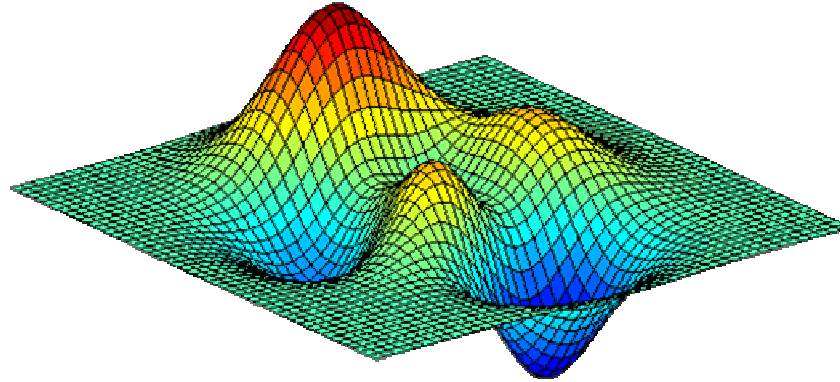

Yeni Başlayanlar için

MATLAB

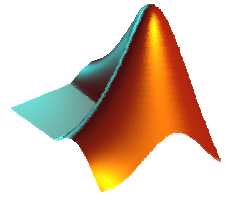
Yardımcı Ders Notları



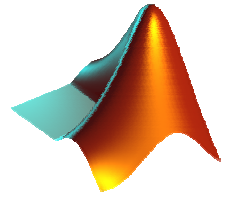
Yrd. Doç. Dr. Cüneyt AYDIN

Yıldız Teknik Üniversitesi

İstanbul-2009

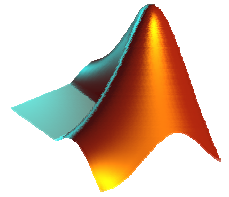


- ④ Giriş
- ④ Matris işlemleri
- ④ Sayı Formatları
- ④ Temel Lineer Cebir İşlemleri
- ④ Diziler (Arrays)
- ④ Programı Dallandıran İfadeler (if-end , switch-case yapıları)
- ④ Döngüler (for-end ve while-end döngüleri)
- ④ Grafik
- ④ Dosya yazdırma-okuma
- ④ Fonksiyon dosyası oluşturma



- @ Doğan, U., (2009), Temel Bilgisayar Bilimleri Ders Notları, YTÜ, Lisans Ders Notları, İstanbul.
- @ Demirel, H., (2005), Dengeleme Hesabı, YTÜ, Lisans Ders Notları, İstanbul.
- @ Uzunoğlu M., vd. (2002), Matlab, Türkmen Kitabevi, İstanbul.
- @ <http://www.mathworks.com/matlabcentral/>
- @ <http://www.mathworks.com/matlabcentral/fileexchange/>

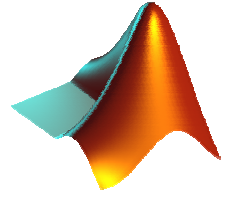
MATLAB (MATrix LABoratory)



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

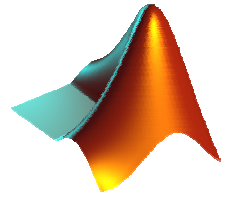
MATLAB, yüksek performanslı bir uygulama yazılımı ve bir programlama dilidir.

- ⊗ 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'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).

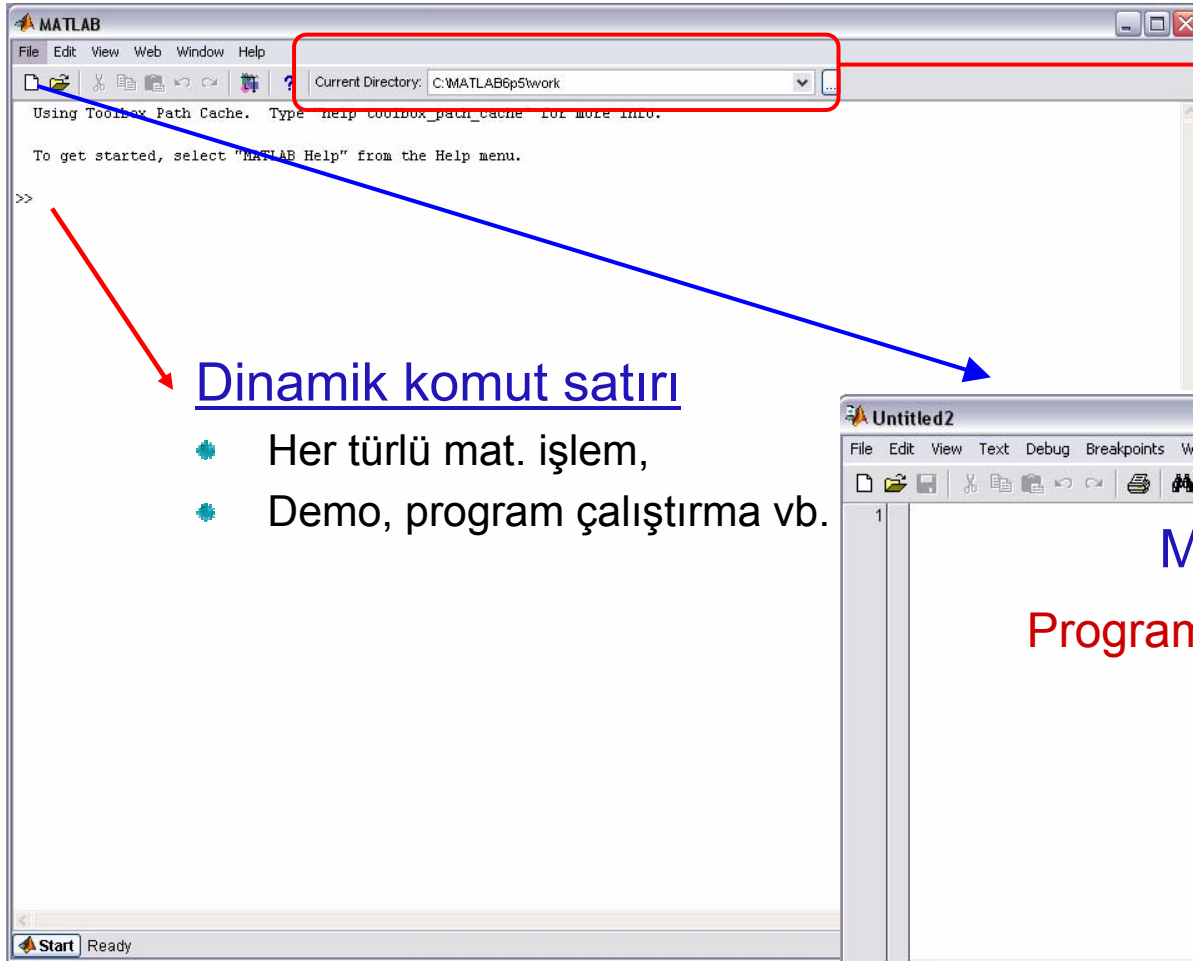
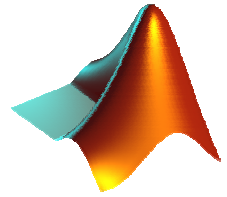
MATLAB (MATrix LABoratory)



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!

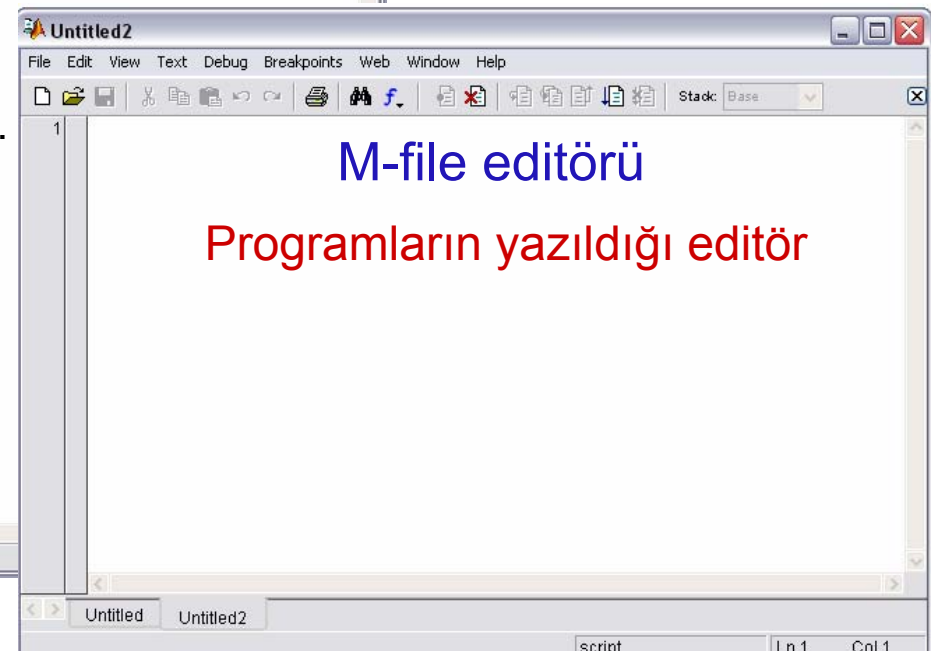
MATLAB/Command window (komut penceresi)



Current directory
(çalışma klasörü)

Dinamik komut satırı

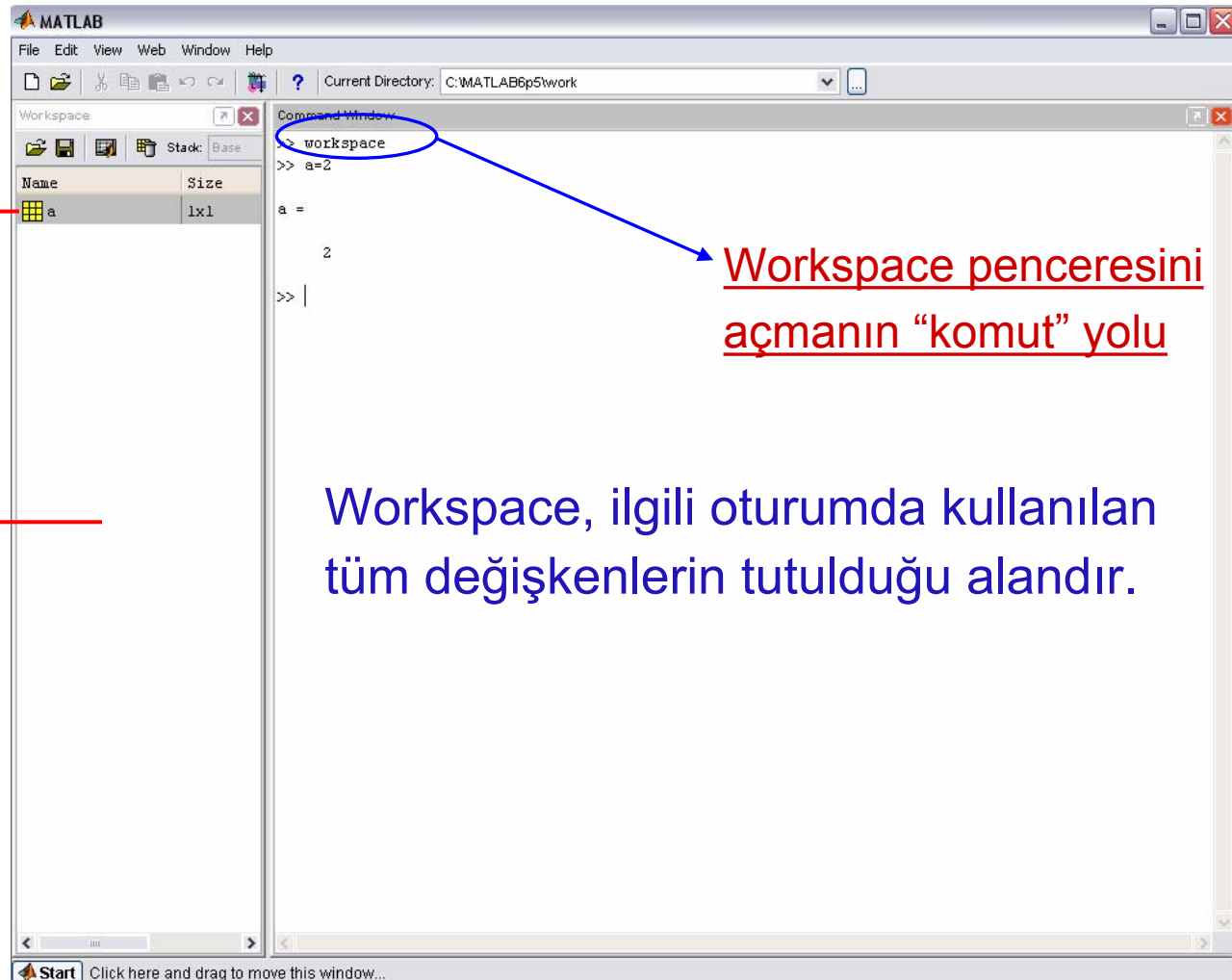
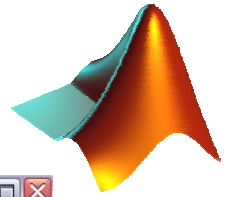
- Her türlü mat. işlem,
- Demo, program çalıştırma vb.



