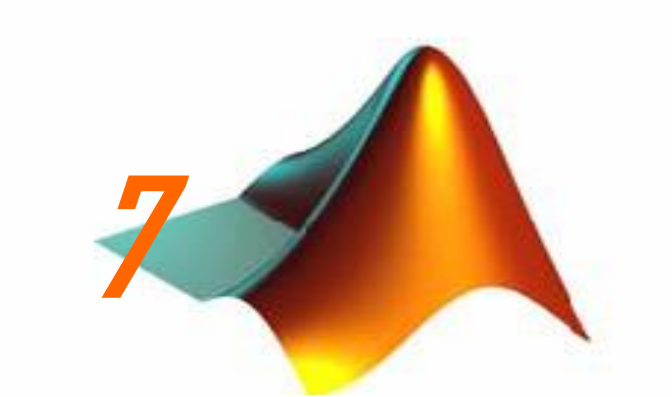
Introduction to MATLAB



BY

TUSHAR KANTI DAS

ASSISTANT PROFESSOR
DEPARTMENT OF MATHEMATICS
J. K. COLLEGE, PURULIA
2014

Outline

- **Introduction**
  - ✓ What is MATLAB?
  - ✓ Matrix and it's application in different areas
- **Operators**
  - ✓ Arithmetic Matrix-Operators
  - ✓ Arithmetic Array- Operators

  and it's application in MATLAB
- **Expressions**

What is MATLAB?

- Acronym stands for **MATRIX LABORATORY.**
- It is an interactive system whose basic data element is a **matrix** that does not require dimensioning.
- It is a technical computing environment for high performance numeric computation and visualization
- Problems and solutions are written mathematically without traditional programming.

Use of MATLAB

- In Mathematics for solution of linear algebra and other advanced courses
- In industry for solving practical engineering and mathematical problems and for research
- all kinds of numerical analysis

MATLAB Tool Boxes

MATLAB features a family of add-on-application-specific solutions called *toolboxes*. These toolboxes are collections of functions written for special applications. It allow to *learn* and *apply* specialized technology:

- Signal Processing
- Control system
- Neural Networks
- Fuzzy Logic
- Power Systems
- Simulink
- Wavelets
- Image Processing
- and many more

Matrix and Vector

*M**atrix** :* a two dimensional ordered array of numbers, variables or functions, given a single name and manipulated as a group.

Vector: an one dimensional ordered array of numbers, variables or functions, given a single name and manipulated as a group.

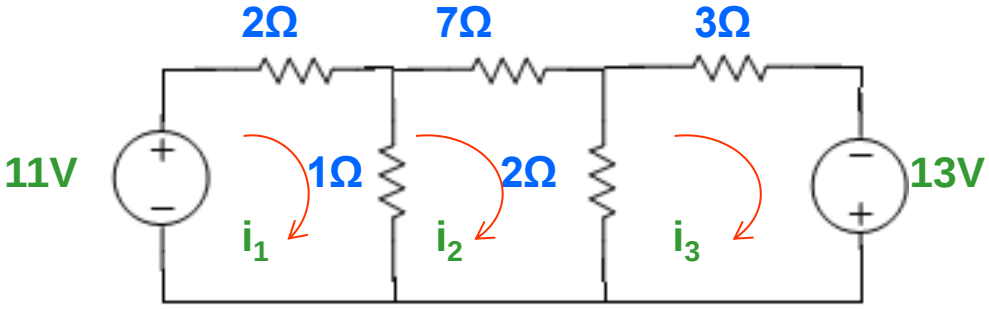
$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Matrix

$$A = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} \qquad B = \begin{bmatrix} a_{11} & a_{12} \end{bmatrix}$$

Vector

Electrical networks



$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \\ \vec{i}_3 \end{bmatrix} = \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vec{v}_3 \end{bmatrix}$$

$$AI = V$$

$$\begin{bmatrix} 3 & -1 & 0 \\ -1 & 10 & -2 \\ 0 & -2 & 5 \end{bmatrix} \begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \\ \vec{i}_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 0 \\ 13 \end{bmatrix}$$

more…

➢ Creat the Matrix A

```
>> A= [3 -1 0;1 10 -2;0 -2 5]
A =
     3  -1   0
     1  10  -2
     0   -2   5
```

(Once we have entered the matrix, it is automatically remembered in the MATLAB workspace. We can refer to it simply as A.)

