

# مقدمه ای بر برنامه نویسی با MATLAB

سعید سهیلی  
استادیار گروه مکانیک  
دانشگاه آزاد اسلامی مشهد

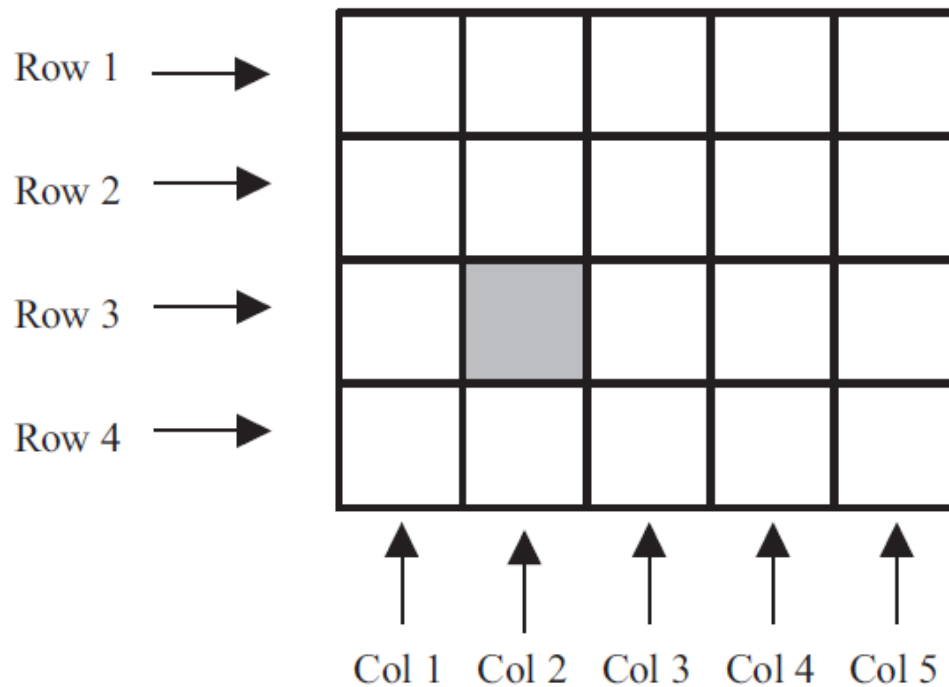
بسمہ اربعہ

فصل دوم

# اصول MATLAB و کار با آرایه ها

# متغیرها و آرایه ها

- اساسی ترین واحد داده در هر برنامه MATLAB آرایه است.
- آرایه مجموعه ای از مقادیر داده است که از سطرها و ستون ها تشکیل شده و دارای یک نام واحد است.

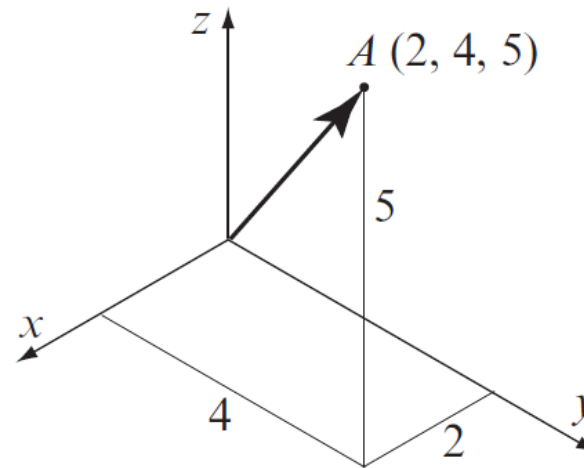


- بردارها: آرایه های یک بعدی
- ماتریس ها: آرایه هایی با دو یا بیش از دو بعد است.

# آرایه های یک بعدی

- تعدادی داده که به صورت سطری یا ستونی ذخیره شده اند.
- موقعیت نقطه در فضا با سه عدد مشخص می شود.

$$\mathbf{r}_A = 2\mathbf{i} + 4\mathbf{j} + 5\mathbf{k}$$



- جمعیت در سال های مختلف

| Year                  | 1984 | 1986 | 1988 | 1990 | 1992 | 1994 | 1996 |
|-----------------------|------|------|------|------|------|------|------|
| Population (millions) | 127  | 130  | 136  | 145  | 158  | 178  | 211  |

- با قرار دادن اعداد داخل براکت و انتساب به نام متغیر

```
variable_name = [ type vector elements ]
```

- بردار سطری: بین اعداد فاصله یا کاما استفاده شود.
- بردار ستونی: بین اعداد سمی کالون (;) یا بعد از هر عدد Enter زده شود.

```
>> yr=[1984 1986 1988 1990 1992 1994 1996]
```

The list of years is assigned to a row vector named yr.

```
yr =
```

```
    1984    1986    1988    1990    1992    1994  
    1996
```

```
>> pop=[127; 130; 136; 145; 158; 178; 211]
```

The population data is assigned to a column vector named pop.

```
pop =
```

```
    127  
    130  
    136  
    145  
    158
```

```
>> pntAH=[2, 4, 5]
```

```
pntAH =
```

```
      2      4      5
```

```
>> pntAV=[2
```

```
4
```

```
5]
```

```
pntAV =
```

```
      2
```

```
      4
```

```
      5
```

```
>>
```

The coordinates of point  $A$  are assigned to a row vector called pntAH.

The coordinates of point  $A$  are assigned to a column vector called pntAV.  
(The **Enter** key is pressed after each element is typed.)

- اگر فاصله بین اعداد ثابت باشد، با مشخص کردن عدد اول، فاصله بین اعداد (گام) و عدد آخر می توان بردار تعریف نمود. (پیش فرض گام عدد یک است)

```
variable_name = [m:q:n]
```

or

```
variable_name = m:q:n
```

(The brackets are optional.)

```
>> x=[1:2:13]
```

First element 1, spacing 2, last element 13.

```
x =
```

```
    1    3    5    7    9   11   13
```

```
>> y=[1.5:0.1:2.1]
```

First element 1.5, spacing 0.1, last element 2.1.

```
y =
```

```
    1.5000    1.6000    1.7000    1.8000    1.9000    2.0000  
    2.1000
```

```
>> z=[-3:7]
```

First element -3, last term 7.  
If spacing is omitted, the default is 1.

```
z =
```

```
    -3    -2    -1     0     1     2     3     4     5     6  
    7
```

```
>> xa=[21:-3:6]
```

First element 21, spacing -3, last term 6.

- می توان با دادن عدد اول، عدد آخر و تعداد تقسیمات نیز بردار تعریف نمود. (پیش فرض تعداد تقسیمات ۱۰۰ عدد است)

```
variable_name = linspace(xi,xf,n)
```

First element      Last element      Number of elements

```
>> va=linspace(0,8,6)
va =
    0    1.6000    3.2000    4.8000    6.4000    8.0000
>> vb=linspace(30,10,11)
vb =
    30    28    26    24    22    20    18    16    14    12    10
>> u=linspace(49.5,0.5)
u =
Columns 1 through 10
    49.5000    49.0051    48.5101    48.0152    47.5202    47.0253
    46.5303    46.0354    45.5404    45.0455
.....
Columns 91 through 100
     4.9545     4.4596     3.9646     3.4697     2.9747     2.4798
     1.9848     1.4899     0.9949     0.5000
```

6 elements, first element 0, last element 8.

11 elements, first element 30, last element 10.

First element 49.5, last element 0.5.

When the number of elements is omitted, the default is 100.

100 elements are displayed.

# آرایه های دو بعدی (ماتریس)

- ماتریس شامل اعداد در سطر و ستون است.
- اعداد یک جدول در یک ماتریس ذخیره می شوند.
- با قرار دادن اعداد داخل براکت و انتساب به نام متغیر ماتریس تعریف می شود

```
variable_name = [ type vector elements ]
```

- ابتدا اعداد اولین سطر وارد می شود. برای رفتن به سطر بعدی از سمی کالون (;) یا Enter استفاده می شود.

```
variable_name=[1st row elements; 2nd row elements; 3rd  
row elements; ... ; last row elements]
```

```
>> a=[5 35 43; 4 76 81; 21 32 40]
```

```
a =
```

```
    5    35    43
    4    76    81
   21    32    40
```

A semicolon is typed before  
a new line is entered.

```
>> b = [7 2 76 33 8
```

```
1 98 6 25 6
5 54 68 9 0]
```

The **Enter** key is pressed  
before a new line is entered.

```
b =
```

```
    7     2    76    33     8
    1    98     6    25     6
    5    54    68     9     0
```

```
>> cd=6; e=3; h=4;
```

Three variables are defined.

```
>> Mat=[e, cd*h, cos(pi/3); h^2, sqrt(h*h/cd), 14]
```

```
Mat =
```

```
    3.0000    24.0000    0.5000
   16.0000     1.6330   14.0000
```

Elements are defined  
by mathematical  
expressions.

```
>>
```

- در تعریف ماتریس تعداد اعداد سطرها حتما باید با هم یکسان باشند.

```
>> A=[1:2:11; 0:5:25; linspace(10,60,6); 67 2 43 68 4 13]
A =
     1         3         5         7         9        11
     0         5        10        15        20        25
    10        20        30        40        50        60
    67         2        43        68         4        13
>>
```

- برخی از توابع برای ایجاد ماتریس ها کمک می کنند.