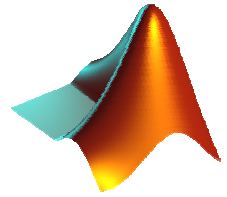
M-file editörü

Programların yazıldığı editör

MATLAB/Workspace (İş alanı)



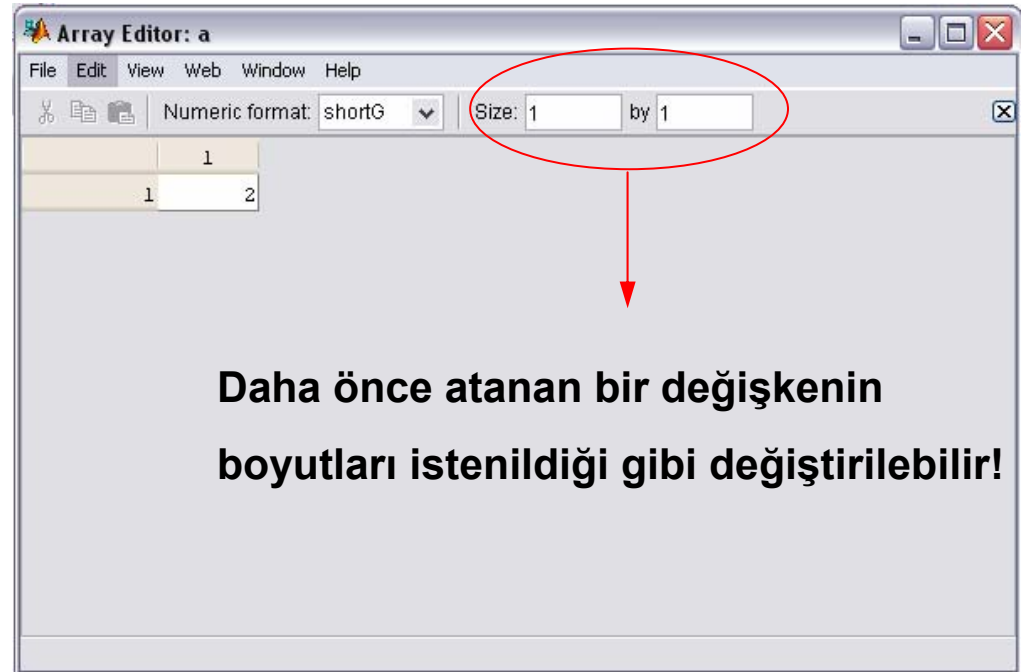
MATLAB/Array Editor (Dizi editörü)



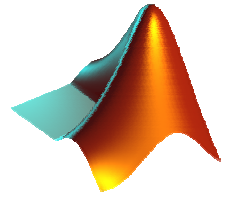
Matris, vektör ve sayılar için excel özelliğindeki editördür.

İki farklı biçimde görüntülenir:

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



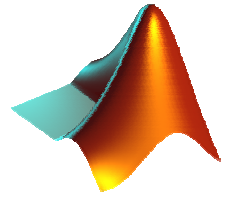
MATLAB/Temel Komutlar



- `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)
- `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.

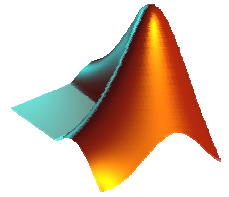
Save ve load komutları, **matris vb. yapıların kaydedilmesi için çok önemlidir.**

MATLAB/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/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]
```

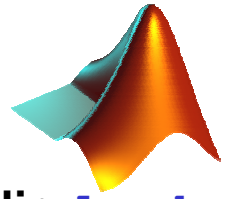
2.yol

```
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
```

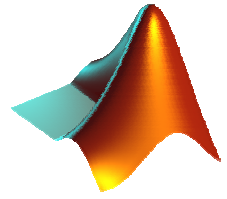
- katsayilar.mat olarak kaydedilen a matrisinin herhangi bir zamanda geri çağırılması için,

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

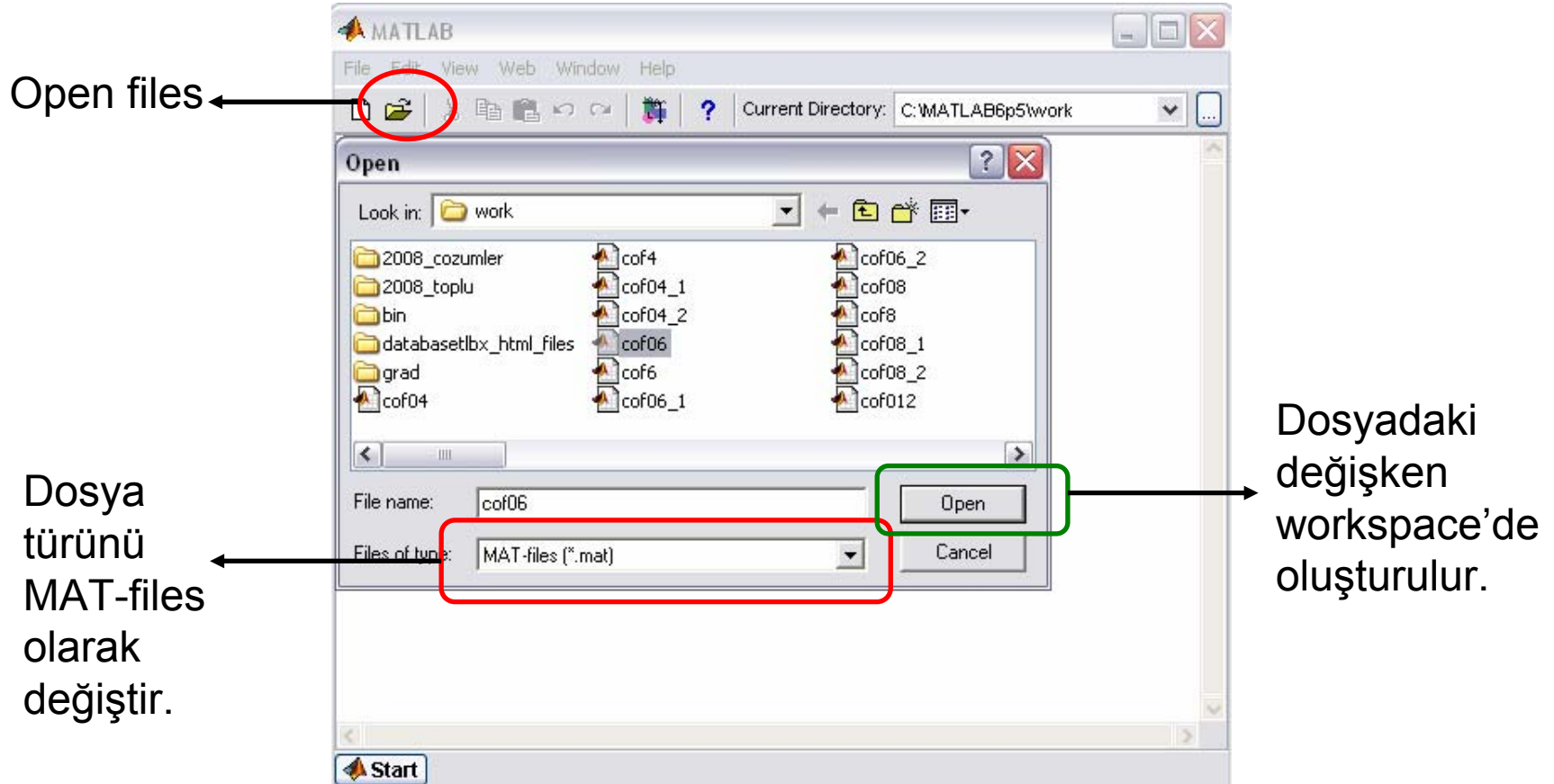
komut dizisi kullanılır. Geri çağırma işleminden sonra, ilgili matris a dizisi olarak workspace’de kaydedilir (workspace’e kaydetme işleminin geçici olduğunu hatırlayınız!)

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

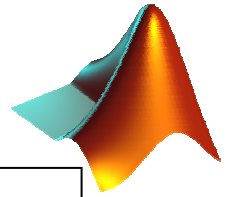
MATLAB/Matrislerin Kaydedilmesi



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



MATLAB/Sayı Formatları



```
>> format long
>> a=1.123000123123123123;
>> a
a =
1.12300012312312
```

16 karakter

```
>> a=100004545.99923423499111;
>> a
a =
1.000045459992342e+008

$10^8$


```

```
>> format short
>> a=1.123000123123123123;
>> a
a =
1.1230
```

6 karakter

```
>> a=100004545.99923423499111;
>> a
a =
1.0000e+008

$10^8$


```

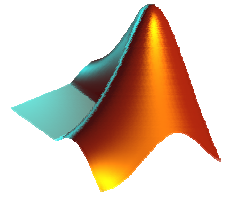
Bir sayının istenilen hanesinin **gösterilmesi** için **fprintf** veya **sprintf** komutları kullanılır:

```
>>a=100004545.99923423499111;
>>fprintf(' %1.10f' ,a)

100004545.9992342300
```

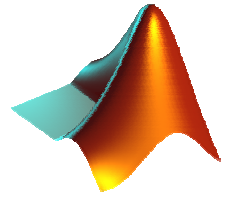
Matris elemanlarının istenilen hanede gösterilmesi için, **printmatrix** fonksiyonu oldukça kullanışlıdır (File-exchange sayfasından download edilebilir!)

MATLAB/Temel lineer cebir komutları



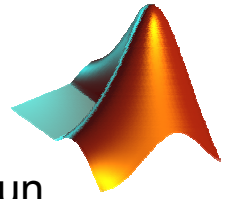
- **`inv(a)`** Bir a kare matrisinin tersini (inversini) alır.
- **`a'`** a matrisinin devriğini (transpozisini) alır.
- **`det(a)`** a matrisinin determinantını hesaplar.
- **`a+b`** Boyutları aynı olan a ve b matrisini toplar.
- **`a-b`** Boyutları aynı olan a ve b matrislerinin farkını alır.
- **`a*b`** 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.

Örneğin,

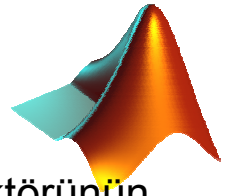
`e=1:1:n`, 1 ile n arasındaki tam sayılardan oluşan bir vektör.

`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.

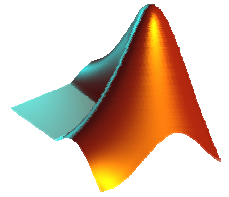
`e=-10:0.1:n`, -10'dan 0.1 artımla n'ye kadar olan sayılardan oluşan bir vektör.

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 =
```

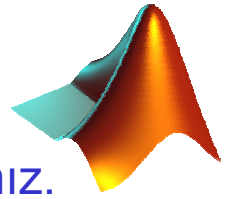
1	1000
3	10
2	5
4	1



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

```
ans =
```

4	1
2	5
3	10
1	1000

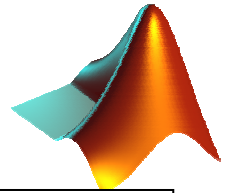


$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 8 & 11 \\ 100 & 1 & 4 \end{bmatrix}$$

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 irdelleyiniz.
- 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 >> inv(A)
ans =

    -0.0288    0.0096    0.0096
    -1.4725    0.6813   -0.0330
     1.0893   -0.4107    0.0179
>> B=ans;
>> A*B
ans =
    1.0000         0    0.0000
         0    1.0000    0.0000
         0    0.0000    1.0000
```

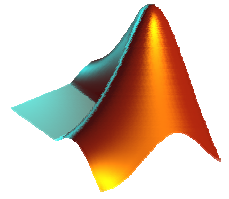
Birim matris

