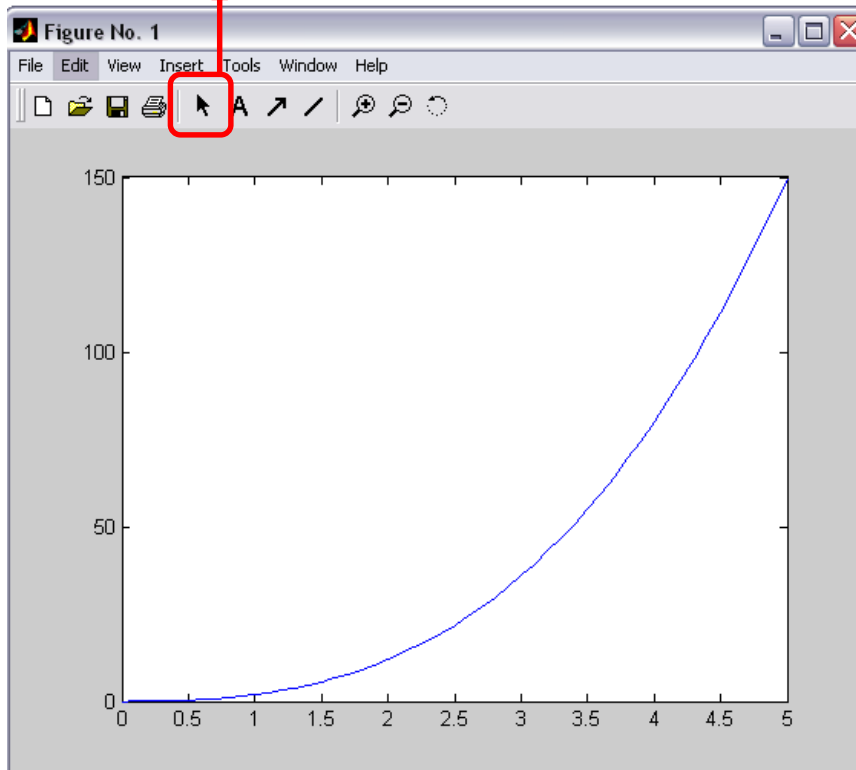
## Üç Boyutlu Koordinat Sistemi



# MATLAB/Grafik

- Matlab'de en temel çizim fonksiyonu **plot**'dur.
- Örneğin,  $x=0:0.1:5$  olan bir dizi vektör elemanlarına karşılık,  $y=x.^3+x.^2$  fonksiyon değerleri hesaplatılsın.
- **plot(x,y)** ile aşağıdaki grafik çizdirilir.

Edit plot

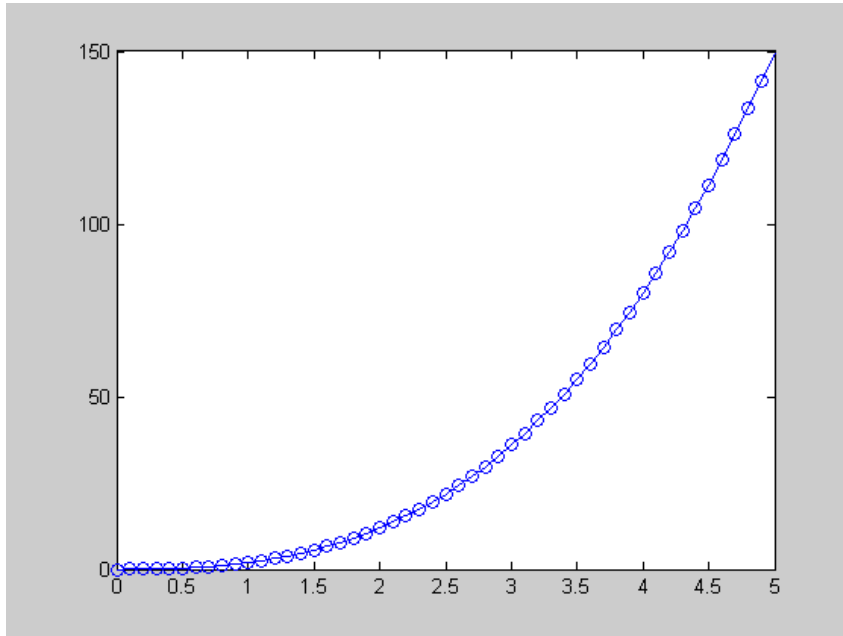


- Çizilen grafiğin üzerinde birçok değişiklik yapmak mümkündür.
- Bunun için **“Edit plot”** düğmesi tıklanır.
- İlgili nesne (çizdirilen eğri, eksenler vb.) iki kez tıklanarak beliren **“Property Editor”** penceresinden istenilen değişiklikler yapılabilir.
- Property Editor penceresinden yapılan her türlü değişikliği, komut olarak yaptırmak mümkündür.

Örneğin, `plot(x,y, '-o')` hem ardışık noktaları şekildeki gibi birleştirir, hem de x,y nokta çiftlerini grafik üzerinde bir “o” sembolü ile işaretler.

# MATLAB/Grafik

- `plot(x,y, '-o')` ile ilgili grafik aşağıdaki gibi olur.

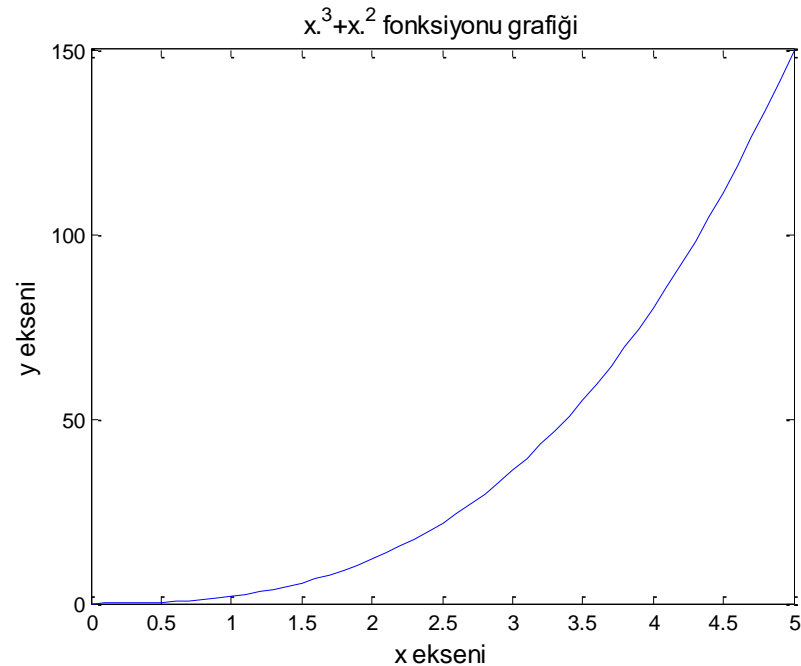


- Aşağıdaki ifadelerle çizimi tekrarlayınız:  
`plot(x,y, '-o')`  
`plot(x,y, '-*')`  
`plot(x,y, '-+')`  
`plot(x,y, '-^')`  
`plot(x,y, '-.')`
- Sözü edilen grafik üzerindeki o,\*,+ gibi sembollere marker denir.
- `plot` fonksiyonu ile ilgili eğrinin rengini değiştirmek de mümkündür:  
`plot(x,y, 'r')` kırmızı (red)  
`plot(x,y, 'k')` siyah  
`plot(x,y, 'b')` mavi (blue)  
`plot(x,y, 'g')` yeşil (green)

# MATLAB/Grafik

- Grafik çizimlerinde grafik için başlık oluşturmak, x-y eksen takımlarını isimlendirmek ihtiyaçları duyabiliriz. Bu durumlarda sırası ile “**title**”, “**xlabel**” ve “**ylabel**” fonksiyonları kullanılır.

```
>> x=[0:0.1:5];  
>> y=x.^3+x.^2;  
>> plot(x,y)  
>> title('x.^3+x.^2 fonksiyonu grafiği')  
>> xlabel('x eksen')  
>> ylabel('y eksen')
```



# MATLAB/Grafik

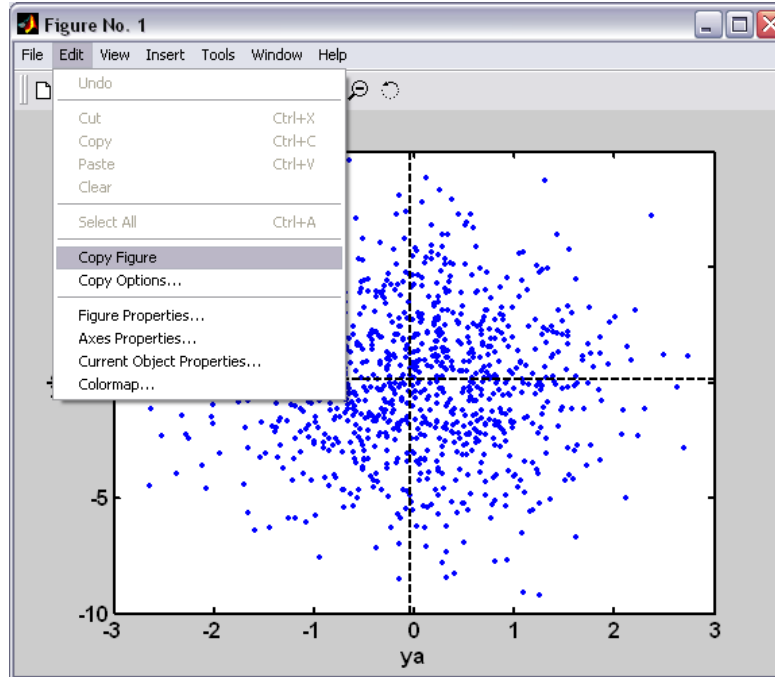
- Matlab' da bazı durumlarda değişkenin ve fonksiyonun sadece ilgilenilen aralıklarda değişiminin çizilmesi istenebilir.

<i>Fonksiyon</i>	<i>Tanımı</i>
<code>axis([xmin xmax ymin ymax])</code>	Eksen takımını verilere göre yeniden yapılandırır.
<code>axis equal</code>	Eksen takımlarını eşit ölçekli ve artımlı yapılandırır.
<code>axis square</code>	Eksen takımlarını eşit uzunlukta kare olarak yapılandırır.
<code>axis normal</code>	axis equal ve axis square fonksiyonlarının etkilerini kaldırır.
<code>axis off</code>	Eksen takımlarını, etiketlerini ve eksen üzerindeki sayıları kaldırır.
<code>axis on</code>	Eksen takımlarını ve diğer özellikleri geri getirir.



# MATLAB/Grafik-Kaydetme ve kopyalama

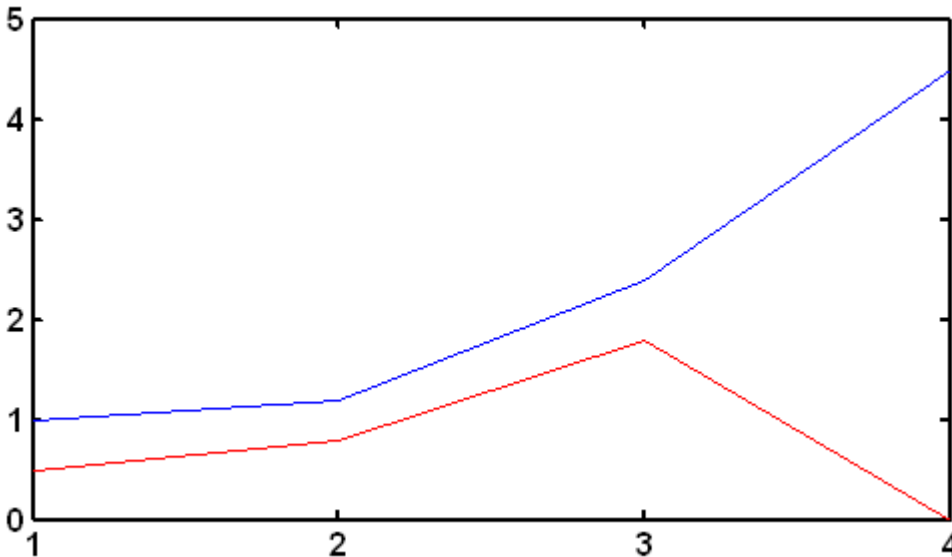
- Çizilen grafikleri kaydetmek için Figure penceresindeki “File” menüsünden “Save” veya “Save As” seçenekleri seçilir. Grafikler, “fig” uzantılı dosyalar olarak ilgili klasöre kaydedilirler.
- Çizilen grafiklerin başka bir ortama aktarılmaları için, “Edit” menüsünün altındaki “Copy Figure” seçeneği seçilir. (Not: Kopyalamanın arka plan rengini ayarlamak için “Copy Options” seçeneğine bakınız.)



## MATLAB/Grafik-Aynı eksen takımına farklı grafikler çizdirme

- Aynı eksen takımına farklı grafikleri çizdirmek için **hold on** ve **hold off** komutları kullanılır. Bu iki komut arasına yazılan her türlü grafiğin çizimi aynı eksen takımında gösterilir.
- Örneğin,  $y_a = [1; 1.2; 2.4; 4.5]$  ve  $y_b = [0.5; 0.8; 1.8; 0]$  vektörleri ile ifade edilen iki farklı ölçü grubunu  $x = [1; 2; 3; 4]$  vektörüne göre aynı eksen takımında çizdirmek için aşağıdaki komutları yazmak yeterlidir:

```
hold on, plot(ya), plot(yb, 'r'), hold off
```



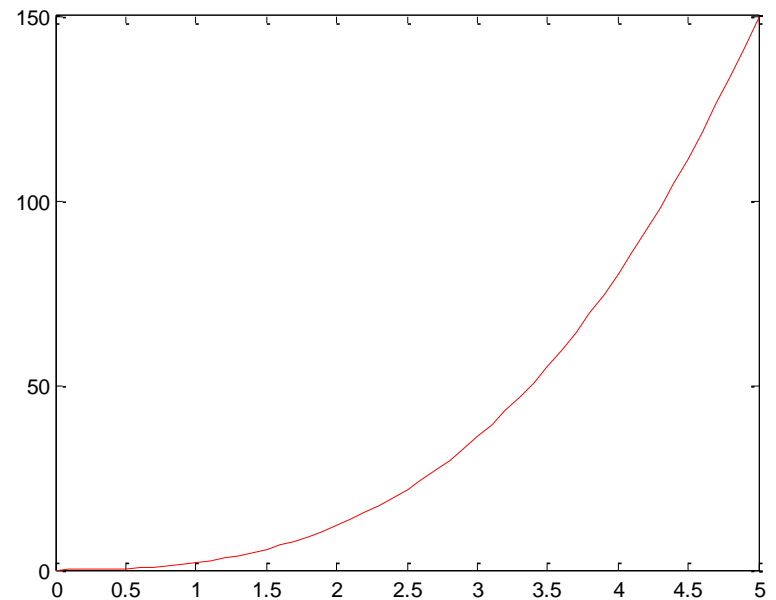
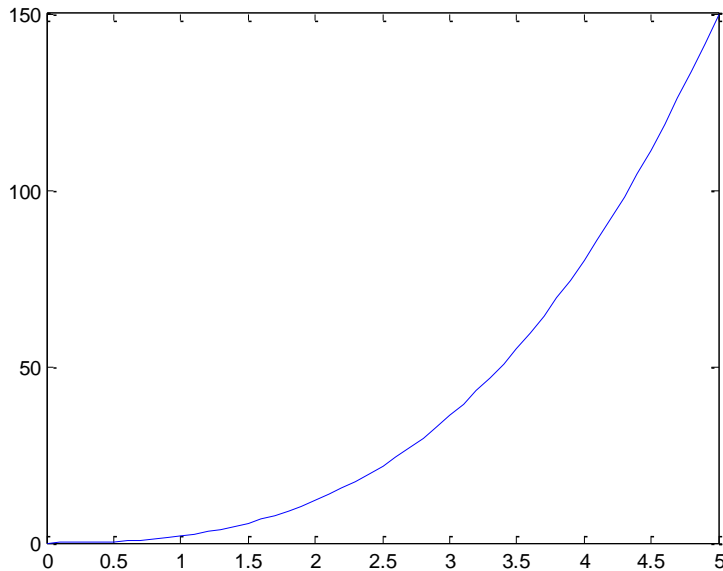
**Not:** Eğer x eksenini, bu örnekte olduğu gibi, y değerlerinin indisini, yani kaçınıcı değer olduğunu, gösteriyorsa, plot fonksiyonunda x'in yeniden belirtilmesine gerek yoktur.

