Chapter 1 : Matrices Course

Definition of a Matrix

- The matrix is a rectangular array (i.e., a table) of elements (numbers or variables) .
- Has ***r*** Rows and ***c*** Columns.
- Named using capital letters.
- Always the First subscript (***r***) is number of rows; second subscript (***c***) is number of columns.
- A matrix (***A***) with ***r*** rows and ***c*** columns is called a matrix of order $r \times c$.

$$\boldsymbol{A_{r\times c}} = \begin{pmatrix} \boldsymbol{a_{11}} & \boldsymbol{a_{12}} & \boldsymbol{a_{13}} & \cdots & \boldsymbol{a_{1c}} \\ \boldsymbol{a_{21}} & \boldsymbol{a_{22}} & \boldsymbol{a_{23}} & \cdots & \boldsymbol{a_{2c}} \\ \boldsymbol{a_{31}} & \boldsymbol{a_{32}} & \boldsymbol{a_{33}} & \cdots & \boldsymbol{a_{3c}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a_{r1}} & \boldsymbol{a_{r2}} & \boldsymbol{a_{r3}} & \cdots & \boldsymbol{a_{r1}} \end{pmatrix}$$

Examples of matrices: Each of the following is examples of matrices

$A_{2\times3} = \begin{pmatrix} 2 & 5 & -1 \\ 3 & 0 & 5 \end{pmatrix}_{2\times3}$ (Rectangular matrix)	$B_{1\times4} = \begin{pmatrix} 1 & 3 & 0 & -2 & 7 \end{pmatrix}_{1\times4}$ (Vector matrix; row vector)	$D_{5\times1} = \begin{pmatrix} 2 \\ 0 \\ 3 \\ 4 \\ 8 \end{pmatrix}_{5\times1}$ (Vector matrix; column vector)	$C_{3\times3} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}_{3\times3}$ (square matrix)
$E_{1\times1} = (6)_{1\times1}$ (one element matrix) (which mean that each number in mathematics is in fact is a 1×1 matrix)			

The matrix element's position

The elements of a matrix are referred to (or labeled by their row number and column number, but always in that order, i.e. the row number (r) followed by the column number(c) (a_{rc}) the general form of the matrix is:

$$A_{r\times c} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1c} \\ a_{21} & a_{22} & \cdots & a_{2c} \\ \vdots & \vdots & \vdots & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rc} \end{pmatrix}$$

Example	Home Work
$A_{3\times4} = \begin{pmatrix} 1 & 3 & 5 & 2 \\ 4 & 0 & -1 & 3 \\ 6 & 7 & 8 & 9 \end{pmatrix}$ $a_{11} = 1$, $a_{12} = 3$, $a_{13} = 5$, $a_{14} = 2$, $a_{21} = 4$, $a_{22} = 0$, $a_{23} = -1$, $a_{24} = 3$, $a_{31} = 6$, $a_{32} = 7$, $a_{33} = 8$, $a_{34} = 9$	$A_{3\times3} = \begin{pmatrix} 3 & 2 & -1 \\ 4 & 6 & 7 \\ -3 & 4 & 8 \end{pmatrix}$ For the above matrix determine the location of the following elements: 6 , 7 , 8 , -1 , 4 , 2, 8, -3

Vectors (one dimensional matrix)

The matrix with only one row or one column is called **a vector**;

- A ***column vector*** is a one-column matrix:

Form	Example
$A_{r\times1} = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{r1} \end{pmatrix}, \quad A_{r\times1} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_r \end{pmatrix}$	$Y_{4\times1} = \begin{pmatrix} -5 \\ 3 \\ 1 \\ 4 \end{pmatrix}$, $y_{21}=3$ or $y_2=3$

- Likewise, A row vector is a one-row matrix:

Form	Example
$B'_{1\times c} = (b_{11}, b_{12}, \dots, b_{1\times c})$ $B'_{1\times c} = (b_1, b_2, \dots, b_n)$	$Y_{4\times1} = (-5 \quad 3 \quad 1 \quad 4)$ $y_{12}=3$ or $y_2=3$

Square Matrix

A **square matrix** is a matrix with an equal number of rows (***r***) and columns (***c***). Since the number of rows and columns are the same, it is said to have **order *n***.

For example the matrix B,

$$B_3 \equiv B_{3\times3} = \begin{pmatrix} -5 & 1 & 3 \\ 2 & 2 & 6 \\ 7 & 3 & -4 \end{pmatrix}$$

Terms used with square matrix

Diagonal of the square matrix:

The **main diagonal** of a **square matrix** are the elements from the **upper left** to the **lower right** of the matrix. Thus **The *main diagonal*** of a square matrix B_{3x3} consists of the elements b_{ii}, i=1,2,…,n. In the example, the main diagonal consists of the elements -5, 2, and -4.

The Trace of a square matrix:

Is the **sum of its diagonal elements**

$$trace(B) = \sum_{i=1}^{n} b_{ii} = -5 + 2 - 4 = -7$$

Matrix Order = No. of Rows × No. of Columns

Special Types of matrices

1. A square matrix is ***Lower-triangular* (L)** if all elements that locate above its main diagonal are 0; for example:

$L_3 = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ 5 & 2 & \mathbf{0} \\ 2 & 0 & 3 \end{pmatrix}$	$L_2 = \begin{pmatrix} 0 & \mathbf{0} \\ 9 & 0 \end{pmatrix}$

2. Similarly, an ***Upper-triangular* (U)** matrix has zeroes below the main diagonal; for example:

$U_3 = \begin{pmatrix} 5 & 5 & 2 \\ \mathbf{0} & 2 & -4 \\ \mathbf{0} & \mathbf{0} & 3 \end{pmatrix}$	$U_2 = \begin{pmatrix} 4 & 5 \\ \mathbf{0} & 0 \end{pmatrix}$

3. A ***Diagonal matrix* (D)** is a square matrix with all off-diagonal elements equal to 0 (at least one of the diagonal elements not equal to zero); for example:

$D_4 = \begin{pmatrix} \mathbf{6} & 0 & 0 & 0 \\ 0 & \mathbf{-2} & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 \\ 0 & 0 & 0 & 7 \end{pmatrix} = diag(6,-2,0,7)$	$D_2 = \begin{pmatrix} \mathbf{1} & 0 \\ 0 & \mathbf{0} \end{pmatrix}$

4. An ***Identity matrix* (I)** is a scalar matrix with ones on the diagonal; for example:

$I_3 = I_{3\times3} = \begin{pmatrix} \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} \end{pmatrix}$	$I_2 = \begin{pmatrix} \mathbf{1} & 0 \\ 0 & \mathbf{1} \end{pmatrix}$

- Square matrix.
- Ones on the main diagonal
- Zeros everywhere else
- Denoted by ***I***. If a subscript is included, it is the order of the identity matrix.
- I is the multiplicative identity for matrices
- Any matrix times the identity matrix is the original matrix.
- Multiplication by the identity matrix is commutative, although the order of the identity may change

5. A ***Zero matrix* (0)** has its entire elements equal to 0; for example:

$0_{4\times3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$0_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

6. **A *Unit vector (1)*** has its entire elements equal to 1; for example:

$1_{4\times1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$	$1_{2\times1} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$1_{1\times3} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$

The operations on matrices

1. **The Transpose of a Matrix**

The transpose of a matrix A, Written as A' or A^T, is the matrix that obtained by writing the rows of A as the columns of A^T.

$$A_{r\times c}=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1c} \\ a_{21} & a_{22} & \cdots & a_{2c} \\ \vdots & \vdots & \vdots & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rc} \end{pmatrix} \rightarrow A^T_{c\times r}=\begin{pmatrix} a_{11} & a_{21} & \dots & a_{r1} \\ a_{12} & a_{22} & \cdots & a_{r2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1c} & a_{2c} & \cdots & a_{cr} \end{pmatrix}$$

Properties of the Transpose operation:

If $X_{m\times n}$ and $Y_{m\times n}$ and $Z_{n\times k}$ are matrices ,then

1. $(X + Y)^T = X^T + Y^T$
2. $(X \times Z)T = ZT \times XT$
3. $(X^T)^T = X$

$$if\ \ A_{2\times 3}=\begin{pmatrix} -1 & 2 & 3 \\ 4 & 0 & -7 \end{pmatrix}\ then\ \ A^T_{3\times 2}=\begin{pmatrix} -1 & 4 \\ 2 & 0 \\ 3 & -7 \end{pmatrix}$$

2. **Row (or Column) Operations**

There are three basic *row* (column) operations which may be performed on matrices. These are:

A. Multiplication of any row by a non-zero number.

Ex / ***$R_3 \rightarrow 4R_3$*** means ***row 3*** is ***changed*** to ***4×row 3***.

B. Interchange of two rows.

Ex / ***$R_1 \leftrightarrow R_2$*** means ***row 1*** is ***interchanged*** with ***row 2***.

C. Replacement of any row (or column) by the result of mathematical operation applied on multi rows (or columns).

Ex / ***$R_2 \rightarrow R_2 + 3R_3$*** means ***row 2*** has ***3×row 3*** added to it.

Example : if $X_{3\times 3}\begin{bmatrix} 1 & 2 & 0 \\ -1 & 3 & 2 \\ 4 & 5 & 3 \end{bmatrix}$ is a matrix find:

1) $R_1 \leftrightarrow R_2$. **2)** $R_2 \rightarrow 2R_2$. **3)** $R_1 \rightarrow R_1+2R_2$.

$R_1 \leftrightarrow R_2 \Rightarrow X_{3\times 3}\begin{bmatrix} -1 & 3 & 2 \\ 1 & 2 & 0 \\ 4 & 5 & 3 \end{bmatrix}$. $R_2 \rightarrow 2R_2 \Rightarrow X_{3\times 3}\begin{bmatrix} 1 & 2 & 0 \\ -2 & 6 & 4 \\ 4 & 5 & 3 \end{bmatrix}$. $R_1 \rightarrow R_1+2R_2 \Rightarrow X_{3\times 3}\begin{bmatrix} -1 & 8 & 4 \\ -1 & 3 & 2 \\ 4 & 5 & 3 \end{bmatrix}$

Homework
if $A_{3\times3}\begin{bmatrix} 1 & 2 & 0 \\ -1 & 3 & 2 \\ 4 & 5 & 3 \end{bmatrix}$ and $B_{3\times3}\begin{bmatrix} -1 & 1 & 3 \\ 2 & 2 & 1 \\ 2 & 3 & -1 \end{bmatrix}$ re matrices find $\boldsymbol{A'}$,$\boldsymbol{B'}$
if $X_{3\times3}\begin{bmatrix} 1 & 2 & 0 \\ -1 & 3 & 2 \\ 4 & 5 & 3 \end{bmatrix}$ is a matrix , find :   l: **1)**$R_2 \leftrightarrow R_3$. **2)**$R_3 \rightarrow 3R_1$. **3)**$R_2 \rightarrow R_3+3R_2$. **4)**$C_2 \leftrightarrow C_3$. **5)**$C_3 \rightarrow 3C_1$. **6)**$C_2 \rightarrow C_3+3C_2$

Mathematical operations on matrices

A. Equality:

Two matrices are equal if and only if :

1. The order of the matrices is the same
2. The corresponding elements of the matrices are the same.

B. Addition (or subtraction):

1. (Constant-matrix) addition (or subtraction)

To add matrices they must have the same order, therefore, if λ is a number and A, B, C are matrices then:

$$C_{rc} = \lambda + A_{rc} = \begin{bmatrix} \lambda+a_{11} & \lambda+a_{12} & \cdots & \lambda+a_{1c} \\ \lambda+a_{21} & \lambda+a_{22} & \cdots & \lambda+a_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda+a_{r1} & \lambda+a_{r2} & \cdots & \lambda+a_{rc} \end{bmatrix}$$

a. (matrix-matrix) addition (or subtraction)

1. Order of the matrices must be the same.
2. Add (or subtract) corresponding elements together.
3. Matrix addition is commutative. تبادلي
4. Matrix addition is associative. تجميعي
5. Matrix subtraction is not commutative (neither is subtraction of real numbers)
6. Matrix subtraction is not associative (neither is subtraction of real numbers)

$$C_{rc} = B_{rc} + A_{rc} = \begin{bmatrix} b_{11}+a_{11} & b_{12}+a_{12} & \cdots & b_{1c}+a_{1c} \\ b_{21}+a_{21} & b_{22}+a_{22} & \cdots & b_{2c}+a_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ b_{r1}+a_{r1} & b_{r2}+a_{r2} & \cdots & b_{rc}+a_{rc} \end{bmatrix}$$

Ex/ Let $\lambda = 2$ and $A_{2\times3} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 2 \end{bmatrix}$, $B_{2\times3} = \begin{bmatrix} 3 & 4 & 5 \\ -1 & 0 & -7 \end{bmatrix}$,

find: 1) $C = \lambda + A$, 2) $C = A + B$, 3) $C = A - B$

1) $C_{2\times3} = \lambda + A_{2\times3} = 2 + \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 4 & 5 \\ 2 & 1 & 4 \end{bmatrix}$

2) $C_{2\times3} = A_{2\times3} + B_{2\times3} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 2 \end{bmatrix} + \begin{bmatrix} 3 & 4 & 5 \\ -1 & 0 & -7 \end{bmatrix} = \begin{bmatrix} 4 & 6 & 8 \\ -1 & -1 & -5 \end{bmatrix}$

3) $C_{2\times3} = A_{2\times3} - B_{2\times3} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 2 \end{bmatrix} - \begin{bmatrix} 3 & 4 & 5 \\ -1 & 0 & -7 \end{bmatrix} = \begin{bmatrix} -2 & -2 & -2 \\ -1 & -1 & 9 \end{bmatrix}$

C. **Multiplication**

1. **(Constant-matrix) multiplication**

If λ is a number and A is a matrix then:-

$$\lambda \times A_{rc} = \begin{bmatrix} \lambda a_{11} & \lambda a_{12} & \cdots & \lambda a_{1c} \\ \lambda a_{21} & \lambda a_{22} & \cdots & \lambda a_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda a_{r1} & \lambda a_{r2} & \cdots & \lambda a_{rc} \end{bmatrix}$$

ex: if $\lambda = 2\,,-1$ and $X_{2\times2} = \begin{bmatrix} -1 & 3 \\ 5 & 2 \end{bmatrix}$, find $\lambda \times X$:

$$When\ \lambda = 2\ \ then\ \lambda \times X_{2\times2} = 2 \times \begin{bmatrix} -1 & 3 \\ 5 & 2 \end{bmatrix} = \begin{bmatrix} -2 & 6 \\ 10 & 4 \end{bmatrix}$$

$When\ \lambda = -1\ \ then\ \lambda \times X_{2\times2} = -1 \times \begin{bmatrix} -1 & 3 \\ 5 & 2 \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ -5 & -2 \end{bmatrix}$ (Matrix negation)

H.W./ if $\lambda = 6$ and $A_{2\times3} = \begin{bmatrix} 6 & 5 & 3 \\ 0 & 2 & -4 \end{bmatrix}$, $B_{3\times3} = \begin{bmatrix} -1 & 3 & 0 \\ 5 & 2 & 1 \end{bmatrix}$,

Find: **1)** $B + \lambda$, **2)** $\lambda - A$, **3)** $A - \lambda$, **4)** $A + B$, **5)** $B + A$, **6)** $A - B$, **7)** $\lambda - A + B$, **8)** $B - \lambda + A$, **9)** $\lambda A + \lambda B$, **10)** $\lambda(A - B)$

1. **(matrix-matrix) Multiplication (Inner Product)**

Matrix product is defined by the inner product, therefore, the matrix product is defined only if the matrix's columns on the left are equal the matrix's rows on the right:

$$C_{r\times q} = A_{r\times n}.B_{n\times q} \ \text{ and } c_{ij} = a_i' \cdot b_j = \sum_{k=1}^{n} a_{ik} b_{kj}$$

The product of two vectors represented by the sum of the product of corresponding elements:

$$\boldsymbol{row\ vector.column\ vector = a'_{1\times n} \cdot b_{n\times 1} = \sum_{i=1}^{n} a_i b_i}$$

– Note: the inner product is defined similarly between two row vectors or two column vectors.