```
⑤ >> a1=A(:,1);a3=A(:,3);
⑥ >> C=diag(diag(A));
⑦ >> a1'*a3
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
>>
```

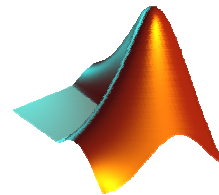


$$\mathbf{B} = \begin{bmatrix} 10 & 5 & 5 \\ 70 & 8 & 7 \\ 10 & 1 & 3 \end{bmatrix}$$

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) $\mathbf{C} = [\mathbf{B} \quad \mathbf{zeros}(3,2)]$ 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

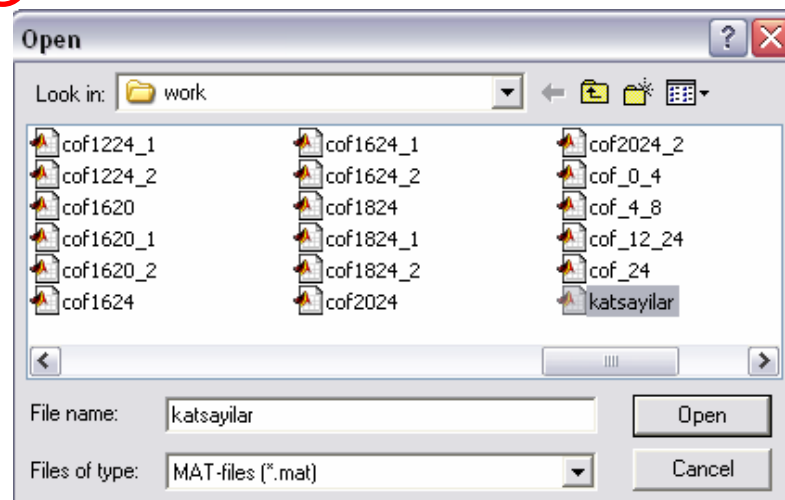
⑧ >> triu(B)

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

>>tril(B)

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

③



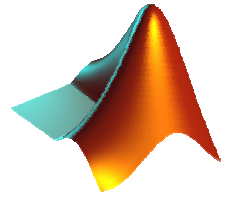
⑨

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

C =

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

MATLAB/Diziler (Arrays)



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

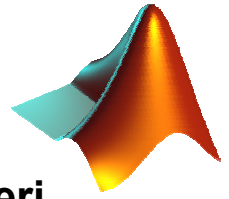
Örnek :

```
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 (Arrays)



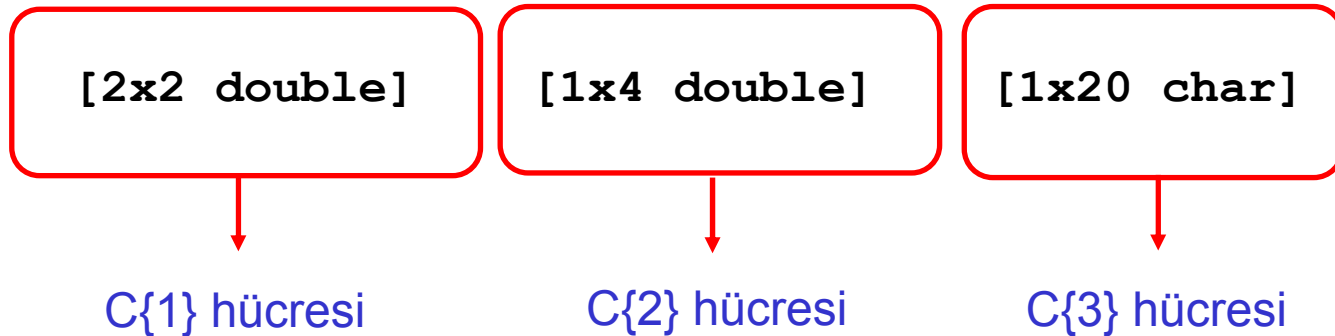
- **Hücre Dizileri (Cell arrays) { } ile tanımlanır.** Böylesi bir dizi, farklı matrisleri aynı isim altında tutmak ve işlemek için kullanılmaktadır.

Örneğin,

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

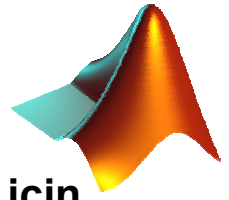
girildiğinde, C bir hücre dizisi olur. Bu hücre geri çağırıldığında,

C =



Her bir hücre ayrı ayrı işlenir.

MATLAB/Diziler (Arrays)



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

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

```
C =
```

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

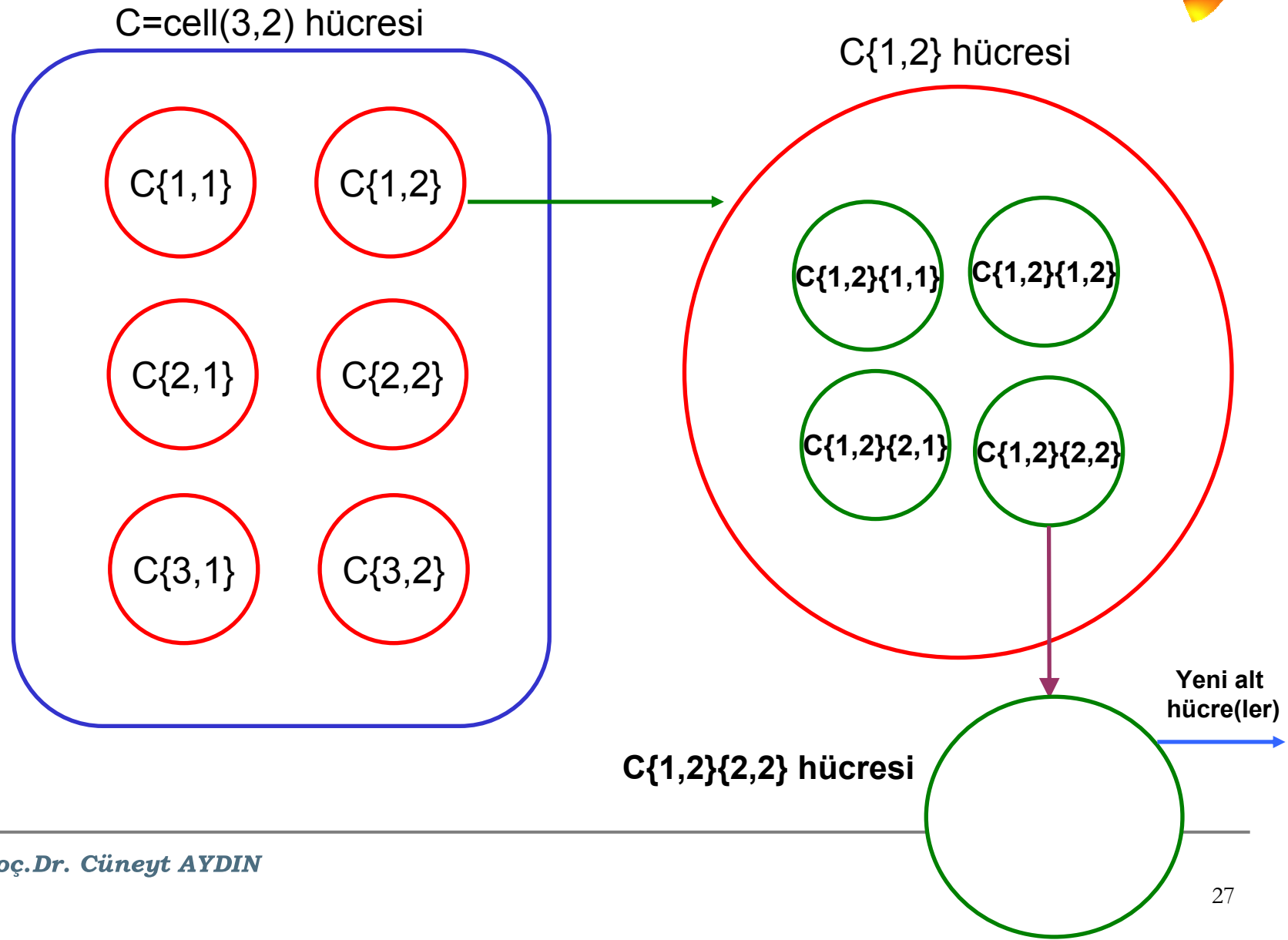
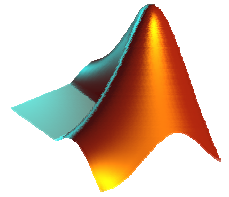
hücresi oluşturulur.

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

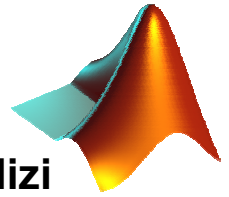
```
C =
```

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

MATLAB/Diziler (Arrays)



MATLAB/Diziler (Arrays)



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

A yapı dizisi çağrıldığında,

```
A.name='Cüneyt';  
A.sname='Aydin';  
A.univ='YTU';  
A.city='Istanbul';  
A.email='caydin@yildiz.edu.tr';  
A.year=2009;
```

ile A, bir structure array olur.

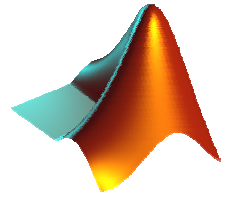
```
>>A
```

```
A =
```

```
    name: 'Cüneyt'  
    sname: 'Aydin'  
    univ: 'YTU'  
    city: 'Istanbul'  
    email: 'caydin@yildiz.edu.tr'  
    year: 2009
```

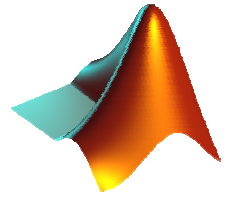
Hücre ve yapı dizileri, mat uzantılı dosyalar olarak, daha önce açıklanan save komutuyla kaydedilip, load komutuyla geri çağrılabilir.

MATLAB/Diziler arasında dönüşümler



- `num2str(a)` Bir a sayısını bir karaktere atama (From **numeric** to (2) **string**)
- `str2num(a)` Karakter olan bir a sayısını sayı değerine atama
- `mat2str(a)` Bir a matrisini bir karakter dizisine atama
- `int2str(a)` Bir a tam sayısını bir karaktere atama
- `char(a)` Bir a hücrelerini bir karakter dizisine atama
- `cellstr(a)` Bir a karakterini bir hücre dizisine atama
- `num2cell(a)` Bir a sayısını bir hücre dizisine atama

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 düşünülmelidir.

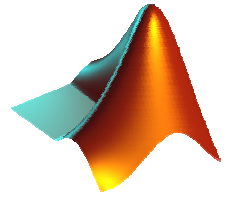
- Bunun daha gelişmiş biçimi, fprintf ile sağlanır:

Diagram illustrating the fprintf function call:

```
fprintf( '%s%1.4f', ('Elde edilen sonuc='), a )
```

The diagram shows the following components and their roles:

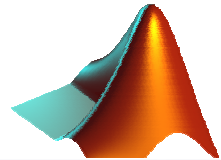
- Karakter** (Character): Points to the format string `'%s%1.4f'`.
- Sayı** (Number): Points to the variable `a`.
- The second argument `('Elde edilen sonuc=')` is enclosed in a green box, indicating it is a character string.



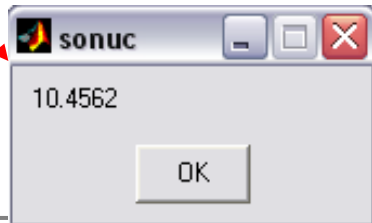
Aşağıdaki işlemleri command window'da yapınız.

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.