**Table 2-1 MATLAB Functions Useful for Initializing Variables**

| Function                      | Purpose                                                                            |
|-------------------------------|------------------------------------------------------------------------------------|
| <code>zeros(n)</code>         | Generates an $n \times n$ matrix of zeros.                                         |
| <code>zeros(m,n)</code>       | Generates an $m \times n$ matrix of zeros.                                         |
| <code>zeros(size(arr))</code> | Generates a matrix of zeros of the same size as <code>arr</code> .                 |
| <code>ones(n)</code>          | Generates an $n \times n$ matrix of ones.                                          |
| <code>ones(m,n)</code>        | Generates an $m \times n$ matrix of ones.                                          |
| <code>ones(size(arr))</code>  | Generates a matrix of ones of the same size as <code>arr</code> .                  |
| <code>eye(n)</code>           | Generates an $n \times n$ identity matrix.                                         |
| <code>eye(m,n)</code>         | Generates an $m \times n$ identity matrix.                                         |
| <code>length(arr)</code>      | Returns the length of a vector, or the longest dimension of a 2-D array.           |
| <code>size(arr)</code>        | Returns two values specifying the number of rows and columns in <code>arr</code> . |

```
>> zr=zeros(3,4)
zr =
    0    0    0    0
    0    0    0    0
    0    0    0    0

>> ne=ones(4,3)
ne =
    1    1    1
    1    1    1
    1    1    1
    1    1    1

>> idn=eye(5)
idn =
    1    0    0    0    0
    0    1    0    0    0
    0    0    1    0    0
    0    0    0    1    0
    0    0    0    0    1
```

- با استفاده از اپراتور ترانهاده می توان سطر را به ستون و بلعکس تبدیل نمود.

```
>> aa=[3 8 1]
```

Define a row vector aa.

```
aa =
```

```
3      8      1
```

```
>> bb=aa'
```

Define a column vector bb as the transpose of vector aa.

```
bb =
```

```
3
```

```
8
```

```
1
```

```
>> C=[2 55 14 8; 21 5 32 11; 41 64 9 1]
```

Define a matrix C with 3 rows and 4 columns.

```
C =
```

```
2      55      14      8
```

```
21      5      32     11
```

```
41     64      9      1
```

```
>> D=C'
```

```
D =
```

```
2      21     41
```

```
55      5     64
```

```
14     32      9
```

```
8      11      1
```

Define a matrix D as the transpose of matrix C. (D has 4 rows and 3 columns.)

```
>>
```

# آدرس دهی

- بردارها: نام متغیر نوشته می شود و شماره عدد دلخواه داخل پرانتز وارد می شود.

```
>> VCT=[35 46 78 23 5 14 81 3 55]
```

Define a vector.

```
VCT =
```

```
    35    46    78    23     5    14    81     3    55
```

```
>> VCT(4)
```

Display the fourth element.

```
ans =  
    23
```

Assign a new value to the sixth element.

```
>> VCT(6)=273
```

The whole vector is displayed.

```
VCT =
```

```
    35    46    78    23     5   273    81     3    55
```

```
>> VCT(2)+VCT(8)
```

Use the vector elements in mathematical expressions.

```
ans =  
    49
```

```
>> VCT(5)^VCT(8)+sqrt(VCT(7))
```

```
ans =
```

```
   134
```

- ماتریس ها: نام ماتریس نوشته می شود و داخل پرانتز سطر و ستون عدد دلخواه وارد می شود.

```
>> MAT=[3 11 6 5; 4 7 10 2; 13 9 0 8]
```

Create a  $3 \times 4$  matrix.

```
MAT =
```

|    |    |    |   |
|----|----|----|---|
| 3  | 11 | 6  | 5 |
| 4  | 7  | 10 | 2 |
| 13 | 9  | 0  | 8 |

```
>> MAT(3,1)=20
```

Assign a new value to the (3,1) element.

```
MAT =
```

|    |    |    |   |
|----|----|----|---|
| 3  | 11 | 6  | 5 |
| 4  | 7  | 10 | 2 |
| 20 | 9  | 0  | 8 |

```
>> MAT(2,4)-MAT(1,2)
```

Use elements in a mathematical expression.

```
ans =
```

```
-9
```

# استفاده از کولون (:) در آدرس دهی

• بردارها:

- $va(:)$  تمامی اعضا بردار  $va$  را نمایش می دهد.
- $va(m:n)$  تمامی اعضا از  $m$  تا  $n$  را نمایش می دهد.

```
>> v=[4 15 8 12 34 2 50 23 11]
```

A vector v is created.

```
v =
```

```
      4      15      8      12      34      2      50      23      11
```

```
>> u=v(3:7)
```

```
u =
```

```
      8      12      34      2      50
```

A vector u is created from the elements 3 through 7 of vector v.

• ماتریس ها:

- $A(:, n)$  تمامی سطرها، عضو  $n$ ام را نمایش می دهد. (ستون  $n$ )
- $A(n, :)$  سطر  $n$  تمامی ستون هایش را نمایش می دهد. (سطر  $n$ )
- $A(:, m:n)$  تمامی سطرها، اعضای ستون  $m$  تا  $n$
- $A(m:n, :)$  تمامی ستون ها، اعضای سطر  $m$  تا  $n$
- $A(m:n, p:q)$  تمامی اعضا سطر  $m$  تا  $n$  و ستون  $p$  تا  $q$

```
>> A=[1 3 5 7 9 11; 2 4 6 8 10 12; 3 6 9 12 15 18; 4 8 12 16  
20 24; 5 10 15 20 25 30]
```

A =

|   |    |    |    |    |    |
|---|----|----|----|----|----|
| 1 | 3  | 5  | 7  | 9  | 11 |
| 2 | 4  | 6  | 8  | 10 | 12 |
| 3 | 6  | 9  | 12 | 15 | 18 |
| 4 | 8  | 12 | 16 | 20 | 24 |
| 5 | 10 | 15 | 20 | 25 | 30 |

```
>> B=A(:, 3)
```

Define a matrix A with  
5 rows and 6 columns.

Define a column  
vector B from the  
elements in all of  
the rows of column  
3 in matrix A.

B =

5  
6  
9  
12  
15

>> C=A(2,:)

C =

2 4 6 8 10 12

>> E=A(2:4,:)

E =

2 4 6 8 10 12  
3 6 9 12 15 18  
4 8 12 16 20 24

>> F=A(1:3,2:4)

F =

3 5 7  
4 6 8  
6 9 12

>>

Define a row vector C from the elements in all of the columns of row 2 in matrix A.

Define a matrix E from the elements in rows 2 through 4 and all the columns in matrix A.

Create a matrix F from the elements in rows 1 through 3 and columns 2 through 4 in matrix A.

```
>> v=4:3:34
```

Create a vector v with 11 elements.

```
v =
```

```
    4    7   10   13   16   19   22   25   28   31   34
```

```
>> u=v([3, 5, 7:10])
```

Create a vector u from the 3rd, the 5th, and the 7th through 10th elements of v.

```
u =
```

```
   10   16   22   25   28   31
```

```
>> A=[10:-1:4; ones(1,7); 2:2:14; zeros(1,7)]
```

```
A =
```

```
   10    9    8    7    6    5    4
    1    1    1    1    1    1    1
    2    4    6    8   10   12   14
    0    0    0    0    0    0    0
```

Create a  $4 \times 7$  matrix A.

```
>> B = A([1,3],[1,3,5:7])
```

Create a matrix B from the 1st and 3rd rows, and 1st, 3rd, and the 5th through 7th columns of A.

```
B =
```

```
   10    8    6    5    4
    2    6   10   12   14
```

# اضافه کردن عدد به متغیر موجود

- اعداد در مکان آدرس داده شده قرار می گیرند.

```
>> DF=1:4
```

Define vector DF with 4 elements.

```
DF =
```

```
1     2     3     4
```

```
>> DF(5:10)=10:5:35
```

Adding 6 elements starting with the 5th.

```
DF =
```

```
1     2     3     4    10    15    20    25    30    35
```

```
>> AD=[5 7 2]
```

Define vector AD with 3 elements.

```
AD =
```

```
5     7     2
```

```
>> AD(8)=4
```