• For example, $v'.w = [2 \quad 0 \quad 1 \quad 3] \cdot \begin{bmatrix} -1 \\ 6 \\ 0 \\ 9 \end{bmatrix} = 2(-1) + 0(6) + 1(0) + 3(9) = 25$

ex: Let $A_{2x3} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ and $I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, find C = A×I.

$$C_{2\times3} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \times \begin{pmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{pmatrix}$$

$C_{2\times 3}$ =	C_{11} = $(a_{11}.i_{11})+(a_{12}.i_{21})+(a_{13}.i_{31})$ $= 1(1) + 2(0) + 3(0)$ $= 1$	C_{12} = $(a_{11}.i_{12})+(a_{12}.i_{22})+(a_{13}.i_{32})$ $= 1(0) + 2(1) + 3(0)$ $= 2$	C_{13} = $(a_{11}.i_{13})+(a_{12}.i_{23})+(a_{13}.i_{33})$ $= 1(0) + 2(0) + 3(1)$ $= 3$
	C_{21} = $(a_{21}.i_{11})+(a_{22}.i_{21})+(a_{23}.i_{31})$ $= 4(1) + 5(0) + 6(0)$ $= 4$	C_{22} = $(a_{21}.i_{12})+(a_{22}.i_{22})+(a_{23}.i_{32})$ $4(0) + 5(1) + 6(0)=$ $= 5$	C_{23} = $(a_{21}.i_{13})+(a_{22}.i_{23})+(a_{23}.i_{33})$ $= 4(0) + 5(0) + 6(1)$ $= 6$

$$\Rightarrow C_{2\times3} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Properties of mathematical operation on Matrices

1. A + B = B + A (*commutativity*) تبادلي
2. A + (B + C) = (A + B) + C (*associativity*) تجميعي
3. A - B = A + (-B) = -(B - A)
4. A - A = 0 (-A is the *additive inverse* of A).
5. A + 0 = A (0 is the *additive identity*).
6. (A + B)' = A' + B' Where A,B,C, and 0 are matrices of the same order, and α, β, 0, and 1 are scalars.
7. A(BC) = (AB)C (associativity).
8. (A + B)C = AC + BC (distributive laws)
9. A(B + C) = AB + AC.
10. $A_{m\times n}\ I_n = I_m\ A_{m\times n} = A$
11. $A_{m\times n}\ 0_{n\times p} = 0_{m\times p}$
12. $0_{q\times m}\ A_{m\times n} = 0_{q\times n}$
13. $\left(A_{m\times n}\ B_{n\times p}\right)' = A'_{m\times n}\ B'_{n\times p}\ \ (transpose\ of\ product)$
14. $(AB\cdots F)' = F'\cdots B'A'$ *If* <u>*AB*</u> *is defined then* <u>*BA*</u> *is not necessary defined*
15. (-1)×A = -A , *-(-A) = A .*
16. αA = Aα (*commutativity*).
17. α (A + B) = αA + αB (*distributive laws*).
18. A(α + β) = Aα + Aβ
19. 0×A = 0 (*zero*)

Determinant of the Matrices

The determinant is a special number associated to any square matrix and it is unique. The determinant of a matrix A, is denoted det(A) or |A|.

Finding the Determinant:

A. For (2 × 2) matrix:

The determinant of a matrix $A_{2\times2} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ has determinant equal to:

|A| = ad − bc

Ex: if $A = \begin{pmatrix} 4 & 3 \\ 6 & 3 \end{pmatrix}$ find |A|

⇒ |A| = (4×3) – (3×6) = 12 – 18 = -6

B. For (3 × 3) matrix:

The determinant of a matrix $A_{3\times3} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$ has determinant equal to:

$$\boxed{\boldsymbol{|A| = +a \times (e.i - f.h) - b \times (d.i - f.g) + c \times (d.h - e.g)}}$$

Ex: if $B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 0 \\ -2 & 4 & 1 \end{pmatrix}$; find the determinant of matrix B;

$$\Rightarrow \boldsymbol{|B| = \left[1 \times \left((5 \times 1) - (0 \times 4)\right)\right] - \left[2 \times \left((4 \times 1) - (0 \times -2)\right)\right]}$$

$$\boldsymbol{+ \left[3 \times \left((4 \times 4) - (5 \times -2)\right)\right]}$$

$$\Rightarrow \boldsymbol{|B| = 1 \times (5 - 0) - 2 \times (4 - 0) + 3 \times (16 + 10) = 5 - 8 + 78 = 75}$$

H.W. find |A| if : 1) $A = \begin{pmatrix} 5 & 4 \\ 3 & 2 \end{pmatrix}$, 2) $A = \begin{pmatrix} 2 & 6 \\ 1 & 3 \end{pmatrix}$

3) $A = \begin{pmatrix} 2 & 3 & 4 \\ 0 & -4 & 1 \\ 2 & -2 & 3 \end{pmatrix}$, 4) $A = \begin{pmatrix} 3 & 0 & 1 \\ -1 & -5 & 1 \\ 0 & -1 & 3 \end{pmatrix}$

Matrix Inversion

In scalar algebra, the number *1* is the **neutral** number to the product process, which can be obtained by dividing the number by itself (i.e. $\frac{x}{x} = x \times \frac{1}{x} = 1$, Where *1/x* called the ***inverse*** to the number x.

In matrices the same idea exist, but instead of number 1, ***I is the <u>matrix</u>*** **<u>neutral to the product process</u>**, therefore, if ***A*** is a matrix then any matrix that multiply by it to produce the ***I*** matrix, that matrix will called the inverse matrix of (A) i.e.(A^{-1}).

If ***A×B = B×A = I***

Then ***A = (1/B) = B^{-1}***

And ***B = (1/A) = A^{-1}***

ex: if $A = \begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix}$ and $B = \begin{pmatrix} 3 & -5 \\ -1 & 2 \end{pmatrix}$,check if (A) is the inverse of (B) and vice-versa.

$$AB = \begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix} \times \begin{pmatrix} 3 & -5 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} (2\times 3) + (5 \times -1) & (2 \times -5) + (5 \times 2) \\ (1 \times 3) + (3 \times -1) & (1 \times -5) + (3 \times 2) \end{pmatrix}$$

$$= \begin{pmatrix} 6-5 & -10+10 \\ 3-3 & -5+6 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2$$

$$BA = \begin{pmatrix} 3 & -5 \\ -1 & 2 \end{pmatrix} \times \begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} (3\times 2) + (-5 \times 1) & (3 \times 5) + (-5 \times 3) \\ (-1 \times 2) + (2 \times 1) & (-1 \times 5) + (2 \times 3) \end{pmatrix}$$

$$= \begin{pmatrix} 6-5 & 15-15 \\ -2+2 & -5+6 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2$$

<u>H.W./</u> check if ***(A)*** is the inverse matrix of ***(B)*** for the following cases:

1. $A = \begin{pmatrix} 1 & 2 \\ 5 & 8 \end{pmatrix}$ And $B = \begin{pmatrix} -4 & 1 \\ 2.5 & -0.5 \end{pmatrix}$,
2. $A = \begin{pmatrix} -3 & 4 \\ 5 & -6 \end{pmatrix}$ And $B = \begin{pmatrix} 3 & 2 \\ 2.5 & 1.5 \end{pmatrix}$,
3. $A = \begin{pmatrix} 2 & 3 \\ 6 & 1 \end{pmatrix}$ And $B = \begin{pmatrix} -2 & 5 \\ 2 & -4 \end{pmatrix}$

<u>Requirements to have an Inverse</u>

1. The matrix must be square (same number of rows and columns).
2. The determinant of the matrix must not be zero. This is instead of the real number not being zero to have an inverse; the determinant must not be zero to have an inverse.
3. A square matrix that has an inverse is called **invertible** or **non-singular**. While the matrix that does not have an inverse is called **singular**.

Finding the Inverse

The inverse of a matrix A will satisfy the equation

$$\boldsymbol{A \times (A^{-1}) = I},$$

$$\boldsymbol{A^{-1} = \frac{1}{|A|} \times \ adj(A) = \frac{adj(A)}{|A|}}$$

A. Finding the inverse of (2 × 2) matrix:

Let A be a (2 × 2) matrix then:

$$A_{2\times 2} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \Rightarrow$$

$$\therefore adj(A) = \begin{pmatrix} \boldsymbol{d} & -b \\ -c & \boldsymbol{a} \end{pmatrix},$$

$$\because \ A^{-1} = \frac{adj(A)}{|A|} = \frac{1}{|A|} \times adj(A) = \frac{1}{ad-bc} \times \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$$\therefore A^{-1} = \begin{pmatrix} \frac{d}{ad-bc} & \frac{-b}{ad-bc} \\ \frac{-c}{ad-bc} & \frac{a}{ad-bc} \end{pmatrix}$$

Ex/ $\boldsymbol{find\ the\ inverse\ of\ matrix\ A} = \begin{pmatrix} \mathbf{2} & \mathbf{3} \\ \mathbf{1} & \mathbf{4} \end{pmatrix}$

$$adj(A) = \begin{pmatrix} 4 & -3 \\ -1 & 2 \end{pmatrix}, |A| = \ (2 \times 4) - (3 \times 1) = 8 - 3 = 5,$$

$$\therefore \ A^{-1} = \frac{adj(A)}{|A|} = \frac{1}{|A|} \times \ adj(A) = \frac{1}{5} \times \begin{pmatrix} 4 & -3 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} \frac{4}{5} & \frac{-3}{5} \\ \frac{-1}{5} & \frac{2}{5} \end{pmatrix}$$

B. <u>Finding the inverse of (3 × 3) matrix:</u>

Let (A) be a (3×3) matrix then to find the inverse for this matrix; we must do the following:

$$A_{3\times3} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

$$\because\ A^{-1}_{3\times3} = \frac{1}{|A|} \times adj(A)$$

1. $|A_{3\times3}| = \ a(ei - fh) - b(di - fg) + c(dh - eg)$ .....(1)

2. $adj(A_{3\times3}) = cof(A_{3\times3})^T$ the *adjoint* of the matrix .....(2)

Where $cof(A_{3\times3})$ is the **cofactor** of the matrix ($A_{3\times3}$) ; which can founded as :

$$cof(A_{3\times3}) = \begin{pmatrix} +|A_{11}| & -|A_{12}| & +|A_{13}| \\ -|A_{21}| & +|A_{22}| & -|A_{23}| \\ +|A_{31}| & -|A_{32}| & +|A_{33}| \end{pmatrix} \qquad .....(3)$$

$$\therefore adj(A_{3\times3}) = \begin{pmatrix} +|A_{11}| & -|A_{12}| & +|A_{13}| \\ -|A_{21}| & +|A_{22}| & -|A_{23}| \\ +|A_{31}| & -|A_{32}| & +|A_{33}| \end{pmatrix}^T = \left(cof(A_{3\times3})\right)^{\boldsymbol{T}}$$

$$|\boldsymbol{A_{11}}| = \begin{pmatrix} \boldsymbol{e} & \boldsymbol{f} \\ \boldsymbol{h} & \boldsymbol{i} \end{pmatrix} = (\boldsymbol{ei} - \boldsymbol{fh})\ ;\ |\boldsymbol{A_{12}}| = \begin{pmatrix} \boldsymbol{d} & \boldsymbol{f} \\ \boldsymbol{g} & \boldsymbol{i} \end{pmatrix} = (\boldsymbol{di} - \boldsymbol{fg})\ ;\ |\boldsymbol{A_{13}}| = \begin{pmatrix} \boldsymbol{d} & \boldsymbol{e} \\ \boldsymbol{g} & \boldsymbol{h} \end{pmatrix} = (\boldsymbol{dh} - \boldsymbol{eg})\ ;$$

$$|\boldsymbol{A_{21}}| = \begin{pmatrix} \boldsymbol{b} & \boldsymbol{c} \\ \boldsymbol{h} & \boldsymbol{i} \end{pmatrix} = (\boldsymbol{bi} - \boldsymbol{ch})\ ;\ |\boldsymbol{A_{22}}| = \begin{pmatrix} \boldsymbol{a} & \boldsymbol{c} \\ \boldsymbol{g} & \boldsymbol{i} \end{pmatrix} = (\boldsymbol{ai} - \boldsymbol{cg})\ ;\ |\boldsymbol{A_{23}}| = \begin{pmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{g} & \boldsymbol{h} \end{pmatrix} = (\boldsymbol{ah} - \boldsymbol{bg})\ ;$$

$$|\boldsymbol{A_{31}}| = \begin{pmatrix} \boldsymbol{b} & \boldsymbol{c} \\ \boldsymbol{e} & \boldsymbol{f} \end{pmatrix} = (\boldsymbol{bf} - \boldsymbol{ce})\ ;\ |\boldsymbol{A_{32}}| = \begin{pmatrix} \boldsymbol{a} & \boldsymbol{c} \\ \boldsymbol{d} & \boldsymbol{f} \end{pmatrix} = (\boldsymbol{af} - \boldsymbol{cd})\ ;\ |\boldsymbol{A_{33}}| = \begin{pmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{d} & \boldsymbol{e} \end{pmatrix} = (\boldsymbol{ae} - \boldsymbol{bd})\ ;$$

$$\therefore\ adj(A_{3\times3}) = \begin{pmatrix} +|A_{11}| & -|A_{12}| & +|A_{13}| \\ -|A_{21}| & +|A_{22}| & -|A_{23}| \\ +|A_{31}| & -|A_{32}| & +|A_{33}| \end{pmatrix}^T$$

$$\Rightarrow\ adj(A_{3\times3}) = \begin{pmatrix} +(\boldsymbol{ei} - \boldsymbol{fh}) & -(\boldsymbol{di} - \boldsymbol{fg}) & +(\boldsymbol{dh} - \boldsymbol{eg}) \\ -(\boldsymbol{bi} - \boldsymbol{ch}) & +(\boldsymbol{ai} - \boldsymbol{cg}) & -(\boldsymbol{ah} - \boldsymbol{bg}) \\ +(\boldsymbol{bf} - \boldsymbol{ce}) & -(\boldsymbol{af} - \boldsymbol{cd}) & +(\boldsymbol{ae} - \boldsymbol{bd}) \end{pmatrix}^T$$

$$\because\ A^{-1} = \frac{1}{|A|} \times adj(A)$$

$$\therefore\ \boldsymbol{A^{-1}} = \left(\frac{\mathbf{1}}{\boldsymbol{a(ei-fh)-b(di-fg)+c(dh-eg)}}\right) \times \begin{pmatrix} +(\boldsymbol{ei-fh}) & -(\boldsymbol{di-fg}) & +(\boldsymbol{dh-eg}) \\ -(\boldsymbol{bi-ch}) & +(\boldsymbol{ai-cg}) & -(\boldsymbol{ah-bg}) \\ +(\boldsymbol{bf-ce}) & -(\boldsymbol{af-cd}) & +(\boldsymbol{ae-bd}) \end{pmatrix}^{\boldsymbol{T}}$$

Ex: find the inverse for the matrix ($A_{3\times3}$) if you know that $A_{3\times3} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 0 \\ 2 & 1 & 4 \end{pmatrix}$