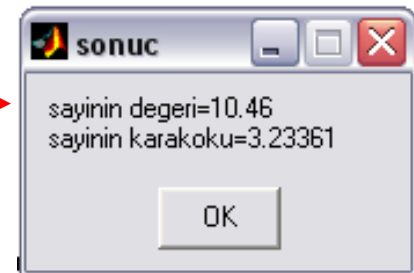
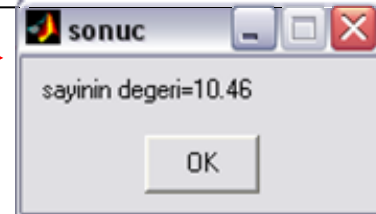
MATLAB/Uygulama-3:Çözüm



```
1 >> a=10.45623;fprintf('%1.3f',a)
10.456
>>
2 >>fprintf('%s%1.2f',('sayinin...
degeri='),a)
3 sayinin degeri=10.46
>>
>>b=sprintf('%s%1.2f',('sayinin...
degeri='),a)
4 b =
sayinin degeri=10.46
>>ischar(b)
ans=
1
>>fprintf('%10.2f',a)
10.46
>>msgbox(num2str(a),'sonuc')
```



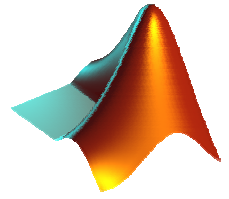
```
7 >> msgbox(b,'sonuc')
>>
8 >>c=sqrt(a)
c =
3.2336
>>b1=sprintf('%s%1.5f',('sayinin...
karakoku='),c);
>>g=char(b,b1);
>>msgbox(g,'sonuc')
```



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')
```

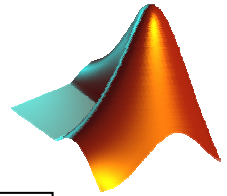
benzer sonucu üretir.



Aşağıdaki işlemleri command window'da yapınız.

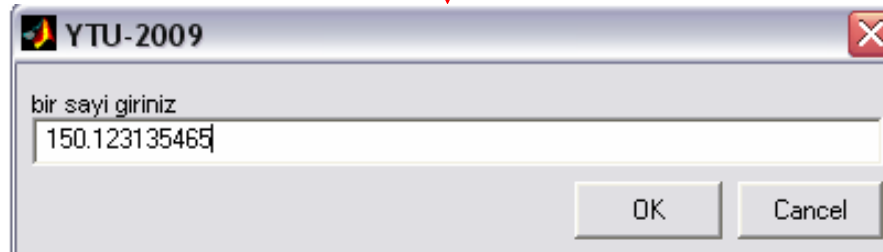
1. Sonraki işlemlerde kullanılacak bir a sayı değerini, inputdlg fonksiyonu ile girdiren komutu yazınız.
2. a değerinin bir sayı olup olmadığını irdelleyiniz.
3. $a*2$ işlemini yapınız. Bu işlemin neden sonuç vermediğini irdelleyiniz.
4. a değerini, gerekli ise, sayı dizisine dönüştürünüz.

MATLAB/Uygulama-4:Çözüm



①

```
>> a=inputdlg('Bir sayi giriniz','YTU-2009')
```



```
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'.
```

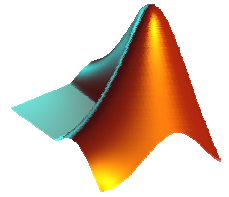
④

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

```
>>
```

```
a =
```

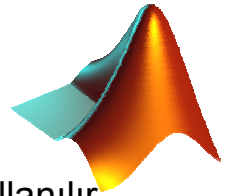
```
150.1231
```



- **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
```

biçimindedir.

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

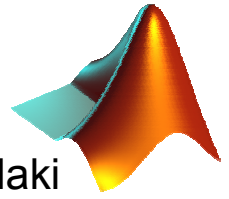
```
a=input(' bir sayi giriniz= ');
if a<0
    a=log(a);
else
a=a;
end
a
```

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

Else yapısı kullanılmasaydı

```
a=input(' bir sayi 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;

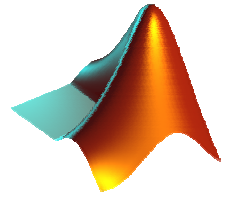
```
switch durum
    case durum1
        işlem1
    case durum2
        işlem2
    otherwise
        işlem3
end
```

→ Kullanımı kişiye bağlıdır.

Örnek: gun degiskeninin, is gunu olup olmadigina 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
```

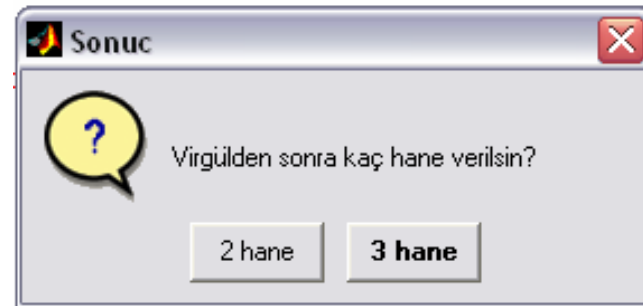
MATLAB/switch,case yapısı



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

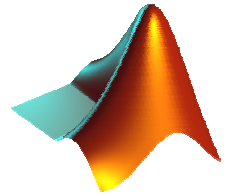
```
a=10.2424542;  
dummy=questdlg('Virgülden sonra kaç..  
hane verilsin?', 'Sonuc', ...  
                '2 hane', '3...  
hane', '3 hane');  
switch dummy  
    case {'2 hane'}  
        fprintf('%1.2f',a)  
    case {'3 hane'}  
        fprintf('%1.3f',a), end
```

questdlg GUI'si



“2 hane” düğmesinin tıklanması durumunda,
 10.24
sonucu görüntülenir.

MATLAB/Uygulama-5



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

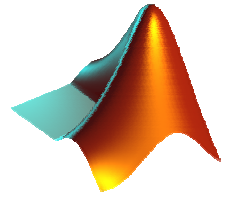
if (DX<0) & (DY>0)
    a=('Aci 2.bolgede');
end

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-6



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

Çözüm

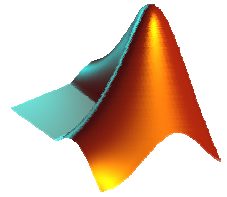
```
clear
clc
X1=input('X1=');Y1=input('Y1=');
X2=input('X2=');Y2=input('Y2=');
DX=X2-X1;DY=Y2-Y1;

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('%s%1.5f%s','(1-2) aciklik acisi=',a,' grad')
```

Dış koşul

MATLAB/Uygulama-7



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
```

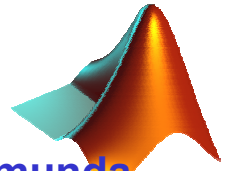
```
fprintf('%s%1.5f%s', '(1-2) aciklik acisi=', a, ' grad')
fprintf('\n')
fprintf('%s%1.3f%s', '(1-2) kenar uzunlugu=', S, ' m')
```

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

Dış koşul

İki ifadeyi alt alta yazdırmak için

MATLAB/Uygulama-9



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

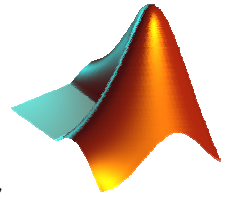
Çözüm

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

if (a>2) | (a<=0)
    menu                                %Kullanıcının 0,negatif veya 3'den büyük sayı girmesi
                                        %durumunda, programı yeniden başlatmak için
end

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

MATLAB/for,end döngüsü



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

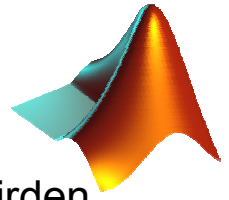
```
for i=1:n      (i→Tam sayı (integer))  
    işlem  
end
```

biçimindedir.

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

```
clear,clc  
N=input('bir sayı giriniz=');  
say=0; %sayac  
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.
(while, end döngüsünü çalıştıran farklı algoritmalara burada değinilmeyecektir).

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

```
clear,clc  
N=input('bir sayi 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.).

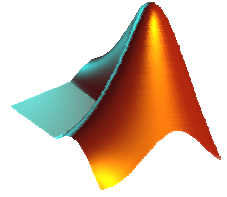
MATLAB/Uygulama-10



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.

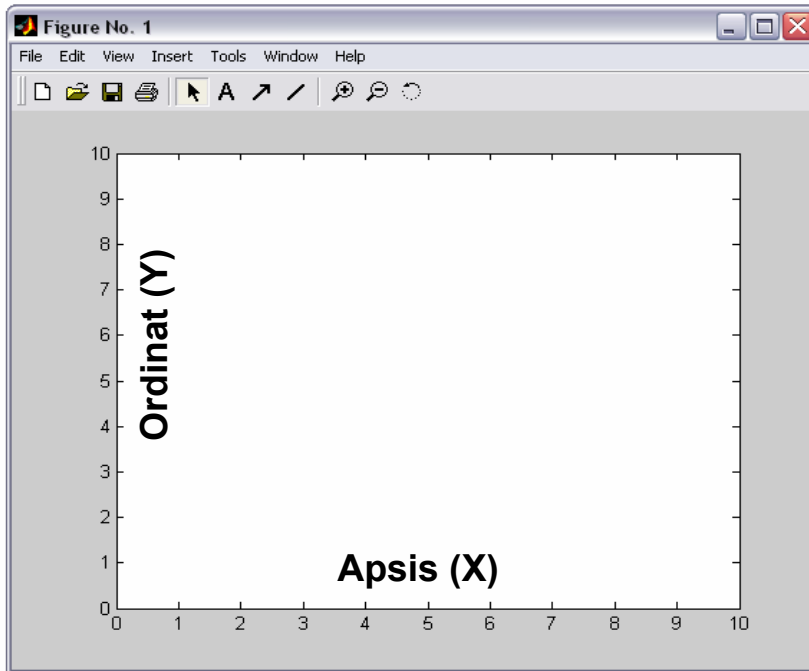
Çözüm