➢ Creat the vector V

```
>>V=[11;0;13]
 V=
      11
       0
      13
```

➢ Generating Matrices

MATLAB provides four functions that generate basic matrices.

zeros	All zeros
ones	All ones
rand	Uniformly distributed random elements
randn	Normally distributed random elements

Examples:

```
>>Z = zeros(2,4)
Z =
     0     0     0     0
     0     0     0     0
```

```
>>F = 5*ones(3,3)
F =
     5     5     5
     5     5     5
     5     5     5

>>N = fix(10*rand(1,10))
N=
     9     2     6     4     8     7     4     0     8     4

>>R = randn(4,4)
R=
    0.6353    0.0860   -0.3210   -1.2316
   -0.6014   -2.0046    1.2366    1.0556
    0.5512   -0.4931   -0.6313   -0.1132
   -1.0998    0.4620   -2.3252    0.3792
```

Arithmetic Matrix-Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

These are executed by MATLAB according to the rules of linear algebra

➢ The summation of a matrix:

```
>> A= [3 -1 0;1 10 -2;0 -2 5]
  A=
       3   -1    0
       1   10   -2
       0   -2    5
  >>sum(A)
  ans =
       4   7   3
```

It gives sum of each column .

➢ The transpose of a matrix:

It changes rows of the original matrix to columns of the new matrix

```
>> A'
  ans =
       3    1    0
      -1   10   -2
       0   -2    5
```

➢ Diagonal of a matrix:

```
>> diag(A)
ans =
     3
    10
     5
```

➢ And the sum of the diagonal:

```
>> sum(diag(A))
ans =
    18
```

➢To calculate the sum of the antidiagonal:

$$A = \begin{pmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{pmatrix}$$

The element in row i and column j of A is denoted by A(i,j).

```
>> A(1,4)+A(2,3)+A(3,2)+A(4,1) ↵

 ans =

       34
```

➢Linear Algebra

```
>>A+A’ ↵

ans =

    32    8   11   17
     8   20   17   23
    11   17   14   26
    17   23   26    2

>>d = det(A) ↵
d =
     0
```

➢ Concatenation

Concatenation is the process of joining small matrices to make bigger ones:

```
>> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1];
```

```
>>B = [A  A+32; A+48  A+16]
```

```
B =

    16     3     2    13    48    35    34    45
     5    10    11     8    37    42    43    40
     9     6     7    12    41    38    39    44
     4    15    14     1    36    47    46    33
    64    51    50    61    32    19    18    29
    53    58    59    56    21    26    27    24
    57    54    55    60    25    22    23    28
    52    63    62    49    20    31    30    17
```

Deleting Rows and Columns

Any row(s) or column(s) of a matrix can be deleted by setting the row or column to a null vector

```
>>X = A;
```

To delete the second column of X

```
>> X(:,2) = [ ]
```

Null vector

```
X =
    16     2    13
     5    11     8
     9     7    12
     4    14     1
```

To delete the second row of X

```
>> x(2,:) = [ ]
x =
    16     2    13
     9     7    12
     4    14     1
```

➢ Extract submatrix of a matrix

```
>>q=[2 3 6 0 5;0 0 20 -4 3;1 2 3 9 8;2 -5 5 -5 6;5 10 15 20 25]

q =

    2    3    6    0    5
    0    0   20   -4    3
    1    2    3    9    8
    2   -5    5   -5    6
    5   10   15   20   25

>> v=[1 4 5]

v =

    1    4    5
```

```
>>q(v,:)
ans =

    2    3    6    0    5
    2   -5    5   -5    6
    5   10   15   20   25
>>q(:,v)
ans =

    2    0    5
    0   -4    3
    1    9    8
    2   -5    6
    5   20   25
```