Solution:

$$A_{3\times3} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 0 \\ 2 & 1 & 4 \end{pmatrix}$$

$$\because\ A^{-1} = \frac{1}{|A|} \times adj(A)$$

$$|A_{3\times3}| = 1(20-0) - 2(16-0) + 3(4-10) = 20 - 32 - 18 = -30$$

$$adj(A_{3\times3}) = \begin{pmatrix} +|A_{11}| & -|A_{12}| & +|A_{13}| \\ -|A_{21}| & +|A_{22}| & -|A_{23}| \\ +|A_{31}| & -|A_{32}| & +|A_{33}| \end{pmatrix}^{T}$$

$$|A_{11}| = \begin{pmatrix} 5 & 0 \\ 1 & 4 \end{pmatrix} = (20-0) = 20\ ; \quad |A_{12}| = \begin{pmatrix} 4 & 0 \\ 2 & 4 \end{pmatrix} = (16-0) = 16\ ;$$
$$|A_{13}| = \begin{pmatrix} 4 & 5 \\ 2 & 1 \end{pmatrix} = (4-10) = -6\ ; \quad |A_{21}| = \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix} = (8-3) = 5\ ;$$
$$|A_{22}| = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} = (4-6) = -2\ ; \quad |A_{23}| = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} = (1-4) = -3\ ;$$
$$|A_{31}| = \begin{pmatrix} 2 & 3 \\ 5 & 0 \end{pmatrix} = (0-15) = -15\ ; \quad |A_{32}| = \begin{pmatrix} 1 & 3 \\ 4 & 0 \end{pmatrix} = (0-12) = -12\ ;$$
$$|A_{33}| = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix} = (5-8) = -3\ ;$$

$$\therefore\ cof(A_{3\times3}) = \begin{pmatrix} +(20) & -(16) & +(-6) \\ -(5) & +(-2) & -(-3) \\ +(-15) & -(-12) & +(-3) \end{pmatrix}$$

$$adj(A_{3\times3}) = cof(A_{3\times3})^{T} = \begin{pmatrix} +(20) & -(16) & +(-6) \\ -(5) & +(-2) & -(-3) \\ +(-15) & -(-12) & +(-3) \end{pmatrix}^{T} = \begin{pmatrix} 20 & -5 & -15 \\ -16 & -2 & 12 \\ -6 & 3 & -3 \end{pmatrix}^{T}$$

$$A^{-1}_{3\times3} = \frac{1}{|A|} \times adj(A) = \frac{1}{-30} \times \begin{pmatrix} 20 & -5 & -15 \\ -16 & -2 & 12 \\ -6 & 3 & -3 \end{pmatrix}$$

$$\Rightarrow A_{3\times3}^{-1} = \begin{pmatrix} \frac{-20}{30} & \frac{5}{30} & \frac{15}{30} \\ \frac{16}{30} & \frac{2}{30} & \frac{-12}{30} \\ \frac{6}{30} & \frac{-3}{30} & \frac{3}{30} \end{pmatrix} = \begin{pmatrix} \frac{-2}{3} & \frac{1}{6} & \frac{1}{2} \\ \frac{8}{15} & \frac{1}{15} & \frac{2}{5} \\ \frac{1}{5} & \frac{-1}{10} & \frac{1}{10} \end{pmatrix}$$

H.W. : find the inverse of the following matrices:

$$A_{3\times3} = \begin{pmatrix} 1 & 0 & 2 \\ 3 & 5 & -1 \\ -2 & 0 & 4 \end{pmatrix}, \qquad B_{3\times3} = \begin{pmatrix} 2 & 1 & 0 \\ -2 & 1 & 3 \\ 1 & 1 & 2 \end{pmatrix}$$

Chapter 2 : MATLAB Course

1. MATLAB is stand for **MAT**rix **LAB**oratory.

2. **MATLAB**
MATLAB is an interactive working environment in which the user can carry out quite complex computational tasks with few commands.

3. **Mathematical Operations**

No.	Operation	Symbol	Example	Priority	Sequence
1	Parenthesis ( )	( )	(2+4)^5	Highest	1
2	power $\mathbf{a}^{b}$	^	2^8=2 ^ 8	Highest	1
3	Division ÷	/ or \	$\frac{56}{8} = {}^{56}/_{8} = 8\backslash 56$	High	2
4	Multiplication ×	*	3.14 * 0.88	High	2
5	Subtraction −	−	90 – 45	Low	3
6	Addition +	+	3 + 22	Low	3

Example	
$\frac{(2^8+4)\times 5^2}{8}$ = ((**2^8**+4)* **5 ^ 2**) / 8	$(\frac{3^2}{2}+\frac{1}{2})^{\frac{1}{2}}$ = ((**3^2**) / 2 + **1 / 2**) ^ (**1 / 2**)
= ((256+4)* **5 ^ 2**) / 8	= (**9 / 2** + 0.5) ^ 0.5
= **(**260 * 25**)** / 8	= (**4.5 + 0.5**) ^ 0.5
= 6500 / 8	= **5 ^ 0.5**
= 812.5	= 2.2361

Home Work		
$\frac{6}{2}+2$	$\frac{6+2}{2}$	$\frac{(2+4^3)\times 3}{{}^{5}/_{2^{-2}}}$

4. **Variables & Constants**

The ***variables*** and ***constants*** in MATLAB are both represented by letter (s) but with one difference:
- The ***variables*** are that which represents numbers.
- The ***Constants*** are that which represents alphabet(s) (i.e. string).

$$\boldsymbol{variables}\begin{cases} x = 12 & \text{(wher x takes 8 byte)} \\ y = 0.5452 & \text{(wher y takes 8 byte)} \\ z = -1 & \text{(wher z takes 8 byte)} \\ r = 0 & \text{(wher r takes 8 byte)} \end{cases}$$

$$\boldsymbol{constants}\begin{cases} x = '12' & \text{(wher x takes 4 byte)} \\ y = '0.5452' & \text{(wher y takes 12 byte)} \\ z = '-1\,' & \text{(wher z takes 4 byte)} \\ r = '0' & \text{(wher r takes 2 byte)} \end{cases}$$

Home Work
Determine the type and the number of bytes for the following: 'John' , 114 , 'and' , 0.007, 7736384 , 'TOMORROW' , '12.1123'

5. Variables & Constants Naming Rules

The variables and constants will obey the same rules in MATLAB, which are:

a. They are case sensitive (i.e. Cost, COST, CoSt, cost, all are differ)
b. Maximum length for the name is 63 letters, any more will be neglected
(Howaboutthisvariablename = 3)
c. Any name must not start with number or contain any space
(HowAboutThis = 20, how_about_this ='d', X3251 = -4.27)

6. Reserved Words in MATLAB

The following words can't be used as variables in MATLAB because MATLAB use them in its operations, these words are:

For	end	if	while	function	return	Elseif	case	otherwise
switch	continuo	else	try	catch	global	Persistent	break	

7. Special Variables Predefined in MATLAB

The following variables are predefined in MATLAB, which are:

No.	Variable	Notes
1	ans	the variable that store the results
2	beep	make the computer to beep
3	pi	the ratio of circle circumstance to its diameter
4	eps	the minimum number that should be added to 1 made the number greater than 1 (eps = 2.2204e-016)
5	inf	refer to infinity (i.e. 1/0)
6	NaN	refer to the input is (*not a number*) (i.e. 0/0)
7	i or j	refer to $\sqrt{-1}$

Starting with MATLAB

- **كيفية فتح برنامج ال MATLAB وغلقه :**

MATLAB (shortcut on desktop)

Or *start → all programs → MATLAB*

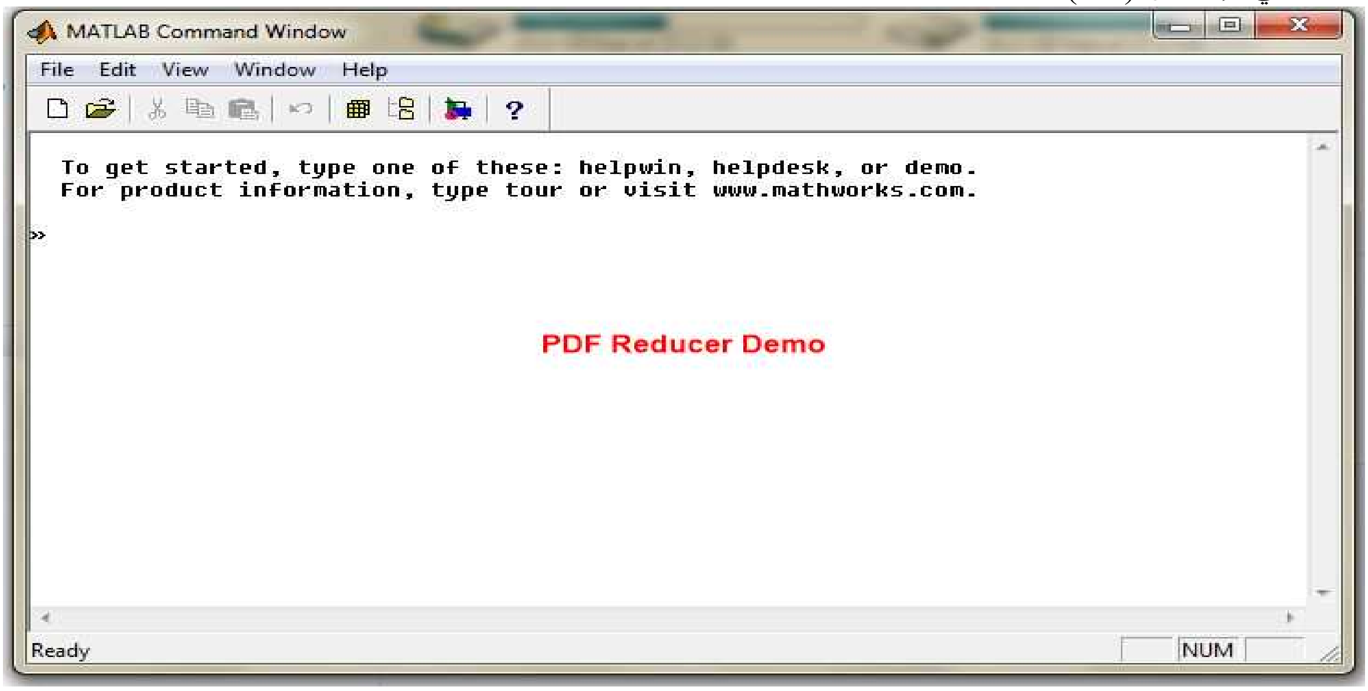
- **غلق برنامج MATLAB**

1. من زر الإغلاق X في نافذة واجهة تطبيق الماتلاب .

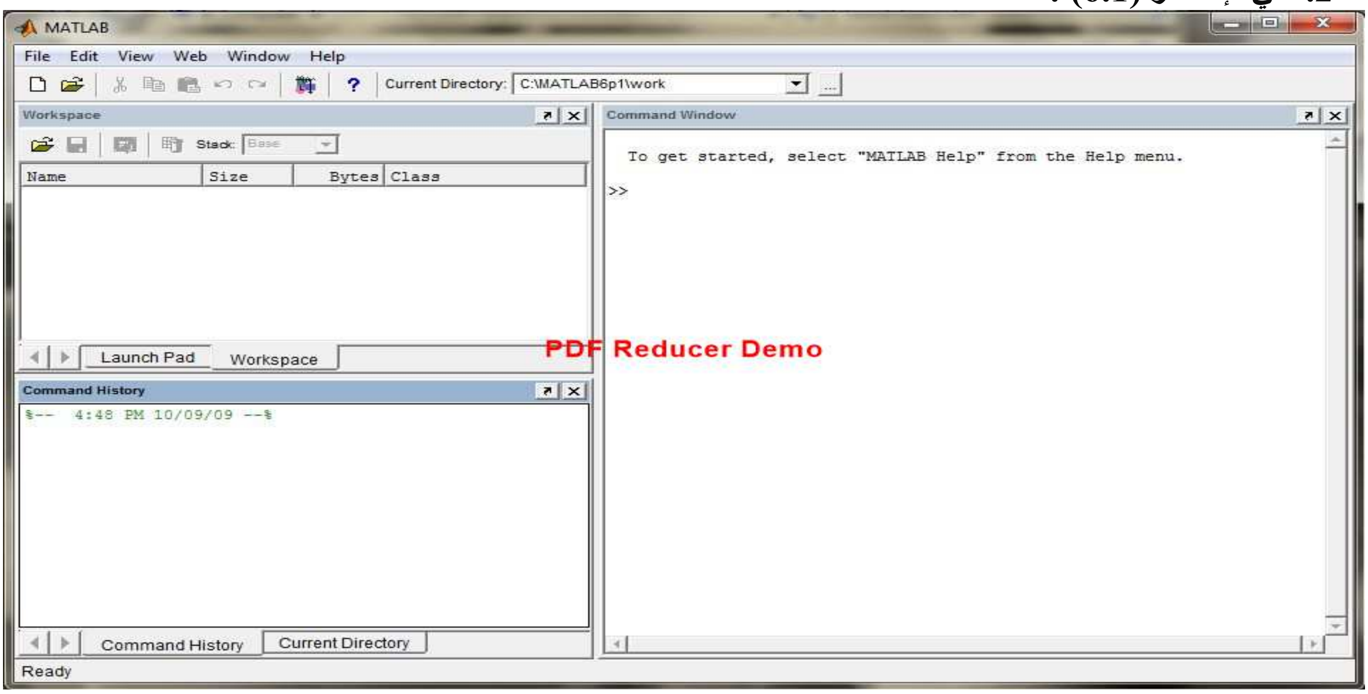
2. غلق برنامج MATLAB من خلال قائمة الملف File ← Exit Matlab

- **التعرف على واجهة التطبيق :**

1. في الإصدار (5.3) :