```
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
```

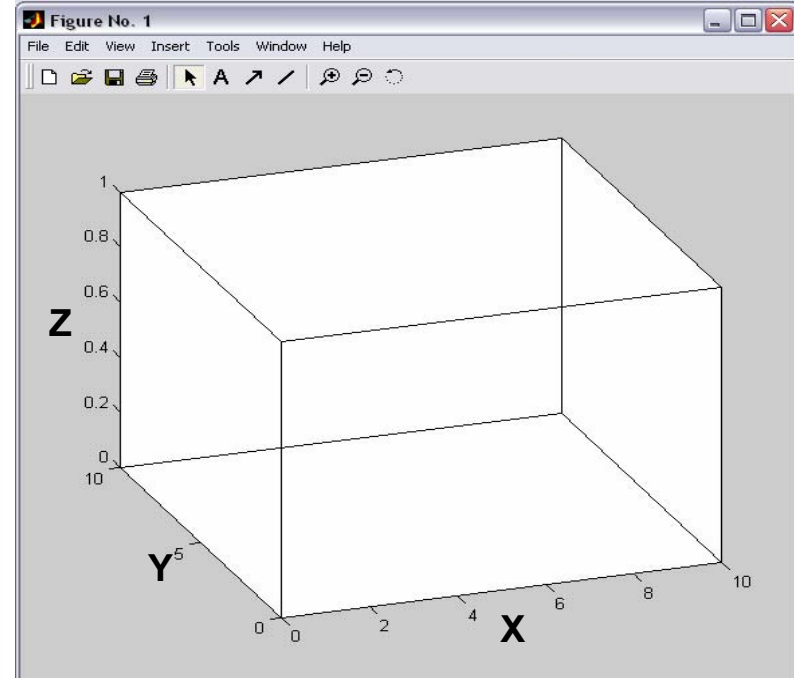


- Matlab'de grafikler “figure” penceresinde çizdirilir.
- İki ve üç boyutlu çizim yanı sıra, kutupsal koordinat sisteminde de çizim olanağı bulunur (bak., polar).

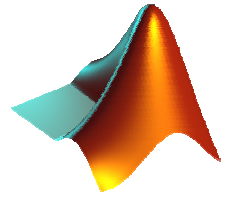
İki Boyutlu Koordinat Sistemi



Üç Boyutlu Koordinat Sistemi

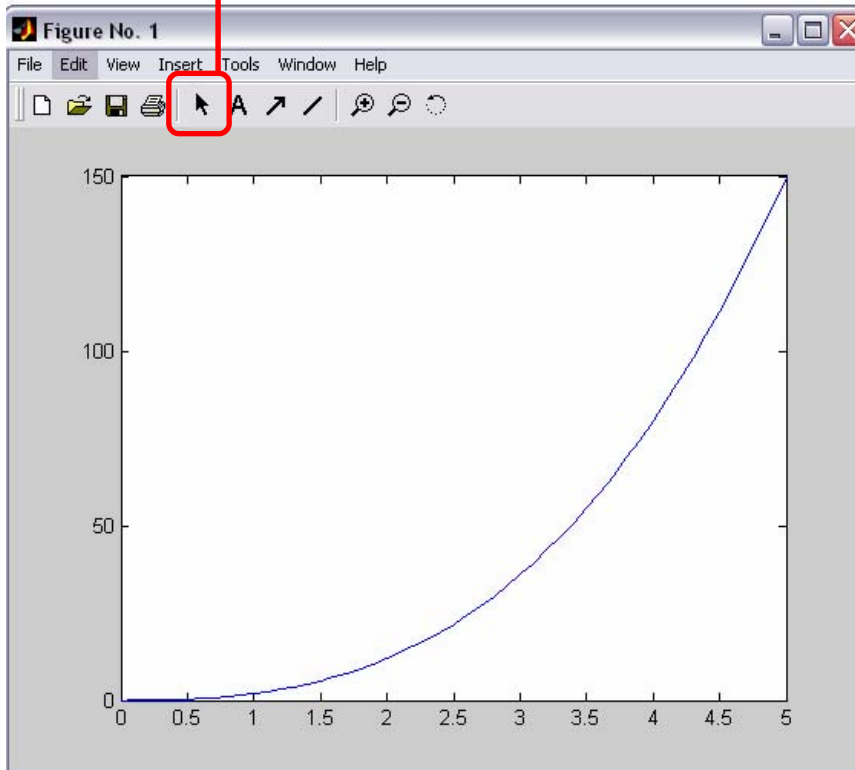


MATLAB/Grafik



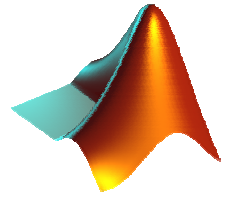
- Matlab'de en temel çizim fonksiyonu **plot**'dur.
- Örneğin, $x=0:0.1:5$ olan bir dizi vektör elemanlarına karşılık, $y=x.^3+x.^2$ fonksiyon değerleri hesaplatılsın.
- `plot(x,y)` ile aşağıdaki grafik çizdirilir.

Edit plot

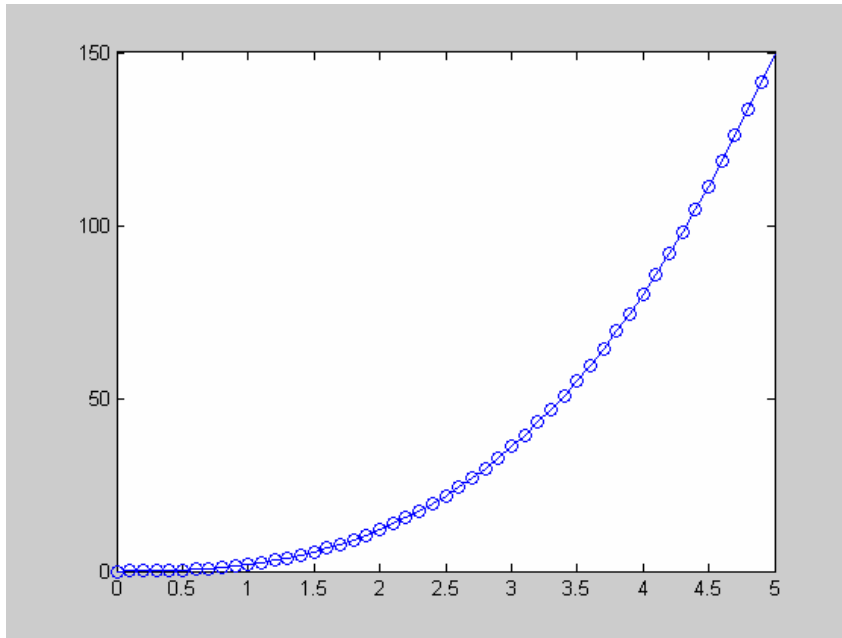


- Çizilen grafiğin üzerinde birçok değişiklik yapmak mümkündür.
- Bunun için **"Edit plot"** düğmesi tıklanır.
- İlgili nesne (çizdirilen eğri, eksenler vb.) iki kez tıklanarak beliren **"Property Editor"** penceresinden istenilen değişiklikler yapılabilir.
- Property Editor penceresinden yapılan her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

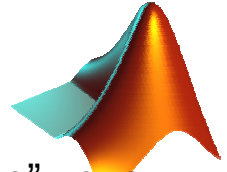


- `plot(x,y, '-o')` ile ilgili grafik aşağıdaki gibi olur.

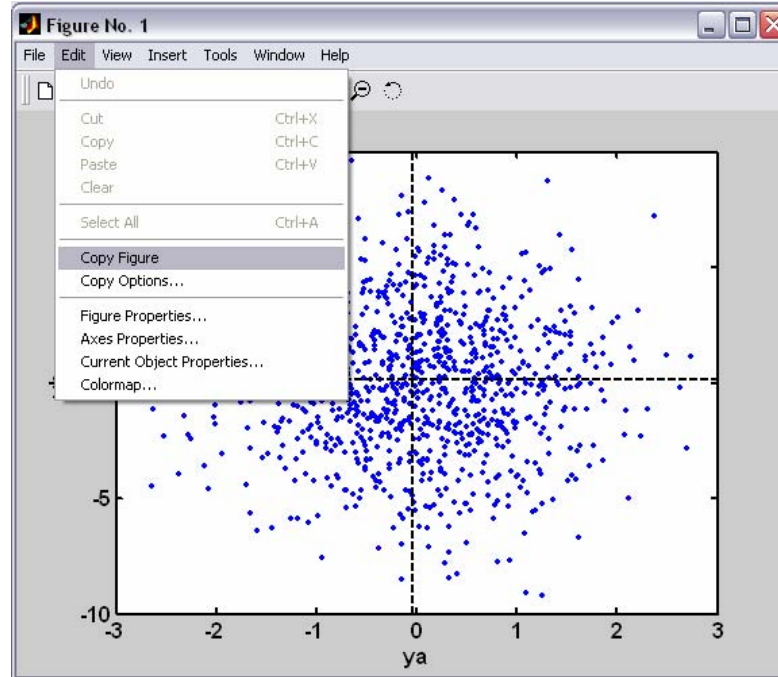


- Aşağıdaki ifadelerle çizimi tekrarlayınız:
`plot(x,y, '-o')`
`plot(x,y, '-*')`
`plot(x,y, '-+')`
`plot(x,y, '-^')`
`plot(x,y, '-.')`
- Sözü edilen grafik üzerindeki o,*,+ gibi sembollere marker denir.
- `plot` fonksiyonu ile ilgili eğrinin rengini değiştirmek de mümkündür:
`plot(x,y, 'r')` kırmızı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

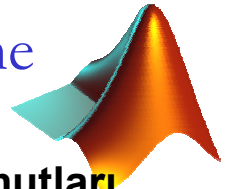
MATLAB/Grafik-Kaydetme ve kopyalama



- 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.)

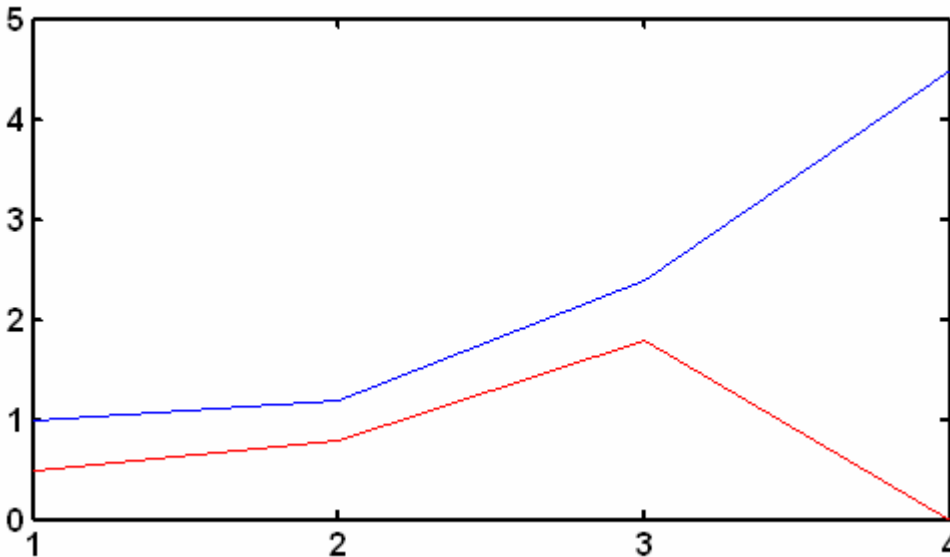


MATLAB/Grafik-Aynı eksen takımına farklı grafikler çizdirme



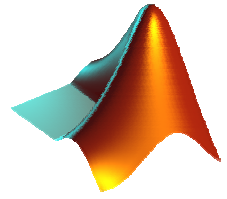
- Aynı eksen takımına farklı grafikleri çizdirmek için **hold on** ve **hold off** komutları kullanılır. Bu iki komut arasına yazılan her türlü grafiğin çizimi aynı eksen takımında gösterilir.
- Örneğin, $y_a = [1; 1.2; 2.4; 4.5]$ ve $y_b = [0.5; 0.8; 1.8; 0]$ vektörleri ile ifade edilen iki farklı ölçü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

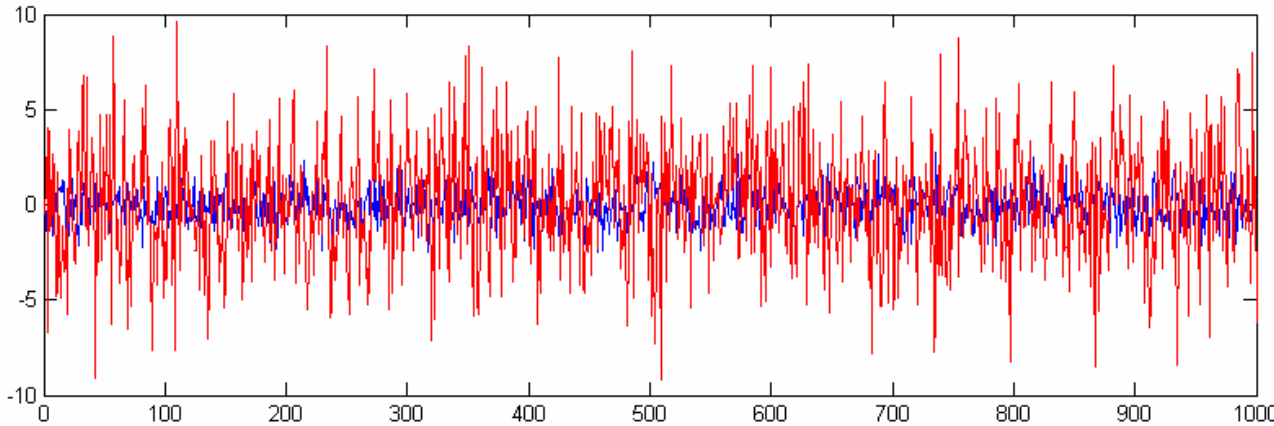
MATLAB/Grafik-Grafik yoluyla bilgi üretme



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:

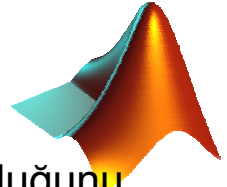
```
hold on, plot(ya), plot(yb, 'r'), hold off