Assign a value to the 8th element.

```
AD =
```

```
5     7     2     0     0     0     0     4
```

MATLAB assigns zeros to the 4th through 7th elements.

```
>> AR(5)=24
```

Assign a value to the 5th element of a new vector.

```
AR =
```

```
0     0     0     0    24
```

MATLAB assigns zeros to the 1st through 4th elements.

```
>>
```

```
>> E=[1 2 3 4; 5 6 7 8]
```

Define a  $2 \times 4$  matrix E.

```
E =
```

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |

```
>> E(3,:)=[10:4:22]
```

Add the vector 10 14 18 22 as the third row of E.

```
E =
```

|    |    |    |    |
|----|----|----|----|
| 1  | 2  | 3  | 4  |
| 5  | 6  | 7  | 8  |
| 10 | 14 | 18 | 22 |

```
>> K=eye(3)
```

Define a  $3 \times 3$  matrix K.

```
K =
```

|   |   |   |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

```
>> G=[E K]
```

Append matrix K to matrix E. The numbers of rows in E and K must be the same.

```
G =
```

|    |    |    |    |   |   |   |
|----|----|----|----|---|---|---|
| 1  | 2  | 3  | 4  | 1 | 0 | 0 |
| 5  | 6  | 7  | 8  | 0 | 1 | 0 |
| 10 | 14 | 18 | 22 | 0 | 0 | 1 |

```
>> AW=[3 6 9; 8 5 11]
```

Define a  $2 \times 3$  matrix.

```
AW =
```

|   |   |    |
|---|---|----|
| 3 | 6 | 9  |
| 8 | 5 | 11 |

```
>> AW(4,5)=17
```

Assign a value to the (4,5) element.

```
AW =
```

|   |   |    |   |    |
|---|---|----|---|----|
| 3 | 6 | 9  | 0 | 0  |
| 8 | 5 | 11 | 0 | 0  |
| 0 | 0 | 0  | 0 | 0  |
| 0 | 0 | 0  | 0 | 17 |

MATLAB changes the matrix size to  $4 \times 5$ , and assigns zeros to the new elements.

```
>> BG(3,4)=15
```

Assign a value to the (3,4) element of a new matrix.

```
BG =
```

|   |   |   |    |
|---|---|---|----|
| 0 | 0 | 0 | 0  |
| 0 | 0 | 0 | 0  |
| 0 | 0 | 0 | 15 |

MATLAB creates a  $3 \times 4$  matrix and assigns zeros to all the elements except  $BG(3,4)$ .

```
>>
```

```
» arr4 = [1 2 3 4; 5 6 7 8; 9 10 11 12]
```

```
arr4 =
```

|   |    |    |    |
|---|----|----|----|
| 1 | 2  | 3  | 4  |
| 5 | 6  | 7  | 8  |
| 9 | 10 | 11 | 12 |

- اگر مقادیر به زیر آرایه نسبت داده شود، فقط مقادیر نسبت داده شده عوض می شوند.

```
» arr4(1:2,[1 4]) = [20 21; 22 23]
```

```
arr4 =
```

|    |    |    |    |
|----|----|----|----|
| 20 | 2  | 3  | 21 |
| 22 | 6  | 7  | 23 |
| 9  | 10 | 11 | 12 |

- اگر مقادیری به آرایه نسبت داده شود، تمامی مقادیر آرایه تغییر می کند

```
» arr4 = [20 21; 22 23]
```

```
arr4 =
```

|    |    |
|----|----|
| 20 | 21 |
| 22 | 23 |

# حذف یک عدد در متغیر

- با آدرس دادن به عدد مورد نظر یک براکت خالی [] به آن نسبت داده می شود.

```
>> kt=[2 8 40 65 3 55 23 15 75 80]
```

kt =  
2 8 40 65 3 55 23 15 75 80

Define a vector with 10 elements.

```
>> kt(6)=[]
```

Eliminate the 6th element.

```
kt =  
2 8 40 65 3 23 15 75 80
```

The vector now has 9 elements.

```
>> kt(3:6)=[]
```

Eliminate elements 3 through 6.

```
kt =  
2 8 15 75 80
```

The vector now has 5 elements.

```
>> mtr=[5 78 4 24 9; 4 0 36 60 12; 56 13 5 89 3]
```

Define a 3×5 matrix.

```
mtr =  
5 78 4 24 9  
4 0 36 60 12  
56 13 5 89 3
```

```
>> mtr(:,2:4)=[]
```

Eliminate all the rows of columns 2 through 4.

```
mtr =  
5 9  
4 12  
56 3
```

```
>>
```

## رشته به عنوان متغیر

- کلمات و یا جملات انگلیسی می توانند به متغیر نسبت داده شوند.

```
>> a='FRty 8'
a =
FRty 8
>> B='My name is John Smith'
B =
My name is John Smith
>>
```

- هر کاراکتر و یا فاصله (شبه اعداد) در آرایه نسبت داده شده ذخیره می شود.

```
>> B(4)
ans =
n
>> B(12)
ans =
J
```

- مشابه بردارهای عددی با ادرس دادن می توان بخشی از آرایه را عوض نمود.

```
>> B(12:15)='Bill'  
B =  
My name is Bill Smith  
>>
```

Using a colon to assign new characters to elements 12 through 15 in the vector B.

# عملیات بر روی مقادیر اسکالر و آرایه ها

- محاسبات در عبارات انتسابی انجام می شود.
  - مقدار عبارت سمت راست تساوی محاسبه و نتیجه به عبارت سمت چپ نسبت داده می شود
- ```
variable_name = expression;  
  
ii = ii + 1;
```
- به طور معمول Matlab عملیات اصلی را با فرض آرایه بودن انجام می دهد.
  - برای انجام عملیات اسکالر قبل از عملگر نقطه گذاشته شود.

\* \*

# عملیات معمول آرایه ای و ماتریسی

| Operation             | MATLAB Form      | Comments                                                                                                                                                               |
|-----------------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Array Addition        | $a + b$          | Array addition and matrix addition are identical.                                                                                                                      |
| Array Subtraction     | $a - b$          | Array subtraction and matrix subtraction are identical.                                                                                                                |
| Array Multiplication  | $a .* b$         | Element-by-element multiplication of $a$ and $b$ . Both arrays must be the same shape, or one of them must be a scalar.                                                |
| Matrix Multiplication | $a * b$          | Matrix multiplication of $a$ and $b$ . The number of columns in $a$ must equal the number of rows in $b$ .                                                             |
| Array Right Division  | $a ./ b$         | Element-by-element division of $a$ and $b$ : $a(i, j) / b(i, j)$ . Both arrays must be the same shape, or one of them must be a scalar.                                |
| Array Left Division   | $a .\ b$         | Element-by-element division of $a$ and $b$ , but with $b$ in the numerator: $b(i, j) / a(i, j)$ . Both arrays must be the same shape, or one of them must be a scalar. |
| Matrix Right Division | $a / b$          | Matrix division defined by $a * \text{inv}(b)$ , where $\text{inv}(b)$ is the inverse of matrix $b$ .                                                                  |
| Matrix Left Division  | $a \backslash b$ | Matrix division defined by $\text{inv}(a) * b$ , where $\text{inv}(a)$ is the inverse of matrix $a$ .                                                                  |
| Array Exponentiation  | $a .^ b$         | Element-by-element exponentiation of $a$ and $b$ : $a(i, j) ^ b(i, j)$ . Both arrays must be the same shape, or one of them must be a scalar.                          |

### Example 2.1—Array and Matrix Operations

Assume that  $a$ ,  $b$ ,  $c$ , and  $d$  are defined as follows:

$$a = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} -1 & 2 \\ 0 & 1 \end{bmatrix}$$

$$c = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$d = 5$$

What is the result of each of the following expressions?

$$(a) \ a + b$$

$$(e) \ a + c$$

$$(b) \ a .* b$$

$$(f) \ a + d$$

$$(c) \ a * b$$

$$(g) \ a .* d$$

$$(d) \ a * c$$

$$(h) \ a * d$$

- از عملگر تقسیم چپ ( \ ) می توان برای حل دستگاه معادلات جبری استفاده نمود.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\text{where } \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \text{ and } \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

$\mathbf{A} \setminus \mathbf{b}$  is defined to be  $\text{inv}(\mathbf{A}) * \mathbf{b}$ ,

$$\begin{aligned} 2x_1 + 5x_2 &= 11 \\ 3x_1 - 2x_2 &= -12 \end{aligned}$$

$$\begin{bmatrix} 2 & 5 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 11 \\ -12 \end{bmatrix}$$

```

» A = [2 5; 3 -2];
» b = [11; -12];
» x = inv(A) * b
x =
    -2.0000
     3.0000