2. **في الإصدار (6.1) :**

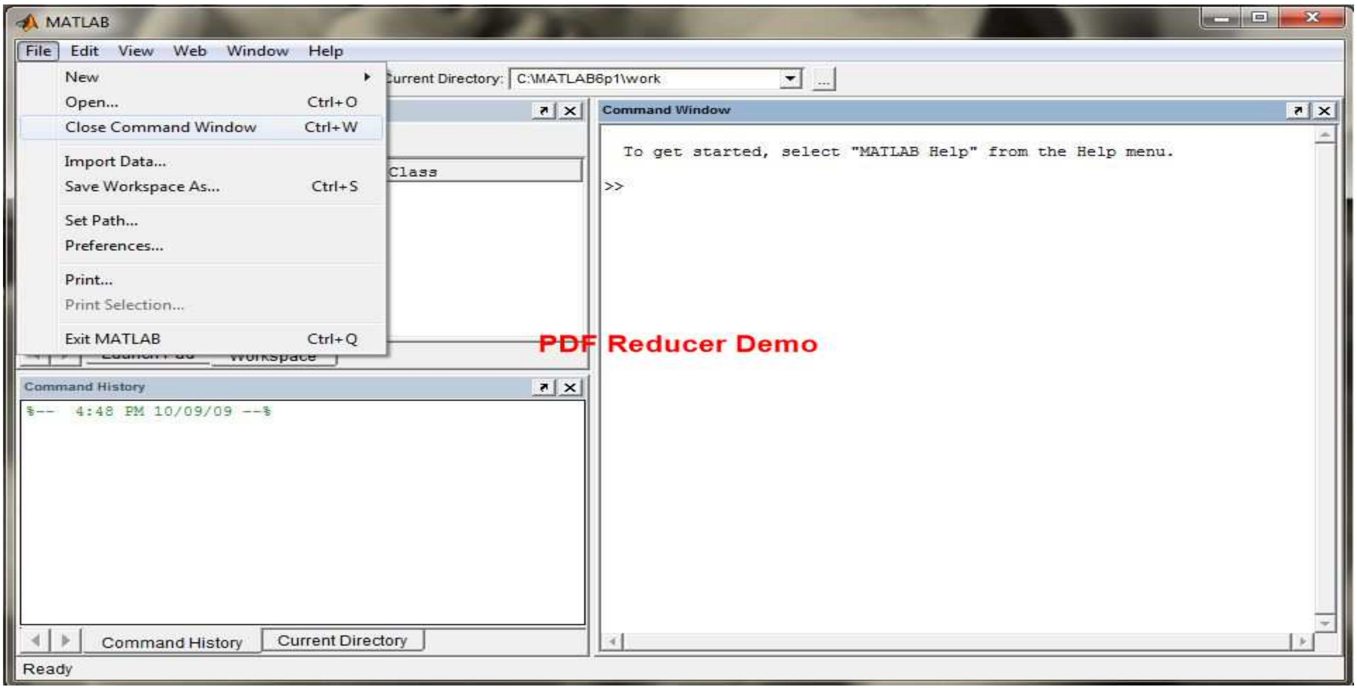


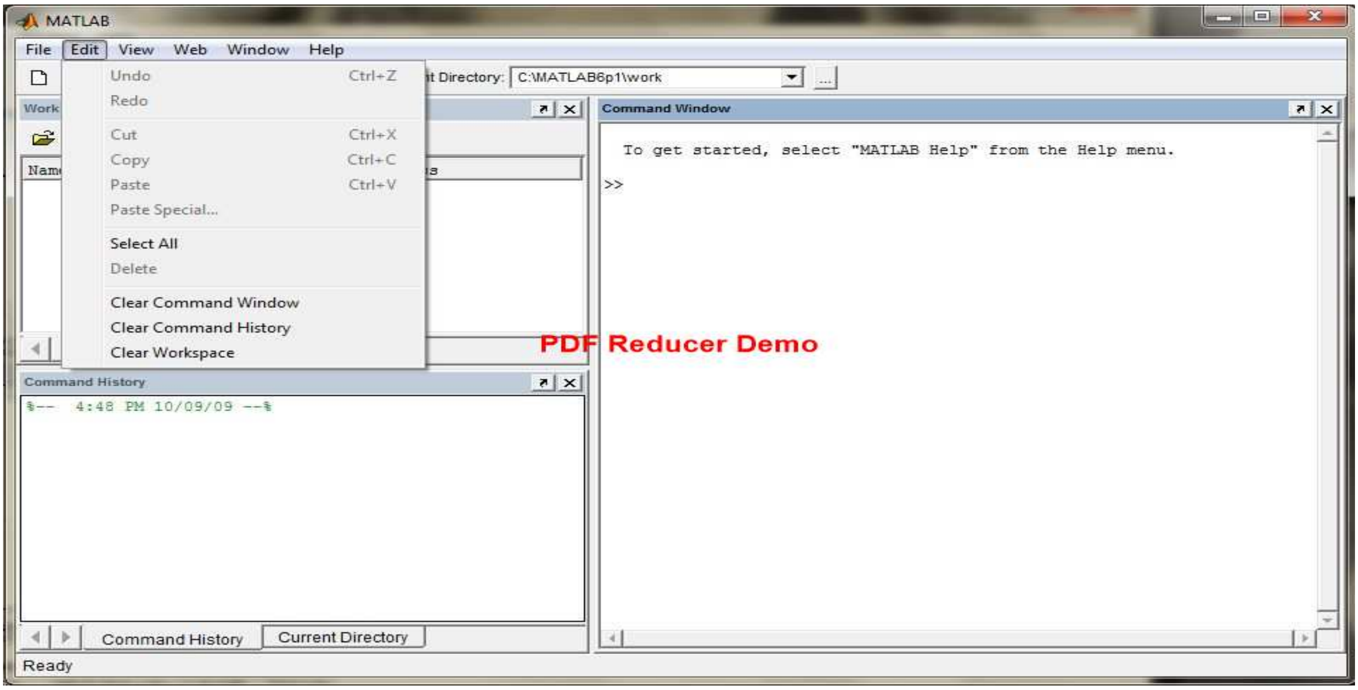
- وتتضمن النوافذ التالية :

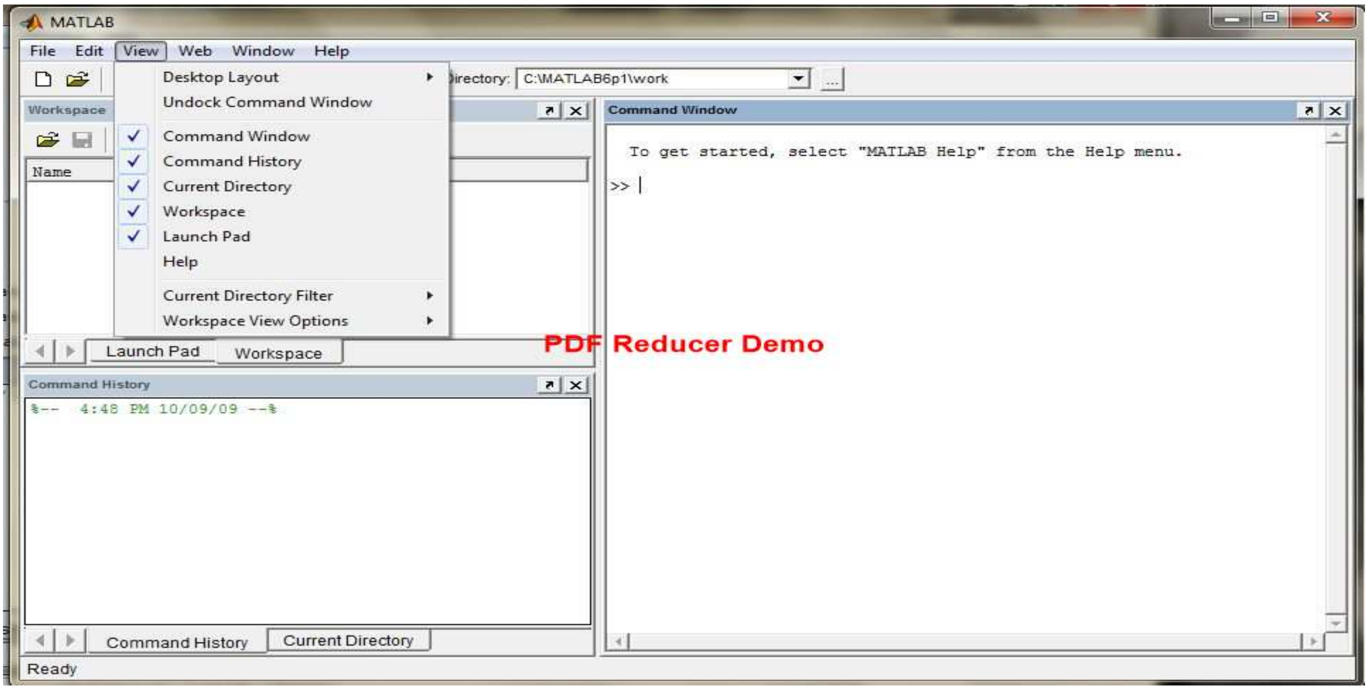
1. نافذة الأوامر (Command Window)	2. فضاء العمل (Workspace)	3. تاريخ الأوامر (Command History)

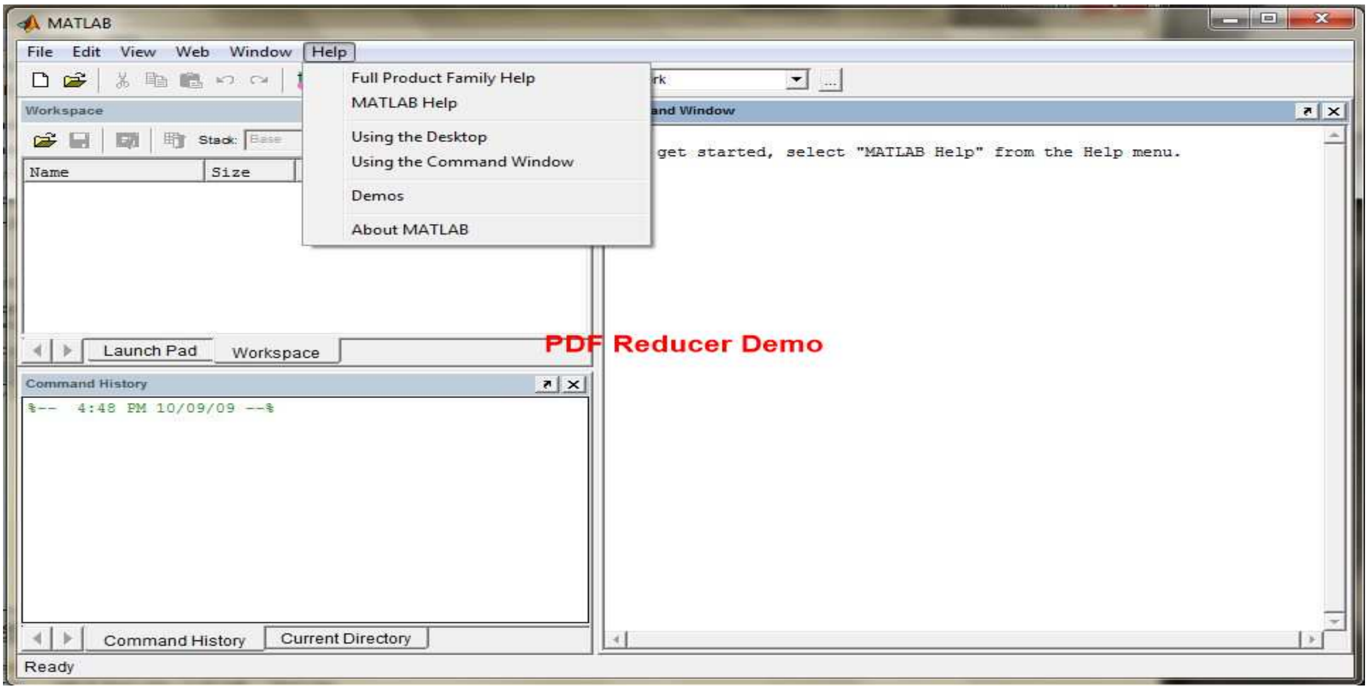
- شريط القوائم : ويتضمن القوائم التالية :

File	Edit	View	Web	Window	Help






MATLAB
File Edit View Web Window Help
Desktop Layout
Undock Command Window
Command Window
Command History
Current Directory
Workspace
Launch Pad
Help
Current Directory Filter
Workspace View Options
Directory: C:\MATLAB6p1\work
Workspace
Name
Command Window
To get started, select "MATLAB Help" from the Help menu.
>>
Launch Pad
Workspace
PDF Reducer Demo
Command History
%-- 4:48 PM 10/09/09 --%
Command History
Current Directory
Ready


MATLAB
File Edit View Web Window Help
Full Product Family Help
MATLAB Help
Using the Desktop
Using the Command Window
Demos
About MATLAB
Workspace
Stack: Base
Name
Size
get started, select "MATLAB Help" from the Help menu.
Launch Pad
Workspace
PDF Reducer Demo
Command History
%-- 4:48 PM 10/09/09 --%
Command History
Current Directory
Ready

Display Formats for Numeric

All the variables store and process in MATLAB in double precision form but to view the number there are eleven forms, which are:

No.	Format	Pi & x = -1.916e-4	Notes
1	*format short (default)*	*3.1416 , –1.9160 e–004*	(4) digits after mantissa
2	*format long*	*3.14159265358979 , –1.916000000000000 e– 004*	(16) digits after mantissa
3	*format short e*	*3.1416e + 000 , –1.9160e– 004*	(4) digits after mantissa with exponential power
4	*format long e*	*3.14159265358979 e+000 , –1.916000000000000 e–004*	(16) digits after mantissa with exponential power
5	*format short g*	*3.1416 , – 0.0001916*	the best between format short and format short e
6	*format long g*	*3.14159265358979 , – 0.0001916*	the best between format long and format long e
7	*format hex*	*400921fb54442d18 ,*	hexadecimal form
8	*format bank*	*3.14 , – 0.00*	two digit after mantissa
9	*format +*	+ –	positive (+) or Negative (–) or zero (0)
10	*format rat*	*355 / 113 – 29/151357*	rational form

Home Work
For the following numbers change the format of displaying number for all possible ways: -358.168e-5 , 196834.5677768129 , -85.23517291 , 1.6e-19

Defining Variables

In MATLAB you can process the data directly in command window, or you can define theme as variables then process theme, like:

Direct Data	Using Variables	
>> 24 + 5 ↵ ans = 29 >> ans – 5 ↵ ans = 24	>> x = 2 ↵ x = 2 >> y = x^2 ↵ y = 4 >> z = x + y ↵ z = 6	>> r = z / 3 ↵ r = 2 >> r * ((z – x) ^ y) ↵ ans = 512

Defining Matrices

In order to define and deal with matrices in MATLAB, there are certain characters must be taken into account:

1. **The brackets [ ]** are used to define a matrix.
2. **Separating the elements** we either use the ***comma (,)*** or the ***space (space)***.

Using Comma	Using Space
>> x = [1,2,3] ↵ x = 1 2 3	>> x = [1 2 3] ↵ x = 1 2 3

3. **To determine the end of the rows** we use **the *semicolon* (;)** or the ***enter*** (↵)

Define Matrix	Define Column Vector
>> x = [1 2 3; 4 5 6; 7 8 9] ↵ x = 1 2 3 4 5 6 7 8 9	>>y=[1↵ 2↵ 3] ↵ y = 1 2 3

Note that the semi-colon also used to hide the executing the result:

Without Using Semicolon	Using Semicolon	Without Using Semicolon	Using Semicolon
>> x = [1 2 3] ↵ x = 1 2 3	>> x = [1 2 3]; ↵	>>2+4↵ ans = 6	>>2+4; ↵

4. **The parentheses ( ) :**

The standard formula of parentheses is:

Operator (Target)

There are two different ways to use the parentheses:

a. To specify certain element in a matrix, the general form is:

A(row, column),

But for vectors A(index).

```
>> x = [4 5 6;3 2 1] ↵
x =  4  5  6
     3  2  1
>> x(1,2) ↵
>> ans =
        5
```

b. To rebuild a new matrix from a predefined matrix depending on the position sequence of the element in that matrix:

```
>> x = [4 5 6]; ↵
>>y =x([2  1  3  2  2]) ↵
y =  5    4    6    5    5
>> z = [-1 0 2;2 4 3;1 0 4] ↵
```

$$z = \begin{matrix} -1 & 0 & 2 \\ 2 & 4 & 3 \\ 1 & 0 & 4 \end{matrix}$$

```
>> r =z([2  5  7  2;1  2  4  9]) ↵
```

$$r = \begin{matrix} 2 & 4 & 2 & 2 \\ -1 & 2 & 0 & 4 \end{matrix}$$

c. To give a detail of the matrix:

```
>> x = [4 5 6]; ↵
>>size(x) ↵
ans =      1     3
>> y = [4 5 6;3 2 1] ↵
```

$$y = \begin{pmatrix} 4 & 5 & 6 \\ 3 & 2 & 1 \end{pmatrix}$$

```
>>size(y) ↵
ans =      2     3
```

5. The colon (:) used in three different ways:

a. To create a vector with defined increment as the following syntax

[*First value: Increment : Last Value*]

Positive Increment	Negative Increment
>> x = [1:2:10] ↵ x = 1 3 5 7 9	>>x = [10:-2:1] ↵ x = 10 8 6 4 2

b. To create a sub matrix of a certain row or column, **<u>or</u>** to re-arrange the matrix in one column.

Specified Row	Specified Column	One Column Matrix
>> x = [1 2 3; 4 5 6; 7 8 9]; ↵ >> y = x(2,:) ↵ y = 4 5 6	>> x = [1 2 3; 4 5 6; 7 8 9]; ↵ >> y = x(:,2) ↵ y = 2 5 8	>> x = [1 2 3; 4 5 6; 7 8 9]; ↵ >> y = x(:)↵ y = 1 4 7 2 5 8 3 6 9 Column after column

c. To create a sub matrix of defined rang of rows or columns, usually the word ***end*** used to indicate the last index in the range.