Alternatif olarak  
`plot(x,ya,x,yb)` veya  
`plotyy(x,ya,x,yb)` kullanılabilir.

# MATLAB/Grafik-Çoklu Grafiklerin Oluşturulması

- **figure** fonksiyonu farklı verilere ait grafiklerin farklı pencerelerde gösterilmesine olanak sağlar.

```
x=[0:0.1:5];  
y1=x.^3+x.^2;  
y2=x.^4+x.^2;  
figure(1)  
plot(x,y1)  
figure(2)  
plot(x,y2,'r')
```

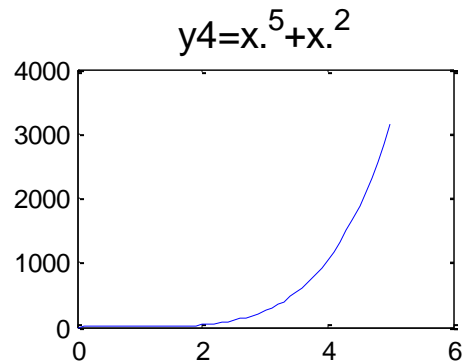
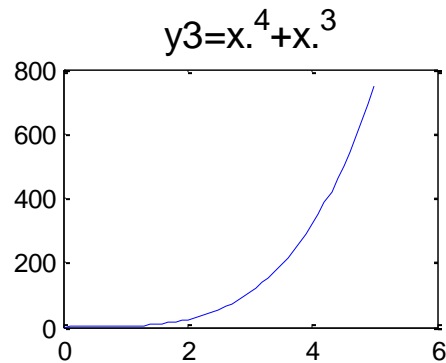
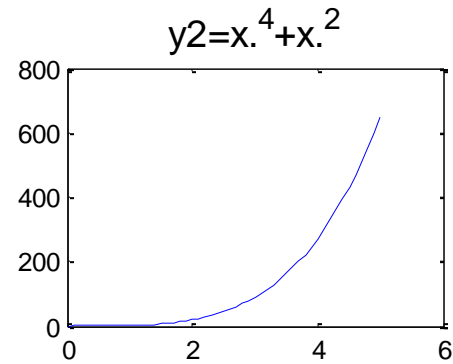
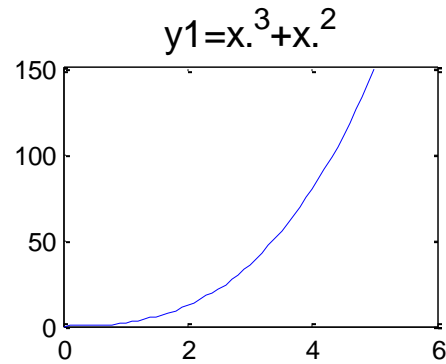


# MATLAB/Grafik-Alt Grafiklerin Oluşturulması

- Matlab' da aynı grafik penceresinde birden fazla grafiğe yer vermek **subplot** fonksiyonu sayesinde mümkün olmaktadır. Genel kullanımı **subplot(a,b,c)** şeklindedir. Bu durumda grafik penceresi **axb** boyutunda yapılandırılır; **c** ise grafik komutlarının işletileceği alt pencere numarasıdır.

```
x=[0:0.1:5];  
y1=x.^3+x.^2;  
y2=x.^4+x.^2;  
y3=x.^4+x.^3;  
y4=x.^5+x.^2;  
subplot(2,2,1)  
plot(x,y1)  
title('y1=x.^3+x.^2','fontsize',14)  
subplot(2,2,2)  
plot(x,y2)  
title('y2=x.^4+x.^2','fontsize',14)  
subplot(2,2,3)  
plot(x,y3)  
title('y3=x.^4+x.^3','fontsize',14)  
subplot(2,2,4)  
plot(x,y4)  
title('y4=x.^5+x.^2','fontsize',14)
```

# MATLAB/Grafik-Alt Grafiklerin Oluşturulması

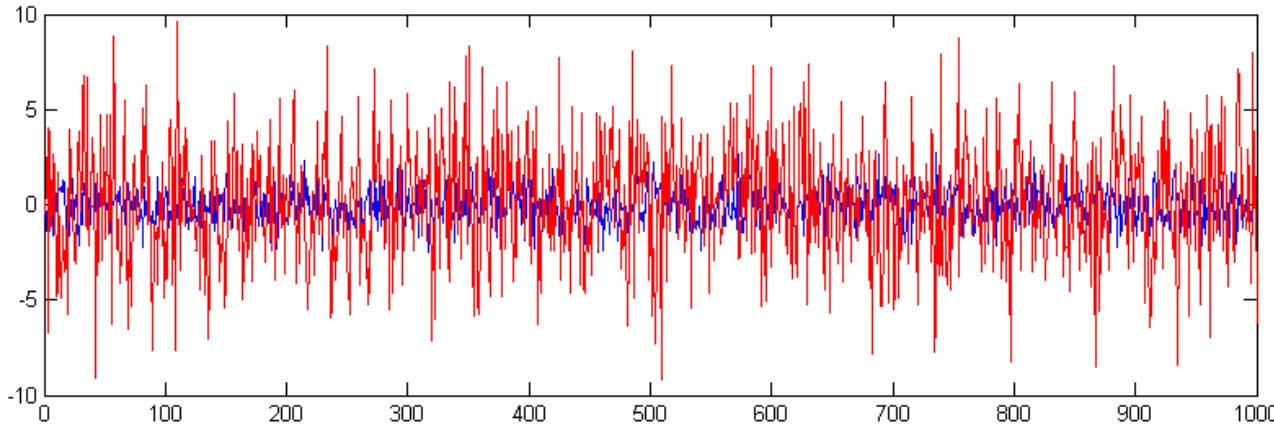


# MATLAB/Grafik-Grafik yoluyla bilgi üretme

**Örnek:** `ya=randn(1000,1)` ve `yb=randn(1000,1)*3` biçiminde iki ölçü grubu oluşturalım. (`randn` fonksiyonu, beklenen değeri 0, standart sapması 1 olan normal dağılmış sayı üretir). `ya`'nın standart sapması 1, `yb`'nin standart sapması ise 3'tür.

Bu ölçülerin, a ve b kaynaklarından elde edildiğini ve de standart sapmalarını bilmediğimizi düşünelim. Hangi ölçü grubunun daha kaliteli olduğunu (standart sapmasının düşük olduğunu) grafik üzerinden görebilmek için, bir önceki örnekteki hold on/hold off komutlarını kullanarak bunları çizdirmek yeterli olacaktır:

```
hold on, plot(ya), plot(yb,'r'),hold off
```



Böylece, kırmızı ile gösterilmiş `yb` ölçülerinin sıfırdan daha çok saptıkları, dolayısıyla standart sapmasının daha yüksek olduğu bilgisi grafik üzerinden kolaylıkla okunabilmektedir.

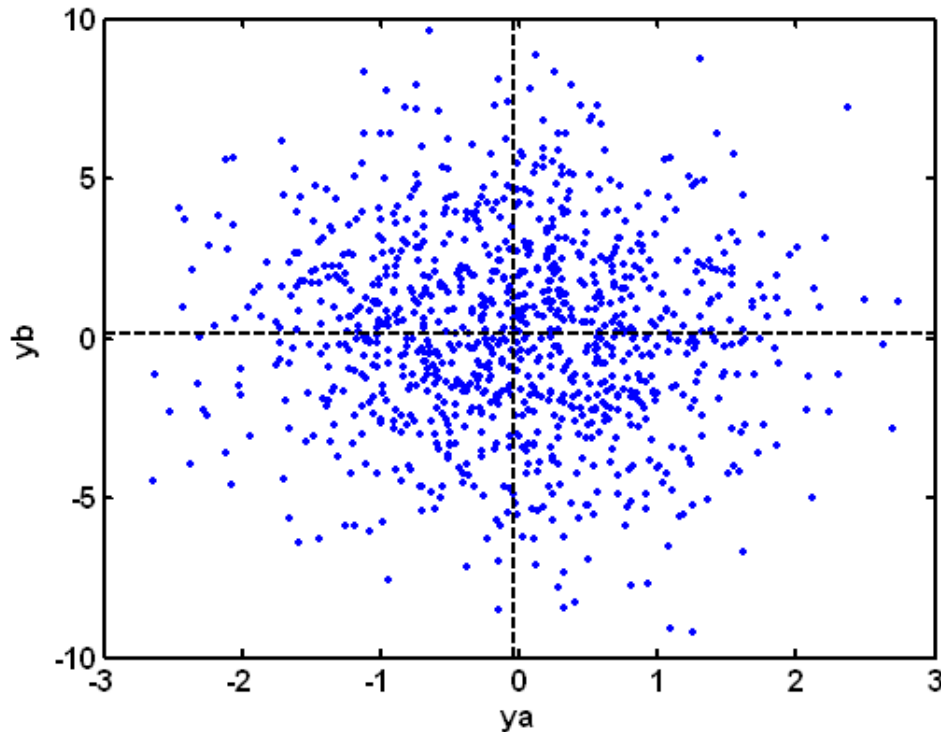
- **Not:** `sqrt(yb'*yb/999)` işlemi, `ya`'nın *deneysel* standart sapmasını verecektir (bkz. İstatistik Ders Notları). Bu değerin "3" *kuramsal* standart sapma değerine yakın olacağına dikkat ediniz.

# MATLAB/Grafik-Grafik yoluyla bilgi üretme

- Bir önceki örnekte kullanılan ya ve yb ölçülerinin birbirleriyle nasıl bir ilişkide olduğunu görmek için,

```
plot(ya,yb,'.')
```

komutunu kullanmak yeterlidir.



- İlgili grafikten, ya ve yb ölçüleri arasında anlamlı bir ilişki (korelasyon) olmadığı bilgisi hemen türetilebilir. Çünkü beklenen değerleri 0 olan bu iki gruba ilişkin ölçü çiftleri, 0 merkezinde düzgün olarak (daire biçiminde) dağılmışlardır.
- İki ölçü grubu arasında korelasyon olabilmesi için, bu nokta bulutunun bir doğru etrafında gözlenmesi gerekir.

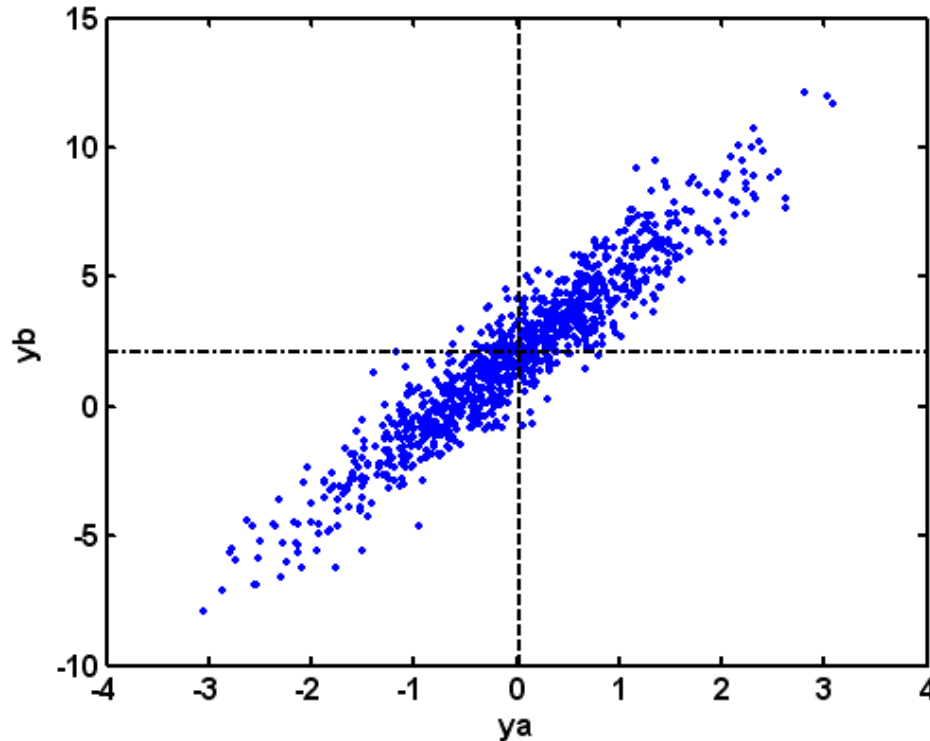


# MATLAB/Grafik-Grafik yoluyla bilgi üretme

- Bir önceki örnekte kullanılan  $y_b$  ölçülerini,  $y_b = 2 + 3 \cdot y_a + \text{randn}(1000, 1) \cdot 1$ , biçiminde  $y_a$  ölçülerine bağlı olarak üretelim. Bu durumda,

```
plot(ya, yb, 'b.'))
```

ile oluşturulan grafikten  $y_a$  ve  $y_b$  ölçüleri arasındaki korelasyonun varlığı hemen görülecektir.



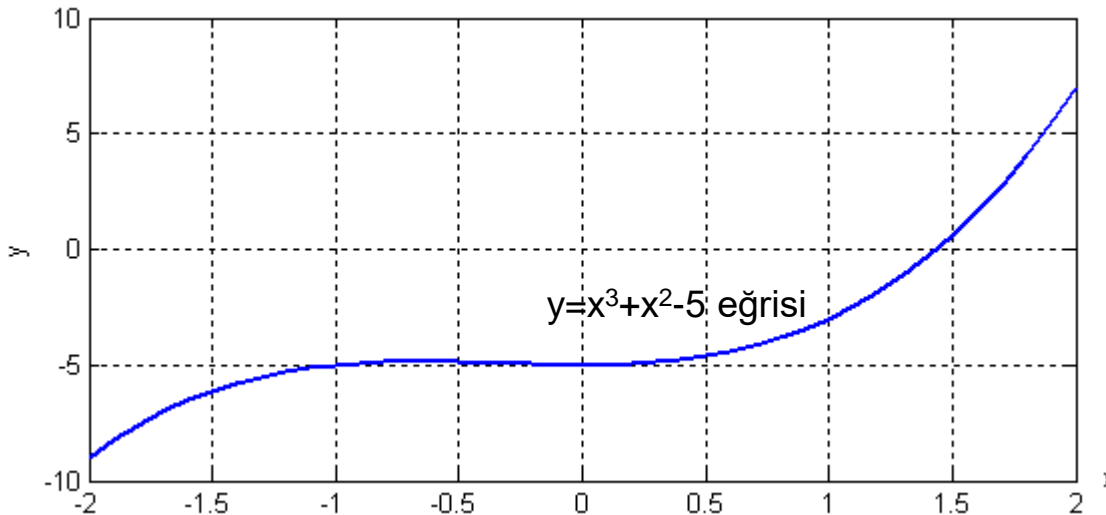


# MATLAB/Grafik-Grafik yoluyla bilgi üretme

- Mühendislik uygulamalarında en çok karşılaşılan problemlerden biri de  $F(x)=0$  biçimindeki bir denklemin ilgili aralıktaki kökünü (fonksiyonu sıfır yapan  $x$  değerini) bulmaktır. Sayısal analizde kullanılan Newton-Raphson gibi yöntemlerde kökün yaklaşık değerine ihtiyaç vardır. Bu yaklaşık değeri bulmak için **grafik çizimi** oldukça kullanışlı olmaktadır. Örneğin,
- $F(x)=x^3+x^2-5=0$  gibi bir denklemin -2 ile 2 arasındaki yaklaşık kökünü bulmak için,  $x=-2:0.1:2$  biçiminde  $x$  değerleri ve  $y=x.^3+x.^2-5$  ile de bu  $x$ 'lere karşılık  $y$  değerleri üretilir.