```



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 *deneyse* standart sapmasını verecektir (bkz. İstatistik Ders Notları). Bu değer "3" *kuramsal* standart sapma değerine yakın olacağına dikkat ediniz.

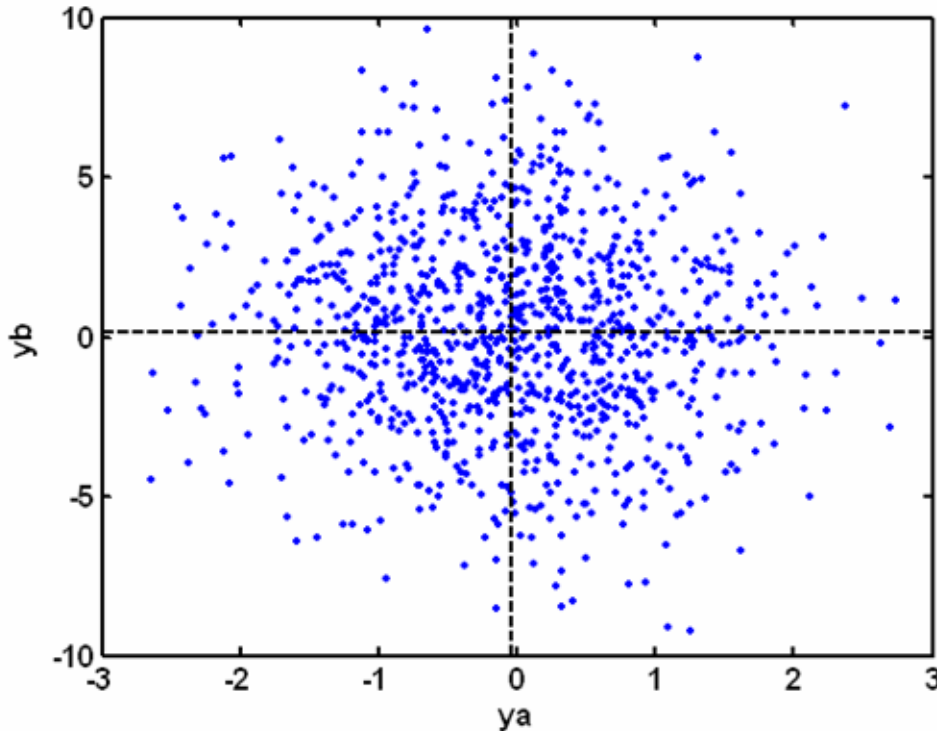
MATLAB/Grafik-Grafik yoluyla bilgi üretme



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

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

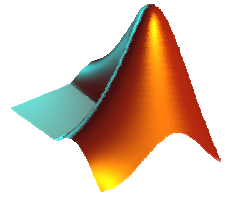
komutunu kullanmak yeterlidir.



- İlgili grafikten, ya ve yb ölçüleri arasında anlamlı bir ilişki (korelasyon) olmadığı bilgisi hemen türetilir. 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.



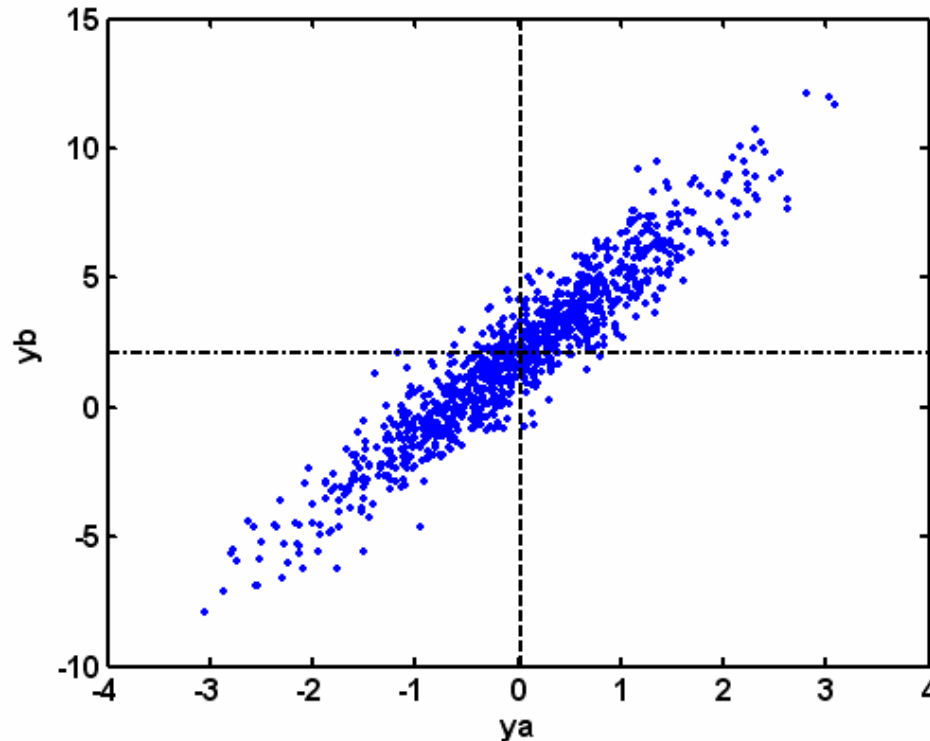
MATLAB/Grafik-Grafik yoluyla bilgi üretme



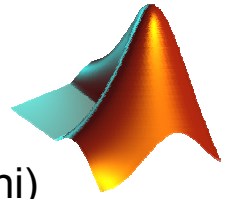
- Bir önceki örnekte kullanılan y_b ölçülerini, $y_b = 2 + 3 \cdot y_a + \text{randn}(1000, 1) \cdot 1$, biçiminde y_a ölçülerine bağlı olarak üretelim. Bu durumda,

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

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



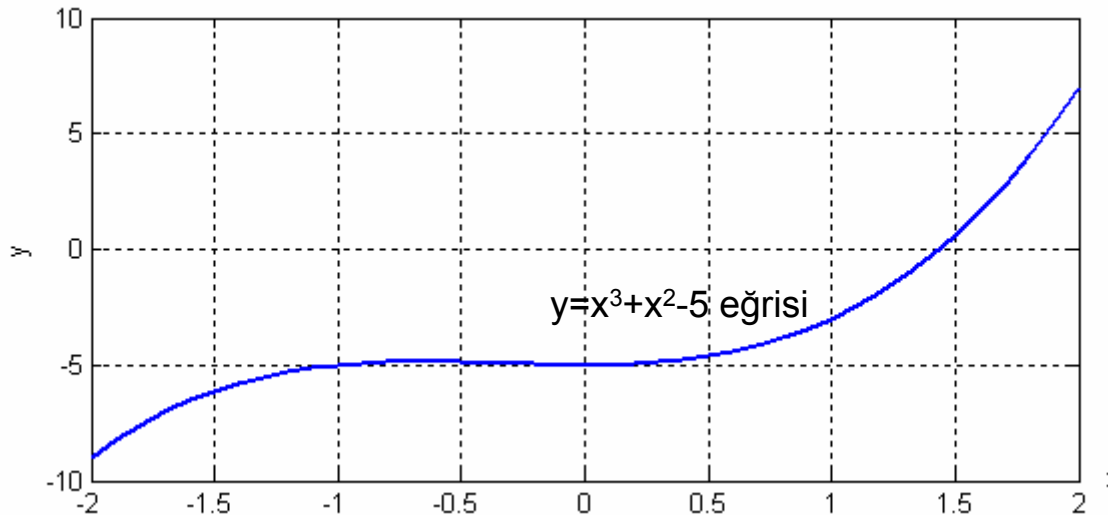
MATLAB/Grafik-Grafik yoluyla bilgi üretme



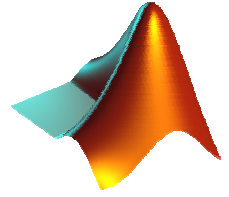
- Mühendislik uygulamalarında en çok karşılaşılan problemlerden biri de $F(x)=0$ biçimindeki bir denklemin ilgili aralıktaki kökünü (fonksiyonu sıfır yapan x değerini) bulmaktır. Sayısal analizde kullanılan Newton-Raphson gibi yöntemlerde kökün yaklaşık değerine ihtiyaç vardır. Bu yaklaşık değeri bulmak 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)

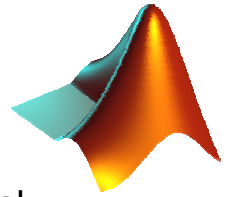


- $y=0$ doğrusunun eğriyi kestiği noktadan, x eksenine hayali bir dik inilirse, bu dikin gösterdiği x değeri, $F(x)$ denklemini sağlayan kök olacaktır. Buradan kökün yaklaşık değerinin 1.4 olduğu sonucuna kolaylıkla ulaşılır.
- **Not:** Figure penceresindeki büyütme özelliği ile, ilgili kesişim noktasına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



- 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!

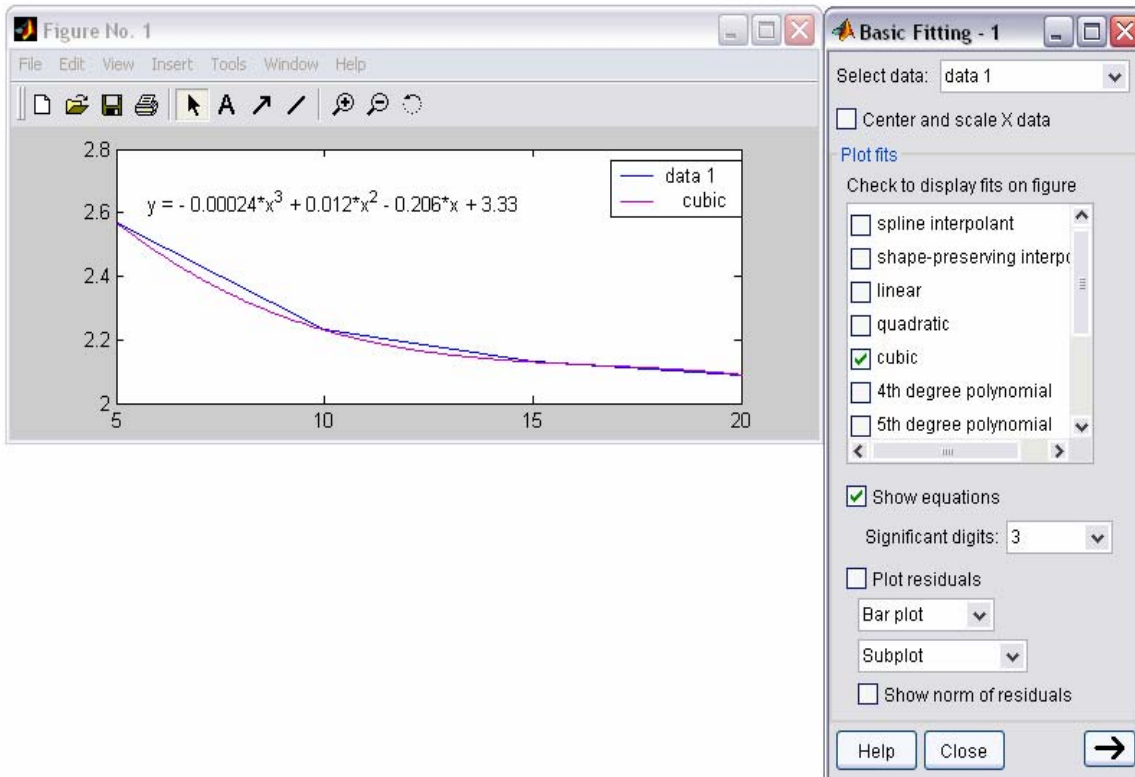
MATLAB/Grafik-Basic Fitting Tool



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.

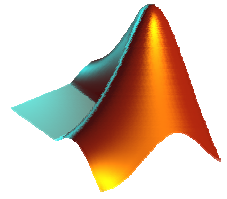
f	5	10	15	20
t	2.57	2.23	2.13	2.09

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

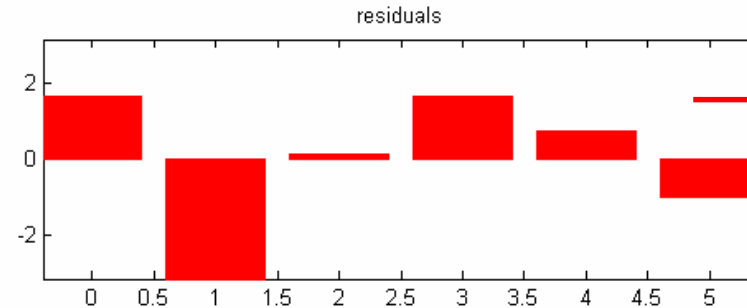
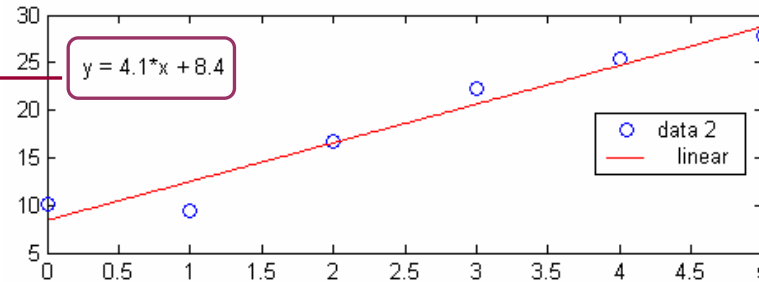


Örnek: Aşağıdaki tabloda, x zamanlarına karşılık y ölçüleri elde edilmiştir. Ölçülere en iyi uyan $y=a+bx$ doğrusunu belirleyiniz (En küçük kareler kestirim yöntemi)