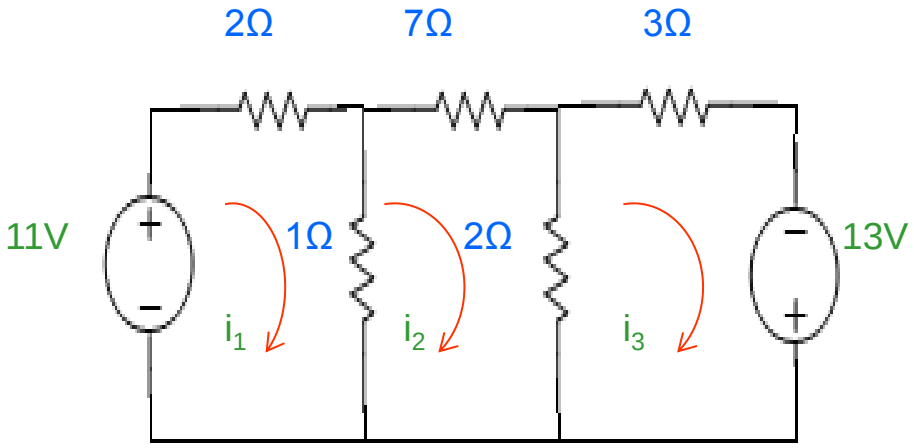
Inverse of a matrix

$AA^{-1} = I$

$A^{-1} = \text{adj } A / \det A$

```
>> A=[5 -3 2;-3 8 4;2 4 -9]
A =
     5    -3     2
    -3     8     4
     2     4    -9

>> B = inv(A)

B =
    0.2005    0.0433    0.0638
    0.0433    0.1116    0.0592
    0.0638    0.0592   -0.0706
```

Determine current in each branch of the network



$$\begin{bmatrix} 3 & -1 & 0 \\ -1 & 10 & -2 \\ 0 & -2 & 5 \end{bmatrix} \begin{bmatrix} \vec{i}_1 \\ \vec{i}_2 \\ \vec{i}_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 0 \\ 13 \end{bmatrix}$$

$$A\vec{I} = \vec{V}$$

$$\vec{I} = A^{-1}\vec{V}$$

```
>>A=[3 -1 0;-1 10 -2;0 -2 5];
>>V=[11;0;13]
>>I=inv(A)*V
I =

    4.0000
    1.0000
    3.0000
```

Solving a linear system

$$5x-3y+2z = 10$$
$$-3x+8y+4z = 20$$
$$2x+4y-9z = 9$$

$$AX = B$$

$$X=A^{-1} B$$

$$\begin{bmatrix} 5 & -3 & 2 \\ -3 & 8 & 4 \\ 2 & 4 & -9 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \\ 9 \end{bmatrix}$$

$$A = \begin{bmatrix} 5 & -3 & 2 \\ -3 & 8 & 4 \\ 2 & 4 & -9 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad B = \begin{bmatrix} 10 \\ 20 \\ 9 \end{bmatrix}$$

```
>>A=[5 -3 2;-3 8 4;2 4 -9]

>>B=[10;20;9]

>>X=inv(A) * B
X =
   3.4442
   3.1982
   1.1868
>>C=A*X   % check the solution
C=
   10.0000
   20.0000
    9.0000
```

Scalar Expansion

```
>> B = A - 2.5 ↵
B =
    0.5000   -3.5000   -2.5000
   -3.5000    7.5000   -4.5000
   -2.5000   -4.5000    2.5000

>> B(1:2,2:3)=0 ↵
B =
    0.5000         0         0
   -3.5000         0         0
   -2.5000   -4.5000    2.5000
```

➢Eigenvalues

The eigenvalues of an n x n matrix A are the roots of the characteristic equation

$$|\lambda I - A| = 0$$

$\lambda^3-18\lambda^2+90\lambda-133=0$ where λ is a scalar.

```
>>
>> A=[3 -1 0;-1 10 -2;0 -2 5];
>> eig(A)
ans =
    2.8133
    4.3709
   10.8157
```

➢Arithmatic Array- Operators

Element –by- element multiplication, division and exponentiation between two matrices or vectors of the *same size* are done by preceding the corresponding arithmetic operators by a period (.)