`plot(x,y),grid on`

komutlarıyla aşağıdaki grafik çizdirilir.(`grid on` komutu şekildeki grid ağını çizer)



- $y=0$  doğrusunun eğriyi kestiği noktadan,  $x$  eksenine hayali bir dik inilirse, bu dikin gösterdiği  $x$  değeri,  $F(x)$  denklemini sağlayan kök olacaktır. Buradan kökün yaklaşık değerinin 1.4 olduğu sonucuna kolaylıkla ulaşılır.
- **Not:** Figure penceresindeki büyütme özelliği ile, ilgili kesişim noktasına zoom yapılarak, yaklaşık kök daha hassas biçimde belirlenir.

# MATLAB/Grafik-Basic Fitting Tool

- Bir mühendis, bir olayı gözler ve gözlem sonucunda elde ettiği ölçüler yoluyla olayı matematiksel eşitliklerle açıklamaya çalışır. Böylesi eşitliklere, kısaca “**model**” adı verilir.
- Figure penceresinde yer alan “Tools” menüsü içindeki “Basic Fitting” seçeneği grafik üzerindeki x ve bunlara karşılık gelen y değerlerini kullanarak, bunlara **en iyi uyan**  $y=f(x)$  polinomunu tanımlar. Böylece oldukça pratik bir biçimde model oluşturulur.
- Burada hatırlatılması gereken iki nokta vardır:

(1) Eğer nokta çifti (x,y) sayısı uydurulan polinomun bilinmeyen sayısına eşitse, bulunan fonksiyon bir enterpolasyon polinomudur.

(**Not:** n. dereceden bir polinomun n+1 adet bilinmeyeni olduğunu hatırlayınız: Örneğin 4. dereceden bir polinom;  $y=ax^4+bx^3+cx^2+dx+e$  dir ve bilinmeyen sayısı 5’dir)

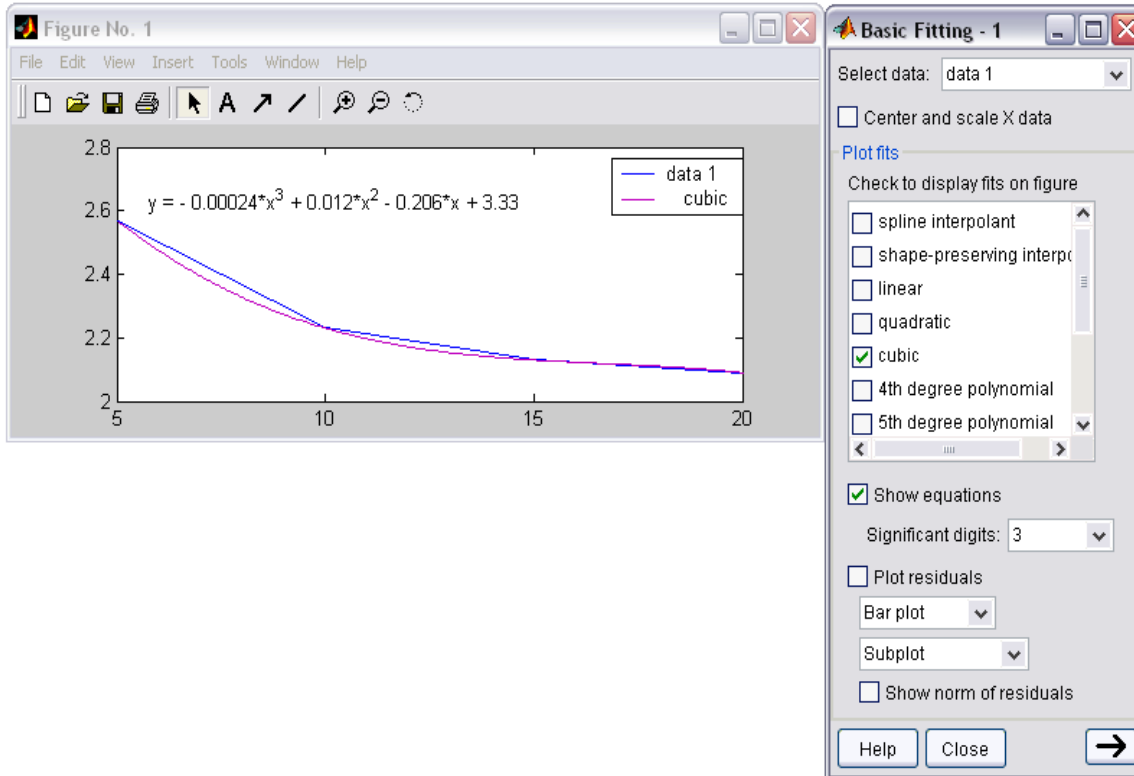
(2) Eğer nokta çifti sayısı, uydurulan polinomun bilinmeyen sayısından fazlaysa en uygun polinom bir “*en küçük kareler*” kestirim yöntemi sonucudur. y değerleri hatalı büyüklüklerse (yani ölçü ise), basic fitting ile uydurulacak polinomun bilinmeyen sayısı her zaman ölçü sayısından küçük olmalıdır!

# MATLAB/Grafik-Basic Fitting Tool

**Örnek:** Aşağıdaki tabloda  $f= 5, 10, 15$  ve  $20$  değerlerine karşılık  $t$ -dağılımının  $\alpha=\%5$  güven sınırları ( $t$ , değerleri) verilmektedir.  $t=af^3+bf^2+cf+d$  polinomunu “basic fitting” özelliğini kullanarak belirleyiniz.  $f=9$  için  $t=2.26$  olduğuna göre elde edilen enterpolasyon polinomun doğruluğunu test ediniz.

f	5	10	15	20
t	2.57	2.23	2.13	2.09

**Çözüm:**  $x=[5;10;15;20]$  ve  $y=[2.57;2.23;2.13;2.09]$  olsun. `plot(x,y)` ile ilgili eğri çizilir.



- Figure penceresindeki “Tools” menüsünden, “Basic Fitting” seçeneği seçilir.
- Açılan, “Basic Fitting” penceresinden ilgili polinom (burada, cubic, yani 3.derece) ve ardından, “show equations” seçeneği işaretlenir.
- Şekil üzerinde gösterilen  $f(x)$  eşitliği, bize enterpolasyon polinomunu vermektedir.
- Bu denklemde,  $x=9$  girilirse,  $y=2.24$  değeri elde edilir.  $f=9$  için  $t=2.26$  olduğu bilindiğine göre, enterpolasyon polinomumuzun doğruluğu-*bu aralık için-%2*’dir.

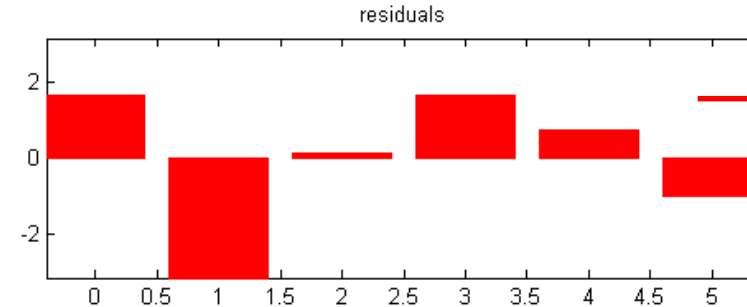
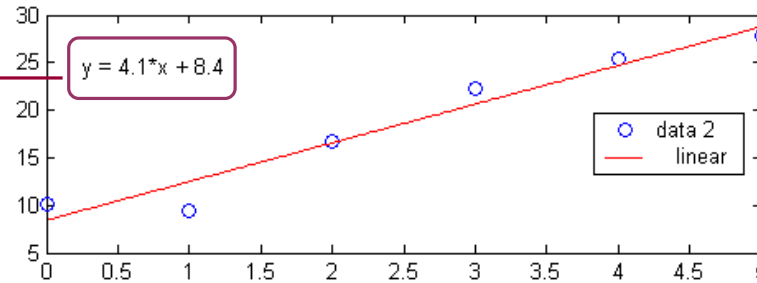
# MATLAB/Grafik-Basic Fitting Tool

**Örnek:** Aşağıdaki tabloda, x zamanlarına karşılık y ölçüleri elde edilmiştir. Ölçülere en iyi uyan  $y=a+bx$  doğrusunu belirleyiniz (En küçük kareler kestirim yöntemi)

x	0	1	2	3	4	5
y	10.06	9.36	16.69	22.28	25.44	27.75

**Çözüm:** Tablodaki değerler x ve y vektörlerine atanır. `plot(x,y,'o')` ile ilgili eğri çizilir. Basic Fitting penceresinde, “linear”, “show equation”, “plot residuals” seçenekleri işaretlendiğinde, aşağıdaki grafik oluşturulur.

En küçük kareler yöntemine göre belirlenen en uygun doğru denklemi (*model*)\*



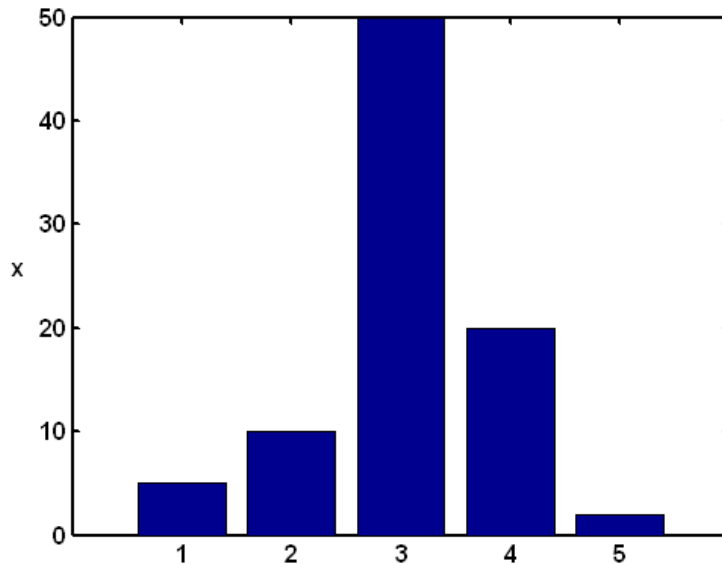
Ölçülerin, belirlenen doğru denkleminden sapmasını (düzeltmeleri-residuals) gösterir.

\* Demirel H (2005), Dengeleme Hesabı Ders Notları, YTÜ, İstanbul

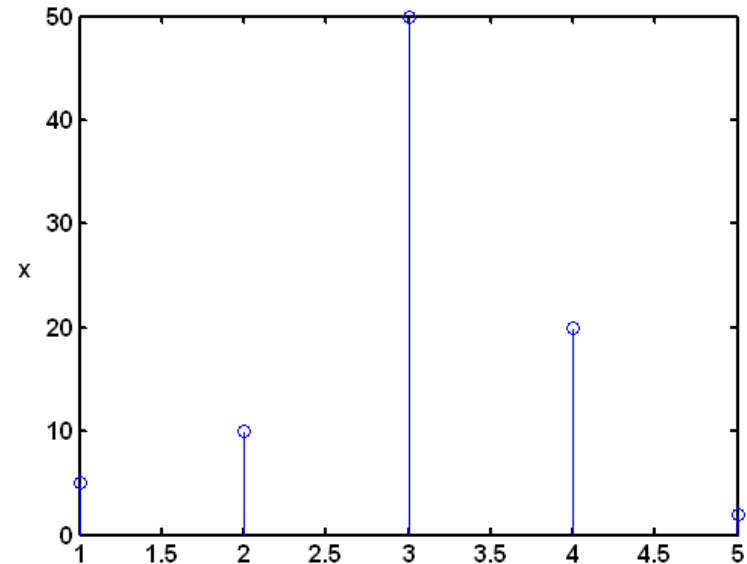
# MATLAB/Grafik-Çubuk (bar) ve stem grafiği

- Matlab'de farklı gösterimlere göre çizim yapmak mümkündür. Bunlardan ikisi bar(...) ve stem(...) çizim fonksiyonlarıdır.
- **Örnek:**  $\mathbf{x}=[5;10;100;20;2]$  vektör elemanlarının bar ve stem grafik olarak göstermek isteyelim. `bar(x)` ve `stem(x)` aşağıdaki grafikleri çizdirecektir.

Bar grafiği



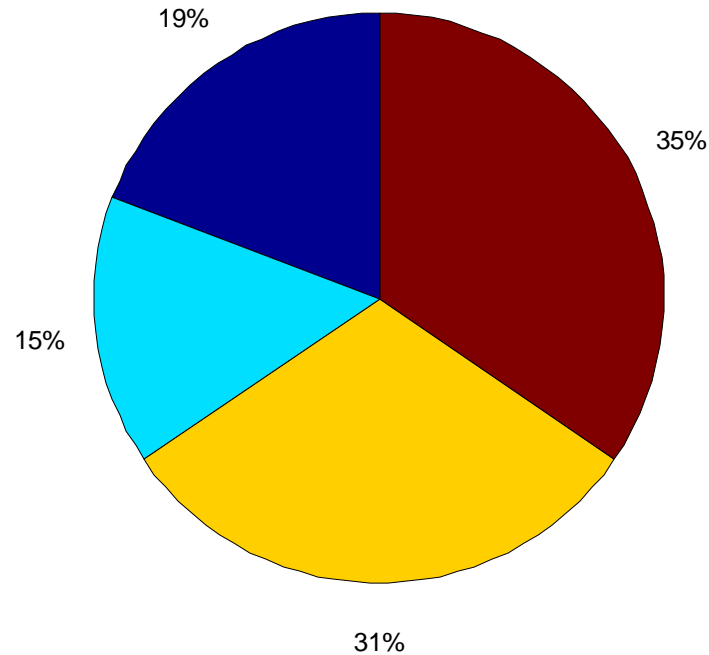
Stem grafiği



# MATLAB/Grafik-pie fonksiyonu

- **pie([dizi elemanları])** fonksiyonu dairesel grafikler oluşturmak için kullanılır.