Define Matrix	Sub Matrix	Sub Matrix Using End
>> x = [1 2 3; 4 5 6; 7 8 9] ↵ x = 1 2 3 4 5 6 7 8 9	>> y = x(2:3,1:3) ↵ y = 4 5 6 7 8 9 **or**	>> y = x(2:end,1:end) ↵ y = 4 5 6 7 8 9

d. to create a new matrix by merging two (or more) predefined matrices

Define Matrices	Define Matrices
>> A = [1 2 3; 4 5 6; 7 8 9]; ↵ >> B = [-1 0 -3; -5 2 11; -2 8 0]; ↵ >> C=[A B] ↵ C= 1 2 3 -1 0 -3 4 5 6 -5 2 11 7 8 9 -2 8 0	>> A = [1 2 3; 4 5 6; 7 8 9]; ↵ >> B = [-1 0 -3; -5 2 11; -2 8 0]; ↵ >> D=[A;B] ↵ D= 1 2 3 4 5 6 7 8 9 -1 0 -3 -5 2 11 -2 8 0

Note/ we can also create a vector with defined increment by using the ***linespace*** function or ***logspace*** functions as the following

i)Using Colon
[First value: Increment: Last Value]
>> x = [0:0.1:1]*pi↵
x = 0 0.3142 0.6283 0.9425 1.2566 1.5708 1.8850 2.1991 2.5133 2.8274 3.1416
ii)Using Linspace
linspace(First value, Last value, Number of values)
>> x = linspace(0,pi,11) ↵
x = 0 0.3142 0.6283 0.9425 1.2566 1.5708 1.8850 2.1991 2.5133 2.8274 3.1416
iii)Using Logspace
logspace(First exponent, Last exponent, Number of values)
>> x = logspace(0,2,11) ↵
x = 1.0000 1.5849 2.5119 3.9811 6.3096 10.0000 15.8489 25.1189 39.8107 63.0957 100.00

Home Works
If $A_{3\times3} = \begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{bmatrix}$, find the following: **1.** Define the matrix A in MATLAB. **2.** Find the value of A(2,3), A(1,3), A(2,1), A(1,1). **3.** Create the following sub matrices from A: $i)B = \begin{bmatrix} 6 & 5 \\ 9 & 8 \end{bmatrix}$, ii)$C = \begin{bmatrix} 2 & 1 \\ 5 & 4 \end{bmatrix}$, iii)$D = \begin{bmatrix} 2 & 1 \\ 5 & 4 \\ 8 & 7 \end{bmatrix}$, iv)$E = \begin{bmatrix} 2 \\ 5 \\ 8 \end{bmatrix}$, v)$F = [3 \quad 2 \quad 1]$. **4.** Create the matrix ***G*** by merging the above ***D*** & ***E*** matrices. **5.** How could you merge the matrices ***D*** & ***F*** in one matrix? **6.** Create and name a (4×2) matrix from the ***B*** & ***C*** matrices.
Write the MATLAB statement to: 1- Find the even numbers in the range (0-1000). 2- Find the odd numbers in the range (1-1000). 3- Divide the range (0-6) to 20 numbers using linear manner. 4- Divide the range (1-1000) to 31 numbers using logarithmic manner.

Matrix and Element - wise Operations (using .)

Operator	Description
+ **Plus**	Addition. ***A*** + ***B*** adds the values stored in variables ***A*** and ***B***. ***A*** and ***B*** must have the same size, unless one is a scalar : (a scalar can be added to a matrix of any size)
- **Minus**	Subtraction. ***A*** **-** ***B*** subtracts the value of ***A*** and ***B***. ***A*** and ***B*** must have the same size, unless one is a scalar: (a scalar can be subtracted from a matrix of any size)

Examples

>>G=[-1 0 1;3 4 5]; ↵ >> G - 2↵ ans = $\begin{matrix} -3 & -2 & -1 \\ 1 & 2 & 3 \end{matrix}$	>>A=[1 2 3;4 5 6]; ↵ >>B=[2 1 5;0 3 4]; ↵ >>A+B↵ ans = $\begin{matrix} 3 & 3 & 8 \\ 4 & 8 & 10 \end{matrix}$	>>A=[1 2 3;4 5 6]; ↵ >>B=[2 1 5;0 3 4]; ↵ >>A+2-B-3↵ ans = $\begin{matrix} -2 & 0 & -3 \\ 3 & 1 & 1 \end{matrix}$

Operator	Description
***** × **multiply**	Matrix multiplication (The inner Product). ***C*** = ***A*** * ***B*** is the linear algebraic product of the matrices ***A*** and ***B***. More precisely, $$C(i,j) = \sum_{k=1}^{n} A(i,k)B(k,j)$$ For non-scalar ***A*** and ***B***, the number of columns of A must be equal to the number of rows of B. A scalar can multiply a matrix of any size.
.*	Element by element multiplication. ***A.*** * ***B*** is the element-by-element product of the matrices ***A and B***. ***A and B*** must have the same size, unless one of them is a scalar.

Examples

>>G=[-1 0 1;3 4 5]; ↵ >> G * 2↵ ans = $\begin{matrix} -2 & 0 & 2 \\ 6 & 8 & 10 \end{matrix}$	>>A=[1 2;-4 5]; ↵ >>B=[2 1;0 -1]; ↵ >>A* B↵ ans = $\begin{matrix} 2 & -1 \\ -8 & -9 \end{matrix}$	>>A=[1 2;-4 5]; ↵ >>B=[2 1;0 -1]; ↵ >>A .*B↵ ans = $\begin{matrix} 2 & 2 \\ 0 & -5 \end{matrix}$

Operator	Description
/ **Division**	Slash or matrix right division. ***B/A*** is roughly the same as ***B * inv(A)***. More precisely, ***B / A = (A' \ B')'.***
./	Matrix right division. ***A . / B*** is the matrix with elements ***A(i , j)/ B(i , j)***. ***A and B*** must have the same size, unless one of them is a scalar.
\	Backslash or matrix left division. If ***A*** is a square matrix, ***A\ B*** is roughly the same as ***inv(A)*B*** , except it is computed in a different way. If ***A*** is an **n-by-n** matrix and ***B*** is a column vector with n components, or a matrix with several such columns, then ***X = A \ B*** is the solution to the equation ***AX = B***. a warning message is displayed if ***A*** is badly scaled or nearly singular.
.\	Matrix left division. ***A. \ B*** is the matrix with elements ***B(i, j) / A(i, j)***. ***A*** and ***B*** must have the same size, unless one of them is a scalar.

Examples

>>G=[-1 0 1;3 4 5]; ↵ **>> G / 2↵** **ans =** $\begin{matrix} -0.5 & 0 & 0.5 \\ 1.5 & 2 & 2.5 \end{matrix}$ $G \div 2 = G \times \frac{1}{2}$	**>>A=[1 2;-4 5]; ↵** **>>B=[2 1;0 -1]; ↵** **>>A / B↵** ≡ A * inv(B) **ans =** $\begin{matrix} 0.5 & -1.5 \\ -2 & -7 \end{matrix}$ $A \times \frac{1}{B}$	**>>A=[1 2;-4 5]; ↵** **>>B=[2 1;0 -1]; ↵** **>>B / A↵** ≡(B * inv(A)) **ans =** **1.0769 -0.2308** **-0.3077 -0.0769** $B \times \frac{1}{A}$
>>G=[-1 0 1;3 4 5]; ↵ **>> 2 \ G↵** ≡(G / 2) **ans =** $\begin{matrix} -0.5 & 0 & 0.5 \\ 1.5 & 2 & 2.5 \end{matrix}$ $G \div 2$	**>>A=[1 2;-4 5]; ↵** **>>B=[2 1;0 -1]; ↵** **>>A \ B↵** ≡ inv(A)*B **ans =** $\begin{matrix} 0.7692 & 0.5385 \\ 0.6154 & 0.2308 \end{matrix}$ $\frac{1}{A} \times B$	**>>A=[1 2;-4 5]; ↵** **>>B=[2 1;0 -1]; ↵** **>>B \ A↵** ≡(inv(B) * A) **ans =** **-1.5000 3.5000** **4.0000 -5.0000** $\frac{1}{B} \times A$

>>A=[1 2;-4 5]; ↵ **>>B=[2 1;0 -1]; ↵** **>>A . / B↵** **ans =** $\frac{1}{2}$ 2 $-inf$ -5	**>>A=[1 2;-4 5]; ↵** **>>B=[2 1;0 -1]; ↵** **>>B . \ A↵** **ans =** $\frac{1}{2}$ 2 $-inf$ -5
>>A=[1 2;-4 5]; ↵ **>>B=[2 1;0 -1]; ↵** **>>B . / A** **ans =** **2.0000 0.5000** **0 -0.2000**	**>>A=[1 2;-4 5]; ↵** **>>B=[2 1;0 -1]; ↵** **>>A . \ B↵** **ans =** **2.0000 0.5000** **0 -0.2000**

Operator	Description
^ **Power**	Matrix power. ***X^p*** is X^p, if ***p*** is a scalar. If ***p*** is an integer, If the integer is negative, ***X*** is inverted first.
.^	Matrix power. ***A . ^ B*** is the matrix with elements ***A(i , j)*** to the ***B(i , j)*** power. ***A*** and ***B*** must have the same size, unless one of them is a scalar.

Examples

>>A=[1 2;-4 5]; ↵ >>A ^ 2 ↵ = **A * A** ans = -7 12 -24 17 = $\begin{pmatrix} 1 & 2 \\ -4 & 5 \end{pmatrix} \times \begin{pmatrix} 1 & 2 \\ -4 & 5 \end{pmatrix}$	>>A=[1 2;-4 5]; ↵ >>A .^ 2 ↵ ans = 1 4 16 25 = $\begin{pmatrix} 1^2 & 2^2 \\ (-4)^2 & 5^2 \end{pmatrix}$

<table>
<tr><th>Operator</th><th colspan="2">Description</th></tr>
<tr><td>'
Transpose</td><td colspan="2">Matrix transpose.
A' is the linear algebraic transpose of A.
For complex matrices, this is the complex conjugate transpose.</td></tr>
<tr><td>.'</td><td colspan="2">Matrix transpose.
A.' is the Matrix transpose of A.
For complex matrices, this does not involve conjugation.</td></tr>
<tr><th colspan="3">Examples</th></tr>
<tr><td>>>A=[1 2;-4 5] ↵
A=
$\begin{matrix} 1 & 2 \\ -4 & 5 \end{matrix}$
>>A'↵
ans =
$\begin{matrix} 1 & -4 \\ 2 & 5 \end{matrix}$
transpose only</td><td>>>B=[1+i 2-2i] ↵
B=
1+i 2-2i
>>B'
ans =
$\begin{matrix} 1-i \\ 2+2i \end{matrix}$
transpose & complex Conjugate</td><td>>>C=[1+i , 2+3i;-i , -2i] ↵
>>C↵
C =
$\begin{matrix} 1+i & 2+3i \\ -i & -2i \end{matrix}$
>>C '↵
ans =
$\begin{matrix} 1-i & +i \\ 2-3i & +2i \end{matrix}$
transpose & complex Conjugate</td></tr>
<tr><td>>>A=[1 2;-4 5] ↵
A=
$\begin{matrix} 1 & 2 \\ -4 & 5 \end{matrix}$
>>A .'↵
ans =
$\begin{matrix} 1 & -4 \\ 2 & 5 \end{matrix}$
transpose only</td><td>>>B=[1+i 2-2i] ↵
B=
1+i 2-2i
>>B .'
ans =
$\begin{matrix} 1+i \\ 2-2i \end{matrix}$
transpose only</td><td>>>C=[1+i , 2+3i;-i , -2i] ↵
C =
$\begin{matrix} 1+i & 2+3i \\ -i & -2i \end{matrix}$
>>C .'↵
ans =
$\begin{matrix} 1+i & -i \\ 2+3i & -2i \end{matrix}$
transpose only</td></tr>
</table>

Homework
You have the matrices $A = \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}, B = \begin{matrix} -1 & 3 \\ 5 & 4 \end{matrix}, X = \begin{matrix} 1+i & 2+2i \\ 1-2i & 2+i \end{matrix}, Y = \begin{matrix} 4 & 2 \\ 1 & 3 \end{matrix}$ **1.** How could you get the conjugate and transpose of matrices ***X*** & ***Y*** ?   **2.** Determine (A.*B) , (Y ^3) , (B.^A) , (B / A) , (A+B)*(Y^2) /3 , (A+B)+(Y^2) /3

Creating Standard Matrices in MATLAB

MATALB provide functions to create standard matrix like I , D, 0,etc.. these functions are :

1. **Ones ( ):** create matrix all its elements are equal to one.

Create Square Matrix	Create m × n Matrix
>> ones(3) ↵ for the square matrices ans = 1 1 1 1 1 1 1 1 1	>> ones(3,4) ↵ for non-square matrices ans = 1 1 1 1 1 1 1 1 1 1 1 1

2. **Zeros ( ):** create matrix all its elements are zeros.

Create Square Matrix	Create m × n Matrix
>> zeros(3) ↵ ans = 0 0 0 0 0 0 0 0 0	>> zeros(3,4) ↵ for non-square matrices ans = 0 0 0 0 0 0 0 0 0 0 0 0

3. **eye ( ):** create unity matrix (I) .

Create Square Matrix	Create m × n Matrix
>> eye(3) ↵ ans = **1** 0 0 0 **1** 0 0 0 **1**	>>eye(3,4) ↵ for non-square matrices ans = **1** 0 0 0 0 **1** 0 0 0 0 **1** 0

4. **rand ( ):** create random matrix its elements are in the range (0 to 1).

Create Square Matrix	Create **m** × **n** Matrix
>> rand(3) ↵ ans = 0.8147 0.9134 0.2785 0.9058 0.6324 0.5469 0.1270 0.0975 0.9575	>> rand (3,1) ↵ for non-square matrices ans = 0.9649 0.1576 0.9706

5. **diag( ) :** is used to create the Diagonal matrices or determine the diagonal of matrix as:

   a. Create diagonal matrix (D), using the predefined vector (V), its position is determine by the value of k.

D=(V, k)

Where the **(k)** is the shift of **(V)** about the diagonal of matrix **(D)**

$$k = \begin{cases} = 0 & \textit{on the main diagonal} \\ > 0 & \textit{upper the main diagonal} \\ < 0 & \textit{under the main diagonal} \end{cases}$$

Create Diagonal Matrix	Create Diagonal Matrix away from main diagonal	
>> V = [1 2 3]; ↵ >>diag(V,0) ↵ ans = **1** 0 0 0 **2** 0 0 0 **3**	>> V = [1 2 3]; ↵ >>diag(V,1) ↵ ans = **0** **1** 0 0 0 **0** **2** 0 0 0 **0** **3** 0 0 0 **0**	>> V = [1 2 3]; >>diag(V,-2) ans = **0** 0 0 0 0 0 **0** 0 0 0 **1** 0 **0** 0 0 0 **2** 0 **0** 0 0 0 **3** 0 **0**

The diagonal vector can be given in raw or column vector (not a matter).

   b. to determine the main diagonal of predefined matrix:
   (Not that the extracted diagonal represented in column vector)

Determine the Matrix Diagonal
>>x = [0 7 4;4 7 4;3 1 6]; x = **0** 7 4 4 **7** 4 3 1 **6** >>diag(x) ans = **0** **7** **6**

6. creation of equal elements matrix there are four ways to do:

Via Multiplication (Very Slow)	Via Addition (Slow)	Via Direct Substitution (Fast)	Via repeat (Very Fast)
>>ones(2,3)*5 ans = 5 5 5 5 5 5	>>5+zeros(2,3) ans = 5 5 5 5 5 5	>>d=5; >>d*(ones(2,3)) ans = 5 5 5 5 5 5	>>repmat(**5**, 2,3) ans = 5 5 5 5 5 5 **Replicate and tile array**

Functions Frequently Used With Matrices

There are few functions that are frequently used with matrices, these functions are:

1. **format** : Set display format for output .

2. **clc** : Clear Command Window (clears all input and output from the Command Window display, giving you a "clean screen." After using **clc**, you cannot use the scroll bar to see the history of functions, but you still can use the up arrow to recall statements from the command history.

3. **clear :** Remove items from workspace, freeing up system memory (clear removes all variables from the workspace, releasing them from system memory).

4. **who** : List variables in workspace (who lists in alphabetical order all variables in the currently active workspace).

5. **whos :** List variables in workspace, with sizes and types.