```

```

» x = A \ b
x =
    -2
     3

```

$$4x - 2y + 6z = 8$$

$$2x + 8y + 2z = 4$$

$$6x + 10y + 3z = 0$$

$$\begin{bmatrix} 4 & -2 & 6 \\ 2 & 8 & 2 \\ 6 & 10 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix}$$

```
>> A=[4 -2 6; 2 8 2; 6 10 3];
```

Solving the form  $AX = B$ .

```
>> B=[8; 4; 0];
```

```
>> X=A\B
```

Solving by using left division:  $X = A \setminus B$ .

```
X =
```

```
    -1.8049
```

```
     0.2927
```

```
     2.6341
```

```
>> Xb=inv(A)*B
```

Solving by using the inverse of  $A$ :  $X = A^{-1}B$ .

```
Xb =
```

```
    -1.8049
```

```
     0.2927
```

```
     2.6341
```

## برخی توابع برای انجام عملیات روی آرایه ها

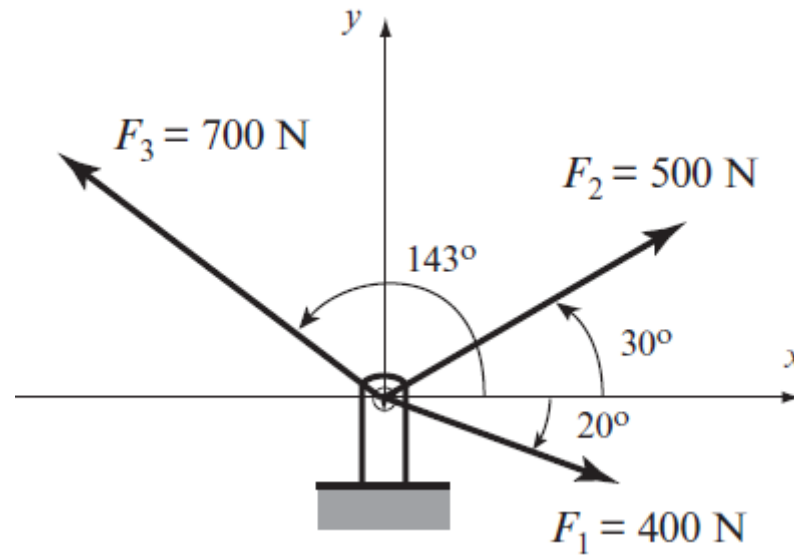
| Function      | Description                                                                                                                              | Example                                                                      |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| mean (A)      | If A is a vector, returns the mean value of the elements of the vector.                                                                  | <pre>&gt;&gt; A = [5 9 2 4]; &gt;&gt; mean(A) ans =     5</pre>              |
| C=max (A)     | If A is a vector, C is the largest element in A. If A is a matrix, C is a row vector containing the largest element of each column of A. | <pre>&gt;&gt; A = [5 9 2 4 11 6 11 11]; &gt;&gt; C = max(A) C =     11</pre> |
| [d,n]=max (A) | If A is a vector, d is the largest element in A, and n is the position of the element (the first if several have the max value).         | <pre>&gt;&gt; [d,n] = max(A) d =     11 n =     5</pre>                      |
| min (A)       | The same as max (A) , but for the smallest element.                                                                                      | <pre>&gt;&gt; A = [5 9 2 4]; &gt;&gt; min(A) ans =     2</pre>               |
| [d,n]=min (A) | The same as [d,n] = max (A) , but for the smallest element.                                                                              |                                                                              |

|                        |                                                                                                            |                                                                                   |
|------------------------|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| <code>sum(A)</code>    | If A is a vector, returns the sum of the elements of the vector.                                           | <pre>&gt;&gt; A=[5 9 2 4]; &gt;&gt; sum(A) ans =     20</pre>                     |
| <code>sort(A)</code>   | If A is a vector, arranges the elements of the vector in ascending order.                                  | <pre>&gt;&gt; A=[5 9 2 4]; &gt;&gt; sort(A) ans =      2     4     5     9</pre>  |
| <code>median(A)</code> | If A is a vector, returns the median value of the elements of the vector.                                  | <pre>&gt;&gt; A=[5 9 2 4]; &gt;&gt; median(A) ans =     4.5000</pre>              |
| <code>std(A)</code>    | If A is a vector, returns the standard deviation of the elements of the vector.                            | <pre>&gt;&gt; A=[5 9 2 4]; &gt;&gt; std(A) ans =     2.9439</pre>                 |
| <code>det(A)</code>    | Returns the determinant of a square matrix A.                                                              | <pre>&gt;&gt; A=[2 4; 3 5]; &gt;&gt; det(A) ans =     -2</pre>                    |
| <code>dot(a,b)</code>  | Calculates the scalar (dot) product of two vectors a and b. The vectors can each be row or column vectors. | <pre>&gt;&gt; a=[1 2 3]; &gt;&gt; b=[3 4 5]; &gt;&gt; dot(a,b) ans =     26</pre> |

|                         |                                                                                                                     |                                                                                                                                                                       |
|-------------------------|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>cross(a,b)</code> | Calculates the cross product of two vectors a and b, (a <b>x</b> b). The two vectors must have each three elements. | <pre>&gt;&gt; a=[1 3 2]; &gt;&gt; b=[2 4 1]; &gt;&gt; cross(a,b) ans =     -5     3    -2</pre>                                                                       |
| <code>inv(A)</code>     | Returns the inverse of a square matrix A.                                                                           | <pre>&gt;&gt; A=[2 -2 1; 3 2 -1; 2 -3 2]; &gt;&gt; inv(A) ans =     0.2000    0.2000          0    -1.6000    0.4000     1.0000    -2.6000    0.4000     2.0000</pre> |

# مثال) نیروی معادل (جمع بردارها)

• بردار نیروی معادل را پیدا کنید.



$$\mathbf{F} = F_x \mathbf{i} + F_y \mathbf{j} = F \cos \theta \mathbf{i} + F \sin \theta \mathbf{j} = F(\cos \theta \mathbf{i} + \sin \theta \mathbf{j})$$

$$F = \sqrt{F_x^2 + F_y^2} \quad \text{and} \quad \tan \theta = \frac{F_y}{F_x}$$

```
% Sample Problem 3-2 solution (script file)
```

```
clear
```

```
F1M=400; F2M=500; F3M=700;
```

Define variables with the magnitude of each vector.

```
Th1=-20; Th2=30; Th3=143;
```

Define variables with the angle of each vector.

```
F1=F1M*[cosd(Th1) sind(Th1)]
```

```
F2=F2M*[cosd(Th2) sind(Th2)]
```

```
F3=F3M*[cosd(Th3) sind(Th3)]
```

Define the three vectors.

```
Ftot=F1+F2+F3
```

Calculate the total force vector.

```
FtotM=sqrt(Ftot(1)^2+Ftot(2)^2)
```

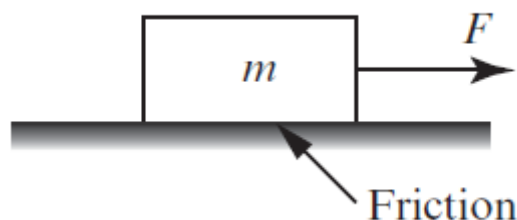
Calculate the magnitude of the total force vector.

```
Th=atand(Ftot(2)/Ftot(1))
```

Calculate the angle of the total force vector.

## مثال) محاسبه اصطکاک (میانگین گیری)

- ضریب اصطکاک با محاسبه نیروی لازم برای به حرکت درآوردن جرم های مختلف قابل محاسبه است.



$$\mu = F/(mg) \quad (g = 9.81 \text{ m/s}^2).$$

| Test          | 1    | 2    | 3  | 4  | 5   | 6   |
|---------------|------|------|----|----|-----|-----|
| Mass $m$ (kg) | 2    | 4    | 5  | 10 | 20  | 50  |
| Force $F$ (N) | 12.5 | 23.5 | 30 | 61 | 117 | 294 |

```
>> m=[2 4 5 10 20 50];
```

Enter the values of m in a vector.

```
>> F=[12.5 23.5 30 61 117 294];
```

Enter the values of F in a vector.

```
>> mu=F./(m*9.81)
```

A value for mu is calculated for each test, using element-by-element calculations.

```
mu =
```

```
    0.6371    0.5989    0.6116    0.6218    0.5963    0.5994
```