Examples:

$u=[u_1\ u_2\ u_3\ldots]$ and $v=[v_1\ v_2\ v_3\ldots]$

u.*v produces $[u_1v_1\ u_2v_2\ u_3v_3\ldots.]$

u./v produces $[u_1/v_1\ u_2/v_2\ u_3/v_3\ldots]$

u.^v produces $[u_1^{v1}, u_2^{v2}, u_3^{v3}\ldots]$

+	Addition
-	Subtraction
.*	Element-by-element multiplication
./	Element-by-element division
.\	Element-by-element left division
.^	Element-by-element power
.'	Unconjugated array transpose

Example1:

$$A.*A=\begin{bmatrix}3 & -1 & 0\\ -1 & 10 & -2\\ 0 & -2 & 5\end{bmatrix}.*\begin{bmatrix}3 & -1 & 0\\ -1 & 10 & -2\\ 0 & -2 & 5\end{bmatrix}$$

```
>> A.*A
ans =
   9    1    0
   1  100    4
   0    4   25
```

Example2: $y = x^2 - 4x$

```
>>x=[1:8]
x=
    1   2   3   4   5   6   7   8
>>y=x.^2 -4*x
y=
    -3   -4   -3   0   5   12   21   32
```

➢ Three different *vector* multiplications:

```
>> x = [1 2 3];          % (1 x 3 matrix)
>> y = [3 2 1];          % (1 x 3 matrix)
>>disp(x'*y)             %(3 x 1) * (1 x 3) = (3 x 3)
     3    2    1
     6    4    2          column*row
     9    6    3
>> disp(x*y')             %(1 x 3) * (3 x 1) = (1 x 1)
    10                      row*column
>> disp(x.*y)            %(1 x 3) .* (1 x 3) = (1 x 3)
     3    4    3       array multiplication
```

The array dot must also be used if we raise each element to a power

```
>> x = [1 2 3]
x =
     1     2     3
>> x.^3
ans =
     1     8    27
```

➢ MATLAB has many built-in functions for analyzing arrays

❑ mean(A)

```
>>A=[5 9 2 4];
>>mean(A)
ans=
      5
```

❑ max(A)

```
>>A=[5 9 2 4 11 6 7 11 0 1];
>>c=max(A)
c=
    11
```

❑ min(A)

```
>>A=[5 9 2 4];
>>min(A)
ans=
      2
```

❑ sort(A)

```
>>A=[5 9 2 4];
>>sort(A)
ans=
      2   4   5   9
```

➢ The colon operator:

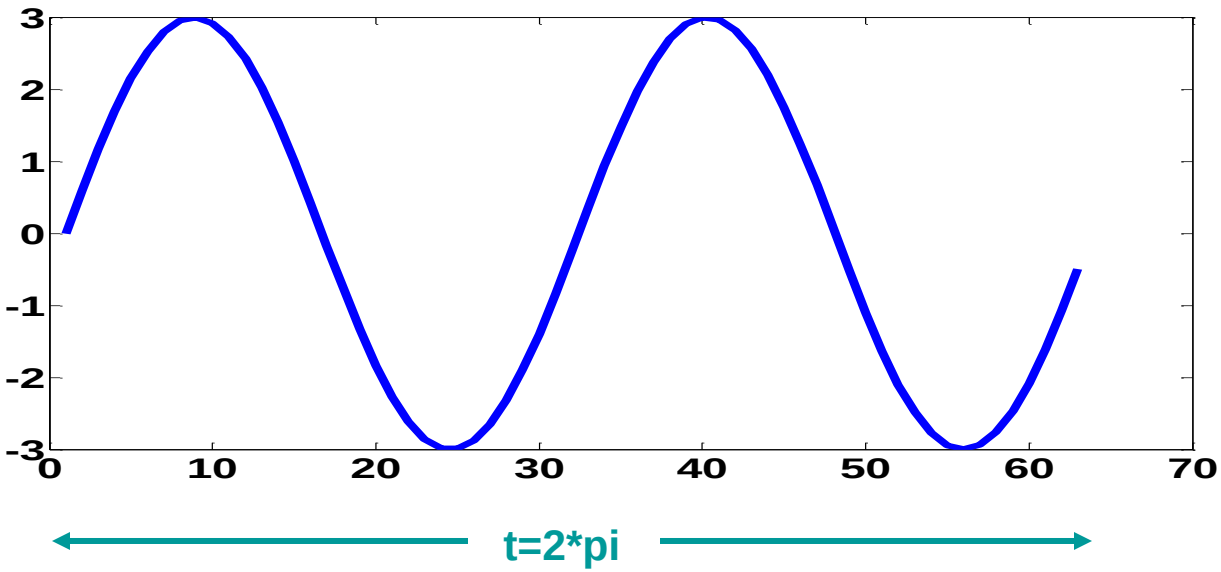
```
>> 1:10
 ans =
       1   2   3   4   5   6   7   8   9   10
 >> 100:-7:50
 ans =
      100   93   86   79   72   65   58   51
```