6. **Length( ): return the Length of vector or largest matrix dimension:**

Length of Vector	Largest Matrix Dimensions
>> x = [1 3 5 7];↵ >> length(x) ans = 4	>> x = [1 2 3;4 5 6];↵ >> length(x) ans = 3

7. **size( ): return the dimensions of a matrix:**

All Dimensions	Two Dimensions	Specified Dimension
>> x = [1 2 3;4 5 6];↵ >> size(x) ↵ ans = 2 3	>> x = [1 2 3;4 5 6]; ↵ >> [rows columns] = size(x) ↵ rows = 2 columns = 3	>> x = [1 2 3;4 5 6]; ↵ >> rows = size(x,1) ↵ rows = 2 >> columns = size(x,2) ↵ columns = 3

8. **numel ( ): return the number of elements in a matrix or a vector:**

Number of Elements in a Vector	Number of Elements in a Matrix
>> x = [1 3 5 7]; ↵ >>numel(x) ans = 4	>> x = [1 2 3;4 5 6]; ↵ >>numel(x) ans = 6

9. max( ): return the maximum number in a vector or in each column in the matrix:

Maximum of a Vector	Maximums in the Columns	Maximum in a Matrix
>> x = [1 3 5 7]; ↵ >>max(x) ans = 7 returns the largest elements along different dimensions of an array	>> x = [1 2 3;4 5 6]; ↵ >> max(x) ans = 4 5 6	>> x = [1 2 3;4 5 6]; ↵ >> max(x(:)) ans = 6 or also >>**max(max(x))**]; ↵
Compare each element of x to a scalar (smaller than this scalar) and replace it by this scalar		
>> x= [2 8 4; 7 3 9] ; ↵ >> max(x,4) ↵ ans = $\boxed{4}$ 8 4 $\boxed{2}$ 8 4 7 $\boxed{4}$ 9 7 $\boxed{3}$ 9	>> x= [2 8 4; 7 3 9] ; ↵ >> max(x,5) ↵ ans = $\boxed{5}$ 8 $\boxed{5}$ $\boxed{2}$ 8 $\boxed{4}$ 7 $\boxed{5}$ 9 7 $\boxed{3}$ 9	

10. Min( ): return the minimum number in a vector or in each column in a matrix

Minimum of a Vector	Minimums in the Columns	Minimum in a Matrix
>> x = [1 3 5 7]; ↵ >>min(x) ans = 1 returns the minimum elements along different dimensions of an array	>> x = [1 2 3;4 5 6]; ↵ >> min(x) ans = 1 2 3	>> x = [1 2 3;4 5 6]; ↵ >> min(x(:)) ans = 1 >>**min(min(x))**; ↵
Compare each element of x to a scalar (smaller than this scalar) and replace it by this scalar		
>> x= [2 8 4; 7 3 9] ; ↵ >> min(x,4) ↵ ans = 2 $\boxed{4}$ 4 2 $\boxed{8}$ 4 $\boxed{4}$ 3 $\boxed{4}$ $\boxed{7}$ 3 $\boxed{9}$	>> x= [2 8 4; 7 3 9] ; ↵ >> min(x,7) ↵ ans = 2 $\boxed{7}$ 4 2 **8** 4 7 3 $\boxed{7}$ 7 3 **9**	

11. mean( ) : return the average value of vector or the averages of each column:

Average of a Vector	Average of the Columns	Average of a Matrix
>> x = [1 3 5 7]; ↵ >>mean(x) ans = 4 returns the average elements along different dimensions of an array	>> x = [1 2 3;4 5 6]; ↵ >> mean(x) ans = 2.5 3.5 4.5	>> x = [1 2 3;4 5 6]; ↵ >> mean(x(:)) ans = 3.5 >>**mean(mean(x))**; ↵

12. sum( ): return the sum of the entries of a vector or the columns of a matrix.

Sum of a Vector	Sum of the Columns	Sum of a Matrix
>> x = [1 3 5 7]; ↵ >>sum(x) ans = 16 returns sums along different dimensions of an array	>> x = [1 2 3;4 5 6]; ↵ >> sum(x) ans = 5 7 9	>> x = [1 2 3;4 5 6]; ↵ >> sum(x(:)) ≡ sum(sum(x)) ans = 21

13. dot(A,B): return the dot product of matrices *A* & *B*

dot prod. of two Vectors	dot prod. of the matrices
>> a = [1 2 3]; b = [4 5 6]; ↵ >>dot(a,b) ↵ (1×4)+ (2×5)+ (3×6) ans = 32	>> x = [1 2 3;4 5 6]; >> y = [4 0 2;1 0 3]; >>dot(x,y) ↵ ans = 8 0 24 (1×4)+(4×1) (2×0)+(5×0) (3×2)+(6×3)

14. Cross(A,B): return the cross product of matrices A & B

Cross prod. of two Vectors

>> a = [1 2 3]; b = [4 5 6]; ↵

>>cross(a,b) ↵ $\equiv a\times b = \begin{array}{ccc} i & j & k \\ 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} = i(2\times 6-3\times 5)-j(1\times 6-3\times 4)+k(1\times 5-2\times 4)$

$= i(-3) - j(-6) + k(-3)$

ans =

-3 6 -3

Cross prod. of the matrices

>> x = [1 2 3;4 5 6]; ↵

>> y = [4 0 2;1 0 3]; ↵

>> cross (x,y) ↵

$$\equiv x\times y = \begin{array}{ccc} i & j & k \\ \hline 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} \times \begin{array}{ccc} i & j & k \\ \hline 4 & 0 & 2 \\ 1 & 0 & 3 \end{array} = \left\{ \begin{array}{ccc} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ 1 & 2 & 3 \\ 4 & 0 & 2 \\ \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ 4 & 5 & 6 \\ 1 & 0 & 3 \end{array} \right. = \begin{array}{l} i(2\times 2-3\times 0)-j(1\times 2-3\times 4)+k(1\times 0-2\times 4) \\ i(5\times 3-6\times 0)-j(4\times 3-6\times 1)+k(4\times 0-5\times 1) \end{array}$$

$$= \begin{array}{ccccc} i(4) & - & j(-10) & + & k(-8) \\ i(15) & - & j(6) & + & k(-5) \end{array}$$

$$\text{ans} = \begin{array}{ccc} 4 & 10 & -8 \\ 15 & -6 & -5 \end{array}$$

15. prod( ): return the Product of array elements.

<table>
<tr><th colspan="2">Prod. of a Vector elements</th><th>Prod. of a Matrix column elements</th></tr>
<tr><td>>> x = [1 3 5 7]; ↵
>>prod(x)
ans =
105

Returns the product of the row elements.</td><td>>> x = [1; 2;3]; ↵
>> prod(x)
ans =
6

Returns the product of the column elements.</td><td>>> x = [1 2 3;4 5 6]; ↵
>> prod(x)
ans =
4 10 18

Returns the products along columns matrix.</td></tr>
<tr><th colspan="3">Prod. of a Matrix elements</th></tr>
<tr><td>>> x = [1 2 3;4 5 6]; ↵
>> prod(prod(x)) ↵
ans =
720</td><td colspan="2">>> x = [1 2 3;4 5 6]; ↵
>> prod(x (:)) ↵
ans =
720</td></tr>
</table>

16.trace( ) : Sum of diagonal elements

```
>>a=[1 2 3;4 5 6;7 8 9] ↵
```

$$a = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

```
>>trace(a) ↵
ans =
      15
```

17. tril( ) : Lower triangular part of matrix

```
>>r=[1 2 3 5;4 5 6 7;7 8 9 2;8 6 4 2]↵
```

$$r = \begin{matrix} 1 & 2 & 3 & 5 \\ 4 & 5 & 6 & 7 \\ 7 & 8 & 9 & 2 \\ 8 & 6 & 4 & 2 \end{matrix}$$

```
>>tril(r) ↵

ans =
          1   0   0   0
          4   5   0   0
          7   8   9   0
          8   6   4   2
```

18.triu( ) : Upper triangular part of matrix

>>r=[1 2 3 5;4 5 6 7;7 8 9 2;8 6 4 2]↵

$$r = \begin{matrix} 1 & 2 & 3 & 5 \\ 4 & 5 & 6 & 7 \\ 7 & 8 & 9 & 2 \\ 8 & 6 & 4 & 2 \end{matrix}$$

>>triu(r) ↵

ans =

$$\begin{matrix} 1 & 2 & 3 & 5 \\ 0 & 5 & 6 & 7 \\ 0 & 0 & 9 & 2 \\ 0 & 0 & 0 & 2 \end{matrix}$$

19.det( ) : Matrix determinant .

>>r=[1 2 3 5;4 5 6 2;7 8 -2 2;8 1 4 2]↵

$$r = \begin{matrix} 1 & 2 & 3 & 5 \\ 4 & 5 & 6 & 7 \\ 7 & 8 & 9 & 2 \\ 8 & 6 & 4 & 2 \end{matrix}$$

>>det(r) ↵

ans =

1.8060e+003

Returns the determinant of the square matrix**.**

20. inv( ) : Matrix inverse.

>>x=[1 2 3;4 5 6;7 8 -2]↵

$$r = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & -2 \end{matrix}$$

>>inv(x) ↵

ans =

$$\begin{matrix} -1.7576 & 0.8485 & -0.0909 \\ 1.5152 & -0.6970 & 0.1818 \\ -0.0909 & 0.1818 & -0.0909 \end{matrix}$$

Returns the inverse of the square matrix**.**

21. fix( ) : rounds the elements of matrix toward zero, resulting in an array of integers

>>x=[-1.7576 0.8485 -5.0909; 1.5152 -3.6970 0.1818] ↵ $x = \begin{matrix} -1.7576 & 0.8485 & -5.0909 \\ 1.5152 & -3.6970 & 0.1818 \end{matrix}$ >>fix(x) ↵ ans = $\begin{matrix} -1 & 0 & -5 \\ 1 & -3 & 0 \end{matrix}$

22. round( ) : Round to nearest integer

>>x=[-1.7576 0.8485 -5.0909; 1.5152 -3.6970 0.1818] ↵ $x = \begin{matrix} -1.7576 & 0.8485 & -5.0909 \\ 1.5152 & -3.6970 & 0.1818 \end{matrix}$ >>fix(x) ↵ ans = $\begin{matrix} -2 & 1 & -5 \\ 2 & -4 & 0 \end{matrix}$ rounds the elements of matrix to the nearest integers.

23. abs( ) : Absolute value and complex magnitude

Integer number	Complex number
>>x= -5 ; ↵ >>abs(x) ↵ ans = 5	>>*x= -3 + 4 i* ; ↵ >>abs(x) ↵ ans = 5 i.e. $abs(x) = \sqrt{(-3)^2 + (4)^2} = \sqrt{9+16} = \sqrt{25} = 5$
$\sqrt{(\mathbf{real}(\mathbf{x}).\,\hat{}\,\mathbf{2} + \mathbf{imag}(\mathbf{x}).\,\hat{}\,\mathbf{2})}$	

24. sqrt ( ) : Square root

Integer number	Imaginary number
>>x= 9 ; ↵ >>sqrt(x) ↵ ans = 3 sqrt(x returns the square root of each element of the array X. For the elements of X that are negative or complex, sqrt(X) produces complex results.	x= -5 ; ↵ sqrt(x) ↵ ans = *0 + 2.2361i*
Square root of matrix	
>>x=[4 9;16 25] ↵ >>x = 4 9 16 25	
>>r= x^2↵ $\equiv \begin{pmatrix} 4 & 9 \\ 16 & 25 \end{pmatrix} \times \begin{pmatrix} 4 & 9 \\ 16 & 25 \end{pmatrix}$ $\equiv \begin{matrix} (4\times 4)+(9\times 16) & (4\times 9)+(9\times 25) \\ (16\times 4)+(25\times 16) & (16\times 9)+(25\times 25) \end{matrix}$ $\equiv \begin{pmatrix} 16+144 & 36+225 \\ 64+400 & 144+625 \end{pmatrix}$ r= 160 261 464 769	>>sqrtm(r) ↵ $\equiv \sqrt{\begin{pmatrix} 160 & 261 \\ 464 & 769 \end{pmatrix}}$ ans = 4 9 16 25 ≡ x
>> q=x.^2↵ $\equiv \begin{pmatrix} 4^2 & 9^2 \\ 16^2 & 25^2 \end{pmatrix}$ q= 16 81 256 625	>>sqrt(q) ↵ $\equiv \begin{pmatrix} \sqrt{16} & \sqrt{81} \\ \sqrt{256} & \sqrt{625} \end{pmatrix}$ ans = 4 9 16 25 ≡ x

25. log( ): natural logarithm (Ln)

natural logarithm of number	natural logarithm of matrix
>>x= -5 ; ↵ >>log(x) ↵ $\equiv ln(x) = ln(-5)$ ans = 1.61	>> y=[1 2 3]; ↵ >> log(y) ↵ $\equiv$ $ln(1)$ $ln(2)$ $ln(3)$ ans = 0 0.69 1.10 The log function operates element-wise on matrix.

26. exp( ): Exponential

Exponential of number	Exponential of matrix
>>x= -5 ; ↵ >>exp(x) ↵ ≡ $e^x = e^{-5}$ ans = 0.006737946999085 >>log(exp(x)) ; ↵ ans = -5	>> y=[1 2 3]; ↵ >> exp(y) ↵ ≡ $(e^1 \quad e^2 \quad e^3)$ ans = 2.72 7.39 20.09 returns the exponential for each element of y >>log(exp(y)) ; ↵ ans = 1 2 3

27. log2( ): computes the base 2 logarithm

>>x= 8 ; ↵ >>log2(x) ↵ (which number must be the power to the base 2 to give 8) ans = 3 ≡ $2^3 = 8$	>> y=[1 2 3]; ↵ >> log2(y) ↵ ≡ $2^x = 1 \quad 2^x = 2 \quad 2^x = 3$ ans = 0 1.0000 1.5850

28. log10( ): computes the base 10 logarithm

logarithm of number	logarithm of matrix
>>x= 1000 ; ↵ >>log10(x) ↵ (which number must be the power to the base 10 to give 1000) ans = 3 ≡ $10^3 = 1000$	>> y=[1 2 3]; ↵ >> log10(y) ↵ ≡ $10^x = \mathbf{1} \quad 10^x = \mathbf{2} \quad 10^x = \mathbf{3}$ ans = 0 0.3010 0.4771

29. conj(x) : returns the complex conjugate of the elements or matrix

Complex number	Complex matrix
>>x= 2 - 3i ; ↵ >>conj(x) ↵ ans = $2 + 3i$	>> y=[1+3i 2 -i;-2-2i 5+i 4i]; ↵ y= $\begin{matrix} 1+3i & 2 & -i \\ -2-2i & 5+i & 4i \end{matrix}$ >> conj(y) ↵ ans = $\begin{matrix} 1-\boldsymbol{3i} & 2 & +\boldsymbol{i} \\ -2+\boldsymbol{2i} & 5-\boldsymbol{i} & -\boldsymbol{4i} \end{matrix}$

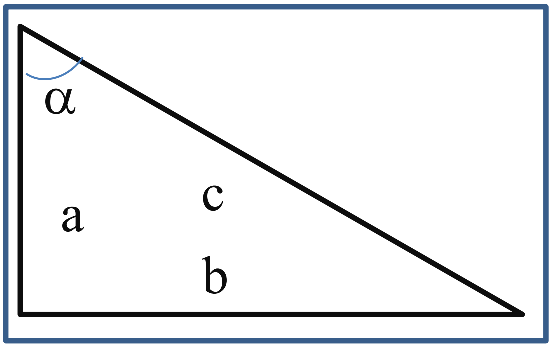
30. real(x) : returns the real part of the elements of the matrix

Complex number	Complex matrix
>>x= 2 - 3i ; ↵ >>real(x) ↵ ans = 2	>> y=[1+3i 2 -i;-2-2i 5+i 4i]; ↵ y= $\begin{matrix} 1+3i & 2 & -i \\ -2-2i & 5+i & 4i \end{matrix}$ >> real(y) ↵ ans = $\begin{matrix} 1 & 2 & 0 \\ -2 & 5 & 0 \end{matrix}$

31. imag(x) : returns the imaginary part of the elements of matrix

Complex number	logarithm of matrix
>>x= 2 - 3i ; ↵ >>imag(x) ↵ ans = 3	>> y=[1+3i 2 -i;-2-2i 5+i 4i]; ↵ y= $\begin{matrix} 1+3i & 2 & -i \\ -2-2i & 5+i & 4i \end{matrix}$ >> imag(y) ↵ ans = $\begin{matrix} 3 & 0 & -1 \\ -2 & 1 & 4 \end{matrix}$

(Trigonometric functions) الدوال المثلثية



Sin(α)	b/c
cos(α)	a/c
tan(α)	Sin(α)/cos(α)=b/a
sec(α)	1/cos(α)=c/a
csc(α)	1/sin(α)=c/b
cot(α)	1/tan(α)