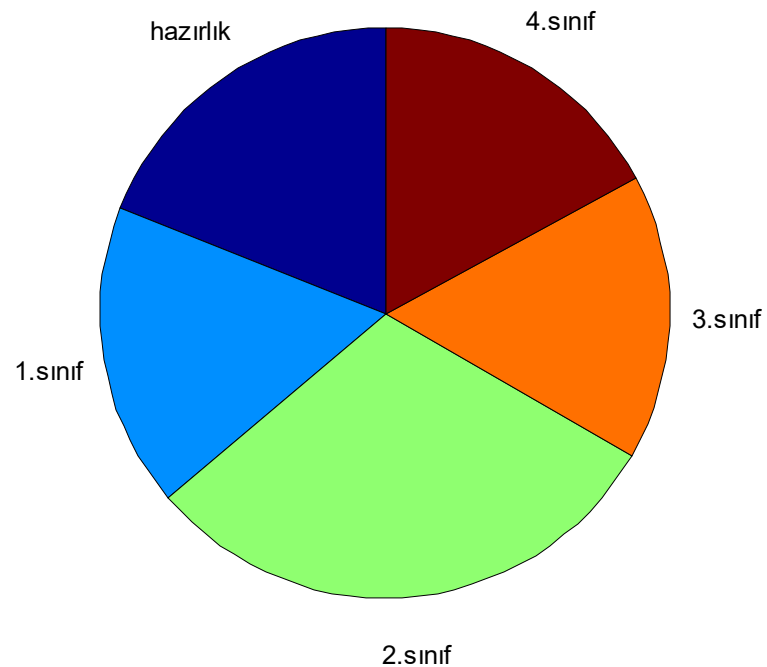
```
a=[5 4 8 9];  
pie(a)
```



# MATLAB/Grafik-pie fonksiyonu

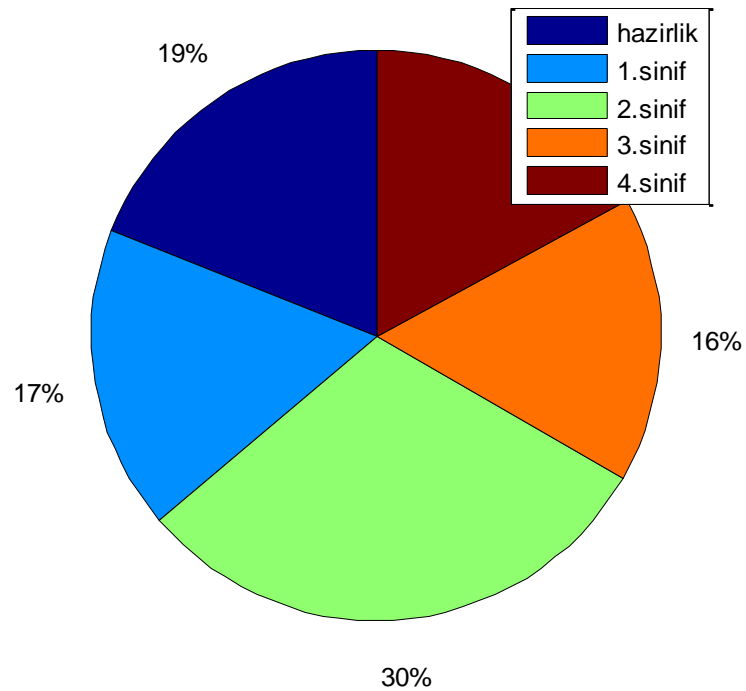
- Sınıflara göre öğrenci sayıları verildiğine göre, bu değerlerin yüzdelik dağılımlarını bir grafik üzerinde gösterelim.

```
clear  
clc  
a=[250, 225, 400, 212, 225];  
b={'hazirlık','1.sinif', '2.sinif','3.sinif','4.sinif'};  
pie(a,b)
```



# MATLAB/Grafik-pie fonksiyonu

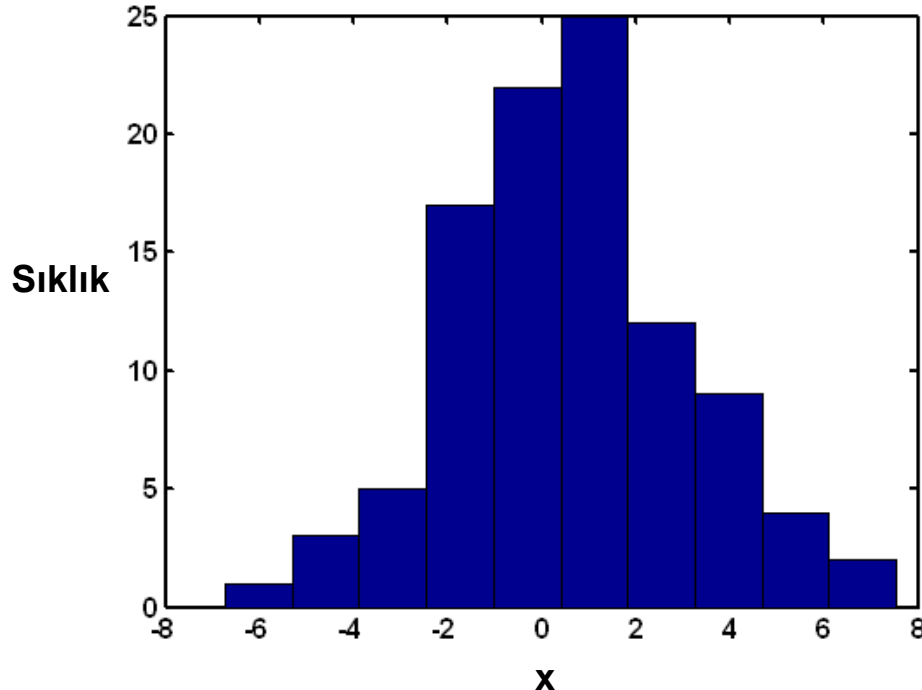
```
clear  
clc  
a=[250, 225, 400, 212, 225];  
pie(a)  
legend('hazirlik','1.sinif', '2.sinif','3.sinif','4.sinif');
```





# MATLAB/Grafik-Histogram

- Ölçülerin hangi istatistiksel dağılıma uyduğunu görebilmek için, frekans (sıklık) değerleri hesaplanır ve histogram grafikleri çizilir.
- Elimizde, aynı dağılımda olduğu bilinen bir x ölçü vektörü varsa, `hist(x)` fonksiyonu otomatik olarak bir histogram grafiği çizer.
- Örneğin, `x=randn(100,1)*3` biçiminde normal dağılmış bir ölçü grubu üretelim. `hist(x)` ile aşağıdaki histogram grafiği oluşturulur (Her bir barın üst noktası birleştirildiğinde oluşan eğrinin bir normal dağılım eğrisi veya diğer adıyla çan eğrisi biçiminde olduğu görülecektir.)

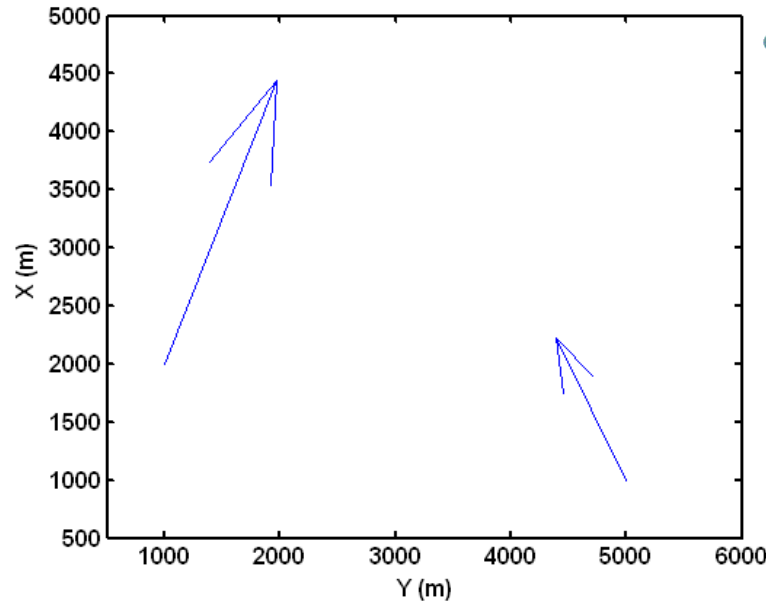


# MATLAB/Grafik-Vektör çizimi

- $x$  ve  $y$  koordinat değerlerine sahip bir noktanın  $dx$  ve  $dy$  kadar yer değiştirdiği düşünölsün. Bu noktadaki  $(dx,dy)$  vektörünü çizdirmek istediğimizde, **quiver** fonksiyonu kullanılır.
- Örneğin, bir jeodezik dik koordinat sisteminde iki noktanın koordinatları  $\mathbf{x}=[1000;2000]$  ,  $\mathbf{y}=[5000;1000]$  vektörleri, bu noktadaki değişimler ise  $d\mathbf{x}=[1;2]$  ve  $d\mathbf{y}=[-0.5;0.8]$  ile tanımlansın.

**quiver(y,x,dy,dx)** (**Not:** Bir jeodezik dik koordinat sisteminde  $x$  ve  $y$ 'nin yer değiştirdiğini hatırlayınız!)

komutu ile bir jeodezik dik koordinat sisteminde vektör çizimi gerçekleştirilir.



- Vektörleri ölçeklendirmek için,  $s$  ölçek faktörü quiver fonksiyonuna beşinci bir değişken olarak eklenmelidir;

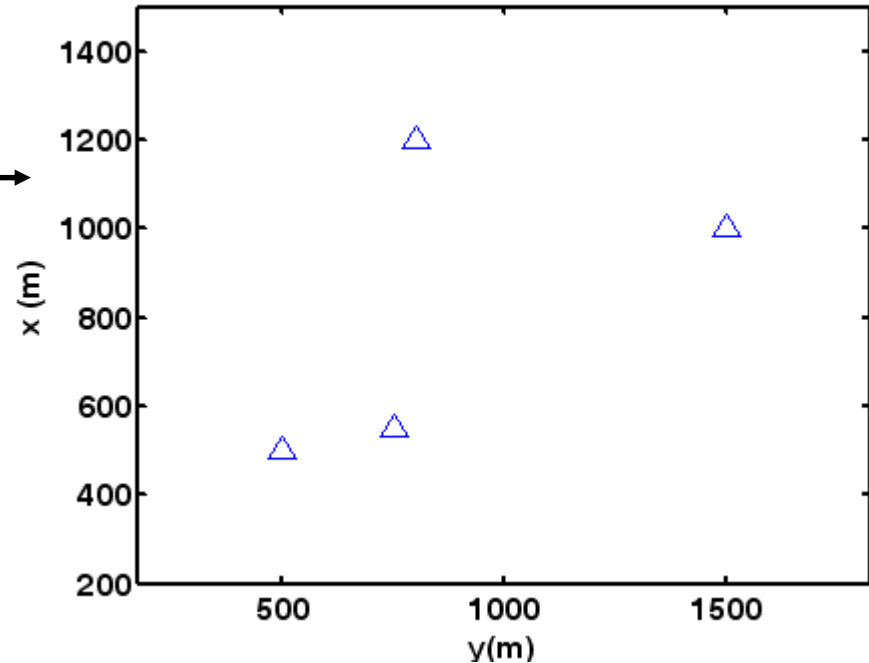
**quiver(y,x,dy,dx,s)**

# MATLAB/Grafik-Kanava Çizimi

- Bir jeodezik dik koordinat sistemindeki x ve y koordinatları verilen jeodezik noktaları, nokta sembolleri üçgen olacak biçimde çizdiriniz.

Nokta	P1	P2	P3	P4
x (m)	500.00	550.00	1000.00	1200.00
y (m)	500.00	750.00	1500.00	800.00

```
plot(y,x,'^')  
axis([200 1700 200 1500])  
axis equal
```

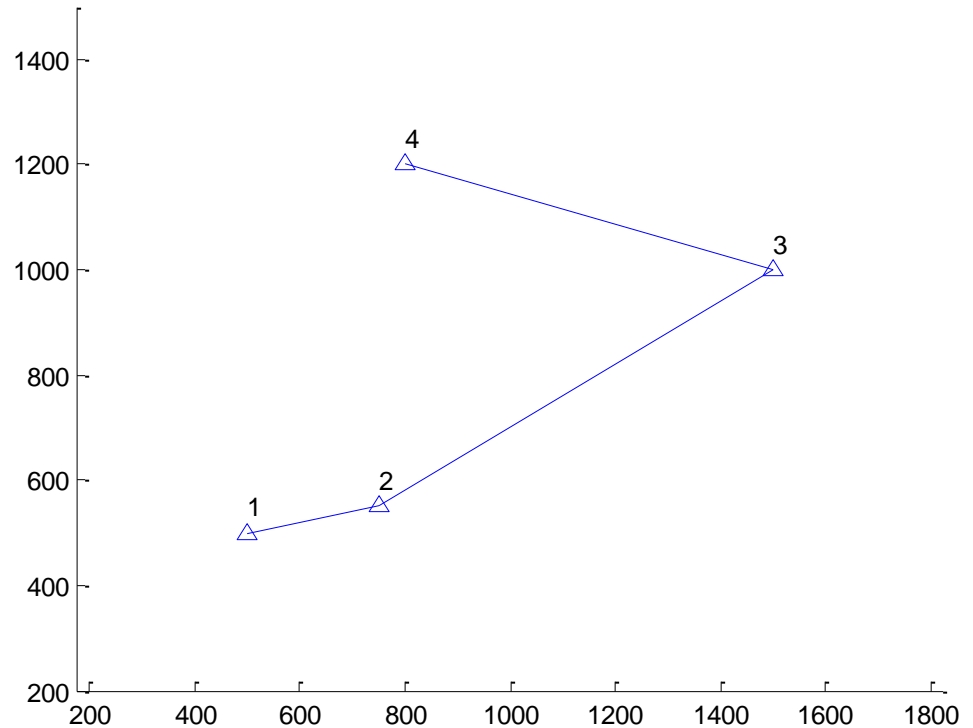


- `axis([Xmin Xmax Ymin Ymax])` fonksiyonu, eksenlerin en küçük ve en büyük değerlerini ayarlar,
- `axis equal` ise x ve y eksenlerindeki ölçek faktörünü (büyüme ve küçülme oranlarını) eşitler.