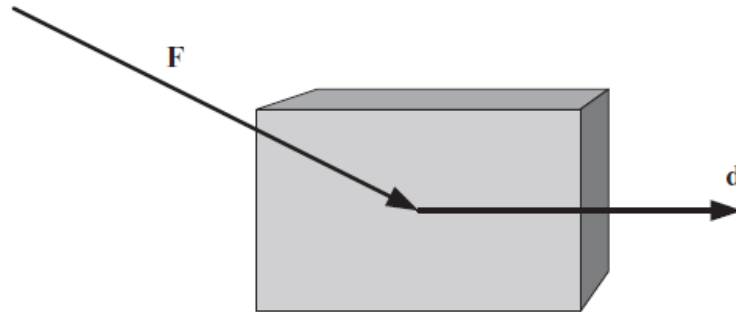
```
>> mu_ave=mean(mu)
```

The average of the elements in the vector mu is determined by using the function mean.

```
mu_ave =  
    0.6109
```

## مثال) کار انجام شده روی ذره (ضرب داخلی)

$$W = \mathbf{F} \cdot \mathbf{d}$$



$$\mathbf{F} = 10 \hat{\mathbf{i}} - 4 \hat{\mathbf{j}} \text{ N} \qquad \mathbf{d} = 5 \hat{\mathbf{i}} \text{ m.}$$

$$W = \mathbf{F} \cdot \mathbf{d} = (10 \hat{\mathbf{i}} - 4 \hat{\mathbf{j}}) \cdot (5 \hat{\mathbf{i}}) = 50 \text{ J}$$

```
>> F = [10 -4];
```

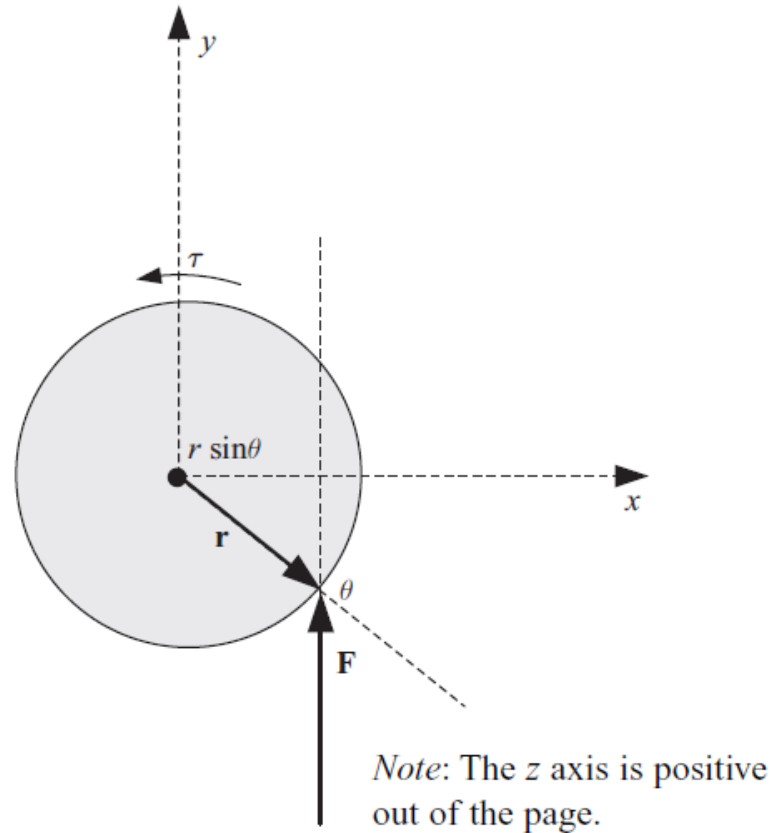
```
>> d = [5 0];
```

```
>> W = dot(F,d)
```

```
W =
```

```
50
```

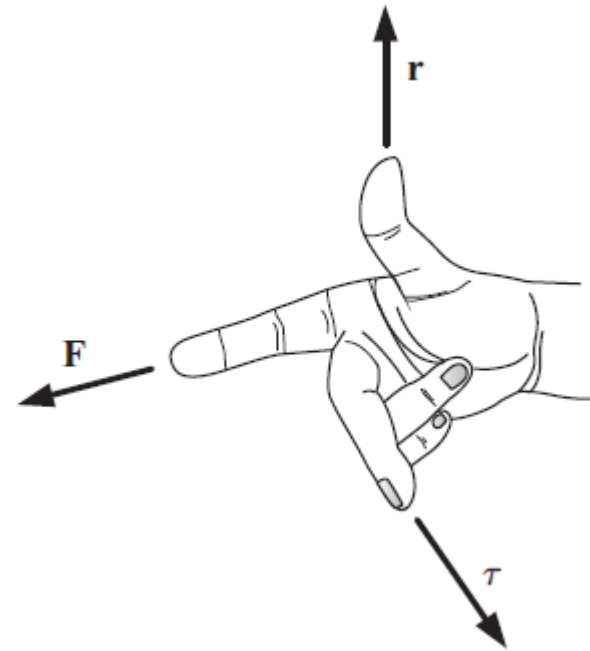
## مثال) گشتاور روی محور موتور (ضرب خارجی)



The torque on an object is a product of the force applied to the object and the perpendicular distance between the line of the force and the point of rotation.

$$\tau = rF \sin \theta$$

$$\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F}$$



• گشتاور وارده را با داشتن بردار نیرو و بردار  $\mathbf{r}$  محاسبه کنید.

$\mathbf{r} = 0.866\hat{\mathbf{i}} - 0.5\hat{\mathbf{j}}\text{ m}$ , and  $\mathbf{F} = 5\hat{\mathbf{j}}\text{ N}$ .

```
>> r = [0.866 -0.5 0];
```

```
>> F = [0 5 0];
```

```
>> tau = cross(r,F)
```

```
tau =
```

```
0 0 4.3300
```