	0° =	**30° =**	**45°=**	**60°=**	**90°=**	**120°=**	**135°=**	**150°=**	**180°=**
Sin(α)	0	1/2	$1/\sqrt{2}$	$\sqrt{3}/2$	1	$\sqrt{3}/2$	$1/\sqrt{2}$	1/2	0
cos(α)	1	$\sqrt{3}/2$	$1/\sqrt{2}$	1/2	0	-1/2	$-1/\sqrt{2}$	$-\sqrt{3}/2$	-1
tan(α)	0	$1/\sqrt{3}$	1	$\sqrt{3}$	∞	$-\sqrt{3}$	-1	$-1/\sqrt{3}$	0
cot(α)	∞	$\sqrt{3}$	1	$1/\sqrt{3}$	0	$-1/\sqrt{3}$	-1	$-\sqrt{3}$	-∞
sec(α)	1	$2/\sqrt{3}$	$\sqrt{2}$	2	∞	-2	$-\sqrt{2}$	$-2/\sqrt{3}$	-1
csc(α)	∞	2	$\sqrt{2}$	$2/\sqrt{3}$	1	$2/\sqrt{3}$	$\sqrt{2}$	2	∞

الدوال المثلثية المعكوسة	
MATLAB	الدالة
asin()	Inverse Sine
acos()	Inverse Cosine
atan()	Inverse Tangent
acot()	Inverse Cotangent
asec()	Inverse Secant
acsc()	Inverse Cosecant

الدوال المثلثية قطع زائد	
MATLAB	الدالة
sinh()	Hyperbolic Sine
cosh()	Hyperbolic Cosine
tanh()	Hyperbolic Tangent
coth()	Hyperbolic Cotangent
sech()	Hyperbolic Secant
csch()	Hyperbolic Cosecant

الدوال المثلثية قطع زائد المعكوسة	
MATLAB	الدالة
asinh()	Inverse Hyperbolic Sine
acosh()	Inverse Hyperbolic Cosine
atanh()	Inverse Hyperbolic Tangent
acoth()	Inverse Hyperbolic Cotangent
asech()	Inverse Hyperbolic Secant
acsch()	Inverse Hyperbolic Cosecant

Drawing Using MATLAB

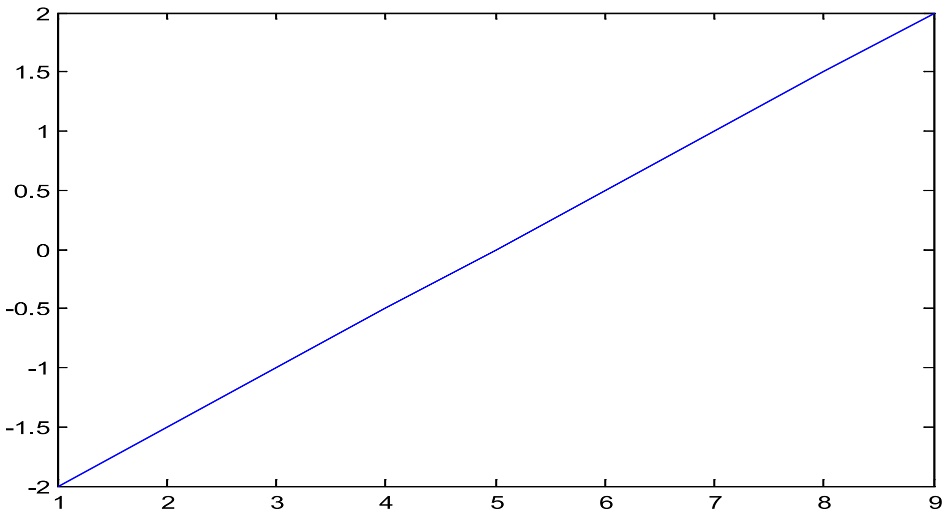
A. The plotting functions

There are standard function that used to draw in MATLAB:

1. **plot( )** : this function plot linear line between two vectors that must be predefined, this function can be used as follows:

a. plot(r) : plots the columns (vector) of (r) versus the index of each value when Y is a real number:

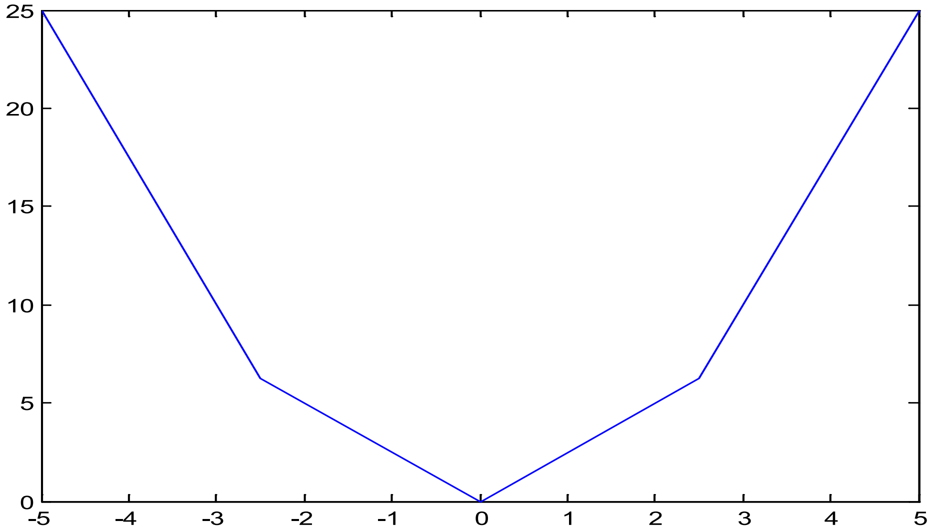
```
X=[ -2 : 0.5: 2]↵
X=
    -2   -1.5   -1   -0.5   0   0.5   1   1.5   2
```



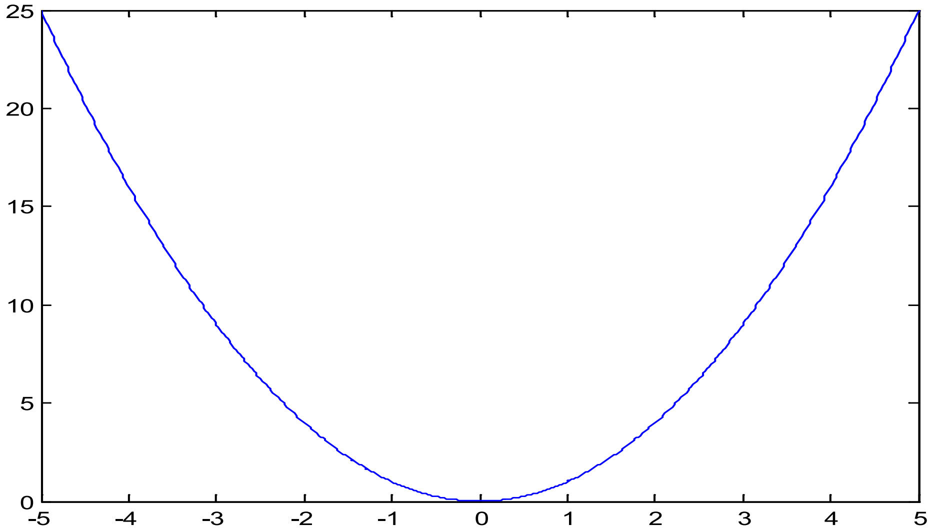
b. plot(X , Y) : plot(X1,Y1,...,Xn,Yn) plots each vector Yn versus vector Xn on the same axes. If one of Yn or Xn is a matrix and the other is a vector, plots the vector versus the matrix row or column with a matching dimension to the vector:

ارسم الدالة ($Y=X^2$) للفترة ($-5 \leq X \leq +5$)

```
x=linspace(-5,5,5); ↵
>> y=x.^2; ↵
>> plot(x,y) ↵
```



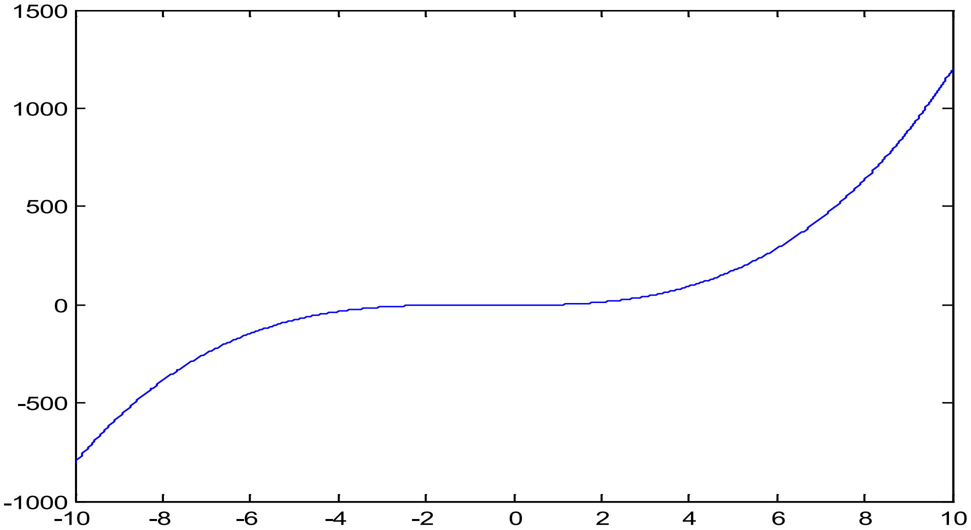

```
x=linspace(-5,5,500); ↵
>> y=x.^2; ↵
>> plot(x,y) ↵
```



نلاحظ انه كلما زادت عدد القيم الواقعة ضمن الفترة المحددة زادت دقت الرسم و اقترب من السلوك النموذجي للدالة .

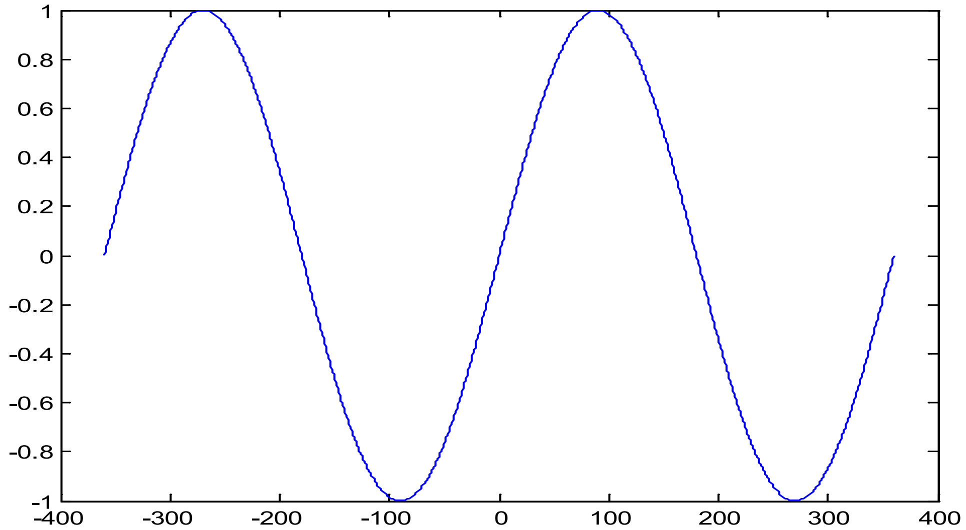
ارسم الدالة $(Y=X^3 + 2X^2 - 3)$ للفترة $(-10 \le X \le +10)$

```
>>X=linspace(-10,10,1000); ↵
>> Y=(X.^3)+(2*(X.^2)) - 3; ↵
>> plot(X,Y) ↵
```



ارسم الدالة $(Y=\sin(X))$ للفترة $(-360^o \le X \le +360^o)$

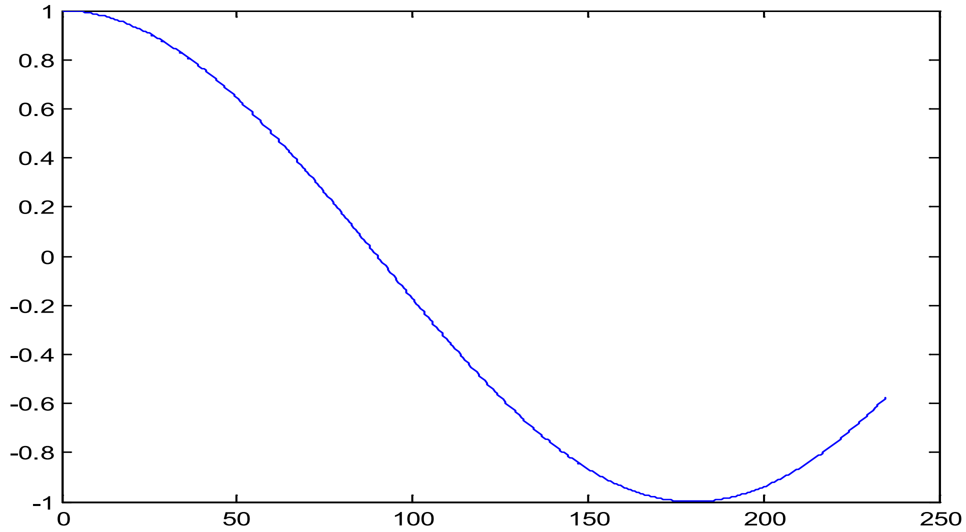
```
>>X=linspace(-360,+360,1000); ↵
>> Y=sin(X.*pi/180); ↵
>> plot(X,Y) ↵
```



يجب ضرب قيم الزوايا داخل الدالة المثلثية (sin , cos , tan ….) في (pi/180) للتحويل من نظام الدرجات (degree ^{o}) الى نظام الزوايا نصف القطرية (radian) كون الماتلاب اصدار (6.1) لايتعرف على الزوايا بنظام الدرجات.

ارسم الدالة (Y=cos (X)) للفترة ($0 \le X \le 235$)

```
>>X=linspace(0, 235,500); ↵
>> Y=cos(X.* pi/180); ↵
>> plot(X,Y) ↵
```



يجب ضرب قيم الزوايا داخل الدالة المثلثية (sin , cos , tan ....) في (pi/180) للتحويل من نظام الدرجات (degree ^{o}) الى نظام الزوايا نصف القطرية (radian) كون الماتلاب اصدار (6.1) لايتعرف على الزوايا بنظام الدرجات.