# MATLAB/Grafik-Line Fonksiyonu

- Koordinatları belirli iki noktayı doğrusal olarak birleştirmek için `line([x1 x2],[y1 y2])` şeklinde kullanılır.

```
clear
clc
ad=[1 2 3 4];
x=[500 550 1000 1200];
y=[500 750 1500 800];
hold on
plot(y,x,'^')
axis([200 1700 200 1500])
axis equal
for i=2:length(x)
    line([y(i-1) y(i)],[x(i-1) x(i)]);
end
for i=1:length(ad)
    text(y(i),(x(i)+50),num2str(ad(i)))
end
hold off
```

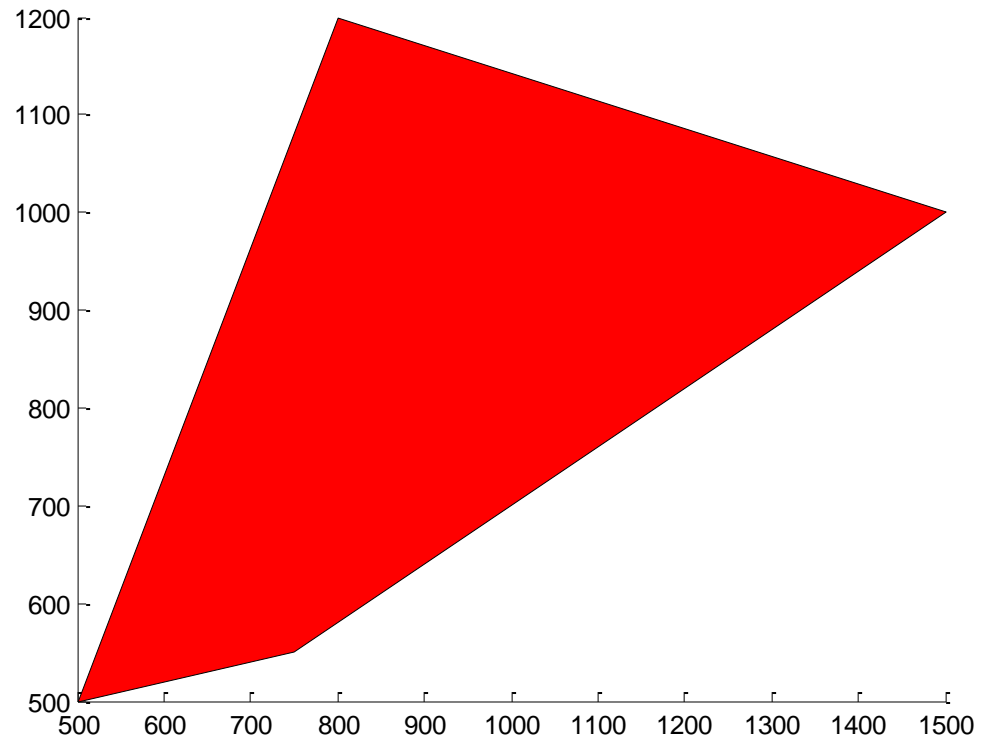


`text(X,Y,'string')`

# MATLAB/Grafik-Patch Fonksiyonu

- Koordinatları belirli alanı doldurmaya yarar. Kullanımı `patch(X,Y,C)` şeklindedir. Burada C taranacak rengi belirler.

```
clear  
clc  
x=[500 550 1000 1200];  
y=[500 750 1500 800];  
patch(y,x,'r')
```



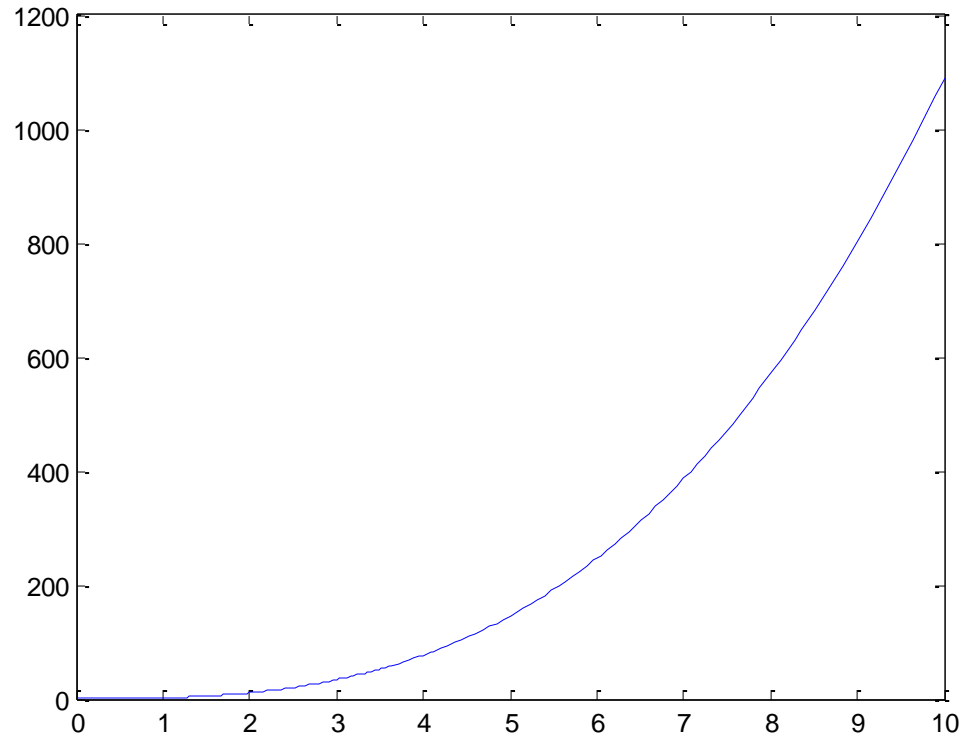
# MATLAB/Grafik-fplot Fonksiyonu

- Belirtilen x ve y aralıklarında karakter dizisi kullanılarak ifade edilen fonksiyonun grafiğini çizdirir. `fplot('fonksiyon',[xmin xmax ymin ymax])` şeklinde kullanılır.

`fplot('x^3+x^2-x+1',[0 10])`



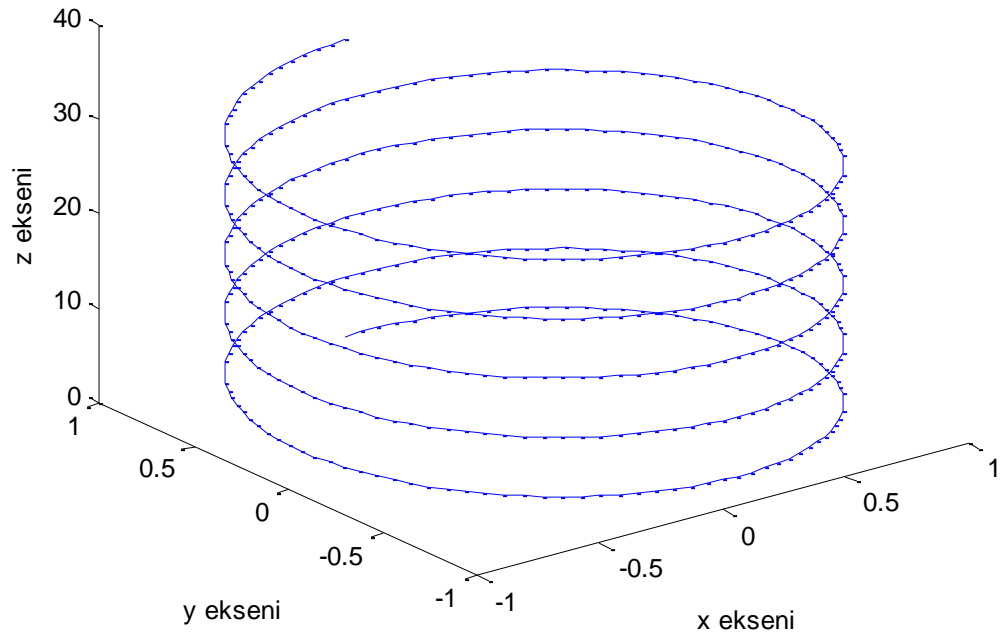
Sınır değer olarak sadece x ekseninin belirtilmesi yeterli olacaktır.



# MATLAB/3B Grafik-plot3 fonksiyonu

- Aynı uzunluktaki üç vektörün birbirlerine göre değişimlerini çizgisel grafiklerle ifade etmemizi sağlar. `plot3(x,y,z)` şeklinde kullanılır.

```
t = 0:pi/50:10*pi;  
plot3(sin(t),cos(t),t);  
xlabel('x eksen');  
ylabel('y eksen');  
zlabel('z eksen');
```

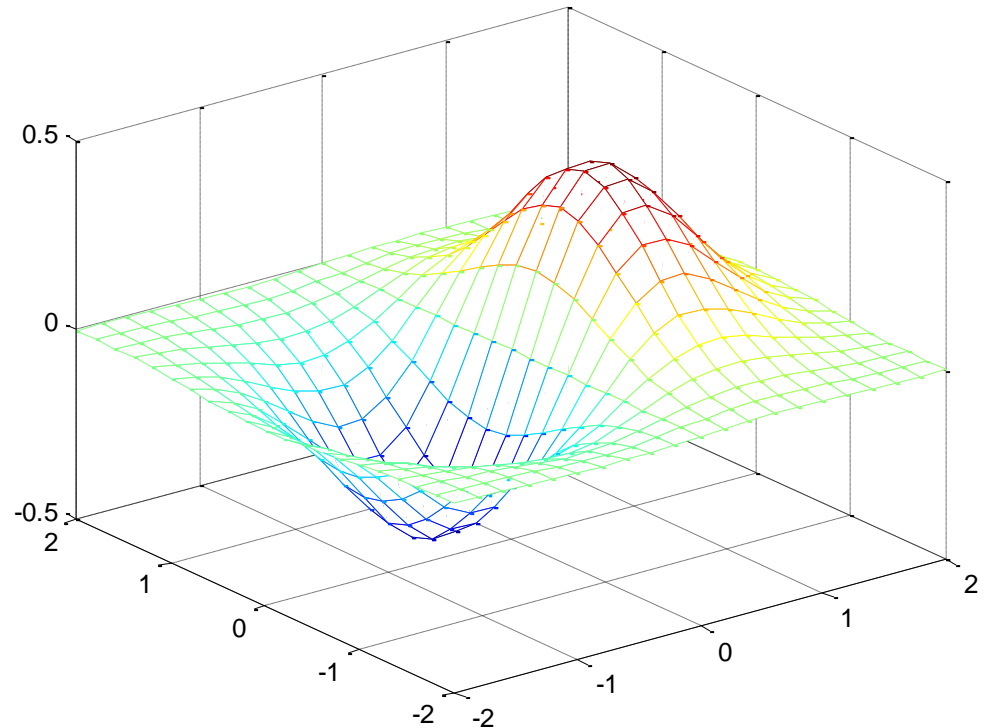


# MATLAB/3B Grafik- meshgrid ve mesh fonksiyonları

- `meshgrid` fonksiyonu  $x$  ve  $y$  vektörleri ile belirtilen alanı,  $X$  ve  $Y$  matrislerine dönüştürür. `[X,Y]=meshgrid(x,y)` şeklinde kullanılır.
- `mesh` fonksiyonu ise  $Z=f(X,Y)$  ile belirli iki değişkenli fonksiyonun belirttiği yüzeyi renkli ağ örgüsü şeklinde çizer. `mesh(X,Y,Z)` şeklinde kullanılır.

$-2 < x < 2$ ,  $-2 < y < 2$ , aralığı için

```
[X,Y] = meshgrid(-2:0.2:2, -2:0.2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
mesh(X,Y,Z)
```



`meshc` ve `meshz` fonksiyonlarını da inceleyelim

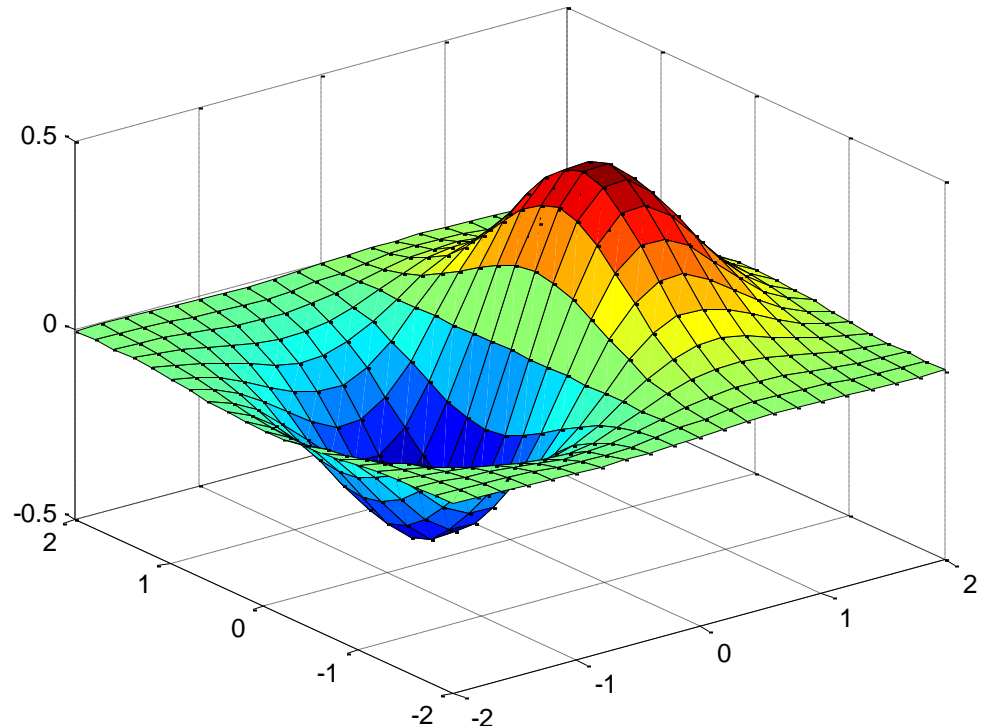


# MATLAB/3B Grafik- surf fonksiyonu

- **surf** fonksiyonu ise  $Z=f(X,Y)$  ile belirli iki değişkenli fonksiyonun belirttiği yüzeyi çizer. **surf(X,Y,Z)** veya **surf(Z)** şeklinde kullanılır.

$-2 < x < 2$ ,  $-2 < y < 2$ , aralığı için

```
[X,Y] = meshgrid(-2:0.2:2, -2:0.2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
surf(X,Y,Z)
```

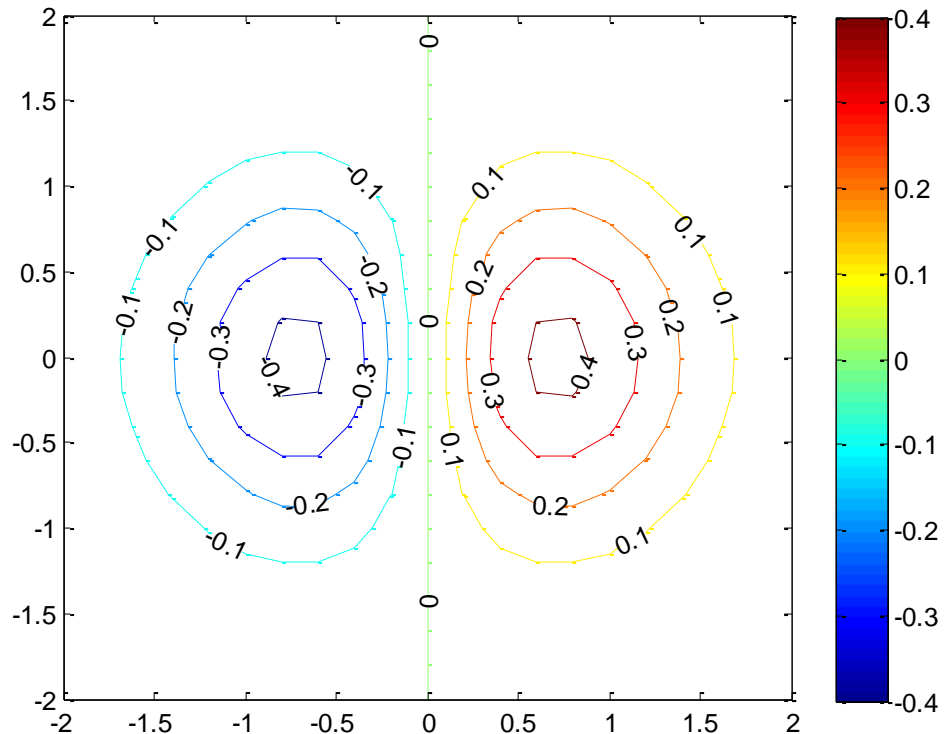


**surf** ve **surfc** fonksiyonlarını da inceleyelim

# MATLAB/3B Grafik- contour fonksiyonu

- Yüzeylere ait eşyüksehti eğrilerinin x-y düzleminde çizdirilebilmesi için **contour** fonksiyonundan yararlanılır. **[C,H]=contour(x,y,z)** şeklinde kullanıldığı durumda eş yüksehti değerlerini [C,H] vektörüne atayacaktır.
- Ayrıca, **clabel(C,H)** komutu ile eşyüksehti değerleri grafik üzerinde gösterilebilir.

```
[X,Y] = meshgrid(-2:0.2:2, -2:0.2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
[C,H]=contour(X,Y,Z);  
clabel(C,H)  
colorbar
```