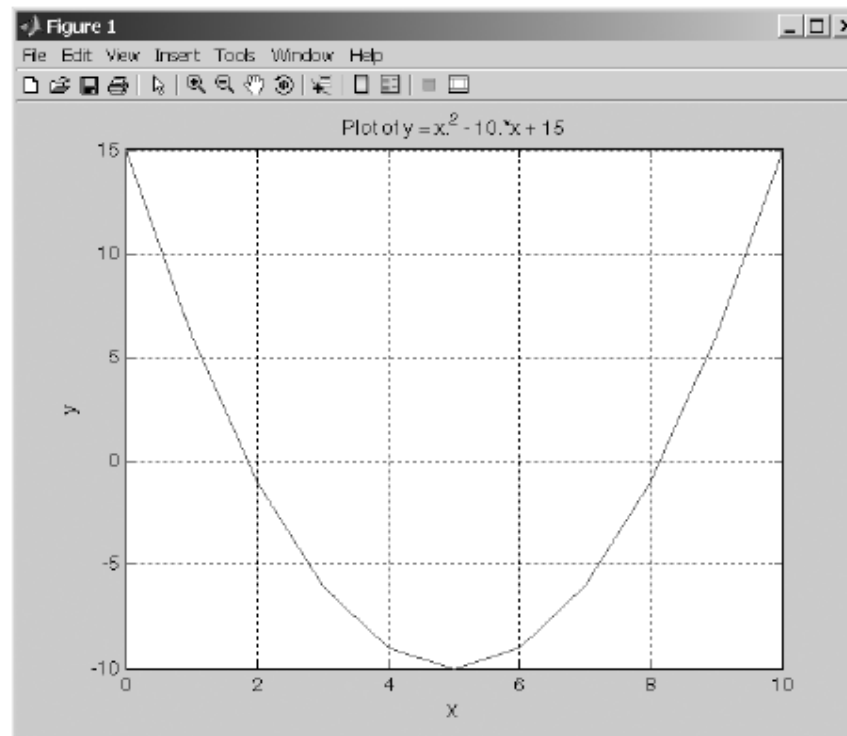
# ترسیم نمودار تکمیلی

- برای رسم نمودار از دستور `plot` استفاده می شود

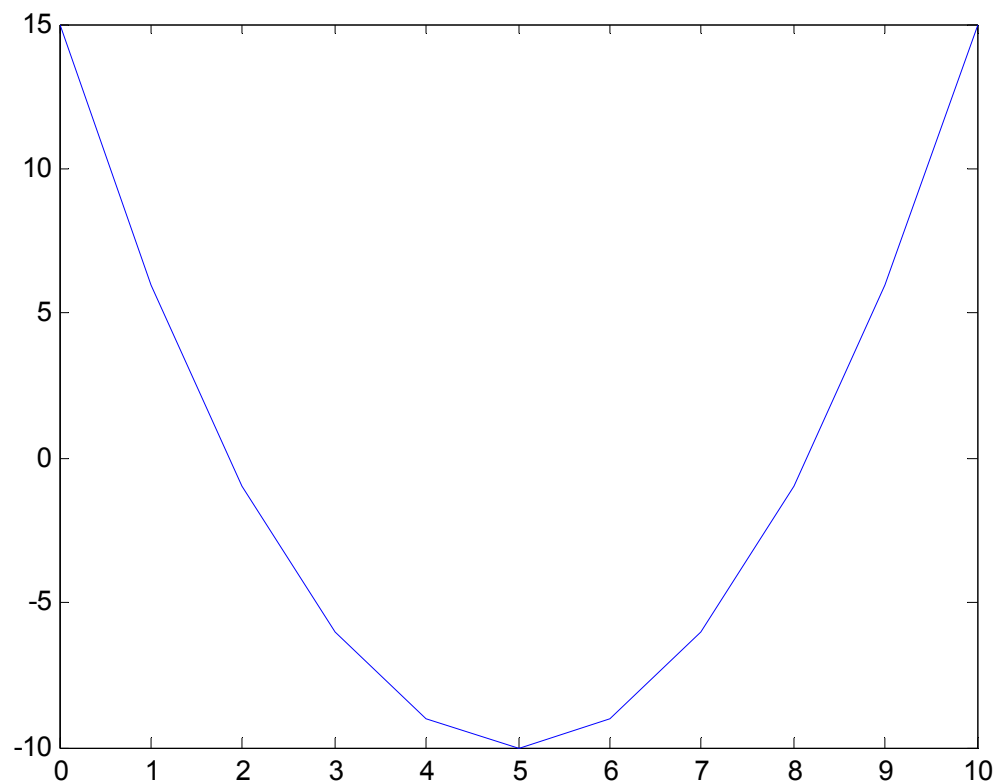
```
x = 0:1:10;  
y = x.^2 - 10.*x + 15;  
plot(x,y);
```

- می توان تیترا برای نمودار و محورها ایجاد کرد

```
title ('Plot of  $y = x.^2 - 10.*x + 15$ ');  
xlabel ('x');  
ylabel ('y');  
grid on;
```



• برای کپی کردن نمودار از Edit و copy figure استفاده کنید

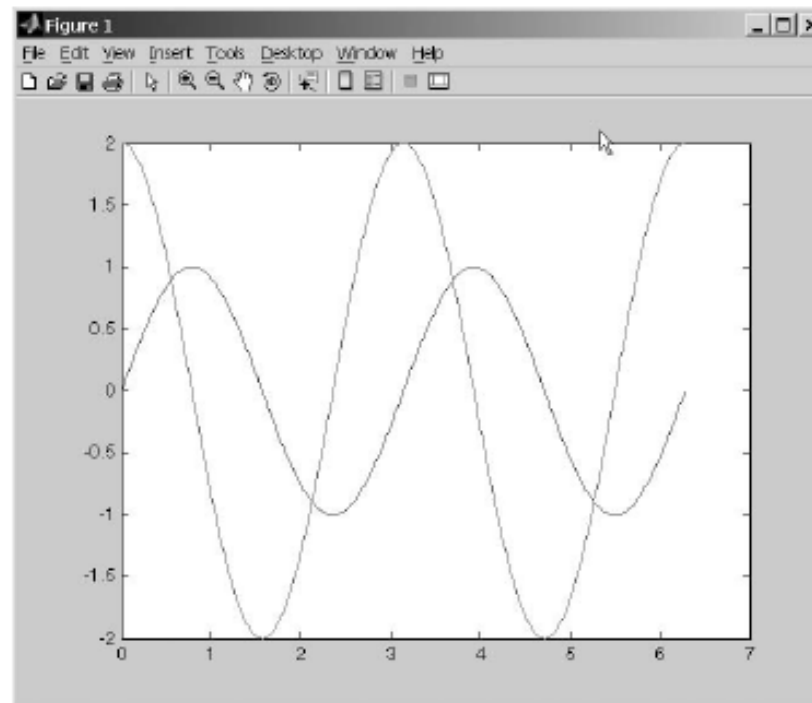


# نمودارهای چندگانه

$$f(x) = \sin 2x$$

$$f'(x) = \frac{d}{dx} \sin 2x = 2 \cos 2x$$

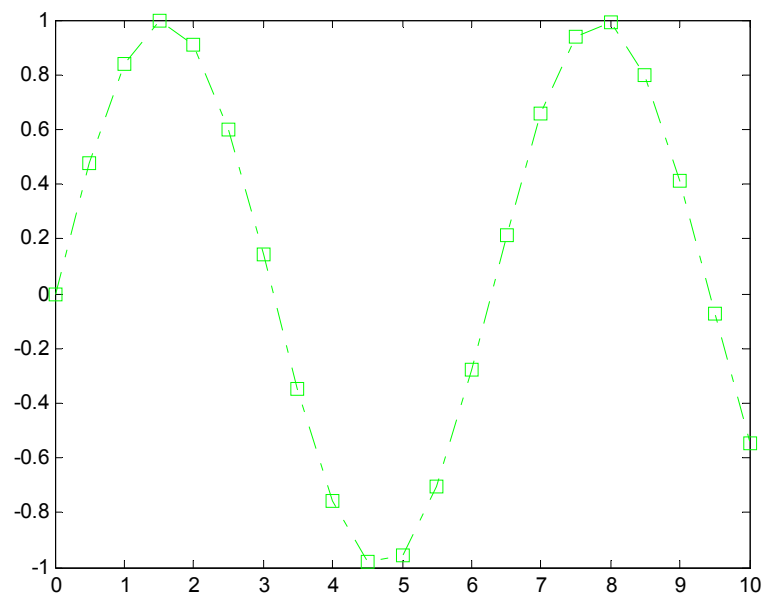
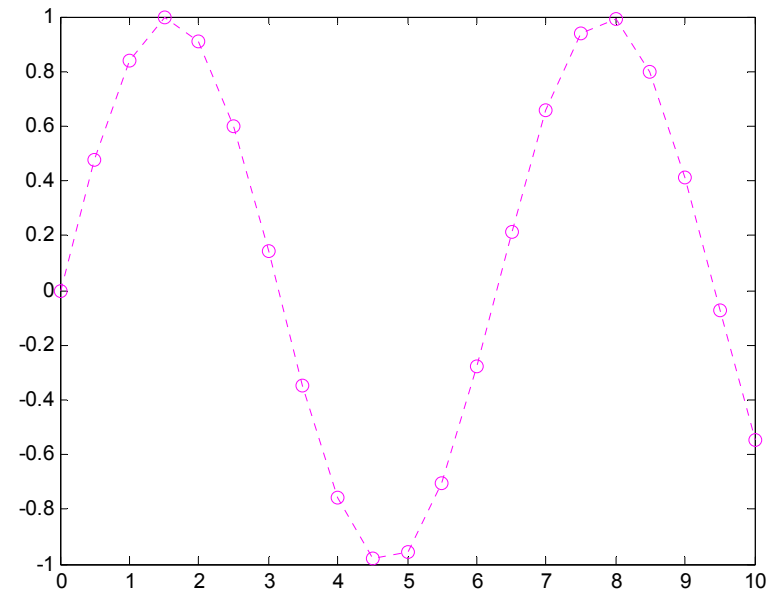
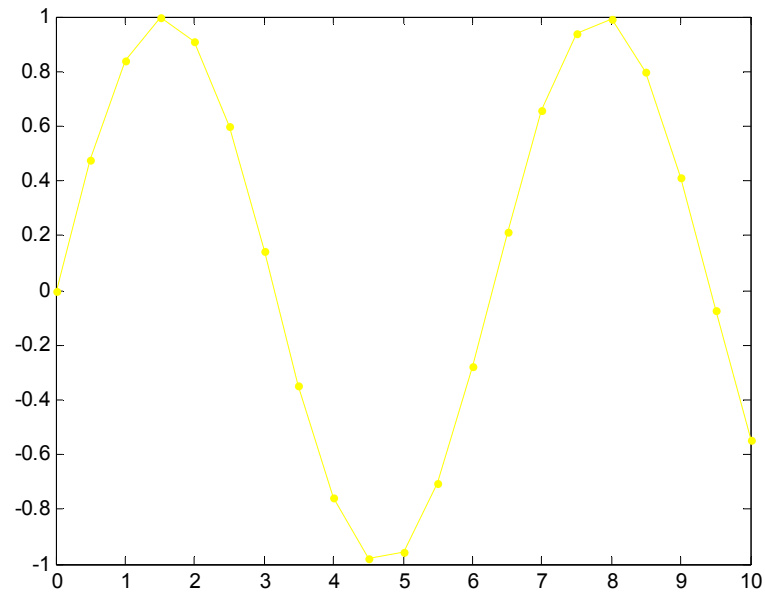
```
x = 0:pi/100:2*pi;  
y1 = sin(2*x);  
y2 = 2*cos(2*x);  
plot(x,y1,x,y2);
```



- بعد از X و Y می توان با وارد کردن سه کاراکتر به ترتیب رنگ خطوط، مدل نشانه گذاری و مدل خطوط را تغییر داد.