B. The plot specifications : the following signs can be used to discriminate the plotted data

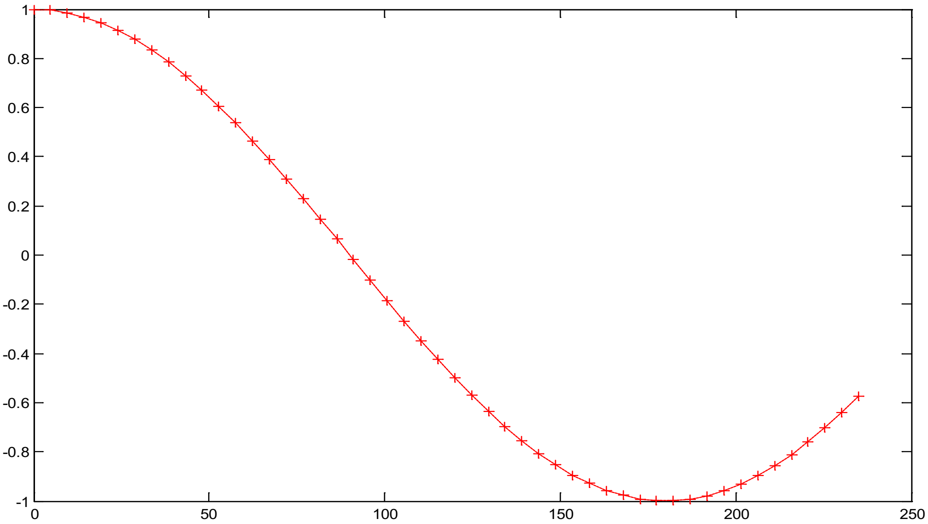
Line Color of line or data point		Data point		Line Type connected between points	
Color	**Meaning**	**Shape**	**Meaning**	**Shape**	**Meaning**
B	Blue ازرق	**.**	Point	**-**	solid
G	Green اخضر	**o**	Circle	**:**	dotted
R	Red احمر	**x**	x-mark	**-.**	dashdot
C	Cyan ازرق سماوي	+	Plus	**--**	dashed
M	Magenta ارجواني	*	Star	**(none)**	no line
Y	Yellow اصفر	**s**	Square		
K	Black اسود	**d**	Diamond		
W	White ابيض	**v**	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	Pentagram		
		h	Hexagram		

The standard form of the plot specifications is as follows:

plot(X ,Y ,'Line color   Data point   Line type')

ارسم الدالة (Y=cos (X)) للفترة (0 ≤ X ≤ 235) على ان تكون نقاط البيانات بشكل علامة زائد و الخط الواصل بينها احمر متقطع

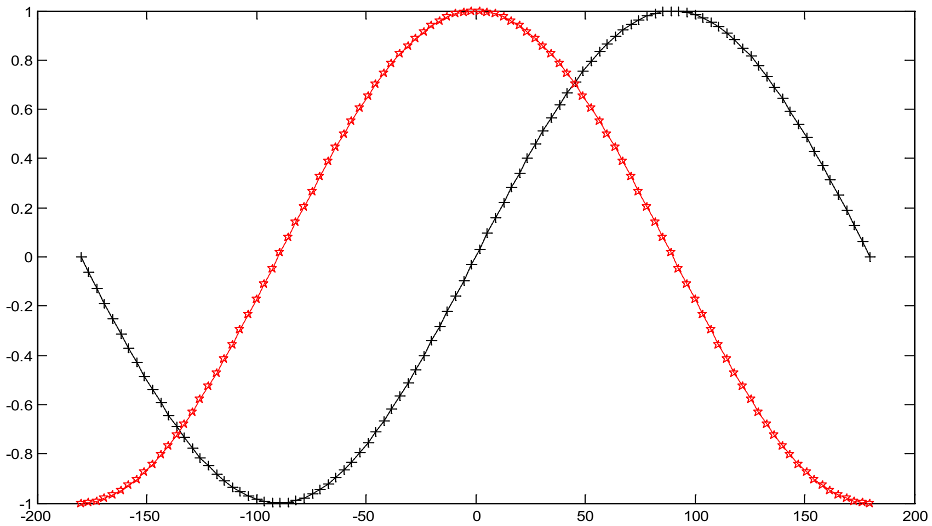
```
>>X=linspace(0, 235,50); ↵
>> Y=cos(X.* pi/180); ↵
>> plot(X,Y, 'k + --) ↵
```



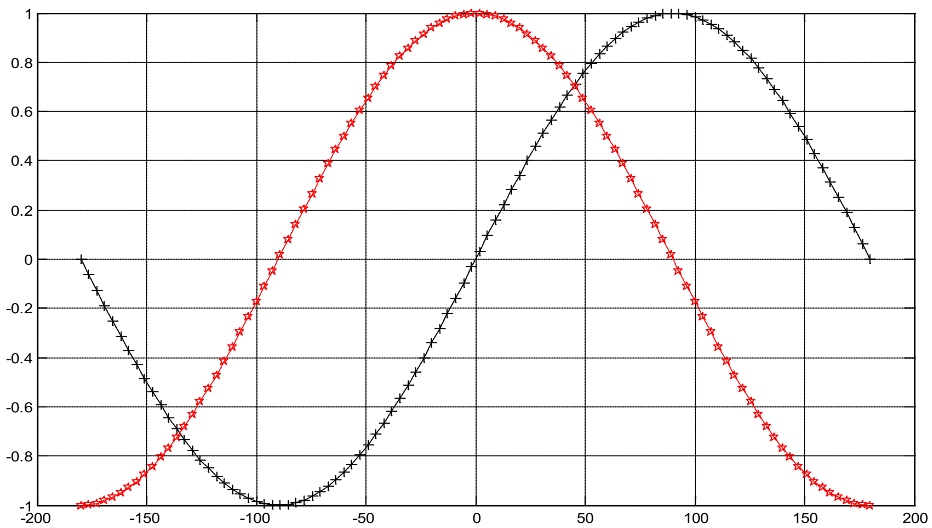
ارسم الدوال التالية :

$$(-180 \le X \le +180) \begin{cases} Y = \sin(X) \\ Z = \cos(X) \end{cases}$$

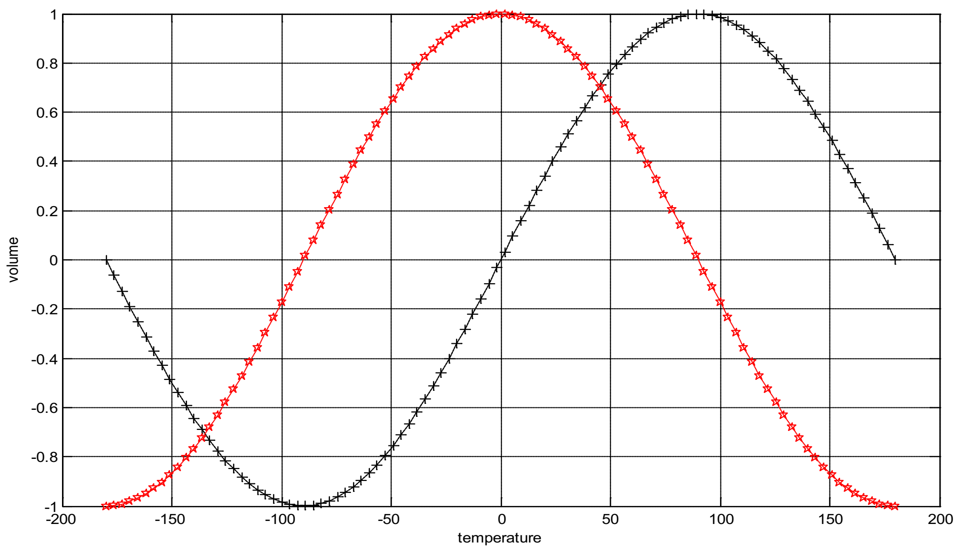
```
>>X=linspace(-180,180,1000); ↵
>> Y=sin(X.* pi/180); ↵
>> Z=cos(X.* pi/180); ↵
>> plot(X,Y, 'k + --' , X,Z, 'r p :') ↵
```



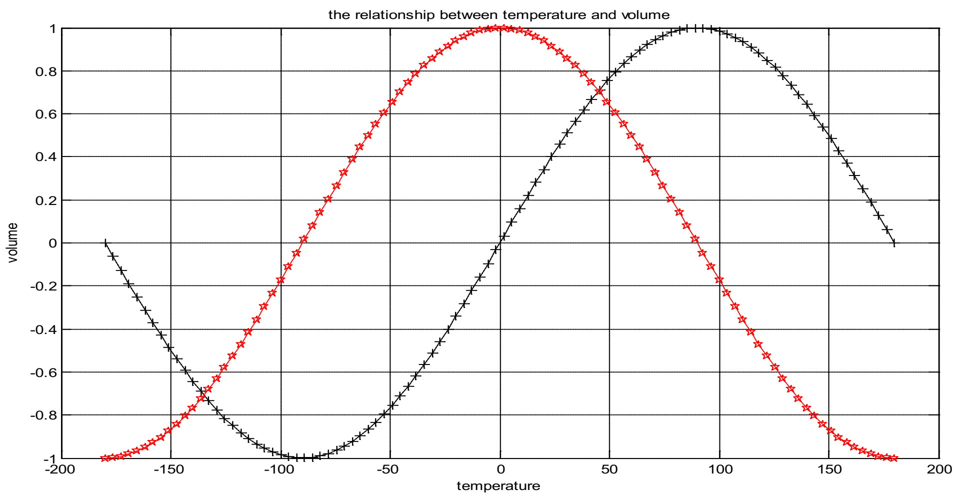
>> grid↵



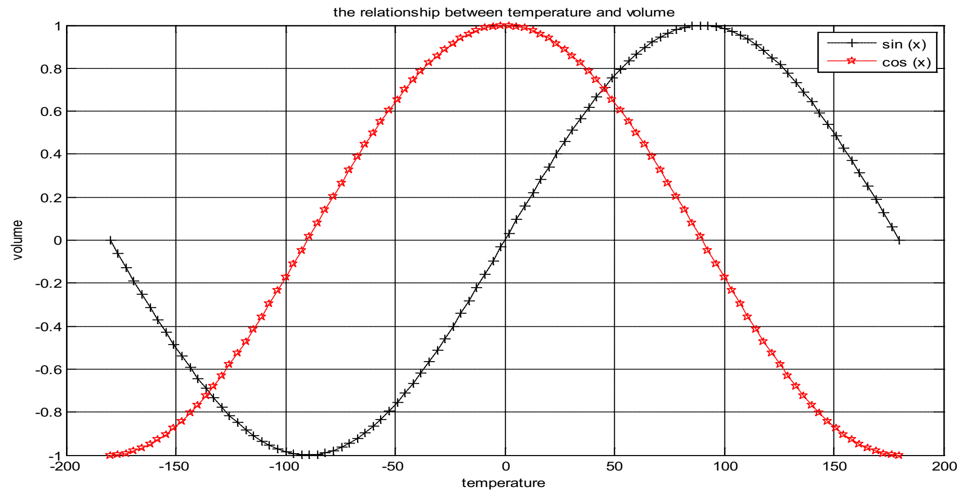
>> xlabel('temperature'), ylabel('volume') ↵



>>title('the relationship between temperature and volume') ↵



>>legend('sin (x)','cos (x)') ↵

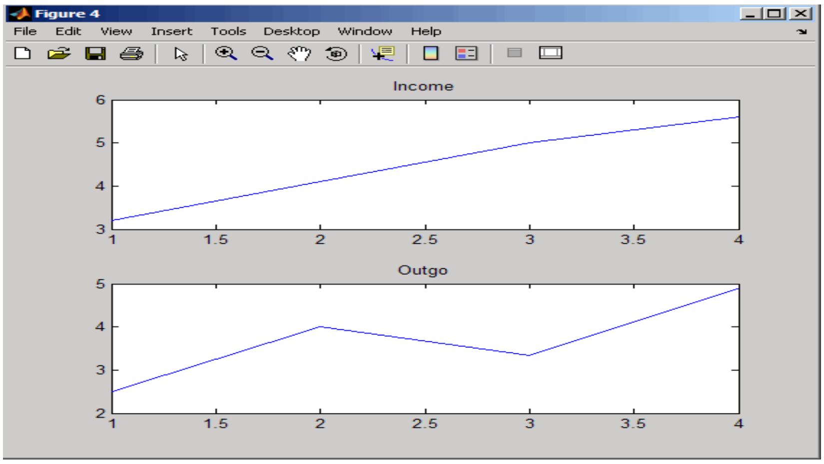


8. Forming a Plot in MATLAB

The following function form the final view the plot

1-

2- xlabel('string'), ylabel('string'), zlabel('string'): label x-, y-, and z-axes.
xlabel('temperature'), ylabel('volume')



9. Logical Comparison and Combining Operators

The logical comparison operator returns false (0) if the logical comparison does not match or true if it exists (1 or nonzero number):

Logical Operator	Meaning
<	less than
< =	less or equal than

>	greater than
> =	greater or equal than
= = (differ than = assignment)	equal to
~ =	not equal
||	or operator
&&	and operator
~	not operator

Let x=2 and y=3 then

= =			
>>x = = y ans = 0	>>x ~ = y&& y > x ans = 1	>>x < = y || 7 ans = 1	>>x > = y ans = 0

10.Control Flow Statements

There are several statements that control the flow of executing program, these statements are:

1- For Loop Statement

For statement is used to repeat executing certain statement(s) for define number of times.

for x=initval:stepval:endval, statements, end

Positive Increment	Without Increment	Negative Increment	Double For Loop
>>x = 2; >>for lcv=1:2:10, y = x^lcv, end y=1 y=8 y=32 y=128 y=512	>>x = 2; >>for lcv=1:4 y = x/lcv end y=2 y=1 y=0.667 y=0.5	>>x = 0; >>for lcv=10:-1:6 y = x +lcv end y=10 y=19 y=27 y=34 y=40	>>for r=1:6 for c=1:4 H(r,c) = (r+c-1) end end >>H

2- While Loop statement

While statement is used to repeat executing certain statement(s) for undefined number of times until the condition becomes false.

while condition, statements, end

>>n = 0; >>a = 1e9; >>while 2^n<a n = n + 1; end, n n = 30	>>x = 10; >>while x>=0.5 x= x/2 end y=5 y=2.5 y=1.25 y=0.625	>>tank = 0; >>gallon = 4.5; >>times = 0; >>while tank <= 200 tank = tank + gallon; times = times + 1; end *tank =202.5 , times = 45*

3- If condition statement

Execute statement(s) if condition is true, otherwise it executes the else statement(s).

if condition, statements, end

OR

if condition, statements
else, statements, end

OR

if condition, statements
elseif condition, statements
elseif condition, statements
:
else, statements, end

>>X = input('input the value of X = '); >>if X>50 'Success' end >>if X<50 'Failed' end	>>X = input('input the value of X = '); >>if X>49 'Success' else 'Failed' end
>>Passed = 0; >>PassedScore = 0; >>Score = round(rand([1 50])*99); >>for lcv=1:50 if Score(lcv)>=50 Passed = Passed + 1; PassedScore = PassedScore + Score(lcv); end end >>Passed >>MeanPassedScore =… PassedScore/Passed	>>Passed = 0; Failed = 0; Fifty = 0; >>PassedScore = 0;FailedScore = 0; >>Score = round(rand([1 50])*99); >>for lcv=1:50 if Score(lcv)>50 Passed = Passed + 1; PassedScore = PassedScore+ Score(lcv); elseif Score(lcv) == 50 Fifty = Fifty + 1; else Failed = Failed + 1; FailedScore = FailedScore+ Score(lcv); end end >>Passed, Failed, Fifty >>MeanPassedScore =PassedScore/Passed >>MeanFailedScore = FailedScore/Failed

4- Break Statement:

Break terminates the execution of a *for* or *while* loop. Statements in the loop that appear after the break statement are not executed.In nested loops, break exits only from the loop in which it occurs. Control passes to the statement that follows the end of that loop.

for u=1:100 x=input('the value of x:'); if x==0, break end y = 3*x^5 –10* x^3 +107* x end	for u=1:10 for v=1:10 if u+v>10 y = (20*u)/(v^2) disp(v); else break end

	end end
x = 5; while x>1 x = x – 0.5; y = sin(x); if x==3, break end end	

5- Continue statement: Pass control to next iteration of for or while loop in which it appears, skipping any remaining statements in the body of the loop.

for u=1:10 x=input('the value of x:'); if (x/2)~=fix(x/2) , continue end y = 3*x^5 –10* x^3 +107* x end	for u=1:10 for v=1:10 if u+v<10 y = (20*u)/(v^2) disp(v); else continue end end end
x = 5; while x<500 if x/5==fix(x/5), continue end y = sin(x);x = x + 1;` end	

6- Switch-End:

The switch selection structure provides an alternative to using the if, elseif, and else commands. The advantage of the switch structure is that in some situations, it yields code that is morereadable.

```
switch expression
        case test expression 1
        commands
        case{test expression 2, test expression 3}
        commands
        .
        .
        .
        otherwise
        commands
end
```

if choice == 1 x = -pi:0.01:pi; elseif choice == 2 x = -pi:0.01:pi; plot(x, sin(x)); end	switch choice case 1 x = -pi:0.01:pi; case 2 x = -pi:0.01:pi; plot(x, sin(x)); end
Interval = input('Input the interval value:'); switch interval < 1 case 1 Result = interval/10; case 0 Result = 0.1; End	x = input('input the distance in meter:'); units = input('input the unit to convert to: '); switch units case {'inch','in'} y = x*0.0254; case{'feet','ft'} y = x*0.3048; case{'meter','m'} y = x; case{'centimeter','cm'} y = x/100; case{'millimeter','mm'} y = x/1000; otherwise disp(['Unknown units: ' units]) y = NaN; end