A(m:n,p) Refers to elements in p^{th} column between rows *m* and *n* of the matrix A.

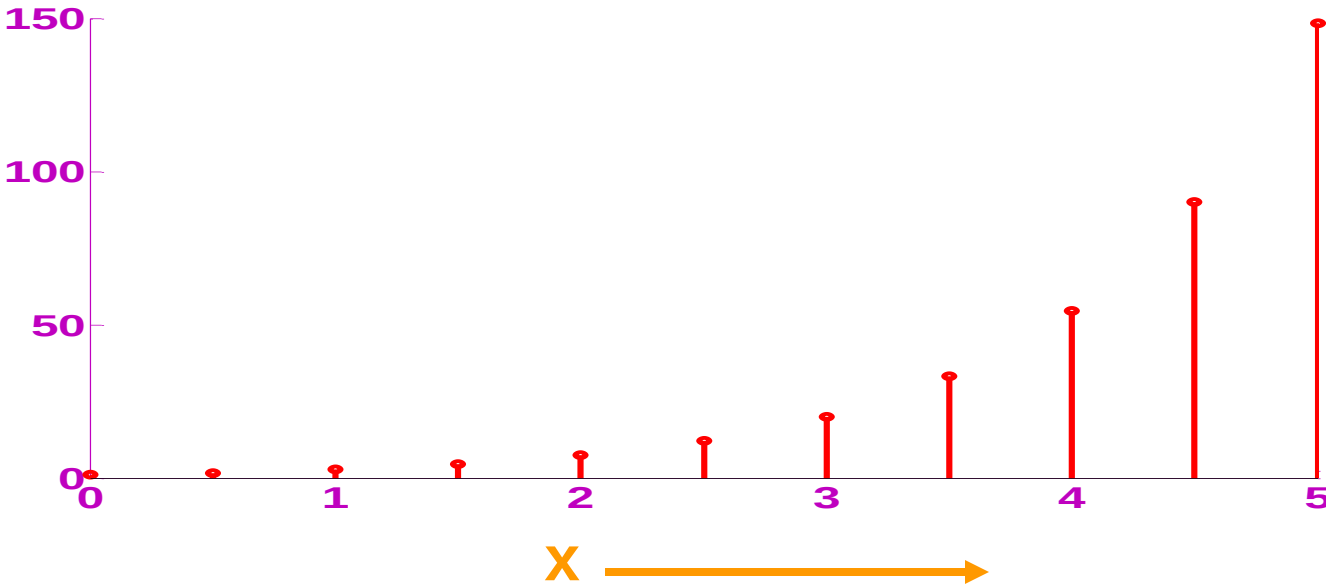
```
>>A=[3 -1 0;-1 10 -2;0 -2 5];
 >>sum(A(1:3,3))
 ans =
         3
```

➢Need of *colon* operator:

V= 3*sin(2*t)