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.

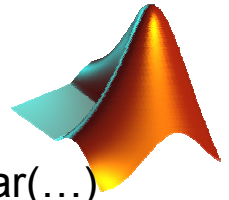
En küçük kareler yöntemine göre belirlenen en uygun doğru denklemi (*model*)*



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

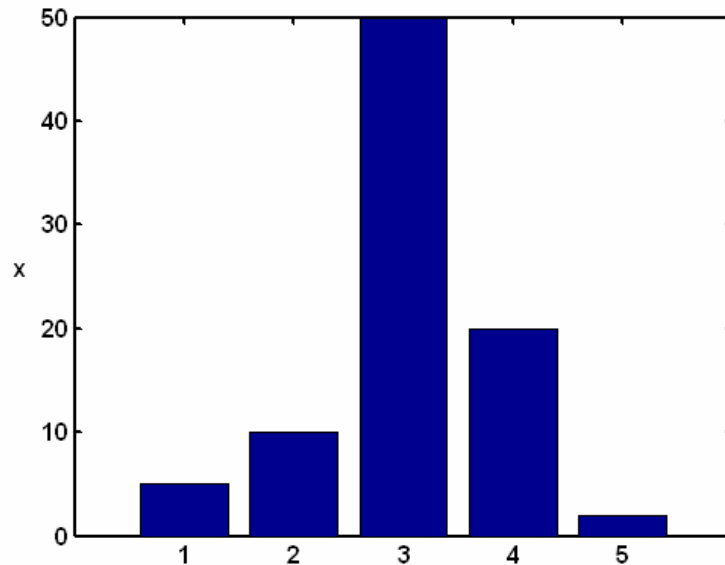
* Demirel H (2005), Dengeleme Hesabı Ders Notları, YTÜ, İstanbul

MATLAB/Grafik-Çubuk (bar) ve stem grafiği

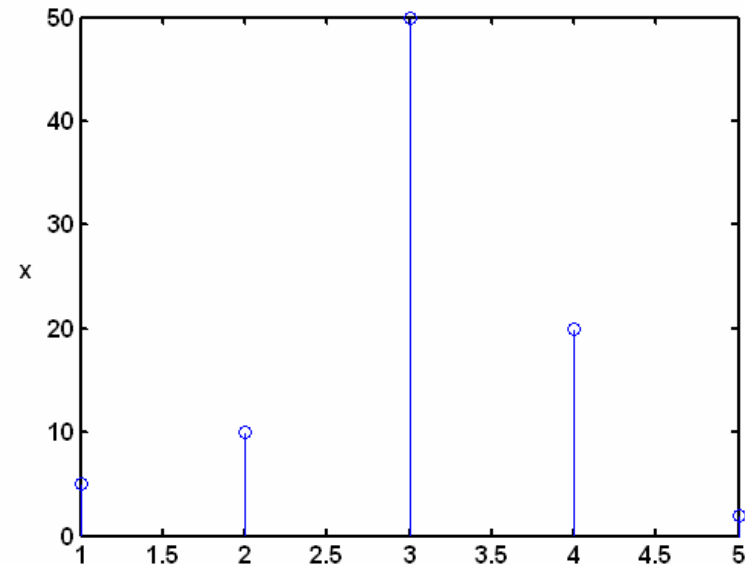


- Matlab'de farklı gösterimlere göre çizim yapmak mümkündür. Bunlardan ikisi `bar(...)` ve `stem(...)` çizim fonksiyonlarıdır.
- Örnek:** $x=[5;10;100;20;2]$ vektör elemanlarının bar ve stem grafik olarak göstermek isteyelim. `bar(x)` ve `stem(x)` aşağıdaki grafikleri çizdirecektir.

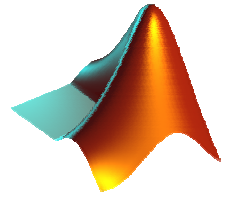
Bar grafiği



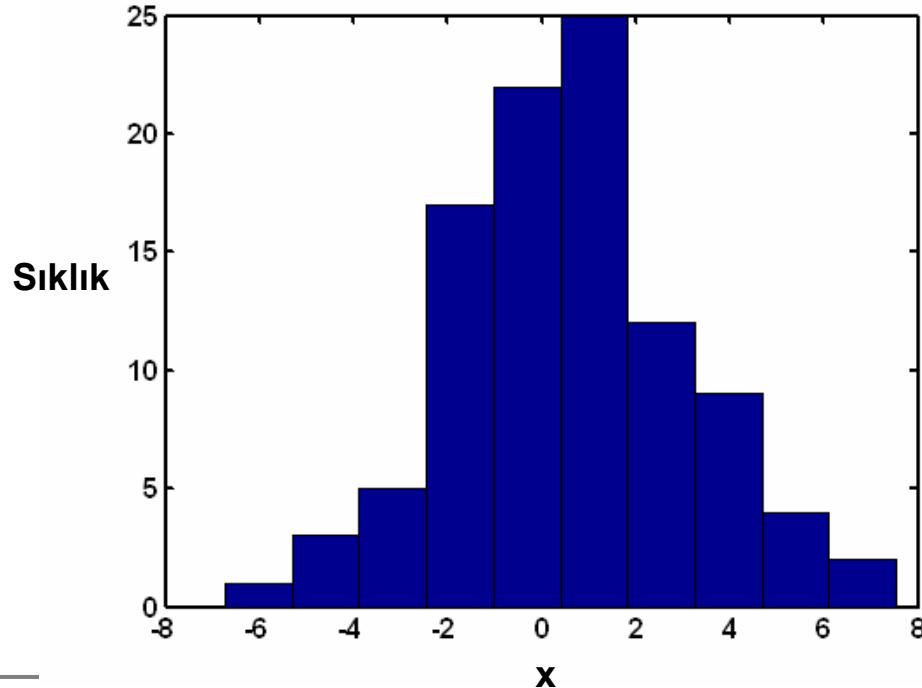
Stem grafiği



MATLAB/Grafik-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.)



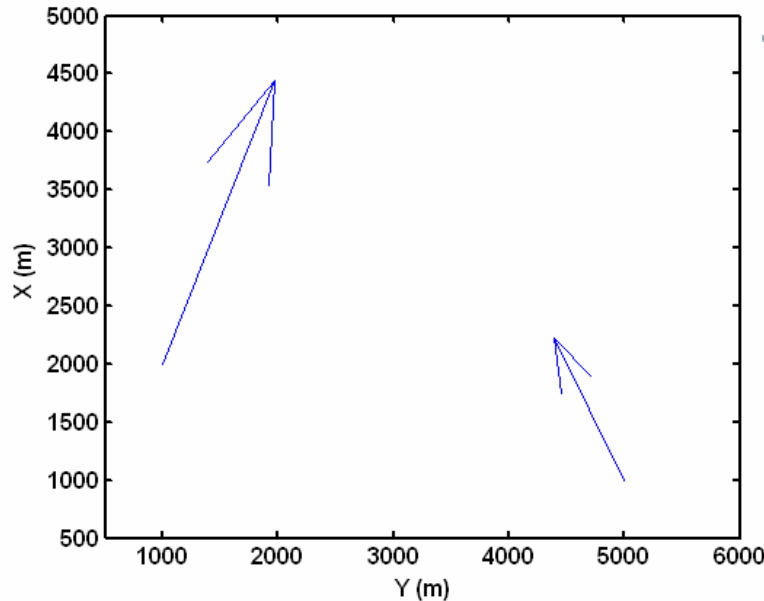
MATLAB/Grafik-Vektör çizimi



- x ve y koordinat değerlerine sahip bir noktanın dx ve dy kadar yer değiştirdiği düşünülün. 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.

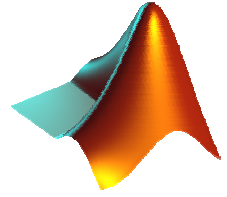
`quiver(y,x,dy,dx)` (**Not:** Bir jeodezik dik koordinat sisteminde x ve y 'nin yer değiştirdiğini hatırlayınız!)

komutu ile bir jeodezik dik koordinat sisteminde vektör çizimi gerçekleştirilir.



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

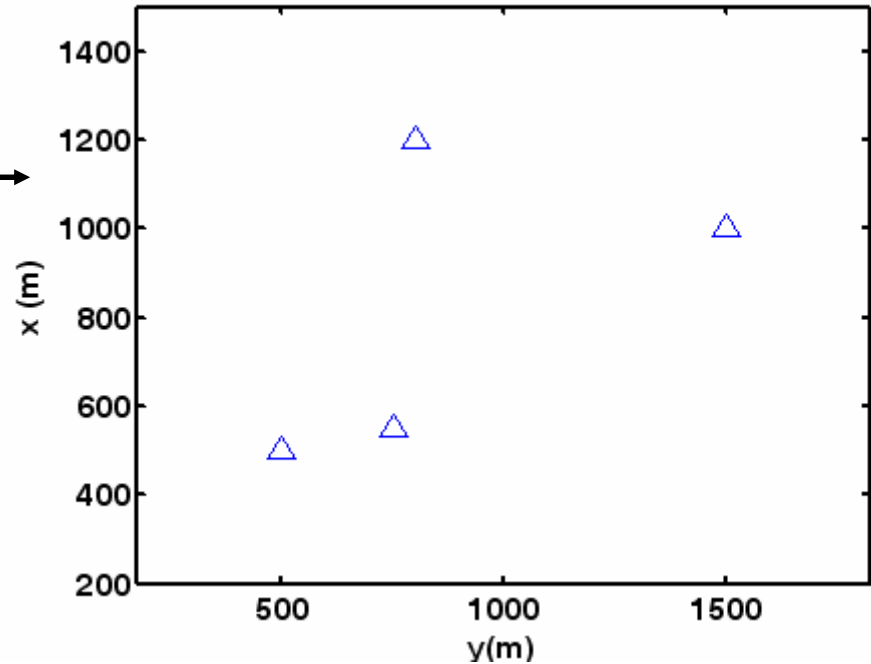
`quiver(y,x,dy,dx,s)`



- 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-İnterpolasyon

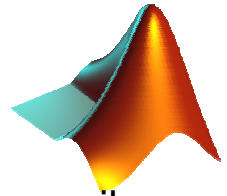


- x ve y koordinatları bilinen noktalara ilişkin üçüncü bir bilgi (örneğin, *yükseklik*, *yükseklik değişimi*, *sıcaklık*, *nem*, *basınç*, *gelgit deformasyonu*, *anomali* vb.) olduğunda, noktaların çevrelediği alanın içindeki hayali noktalar için bu bilgiler, çeşitli matematiksel yaklaşımlarla üretilebilir: 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** noktadaki 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 gibi interpolasyon yöntemleri de bulunur)

MATLAB/Grafik-İnterpolasyon



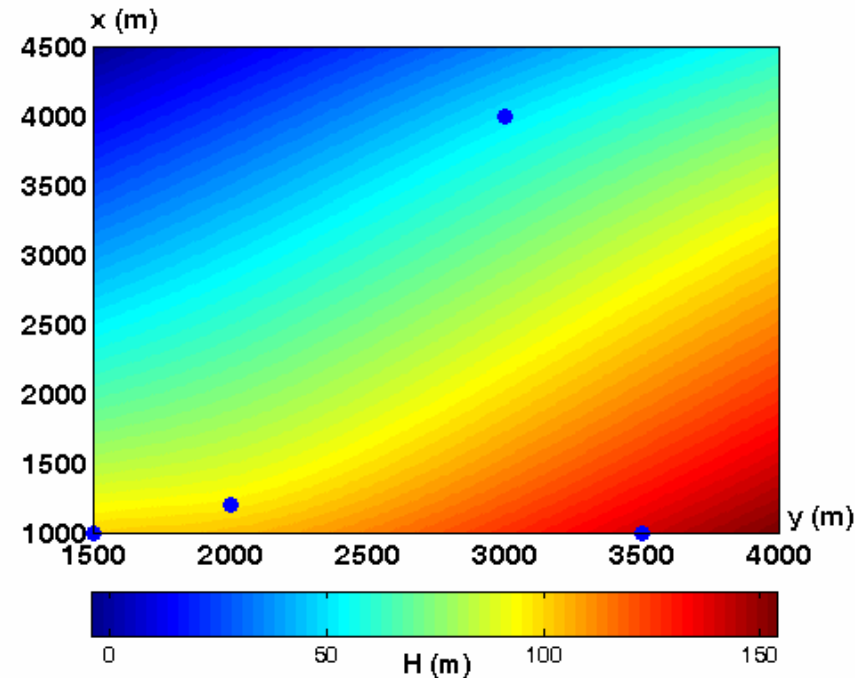
- **Örnek:** Nokta koordinatları, $x=[1000;1200;4000;1000]$, $y=[1500; 2000;3000;3500]$ ile ve bu noktaların yükseklikleri, $H=[100.0000;95.9850;50.5000;140.1200]$ ile tanımlansın. Bölgeyi 10 m'lik gridlere bölerek, bölgenin yükseklik değerlerini gösteren bir renk haritası hazırlayınız.