| Color |         | Marker Style |                  | Line Style |          |
|-------|---------|--------------|------------------|------------|----------|
| y     | yellow  | .            | point            | -          | solid    |
| m     | magenta | o            | circle           | :          | dotted   |
| c     | cyan    | x            | x-mark           | -.         | dash-dot |
| r     | red     | +            | plus             | - -        | dashed   |
| g     | green   | *            | star             | <none>     | no lines |
| b     | blue    | s            | square           |            |          |
| w     | white   | d            | diamond          |            |          |
| k     | black   | v            | triangle (down)  |            |          |
|       |         | ^            | triangle (up)    |            |          |
|       |         | <            | triangle (left)  |            |          |
|       |         | >            | triangle (right) |            |          |
|       |         | p            | pentagram        |            |          |
|       |         | h            | hexagram         |            |          |
|       |         | <none>       | no marker        |            |          |

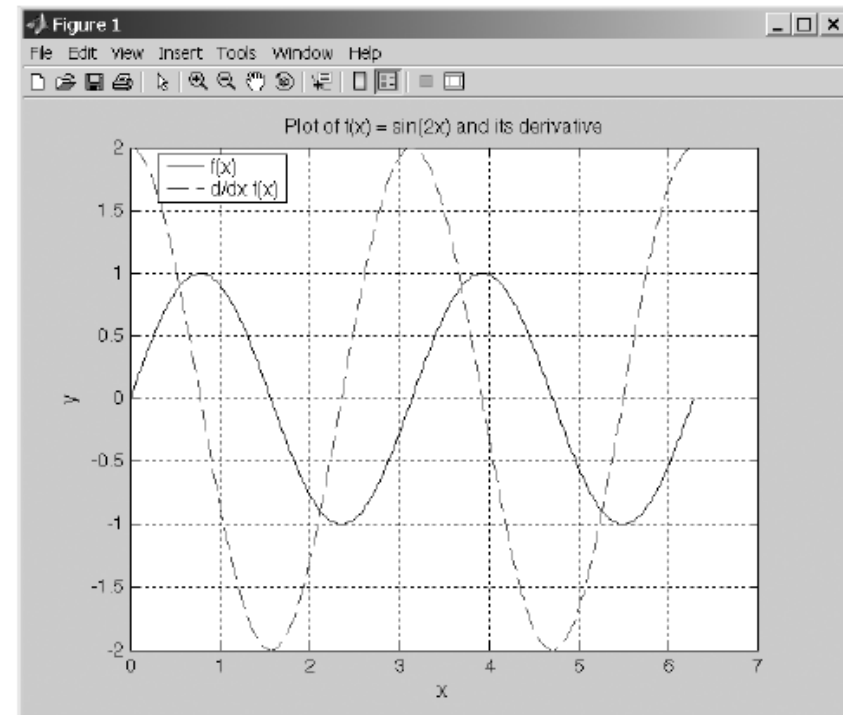
- >> plot(x,y, 'y.-')
- >> plot(x,y, 'mo:')
- >> plot(x,y, 'gs-.')



- با دستور **legend** برای نمودارها می توان راهنما ایجاد کرد.

```
legend('string1','string2', . . . , pos)
```

```
x = 0:pi/100:2*pi;  
y1 = sin(2*x);  
y2 = 2*cos(2*x);  
plot(x,y1,'k-',x,y2,'b--');  
title ('Plot of f(x) = sin(2x) and its derivative');  
xlabel ('x');  
ylabel ('y');  
legend ('f(x)', 'd/dx f(x)', 't1')  
grid on;
```



# مقداردهی به وسیله ورودی صفحه کلید

- می توان در حین اجرای برنامه از کاربر مقدار متغیر را دریافت نمود.

```
» in1 = input('Enter data: ');  
Enter data: 1.23
```

داده ها به صورت  
رشته ذخیره می  
شوند

```
» in2 = input('Enter data: ','s');  
Enter data: 1.23
```



# نمایش خروجی با دستور disp

- برای نمایش مقادیر عددی و رشته ای استفاده می شود.

`disp(name of a variable)` or `disp('text as string')`

```
>> abc = [5 9 1; 7 2 4]
```

A 2×3 array is assigned to variable abc.

```
>> disp(abc)
```

The disp command is used to display the abc array.

```
5     9     1
7     2     4
```

The array is displayed without its name.

```
>> disp('The problem has no solution.')
```

The disp command is used to display a message.

```
The problem has no solution.
>>
```

|                                                       |                                                           |
|-------------------------------------------------------|-----------------------------------------------------------|
| <code>yr=[1984 1986 1988 1990 1992 1994 1996];</code> | The population data is entered in two row vectors.        |
| <code>pop=[127 130 136 145 158 178 211];</code>       |                                                           |
| <code>tableYP(:,1)=yr'</code>                         | yr is entered as the first column in the array tableYP.   |
| <code>tableYP(:,2)=pop';</code>                       | pop is entered as the second column in the array tableYP. |
| <code>disp('           YEAR       POPULATION')</code> | Display heading (first line).                             |
| <code>disp('                   (MILLIONS)')</code>    | Display heading (second line).                            |
| <code>disp(' ')</code>                                | Display an empty line.                                    |
| <code>disp(tableYP)</code>                            | Display the array tableYP.                                |

- می توان با کمک دستور `num2str` مقادیر عددی را در کنار رشته و کاراکتر نمایش داد

```
str = ['The value of pi = ' num2str(pi)];  
disp (str);
```

```
the value of pi = 3.1416
```

# برنامه تبدیل دما

- برنامه ای بنویسید که با دریافت دما بر حسب فارنهایت آن را به کلوین تبدیل کند.

$$T_K = \left[ \frac{5}{9}(T_F - 32.0) \right] + 273.15$$

- برنامه باید مراحل زیر را انجام دهد.
  - درخواست وارد کردن دما بر حسب فارنهایت
  - دریافت دمای ورودی
  - محاسبه دما بر حسب کلوین
  - نمایش نتایج

- قدم اول) دریافت ورودی های مساله

```
temp_f = input('Enter the temperature in degrees Fahrenheit:');
```

- قدم دوم) انجام محاسبات

```
temp_k = (5/9) * (temp_f - 32) + 273.15;
```

- قدم سوم) نمایش نتایج

نمایش عبارت  
داخل کوتیشن

نمایش عبارت  
داخل کوتیشن

```
disp([ num2str(temp_f) 'degrees Fahrenheit = ' num2str(temp_k) 'kelvin');
```

تبدیل عدد  
دمای فارنهایت  
به رشته

تبدیل عدد  
دمای کلوین به  
رشته

# برنام کامل به همراه توضیحات برای کاربر

```
% Script file: temp_conversion
%
% Purpose:
%   To convert an input temperature from degrees Fahrenheit to
%   an output temperature in kelvin.
%
% Record of revisions:
%   Date          Engineer      Description of change
%   ====          =====
%   01/03/10      S. J. Chapman  Original code
%
% Define variables:
%   temp_f  -- Temperature in degrees Fahrenheit
%   temp_k  -- Temperature in kelvin
%
% Prompt the user for the input temperature.
temp_f = input('Enter the temperature in degrees Fahrenheit:');

% Convert to kelvin.
temp_k = (5/9) * (temp_f - 32) + 273.15;

% Write out the result.
disp([ num2str(temp_f) 'degrees Fahrenheit = ' num2str(temp_k) 'kelvin');
```

# عیب یابی (Debugging) برنامه ها

- یک قاعده کلی: اگر برنامه ای با هر اندازه بنویسید در اولین بار که آن را امتحان می کنید کار نخواهد کرد.
- سه نوع خطای رایج:

1. Syntax error (خطای دستوری): خطا در عبارات MATLAB مانند نقطه گذاری، بالانس نبودن پرانتز

```
x = (y + 3) / 2);
```

```
>> test
```

```
??? x = (y + 3) / 2)
```

```
|
```

```
Missing operator, comma, or semi-colon.
```

2. Run-time error (خطا در حین اجرا): انجام یک عمل ریاضی غیرمجاز در حین اجرای برنامه به عنوان مثال تقسیم بر عدد صفر

3. Logical error (خطای منطقی): برنامه با موفقیت کامپایل شده ولی جواب اشتباه تولید می کند.

- اگر اندازه عبارات انتسابی بزرگ است، آن را به چندین عبارت کوچکتر تقسیم کنید.

- اگر در ترتیب انجام محاسبات شک دارید از پرانتز استفاده کنید.

- بررسی کنید که تمامی متغیرها را درست مقداردهی کردید

- بررسی کنید که واحدها درست استفاده شده اند.

- اگر برنامه هنوز اشتباه جواب می دهد، برنامه را خط به خط اجرا کنید.

- اگر برنامه بازهم درست پاسخ نداد، آن را به شخص دیگری نشان دهید. معمولاً برنامه نویس به صورت ناخودآگاه از اشتباهات عبور می کند.