$Y=e^{x}$



Example:

```
>> t=0:0.5:2*pi;
 >>v=3*sin(2*t)
  v =
       0   2.5244   2.7279   0.4234  -2.2704  -2.8768  -0.8382
           1.9710 2.9681   1.2364  -1.6321  -3.0000  -1.6097

>> x=0:0.5:5;
>> y=exp(x)
y =
   1.0000   1.6487   2.7183   4.4817   7.3891  12.1825
  20.0855  33.1155  54.5982  90.0171 148.4132
```

Creating a vector with constant spacing by specifying the first and last terms, and the number of terms:

Variable_name= linspace (xi , xf , n)

Example:

```
>>X=linspace(30,10,11)
X=
  30   28   26   24   22   20   18   16   14   12   10
```

Expressions

- MATLAB provides mathematical expressions.

Examples:

```
>> a = abs(3+4i)
a =
    5
>> b = (1+sqrt(5))/2
b =
   1.6180
```

- The building blocks of expressions are:
  - Variables
  - Numbers
  - Operators
  - Functions

Variable names consist of a letter, followed by any number of letters, digits, or underscores.

For example, a = 25; A=30; var_as=35;

If the variable already exists, MATLAB changes it's contents.

MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters.

➢Numbers

- MATLAB uses conventional decimal notation.
- ***Scientific notation*** uses the letter **e** to specify a power-of-ten scale factor.
- ***Imaginary numbers*** use either **i** or **j** as a suffix.

Example:

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

➢ Functions

Example: abs, sqrt, exp, sin etc.

```
>> help elfun
Elementary math functions.
    Trigonometric.
    sin - Sine.
    sinh - Hyperbolic sine.
    asin - Inverse sine.
    asinh - Inverse hyperbolic sine.
    cos - Cosine.
    cosh - Hyperbolic cosine.
    ...........................................................
```

➢Some useful constants

pi 3.14159265…
i Imaginary unit,
j Same as i
Inf Infinity (e.g. x/0;)
NaN Not-a-number (e.g. 0/0 or Inf-Inf)

➢DON'T USE MATLAB FUNCTION NAMES AS VARIABLES!

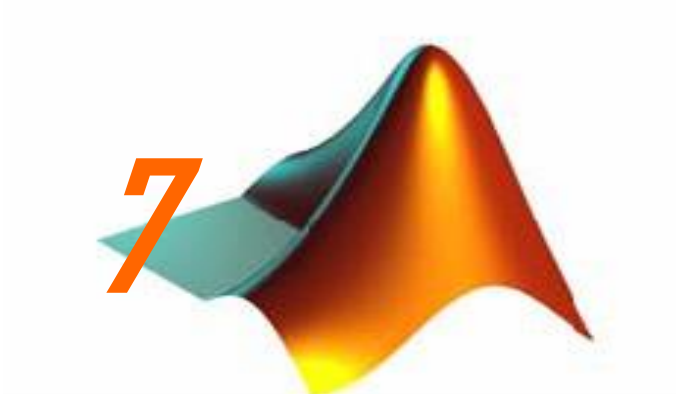
```
>> exp=3 ↵
exp=
        3
>>y=exp(3) ↵
??? Index exceeds matrix dimensions.
```

THANK YOU

dastushar74@gmail.com

Questions?

MATLAB
programming and graphics



BY

TUSHAR KANTI DAS

ASST. PROFESSOR

DEPARTMENT OF MATHEMATICS.