# MATLAB/Grafik-İnterpolasyon

- x ve y koordinatları bilinen noktalara ilişkin üçüncü bir bilgi (örneğin, *yükseklik*, *yükseklik değişimi*, *sıcaklık*, *nem*, *basınç*, *gelgit deformasyonu*, *anomali* vb.) olduğunda, noktaların çevrelediği alanın içindeki hayali noktalar için bu bilgiler, çeşitli matematiksel yaklaşımlarla üretilebilir: Bu işleme, kısaca, **interpolasyon** denir.
- Böylesi hayali noktalar, alanın içindeki belirli büyüklükteki kareler ağının köşe noktaları olabilir. Bu noktalara grid noktaları denir. Matlab'de, sonraki interpolasyon işlemlerinde kullanılmak üzere, **meshgrid** fonksiyonu ile bu noktaların x-y koordinatları belirlenir.
- Öncelikle, x ve y eksenleri, söz konusu karenin kenar büyüklüğü kadar parçalara ayrılır. Örneğin, kenar büyüklüğü 10 m olsun: Böylece eksenler aşağıdaki biçimde 10 m'lik parçalara bölünür;

```
x1=xmin:10:xmax;   y1=ymin:10:ymax
```

- Daha sonra, **[XI, YI]=meshgrid(x1,y1)** ile gridlerin köşe nokta koordinatları XI ve YI matrislerine yazdırılır.
- **HI=griddata(y,x,H,YI,XI,'v4')** fonksiyonu ile x ve y koordinatlarına sahip **jeodezik** noktalardaki üçüncü bilginin toplandığı H, koordinatları XI ve YI'da tanımlanmış grid noktaları için v4 yöntemiyle interpolate edilir; grid noktalarına ilişkin üçüncü bilgi HI vektöründe toplanır. (**Not:** v4 yönteminden başka, cubic, linear, nearest gibi interpolasyon yöntemleri de bulunur)

# MATLAB/Grafik-İnterpolasyon

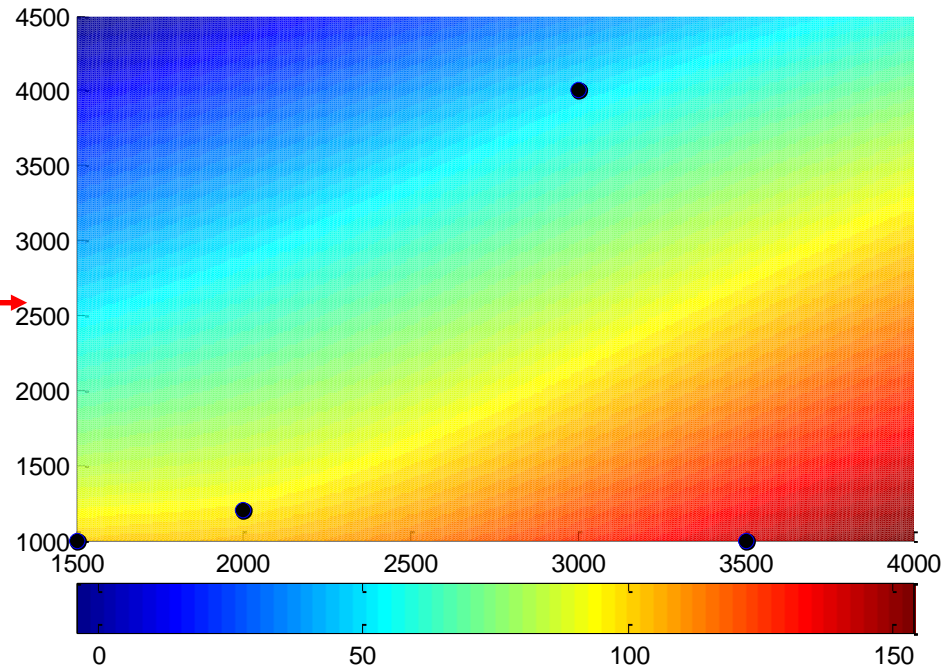
- **Örnek:** Nokta koordinatları,  $\mathbf{x}=[1000;1200;4000;1000]$ ,  $\mathbf{y}=[1500; 2000;3000;3500]$  ile ve bu noktaların yükseklikleri,  $\mathbf{H}=[100.0000;95.9850;50.5000;140.1200]$  ile tanımlansın. Bölgeyi 10 m'lik gridlere bölerek, bölgenin yükseklik değerlerini gösteren bir renk haritası hazırlayınız.

```
clear,clc
x=[1000;1200;4000;1000];
y=[1500;2000;3000;3500];
H=[100.0000;95.9850;50.5000;140.1200];
x1=1000:10:4500;
y1=1500:10:4000;

[XI,YI]=meshgrid(x1,y1);

Hl=griddata(y,x,H,YI,XI,'v4');
hold on,

pcolor(YI,XI,Hl),
shading interp
colorbar('SouthOutside')
plot(y,x,'o','MarkerFacecolor','k')
hold off
```



**Not:** v4 interpolasyonu yerine, diğer interpolasyon yöntemlerini kullanarak aradaki farkları irdeleyiniz.

# MATLAB/Dosya Yazdırma-Okuma

- Program sonuçlarının otomatik olarak farklı bir dosyaya yazdırılması veya bir dosyadaki bilgilerin okunarak program içerisinde kullanılması, programcılıkta oldukça sık başvurulan çıktı alma ve veri girişi yöntemleridir.
- Dosya yazdırma, çıktı almaya; Dosya okuma ise veri girişine karşılık olan işlemlerdir.
- Matlab'de dosya yazdırma, en basit biçimde, **diary** komutuyla gerçekleştirilir. Kullanımı ise aşağıdaki biçimdedir;

```
a=10;  
diary sonuc.txt  
    disp('-----')  
    disp(a)  
diary end
```

- İki diary komutu arasındaki **"command window" da yazdırılacak her türlü bilgi**, sonuc.txt dosyasına yazdırılır. sonuc.txt dosyası, mevcut klasörün içinde oluşturulur,
- Örnekteki, sonuc.txt dosyası yerine başka dosya tür ve isimleri kullanılabilir.
- Yazdırılacak olan dosya, daha önce oluşturulmuş bir dosya ise, çıktı dosyanın içindeki metnin altına yazdırılır.

# MATLAB/Dosya Yazdırma-Okuma

- Daha gelişmiş dosya yazdırma, `fopen`, `fprintf` ve `fclose` fonksiyonlarının kullanımı ile gerçekleştirilir.
- Bu fonksiyonlar ile dosya yazdırmada, yazdırılacak olan metnin “command window” da gösterilmesine gerek yoktur.
- `fopen`, program çıktılarının yazdırılacağı dosyayı açar, `fprintf` yazdırır ve `fclose` ise yazdırma işlemini sonlandırır.

```
ifade=fopen('dosya_adi', 'izin')
fprintf(ifade,'aciklama',değişken)
fclose(ifade)
```

- Örneğin, bir a kenarı programda hesaplatılmış olsun. Bu programın a çıktısını, kenar.txt isimli bir dosyaya yazdırmak için, aşağıdaki kodlar düşünülür;

```
a=150.0234234;
fid=fopen('kenar.txt','w');
fprintf(fid,'kenar uzunluğu=%1.4f',a);
fclose(fid);
```

→ `w` bu dosyanın üzerine yazılacağını gösterir.

# MATLAB/Dosya Yazdırma-Okuma

## Dosya yazdırma ve okumada kullanılan izinler

Tip	Açıklama
'r'	Dosyayı sadece okumaya açar. (Yazma işlemine izin vermez)
'r+'	Dosyayı yazmaya ve okumaya açar.
'w'	Var olan bir veri dosyası içindeki bilgileri siler, dosya yoksa oluşturur ve dosyayı yazmaya açar.
'w+'	Var olan bir veri dosyası içindeki bilgileri siler, dosya yoksa oluşturur ve dosyayı okumaya ve yazmaya açar.
'a'	Var olan bir veri dosyasını yazmak için açar, dosya yoksa oluşturur ve girilecek bilgileri dosya sonuna ekler.

# MATLAB/Dosya Yazdırma-Okuma

- **Örnek:** a=[3.12356 4.12456 1;5.8463 6.45111 2;4 5 6] biçiminde verilen bir a matrisini, elemanları virgülden sonra 4 hane olacak biçimde, mat.out dosyasına yazdıran bir program yazınız.

```
a=[3.12356 4.12456 1;5.8463 6.45111 2;4 5 6]
fid = fopen('mat.out','w');
fprintf(fid,'%1.4f%10.4f%10.4f\n',a);
fclose(fid);
```

- **Örnek:** kenar=1500.123 m ve aciklik=103.3367 grad olan değişkenleri, sonuc.out dosyasına alt alta yazdırınız.

```
kenar=1500.123;
aciklik=103.3367;
fid=fopen('sonuc.out','w');
fprintf(fid,'kenar=%1.3f m\n',kenar);
fprintf(fid,'aciklik=%1.4f grad',aciklik);
fclose(fid)
```



# MATLAB/Dosya Yazdırma-Okuma

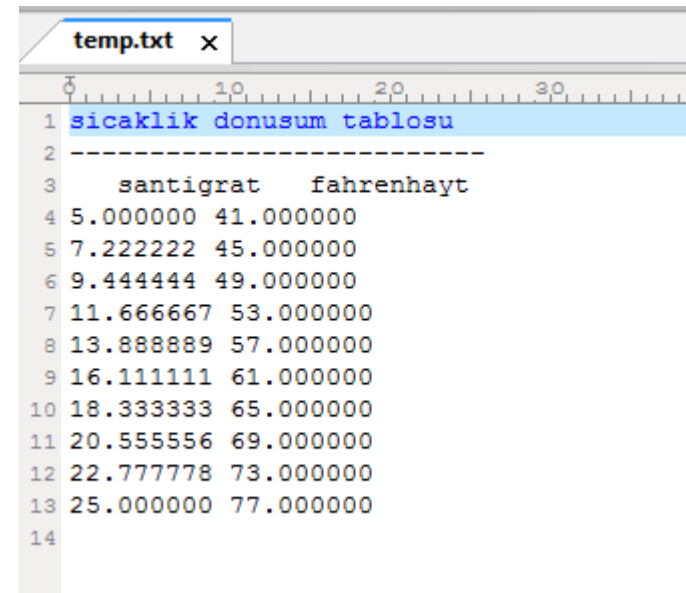
- Fahrenheit ve santigrat değerleri arasında istenilen bir aralıkta dönüşüm yapan ve sonuçları .txt dosyasına yazdıran matlab kodunu yazınız.
- fahrenheit=1.8\*santigrat+32;**

```
Tbasla=input('Ilk sicaklik degerinin yaziniz:');
Tson=input('Son sicaklik degerinin yaziniz:');
nTemp=input('Kac deger istediginizi giriniz:');

santigrat=linspace(Tbasla,Tson,nTemp);

fahrenheit=1.8*santigrat+32;

fid=fopen('temp.txt','w+');
fprintf(fid,'sicaklik donusum tablosu\n');
fprintf(fid,'-----\n');
fprintf(fid,' santigrat  fahrenheit\n');
for k=1:nTemp
    fprintf(fid, '%f %f \n', santigrat(k), fahrenheit(k) );
end
fclose(fid);
```



```
temp.txt x
1  sicaklik donusum tablosu
2  -----
3      santigrat  fahrenheit
4  5.000000  41.000000
5  7.222222  45.000000
6  9.444444  49.000000
7  11.666667  53.000000
8  13.888889  57.000000
9  16.111111  61.000000
10 18.333333  65.000000
11 20.555556  69.000000
12 22.777778  73.000000
13 25.000000  77.000000
14
```

# MATLAB/Dosya Yazdırma-Okuma

- Veri dosyalarının okunması amacı ile `fscanf` fonksiyonu kullanılabilir.
- `[dizi,sayi]=fscanf(ifade,'format',alan)` şeklinde kullanımı vardır.
- Bir önceki uygulamada `mat.out` dosyasına yazdırdığımız matrisi tekrar Matlab ortamında okutalım.

```
fid=fopen('mat.out','r+');  
[dizi,sayi]=fscanf(fid,'%f',inf)
```

dizi =

3.1236  
5.8463  
4.0000  
4.1246  
6.4511  
5.0000  
1.0000  
2.0000  
6.0000

sayi =

9

```
fid=fopen('mat.out','r+');  
[dizi,sayi]=fscanf(fid,'%f',[3 3])
```

dizi =

3.1236	4.1246	1.0000
5.8463	6.4511	2.0000
4.0000	5.0000	6.0000

sayi =

9

# MATLAB/Dosya Yazdırma-Okuma

- Matlab'de dosyaların içindeki kolon yapısındaki metinlerin okunması için `textread` fonksiyonu bulunmaktadır. Örneğin, aşağıda koordinat.txt dosyasındaki verilerin okunması istensin:

P1	1000.1234	1300.23423
P2	1300.5673	1450.98563
P3	2000.1500	2000.11000
P4	3500.3100	1000.12000

Nokta isimleri    x koordinatları    y koordinatları

- Bunun için,

```
[nokta,x,y]=textread('koordinat.txt','%s%f%f')
```

fonksiyonu kullanılır. `nokta`, nokta isimlerini içeren bir hücre dizisi; `x`, x koordinat vektörü ve `y`, y koordinat vektörü olarak atanır.

```
[a, b, c,...]=textread('dosya_adi','format')
```

# MATLAB/Dosya Yazdırma-Okuma

- **Örnek:** Koordinat dosyası, aşağıdaki gibi olan bir koordinat.txt dosyasından, nokta isimlerini, x ve y koordinatlarını textread fonksiyonu kullanarak uygun değişkenlere atayınız.

Nirengi koordinatları		
NN	x (m)	y (m)
P1	1000.1234	1300.23423
P2	1300.5673	1450.98563
P3	2000.1500	2000.11000
P4	3500.3100	1000.12000

[nokta,x,y]=textread('koordinat.txt','%s%f%f','headerlines',2)

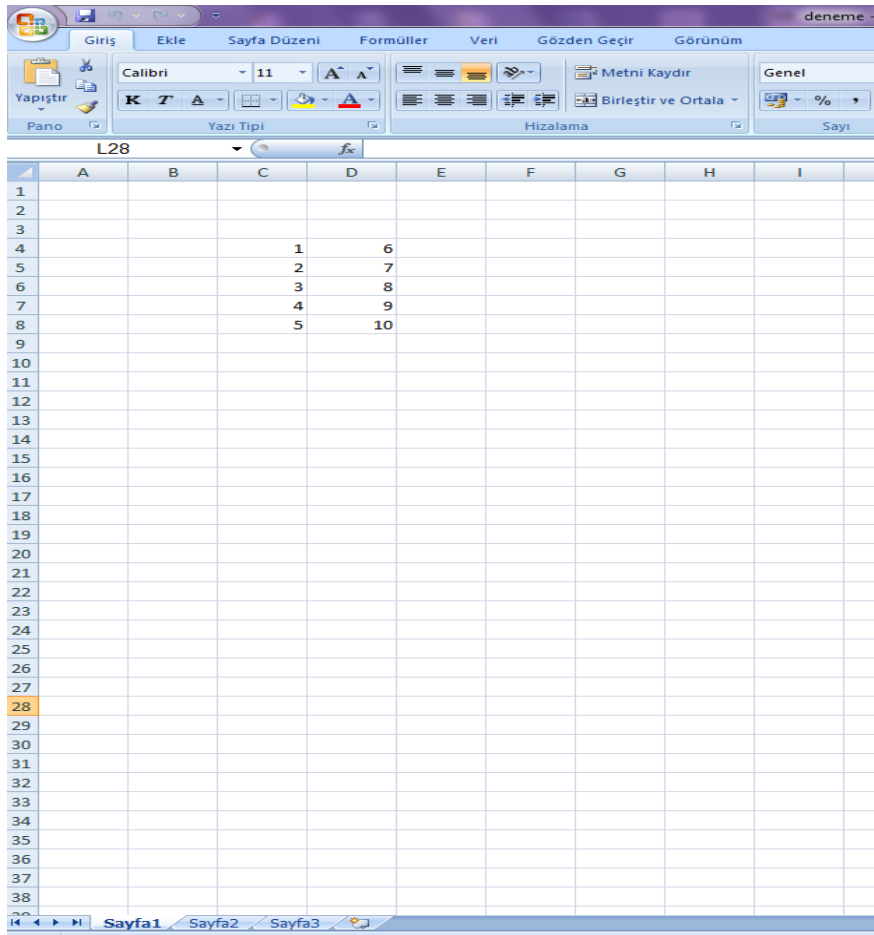
'headerlines' komutu ve ardından gelen sayı, dosyanın başlangıcından itibaren kaç tane satırın dikkate alınmayacağını gösterir.

koordinat.txt dosyasında ilk iki satır alınmadan nokta isimleri, x ve y koordinatları okunmuştur.