```
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;

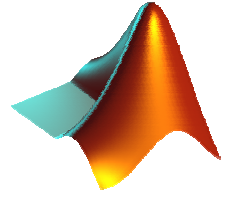
[XI,YI]=meshgrid(x1,y1);

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

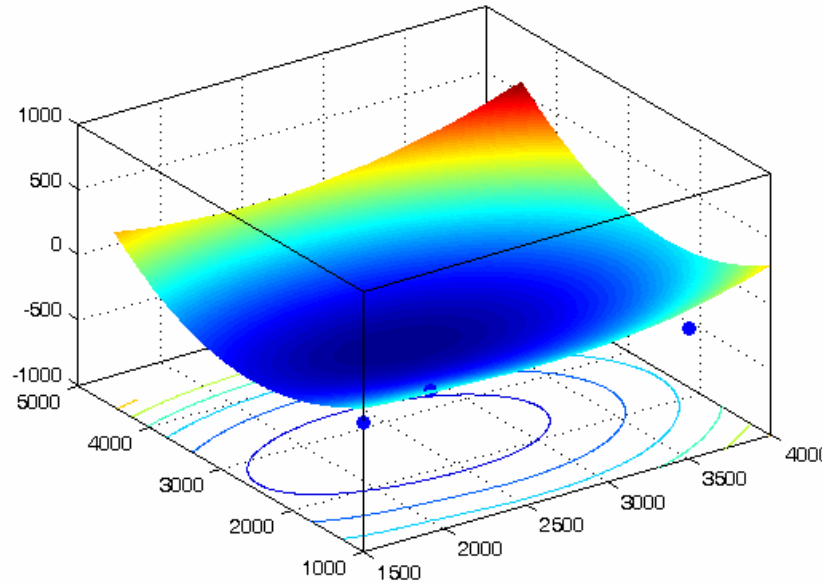
pcolor(YI,XI,HI),
shading interp
colormap('jet')
plot(y,x,'o','MarkerFacecolor','b')
hold off
```



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

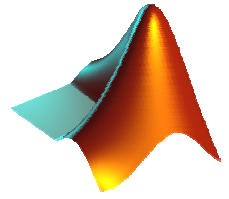


- Contour (örneğin, eş yükselti eğrileri) haritası için `contour` ve `clabel` fonksiyonlarına,
- Üç boyutlu yüzey çizimleri için `mesh`, `surf`, `surfl` ve `surf` fonksiyonlarına bakınız.



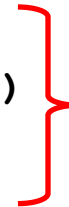
`surf` fonksiyonu ile oluşturulmuş bir yüzey grafiği

MATLAB/Dosya Yazdırma-Okuma



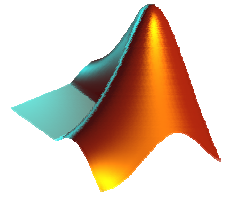
- Program sonuçlarının otomatik olarak farklı bir dosyaya yazdırılması veya bir dosyadaki bilgilerin okunarak program içerisinde kullanılması, programcılıkta oldukça sık başvurulanan çıktı alma ve veri girişi yöntemleridir.
- Dosya yazdırma, çıktı almaya; Dosya okuma ise veri girişine karşılık olan işlemlerdir.
- 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

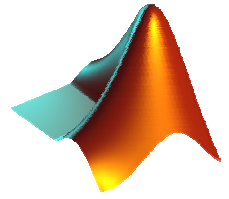


- Daha gelişmiş dosya yazdırma, fopen, fprintf ve fclose fonksiyonlarının kullanımı ile gerçekleştirilir.
- Bu fonksiyonlar ile dosya yazdırmada, yazdırılacak olan metnin “command window” da gösterilmesine gerek yoktur.
- fopen, program çıktılarının yazdırılacağı dosyayı açar, fprintf yazdırır ve fclose ise yazdırma işlemini sonlandırır.
- 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,'%s%1.4f','kenar uzunlugu=',a);  
fclose(fid);
```

→ **w** bu dosyanın üzerine yazılacağını gösterir.
(**a** “mevcut metnin altına yaz” demektir)

MATLAB/Dosya Yazdırma-Okuma



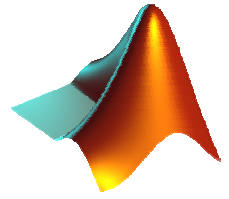
- **Örnek:** `a=[3.12356 4.12456 1;5.8463 6.45111 2;4 5 6]` biçiminde verilen bir a matrisini, elemanları virgülden sonra 4 hane olacak biçimde, `mat.out` dosyasına yazdıran bir program yazınız.

```
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,'%s%1.3f%s\r\n','kenar=',kenar,' m');
fprintf(fid,'%s%1.4f%s','aciklik=',aciklik,' grad');
fclose(fid)
```

MATLAB/Dosya Yazdırma-Okuma



- Matlab'de dosyaların içindeki metnin 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

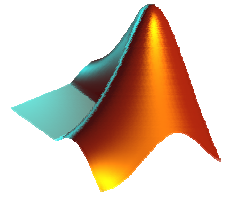
Nokta isimleri x koordinatları y koordinatları

- 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.

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.

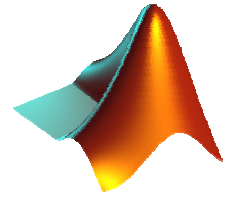
Nirengi koordinatları		
NN	x (m)	y (m)
P1	1000.1234	1300.23423
P2	1300.5673	1450.98563
P3	2000.1500	2000.11000
P4	3500.3100	1000.12000

```
[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/Fonksiyon Dosyası Oluşturma

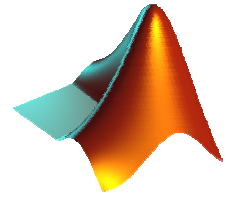


- Matlab fonksiyonları (örneğin, inv, disp, num2str,det, textread...) kullanıcının bir başka programa gerek duymaksızın temel işlemleri kolayca yapabilmesini sağlar.
- Kullanıcılar kendi fonksiyonlarını geliştirebilirler.
- Fonksiyonlar, bir m-dosyası biçiminde saklanır ve bu dosyalara, fonksiyon dosyaları 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.

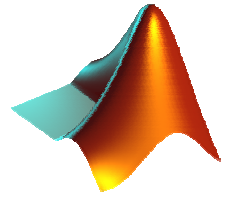
```
C:\MATLAB6p5\work\kenar.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: X
1 function S=kenar(X1,Y1,X2,Y2)
2
3 %S=kenar(X1,Y1,X2,Y2) fonksiyonu 1 ve 2 noktaları
4 %arasındaki kenar uzunluğunu hesaplar.
5 %YTU-2009
6
7 dx=X2-X1;
8 dy=Y2-Y1;
9 S=sqrt(dx^2+dy^2);
10 end
kenar Ln 4 Col 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.

```
C:\MATLAB6p5\work\aci_kenar.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: X
1 function [a,S]=aci_kenar(X1,Y1,X2,Y2)
2
3 %[a,S]=aci_kenar(X1,Y1,X2,Y2) fonksiyonu, (1-2) aciklik
4 %acisini(a) ve 1-2 kenar uzunlugunu(S) hesaplar.
5 |
6 %YTU-2009
7
8 DX=X2-X1;DY=Y2-Y1;
9 if (DX~=0)&(DY~=0),a=atan(DY/DX);a=a*200/pi;
10 if (DX>0)&(DY>0),a=a;end
11 if (DX<0)&(DY>0),a=a+200;end
12 if (DX<0)&(DY<0),a=a+200;end
13 if (DX>0)&(DY<0),a=a+400;end
14 end
15
16 if (DX==0)&(DY>0),a=100;end
17 if (DX==0)&(DY<0),a=300;end
18 if (DX>0)&(DY==0),a=0;end
19 if (DX<0)&(DY==0),a=200;end
20
21 S=sqrt(DX^2+DY^2);
22
23 end
aci_kenar Ln 5 Col 1
```

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 Hızlı Erişim Kılavuzu

Temel Komutlar

help y	Bir y fonksiyonu için yardım
clear	Atanan tüm değişkenlerin silinmesi
clear x	Bir x değişkeninin silinmesi
pwd	Çalışma klasörü yolu
demo	Matlab demo penceresi
save	Matris kaydetme
load	Matris geri çağırma
clc	Çalışma penceresinin temizlenmesi

Matematiksel Operatörler

+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
^	Üst alma
.*	Elemanter çarpım
./	Elemanter bölme
.^	Elemanter üst alma
sqrt	Kök alma
abs	Mutlak değer

Mantıksal Operatörler

&	Ve
	Veya
~	Değil
/	Bölme

Karar Operatörleri

>	Büyüktür
<	Küçüktür
>=	Büyük eşittir
<=	Küçük eşittir
==	Eşittir
~=	Eşit değildir

Semboller

%	Açıklama getirme ifadesi
[]	Matris girme ifadesi
()	İndis ve değer girme ifadesi
=	Değişken atama ifadesi
;	Matrislerde satır ayırıcı. Ayrıca, içinde "=" bulunan bir işlemin gösterilmemesi
:	Kolon ifadesi
ans	Değişken atanmamış en son işlem için varsayılan değişken ismi (answer)
{ }	Hücre dizisi
x.adi	x yapı dizisi

Sabit terimler

pi	π sayısı
eps	2.2204e-016 sayısı
inf	Sonsuz (belirsiz) ifadesi

Trigonometrik Fonksiyonlar

sin, cos, tan, cot	Trig. Fonksiyonlar
asin, acos, atan, acot	Ters Trig. ifadeler

Logaritmik Fonksiyonlar

log	Doğal logaritma
exp	Ekspansiyel

Lineer Cebir Fonksiyonları

det(x)	Bir x matrisinin determinantı
inv(x)	Bir x matrisinin tersi
trace(x)	Bir x matrisinin izi
diag(x)	Bir x matrisinin köşegen elemanları
diag(y)	Köşegenleri, bir y vektörünün elemanları olan köşegen matris
zeros	Sıfır matris
ones	Birler matrisi
eye	Birim matris
eig	Özdeğer ve özvektör bulma

Matlab Hızlı Erişim Kılavuzu**Yazdırma Fonksiyonları**

fprintf	Bir ifadenin yazdırılması
sprintf	Bir ifadenin bir karakter dizisine atanması
disp	Bir ifadenin ekran çıktısı olarak gösterilmesi

Karakter dizisi fonksiyonları

num2str(x)	Bir x sayısını bir karaktere atama
str2num(x)	Karakter olan bir x sayısını sayı değerine atama
char(a)	a hücrelerini bir karaktere atama
char(s1,s2,...)	s1,s2,... karakterlerinden yeni bir karakter dizisi oluşturma
lower(s)	Bir s karakter dizisinin tüm elemanlarını küçük harf yapma
upper(s)	Bir s karakter dizisinin tüm elemanlarını büyük harf yapma
isnumeric	Bir değişkenin sayı olup olmadığını sorgulama
ischar	Bir değişkenin karakter olup olmadığını sorgulama
iscell	Bir değişkenin hücre olup olmadığını sorgulama

Hazır GUI'ler

msgbox	İleti penceresi
inputdlg	Değer girme penceresi
questdlg	Soru diyalog penceresi
uigetfile	Open file diyalog penceresi
uigetdir	Open directory diyalog pen.
Uiputfile	Save file diyalog penceresi

Sayı yuvarlatma fonksiyonları

fix	Sıfıra yuvarlatma
floor	Negatif sonsuza yuvarlatma
ceil	Pozitif sonsuza yuvarlatma
round	En yakın tam sayıya yuvarlatma

Bazı matematiksel fonksiyonlar

sum(x)	Bir x vektör elemanlarının toplamı
diff(x)	Bir x vektör elemanlarının ardışık farklandırılması
mean(x)	Bir x vektör elemanlarının ortalaması
median(x)	Bir x vektör elemanlarının orta değeri (medyanı)
sort(x)	Bir x vektör elemanlarının küçükten büyüğe sıralanması
max,min	En büyük ve en küçük değer bulma
sortrows	İlgili sütuna göre küçükten büyüğe sıralama

Programlama

if/end	Eğer koşulu
for/next	Döngü
while/end	while döngüsü
input	Değişken girdirme

Çizim

plot	2 boyutlu grafik
plot3	3 boyutlu grafik
hist,bar	Histogram ve bar grafikleri
surf, mesh	Yüzey grafikleri
figure	Çizim penceresi oluşturma (İki veya daha fazla sayıdaki grafikleri ayrı pencerelerde göstermek için)
hold on, hold off	Farklı grafikleri aynı eksen takımında çizdirme komutları
axis	Eksen komutu
axis equal	Eksenleri eşit ölçek katsayısı ile ölçeklendirme
stem	Çubuk grafik
plotyy	Çift y eksenli grafik
errorbar	Hata bar grafiği