# نکاتی برای برنامه نویسی

- از نام های با معنی برای متغیرها استفاده کنید. Time, day, mass, ...
- یک لغت نامه داده برای هر برنامه ایجاد کنید.
- از حروف فقط کوچک یا فقط بزرگ برای نام متغیرهای استفاده کنید.
- از ; در انتهای فرامین استفاده کنید
- از پرانتز برای ساده تر کردن معادلات استفاده کنید.
- همیشه واحد مناسب مقادیر را به همراه آن ها نشان دهید.
- در ابتدای برنامه از دستورات `clc`, `clear` استفاده کنید.
- از قابلیت رسم نمودار برای نشان دادن نتایج برنامه ها استفاده کنید.
- نام بامعنی برای برنامه ها انتخاب کنید.
- برنامه های مرتبط را در یک پوشه و جدا از سایر برنامه قرار دهید.

# خلاصه دستورات و توابع

## Special Symbols

---

|     |                                                                                                                                                   |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------|
| [ ] | Array constructor.                                                                                                                                |
| ( ) | Forms subscripts.                                                                                                                                 |
| ' ' | Marks the limits of a character string.                                                                                                           |
| ,   | <ol style="list-style-type: none"><li>1. Separates subscripts or matrix elements.</li><li>2. Separates assignment statements on a line.</li></ol> |

|    |                                                                                                                                                                                      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ,  | Separates subscripts or matrix elements.                                                                                                                                             |
| ;  | <ol style="list-style-type: none"> <li>1. Suppresses echoing in Command Window.</li> <li>2. Separates matrix rows.</li> <li>3. Separates assignment statements on a line.</li> </ol> |
| %  | Marks the beginning of a comment.                                                                                                                                                    |
| :  | Colon operator, used to create shorthand lists.                                                                                                                                      |
| +  | Array and matrix addition.                                                                                                                                                           |
| -  | Array and matrix subtraction.                                                                                                                                                        |
| .* | Array multiplication.                                                                                                                                                                |
| *  | Matrix multiplication.                                                                                                                                                               |
| ./ | Array right division.                                                                                                                                                                |
| .\ | Array left division.                                                                                                                                                                 |
| /  | Matrix right division.                                                                                                                                                               |
| \  | Matrix left division.                                                                                                                                                                |
| .^ | Array exponentiation.                                                                                                                                                                |
| '  | Transpose operator.                                                                                                                                                                  |

## Commands and Functions

|                         |                                                                                                                                      |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <code>...</code>        | Continues a MATLAB statement on the following line.                                                                                  |
| <code>abs(x)</code>     | Calculates the absolute value of $x$ .                                                                                               |
| <code>ans</code>        | Default variable used to store the result of expressions not assigned to another variable.                                           |
| <code>acos(x)</code>    | Calculates the inverse cosine of $x$ . The resulting angle is in radians between 0 and $\pi$ .                                       |
| <code>asin(x)</code>    | Calculates the inverse sine of $x$ . The resulting angle is in radians between $-\pi/2$ and $\pi/2$ .                                |
| <code>atan(x)</code>    | Calculates the inverse tangent of $x$ . The resulting angle is in radians between $-\pi/2$ and $\pi/2$ .                             |
| <code>atan2(y,x)</code> | Calculates the inverse tangent of $y/x$ , valid over the entire circle. The resulting angle is in radians between $-\pi$ and $\pi$ . |
| <code>ceil(x)</code>    | Rounds $x$ to the nearest integer towards positive infinity <code>ceil(3.1) = 4</code> and <code>ceil(-3.1) = -3</code>              |
| <code>char</code>       | Converts a matrix of numbers into a character string. For ASCII characters, the matrix should contain numbers $\leq 127$ .           |
| <code>clock</code>      | Current time.                                                                                                                        |
| <code>cos(x)</code>     | Calculates cosine of $x$ , where $x$ is in radians.                                                                                  |
| <code>cross</code>      | Calculates the cross product of two vectors.                                                                                         |
| <code>date</code>       | Current date.                                                                                                                        |

|                             |                                                                                                                           |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <code>disp</code>           | Displays data in Command Window.                                                                                          |
| <code>doc</code>            | Open HTML Help Desk directly at a particular function description.                                                        |
| <code>dot</code>            | Calculates the dot product of two vectors.                                                                                |
| <code>double</code>         | Converts a character string into a matrix of numbers.                                                                     |
| <code>eps</code>            | Represents machine precision.                                                                                             |
| <code>exp(x)</code>         | Calculates $e^x$ .                                                                                                        |
| <code>eye(m,n)</code>       | Generates an identity matrix.                                                                                             |
| <code>fix(x)</code>         | Rounds $x$ to the nearest integer towards zero: <code>fix(3.1) = 3</code> and <code>fix(-3.1) = -3</code> .               |
| <code>floor(x)</code>       | Rounds $x$ to the nearest integer towards minus infinity: <code>floor(3.1) = 3</code> and <code>floor(-3.1) = -4</code> . |
| <code>format +</code>       | Print + and - signs only.                                                                                                 |
| <code>format bank</code>    | Print in “dollars and cents” format.                                                                                      |
| <code>format compact</code> | Suppress extra linefeeds in output.                                                                                       |
| <code>format hex</code>     | Print hexadecimal display of bits.                                                                                        |
| <code>format long</code>    | Print with 14 digits after the decimal.                                                                                   |
| <code>format long e</code>  | Print with 15 digits plus exponent.                                                                                       |
| <code>format long g</code>  | Print with 15 digits with or without exponent.                                                                            |
| <code>format loose</code>   | Print with extra linefeeds in output.                                                                                     |
| <code>format rat</code>     | Print as an approximate ratio of small integers.                                                                          |
| <code>format short</code>   | Print with 4 digits after the decimal.                                                                                    |
| <code>format short e</code> | Print with 5 digits plus exponent.                                                                                        |
| <code>format short g</code> | Print with 5 digits with or without exponent.                                                                             |

|                          |                                                                                      |
|--------------------------|--------------------------------------------------------------------------------------|
| <code>fprintf</code>     | Print formatted information.                                                         |
| <code>grid</code>        | Add or remove a grid from a plot.                                                    |
| <code>i</code>           | $\sqrt{-1}$ .                                                                        |
| <code>Inf</code>         | Represents machine infinity ( $\infty$ ).                                            |
| <code>input</code>       | Writes a prompt and reads a value from the keyboard.                                 |
| <code>int2str</code>     | Converts $x$ into an integer character string                                        |
| <code>j</code>           | $\sqrt{-1}$ .                                                                        |
| <code>legend</code>      | Adds a legend to a plot.                                                             |
| <code>length(arr)</code> | Returns the length of a vector or the longest dimension of a two-dimensional array.  |
| <code>load</code>        | Load data from a file.                                                               |
| <code>log(x)</code>      | Calculates the natural logarithm of $x$ .                                            |
| <code>loglog</code>      | Generates a log-log plot.                                                            |
| <code>lookfor</code>     | Looks for a matching term in the one-line MATLAB function descriptions.              |
| <code>max(x)</code>      | Returns the maximum value in vector $x$ , and optionally the location of that value. |

|                         |                                                                                      |
|-------------------------|--------------------------------------------------------------------------------------|
| <code>min(x)</code>     | Returns the minimum value in vector $x$ , and optionally the location of that value. |
| <code>mod(m,n)</code>   | Remainder or modulo function.                                                        |
| <code>NaN</code>        | Represents not-a-number.                                                             |
| <code>num2str(x)</code> | Converts $x$ into a character string.                                                |
| <code>ones(m,n)</code>  | Generates an array of ones.                                                          |
| <code>pi</code>         | Represents the number $\pi$ .                                                        |
| <code>plot</code>       | Generates a linear $xy$ plot.                                                        |
| <code>print</code>      | Prints a Figure Window.                                                              |
| <code>round(x)</code>   | Rounds $x$ to the nearest integer.                                                   |
| <code>save</code>       | Saves data from workspace into a file.                                               |
| <code>semilogx</code>   | Generates a log-linear plot.                                                         |
| <code>semilogy</code>   | Generates a linear-log plot.                                                         |
| <code>sin(x)</code>     | Calculates sine of $x$ , where $x$ is in radians.                                    |
| <code>size</code>       | Get number of rows and columns in an array.                                          |
| <code>sqrt</code>       | Calculates the square root of a number.                                              |
| <code>str2num</code>    | Converts a character string into a number.                                           |
| <code>tan(x)</code>     | Calculates tangent of $x$ , where $x$ is in radians.                                 |
| <code>title</code>      | Adds a title to a plot.                                                              |
| <code>zeros</code>      | Generates an array of zeros.                                                         |

---