# MATLAB/Dosya Yazdırma-Okuma

- Excel' den veri okutmak amacıyla `xlsread` fonksiyonu kullanılır.

```
num = xlsread('filename', sheet, 'range')
```



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4			1	6					
5			2	7					
6			3	8					
7			4	9					
8			5	10					
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									

```
A = xlsread('deneme.xlsx', 1, 'C4:D7')
```

A =

1	6
2	7
3	8
4	9

# MATLAB/Fonksiyon Dosyası Oluşturma

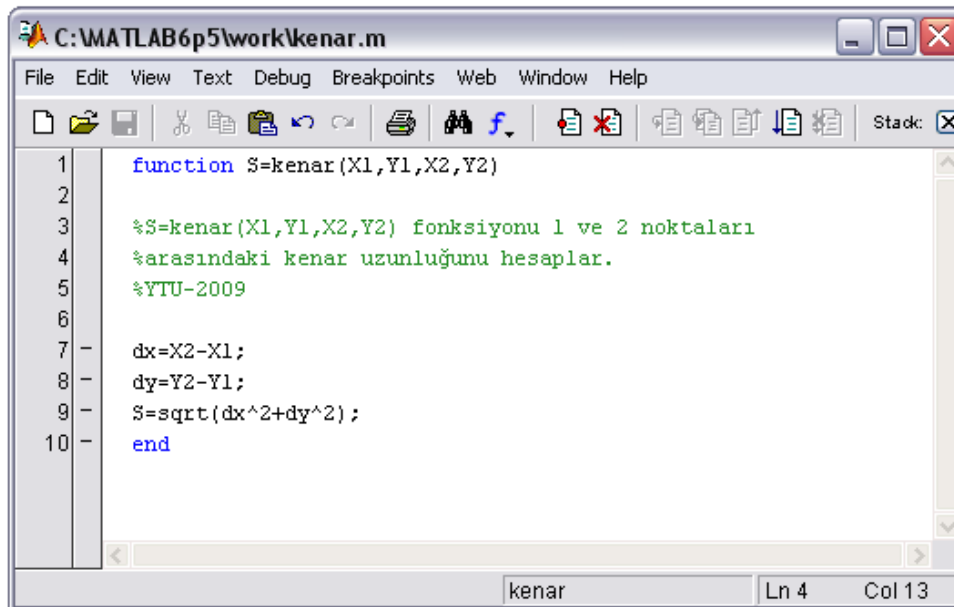
- Matlab fonksiyonları (örneğin, inv, disp, num2str,det, textread...) kullanıcının bir başka programa gerek duymaksızın temel işlemleri kolayca yapabilmesini sağlar.
- Kullanıcılar kendi fonksiyonlarını geliştirebilirler.
- Fonksiyonlar, bir m-dosyası biçiminde saklanır ve bu dosyalara, fonksiyon dosyaları adı verilir.
- Fonksiyon oluşturma'nın iki önemli getirisi vardır:

**Sürekli olarak uygulanan bir işlem için kod tekrarını önler (örneğin, açıklık açısı için yazılmış bir fonksiyon `aciklik` ise, programın ilgili yerinde “`aciklik(x1 ,y1 ,x2 ,y2)`” fonksiyonu (1-2) veya (2-1) `aciklik` acisini doğrudan üretecektir.**

**Fonksiyonlarda kullanılan değişkenler yereldir (local variables). Yani workspace içinde diğer program türlerinin değişkenleri (global variables) gibi yer işgal etmezler.**

# MATLAB/Fonksiyon Dosyası Oluşturma

- **Örnek:** Koordinatları bilinen iki nokta arasındaki yatay uzunluğu hesaplayan kenar isimli bir fonksiyon oluşturunuz.



The image shows a MATLAB Editor window titled 'C:\MATLAB6p5\work\kenar.m'. The window contains the following code:

```
1 function S=kenar(X1,Y1,X2,Y2)
2
3 %S=kenar(X1,Y1,X2,Y2) fonksiyonu 1 ve 2 noktaları
4 %arasındaki kenar uzunluğunu hesaplar.
5 %YTU-2009
6
7 dx=X2-X1;
8 dy=Y2-Y1;
9 S=sqrt(dx^2+dy^2);
10 end
```

The status bar at the bottom indicates the cursor is at line 4, column 13.

- Fonksiyonların, biçim olarak, diğer programlardan tek farkı,  
  
`function output=fonk_ismi(input)`  
  
ile başlaması ve fonksiyon dosyasının sonunda `end` ile bitmesidir.
- function komutunun bulunduğu ilk satırdan hemen sonra gelen açıklama (comment) satırları, ilgili fonksiyonun “yardım” metinleridir.
- Fonksiyon ismiyle, fonksiyon dosyasının ismi aynı olmalıdır.

# MATLAB/Fonksiyon Dosyası Oluşturma

- **Örnek:** Hem açıklık açısını hem de kenar uzunluğunu üreten **aci\_kenar** isimli bir fonksiyon oluşturunuz.

```
function [ a,S ] = aci_kenar( X1,Y1,X2,Y2)
%[a,S]=aci_kenar(X1,Y1,X2,Y2) fonksiyonu (1-2) açıklık açısını
%(a) ve 1-2 kenar uzunlugunu hesaplar

DX=X2-X1;DY=Y2-Y1;

if (DX~=0)&(DY~=0),a=atan(DY/DX);a=a*200/pi;
    if (DX>0)&(DY>0),a=a;end
    if (DX<0)&(DY>0),a=a+200;end
    if (DX<0)&(DY<0),a=a+200;end
    if (DX>0)&(DY<0),a=a+400;end
end

if (DX==0)&(DY>0),a=100;end
if (DX==0)&(DY<0),a=300;end
if (DX>=0)&(DY==0),a=0;end
if (DX<0)&(DY==0),a=200;end

S=sqrt(DX^2+DY^2);%kenar

end
```

- Bir fonksiyonun birden fazla çıktısı olabilir. Bu örnekte a ve S gibi iki çıktı bulunmaktadır.
- a, açıklık açısını, S ise kenar uzunluğunu göstermektedir.
- aci\_kenar(X1,Y1,X2,Y2) komutuyla, ilk output, yani açıklık açısını belirten a değişkeni üretilir.



# MATLAB/ Derleyici (Compiler)

- C, Pascal, Basic gibi yüksek veya orta düzey dillerde yazılan bir programın çalışabilmesi için önce makine diline çevrilmesi gerekir; bu işleme derleme (compilation) denir. Derleme işini yapan programlar da derleyici (compiler) olarak adlandırılır.
- MATLAB'de sadece m-fonksiyon veya grafik kullanıcı arayüzü (GUI) şeklindeki programlar derlenebilmektedir.
- MATLAB derleyicisi (mcc) ile MATLAB'de oluşturulan m-fonksiyon dosyaları C/C++ dosyalarına, .c dosyalar da MATLAB formatına dönüştürülebilir.

**>> mcc -m dosya\_adi**

yazımı ile dosya\_adi.m adlı dosya kendi başına çalışabilen dosya\_adi.exe haline ve C koduna dönüştürülür.

**>> mcc -p dosya\_adi**

yazımı ile dosya\_adi.m adlı dosya kendi başına çalışabilen dosya\_adi.exe haline ve C++ koduna dönüştürülür.

# MATLAB/ Derleyici (Compiler)

- Klavyeden girilen iki sayının toplamını hesaplayan programın derlenerek .exe file haline getirilmesi

%iki sayinin toplamini hesaplayan program

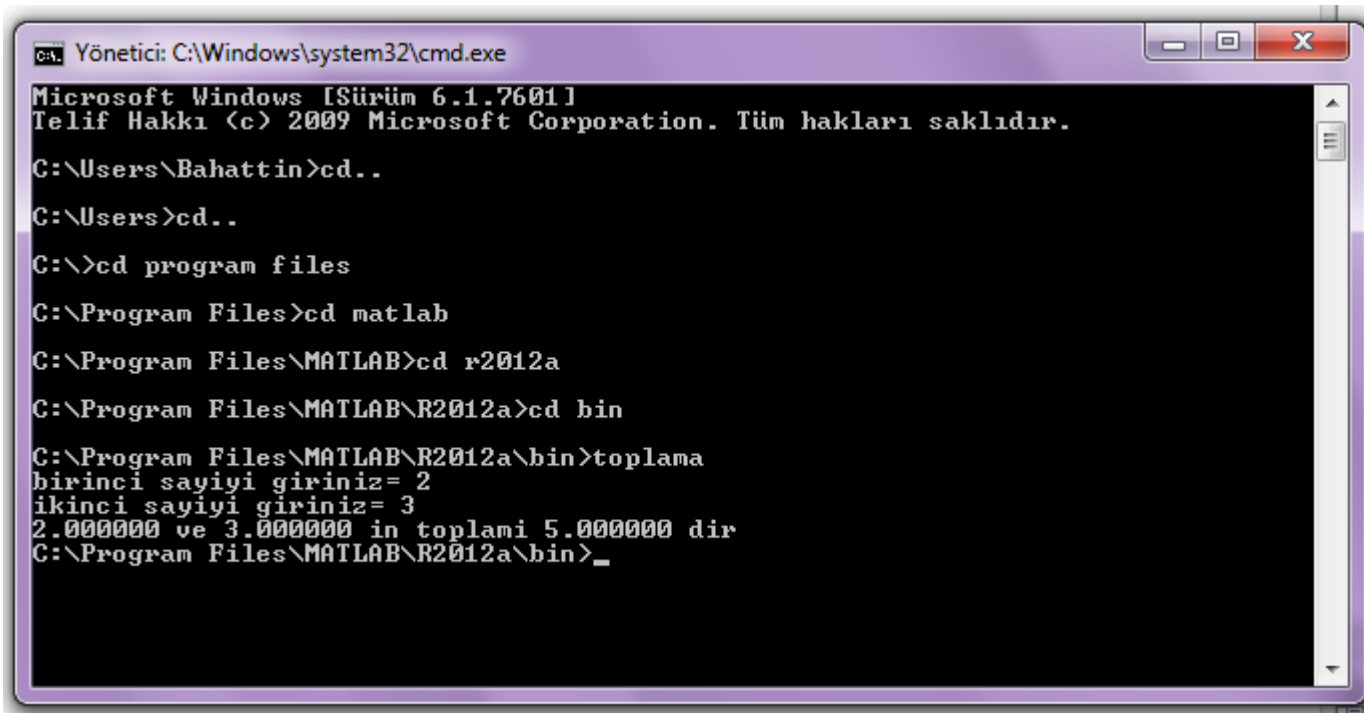
a=input('birinci sayiyi giriniz= ');

b=input('ikinci sayiyi giriniz= ');

c=a+b;

fprintf('%f ve %f in toplami %f dir',a,b,c)

- >>mcc -m toplama (komut satırına yazılarak exe dosyası oluşturulur)

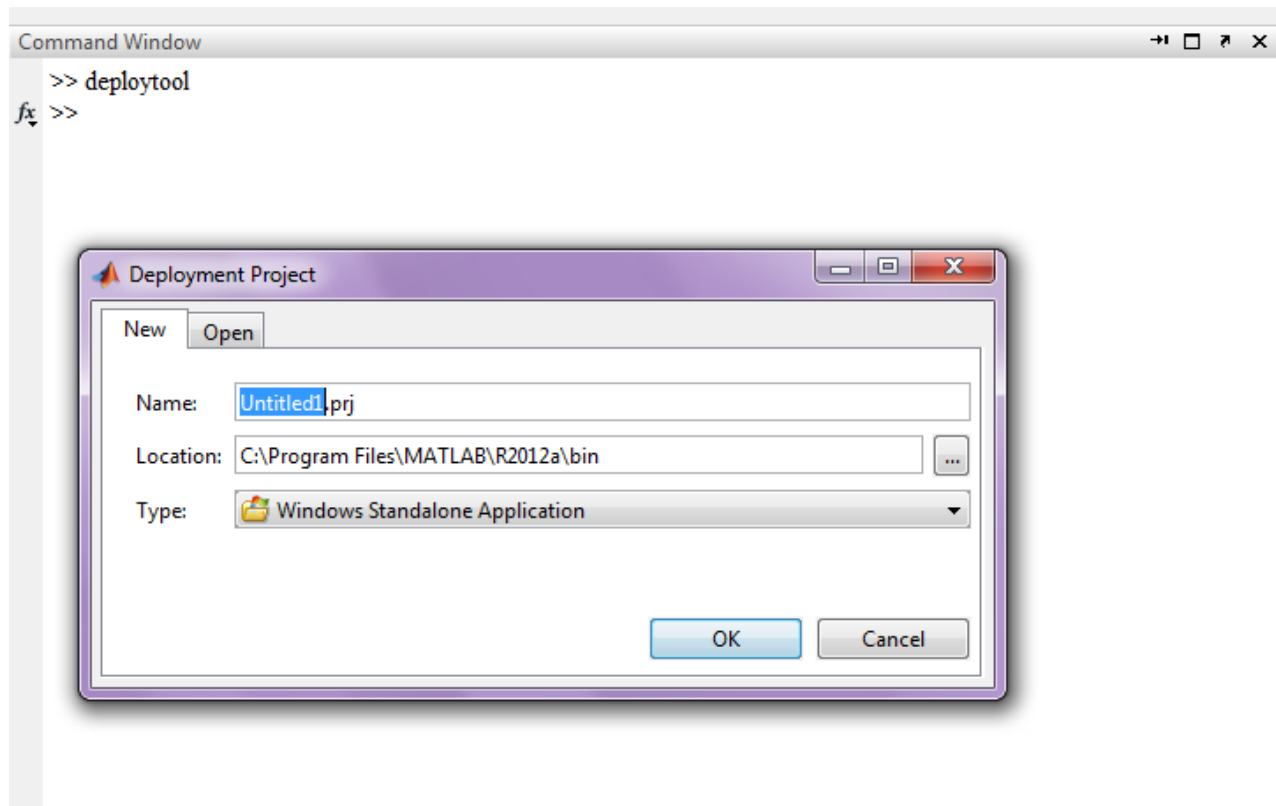


```
Yönetici: C:\Windows\system32\cmd.exe
Microsoft Windows [Sürüm 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.

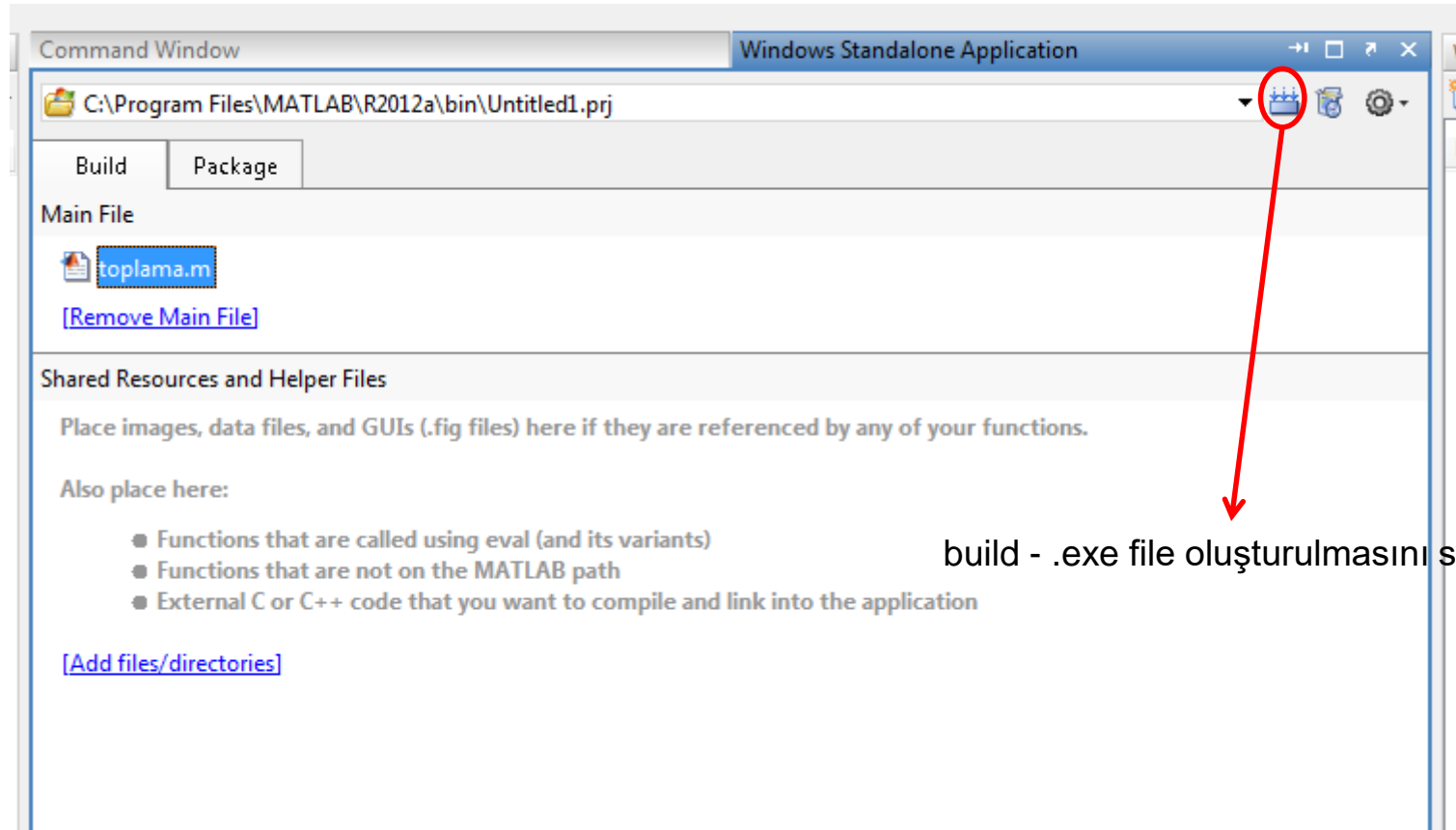
C:\Users\Bahattin>cd..
C:\Users>cd..
C:\>cd program files
C:\Program Files>cd matlab
C:\Program Files\MATLAB>cd r2012a
C:\Program Files\MATLAB\R2012a>cd bin
C:\Program Files\MATLAB\R2012a\bin>toplama
birinci sayiyi giriniz= 2
ikinci sayiyi giriniz= 3
2.000000 ve 3.000000 in toplami 5.000000 dir
C:\Program Files\MATLAB\R2012a\bin>_
```

# MATLAB/ Derleyici (deploytool)

- Oluşturulan .exe dosyasının MATLAB kurulu olmayan bir bilgisayarda çalışabilmesi için **MATLAB Compiler Runtime (MCR)** programının kurulması gerekir. MCR farklı MATLAB versiyonları için değişiklik gösterir. exe dosyası hangi MATLAB versiyonunda derlendiyse ilgili MCR bilgisayarda kurulmalıdır. Kullanıcıya .exe uzantılı dosyanın çalışabilmesi için gerekli tüm dosyaların verilmesi gerekir. Bu amaçla **deploytool** fonksiyonu kullanılabilir.

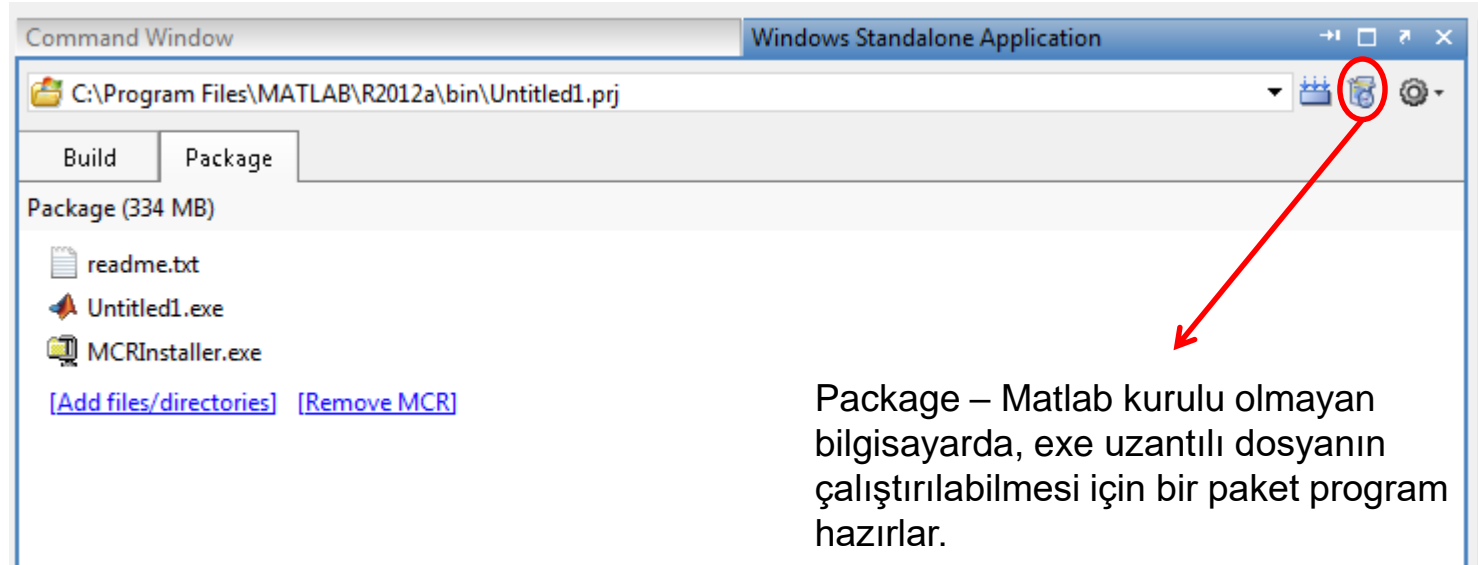


# MATLAB/ Derleyici (deploytool)



build - .exe file oluşturulmasını sağlar.

# MATLAB/ Derleyici (deploytool)



Package – Matlab kurulu olmayan bilgisayarda, exe uzantılı dosyanın çalıştırılabilmesi için bir paket program hazırlar.

Sonuç olarak ....**pkg.exe** uzantılı bir dosya oluşturulur. Bu dosya çalıştırıldığında bilgisayarda MATLAB' de derlenen .exe uzantılı programın çalıştırılabilmesi için gerekli programı kurar.

# MATLAB/Uygulama-15

Aşağıda bir kenara ait 15 ölçü kenar.txt dosyasında verilmektedir. Bu ölçülerin ortalamasını, her bir ölçünün ortalamadan farkını (düzeltmeleri), ölçülerin standart sapmasını hesaplayan,  $|düzeltme| > 3 \times \text{standart sapma}$  olan ölçüleri ölçü kümesinden silen ve geriye kalan ölçüleri temizolcu.txt dosyasına yazdıran matlab kodunu yazınız.

## Olculer

15.538  
16.834  
12.741  
15.862  
15.319  
13.692  
14.566  
15.343  
18.578  
17.769  
13.650  
18.035  
25.725  
14.937  
15.715

**Duzeltme=ortalama – ölçü**

**Standart sapma= $([duzeltme^2]/(\text{ölçü sayisi}-1))^{(1/2)}$**

# MATLAB/Uygulama-15

```
clear
clc
a=textread('kenar.txt','%f','headerlines',3); %textread fonksiyonu ile veriler okutuluyor.
orta=mean(a); %mean fonksiyonu ile ölçülerin ortalamasi bulunuyor.

for i=1:length(a) % for döngüsüyle duzeltmelerin hesaplanmasi yapiliyor.
    duzeltme(i,1)=orta-a(i);
end
stan=sqrt(duzeltme*duzeltme/(length(a)-1)); %ölçülerin standart sapmasi hesaplaniyor.

hata=0;
artim=0;
for i=1:length(duzeltme) %duzeltmesi standart sapmasindan büyük ölçüler bulunuyor.
    if abs(duzeltme(i))>3*stan
        artim=artim+1;
        hata(artim)=i;
    end
end

hata=sort(hata)
for i=1:artim % hatali olcu siliniyor.
    a(hata(artim+1-i),:)=[];
end

veri=fopen('temizolcu.txt','w+') % temiz olculer yazdiriliyor.
fprintf(veri,'%1.3f\n',a)
fclose(veri)
```

# MATLAB/Uygulama-16

Küre üzerinde açı değeri olarak verilen yay değerini uzunluğa çeviren matlab kodunu fonksiyon dosyası şeklinde yazınız.

```
function [ S] = yaykenari(aci,R,ro )  
%[ S] = yaykenari(aci,R,ro ) fonksiyonu acisal olarak  
%girilen yay deđerinin uzunlugunu  
hesaplamaktadır.  
% aci ifadesi hesap yapılacak aci deđerini, R  
deđerı kürenin yaricapini,  
% ro deđerı 180/pi veya 200/pi deđerini ifade eder.  
  
S=aci/ro*R;  
  
end
```



# MATLAB/Uygulama-17

Dik kenarları verilen bir dik üçgende hipotenüsü ve alanı hesaplayan matlab kodunu fonksiyon dosyası şeklinde yazınız.

```
function [ hipo,alan ] = hipoalan( a,b )  
%[ hipo,alan ] = hipoalan( a,b ) fonksiyonu bir dik üçgende dik kenarlar  
%girildiginde hipotenüsü ve alanı hesaplar.  
  
hipo=sqrt(a^2+b^2);  
alan=a*b/2;  
  
end
```

# MATLAB/Uygulama-18

Bir sınıfta 20 öğrenci ve bu öğrencilerin yaş dağılımlarının 18, 19, 20 olduğu bilinmektedir. Klavyeden giriş yapılarak her bir yaş grubunda kaç öğrenci olduğunu hesaplayan ve bar olarak çizdiren matlab kodunu yazınız.

```
clear, clc
say18=0;say19=0;say20=0;
ogrencisayisi=20;
sayac=0;
while sayac<ogrencisayisi
    osay=input('ogrenci yasini giriniz: ');
    if osay==18
        say18=say18+1;
    end
    if osay==19
        say19=say19+1;
    end
    if osay==20
        say20=say20+1;
    end
    sayac=sayac+1;
    disp([num2str(sayac) 'ogrenci girisi yapildi'])
end

bar([18 19 20],[say18 say19 say20])
xlabel('ogrenci yaslari')
ylabel('ogrenci toplam sayilari')
```

# MATLAB/Uygulama-19

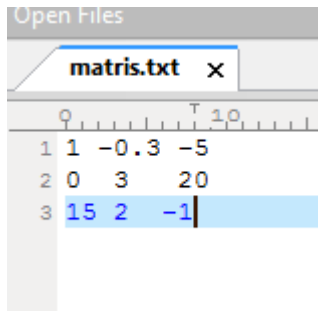
İki değişkenli  $f(x,y)$  fonksiyonu tanımlanmıştır. Kullanıcıdan  $x$  ve  $y$  değerlerini isteyerek fonksiyonun alacağı değeri bulan matlab kodunu yazınız.

$$f(x,y) = \begin{cases} \frac{x}{y} & x \geq 0 \text{ ve } y \geq 0 \\ \frac{x}{5} + y & x \geq 0 \text{ ve } y < 0 \\ x + \frac{y}{5} & x < 0 \text{ ve } y \geq 0 \\ \frac{x}{5} + \frac{y}{5} & x < 0 \text{ ve } y < 0 \end{cases}$$

```
clear
clc
x=input('x degerini giriniz= ');
y=input('y degerini giriniz= ');
if x>=0 & y>=0
    f=x/y;
elseif x>=0 & y<0
    f=x/5+y;
elseif x<0 & y>=0
    f=x+y/5;
elseif x<0 & y<0
    f=x/5+y/5;
end
fprintf('%1.4f ve %1.4f degerlerine gore fonksiyon %1.4f degerini almaktadır.\n',x,y,f)
```

# MATLAB/Uygulama-20

matris.txt dosyasında kayıtlı olan matris veya vektörün negatif ve pozitif elemanlarının sayısını veren matlab kodunu yazınız.



The screenshot shows a text editor window titled 'matris.txt'. It contains a 3x3 matrix with the following values:

1	1	-0.3	-5
2	0	3	20
3	15	2	-1

```
clear
clc
veri=fopen('matris.txt','r+');
a=fscanf(veri,'%f',[3 3]);
fclose(veri);
[nsatir nsutun]=size(a);
pozitif=0;
negatif=0;
for i=1:nsatir
    for j=1:nsutun
        if a(i,j)<0
            negatif=negatif+1;
        else
            pozitif=pozitif+1;
        end
    end
end
```

# MATLAB/Uygulama-21

Klavyeden girilen bir sinir değerine göre 0' dan farklı kaç tane ardışık tamsayı değerinin eşit veya daha düşük bir değere ulaşacağını bulan matlab kodunu yazınız. Ayrıca daha düşük bir değere ulaşıyorsa bu değeri de hesaplayınız.

```
clear
clc
sinir=input('sinir degeri giriniz= ');
sayac=0;
toplam=0;
while toplam <=sinir
    sayac=sayac+1;
    toplam=toplam+sayac;
end

if toplam==sinir
    ulasilan=sinir;
    sonsayac=sayac-1;
elseif toplam>sinir
    ulasilan=toplam-sayac;
    sonsayac=sayac-1;
end
fprintf('%d sayisina %d adet ardisik tamsayinin toplami sonucunda ulasilabilmektedir\n',ulasilan,sonsayac)
```