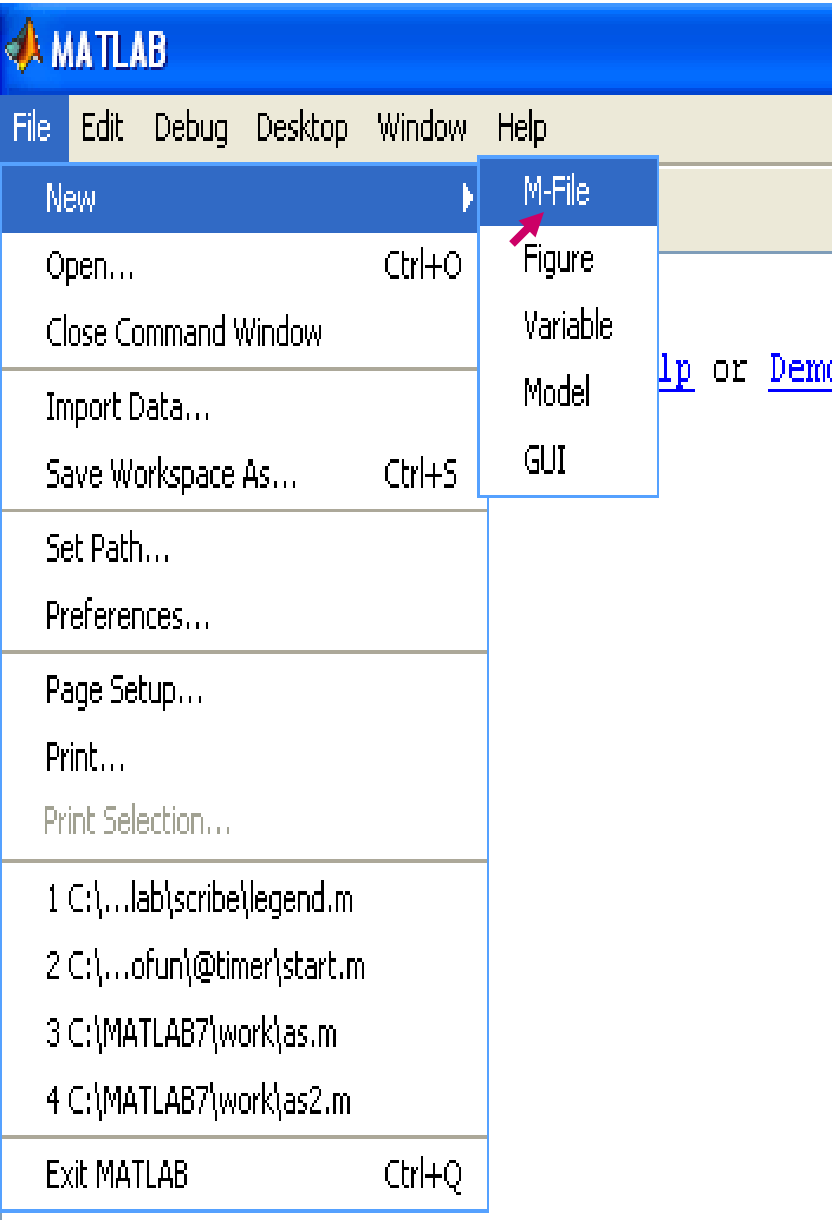
J. K. College, Purulia

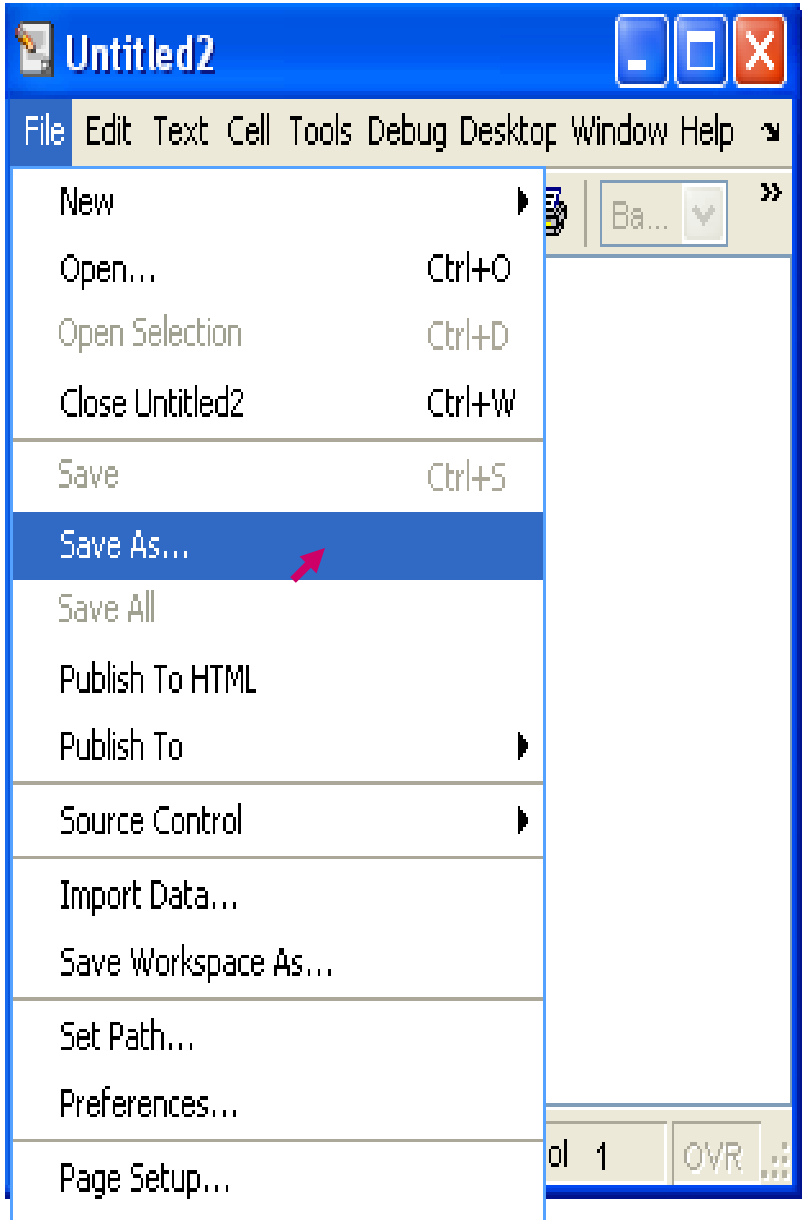
2008

Outline

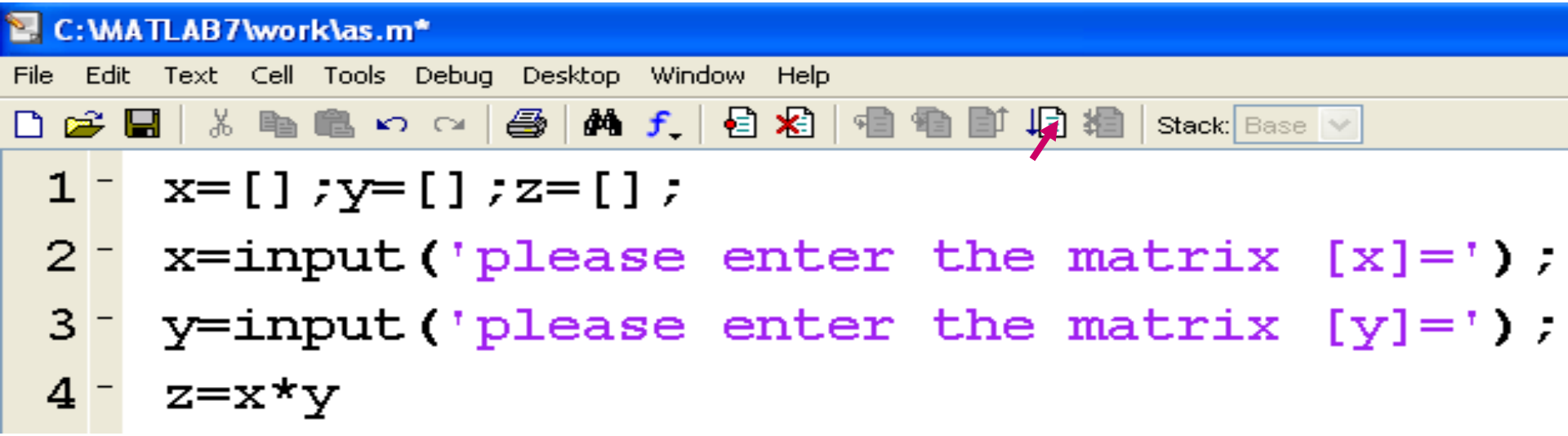
- Opening M-Files
- Structure of MATLAB programming
- Programming
- Concatenation of strings
- Vectorization
- Basic Graphics

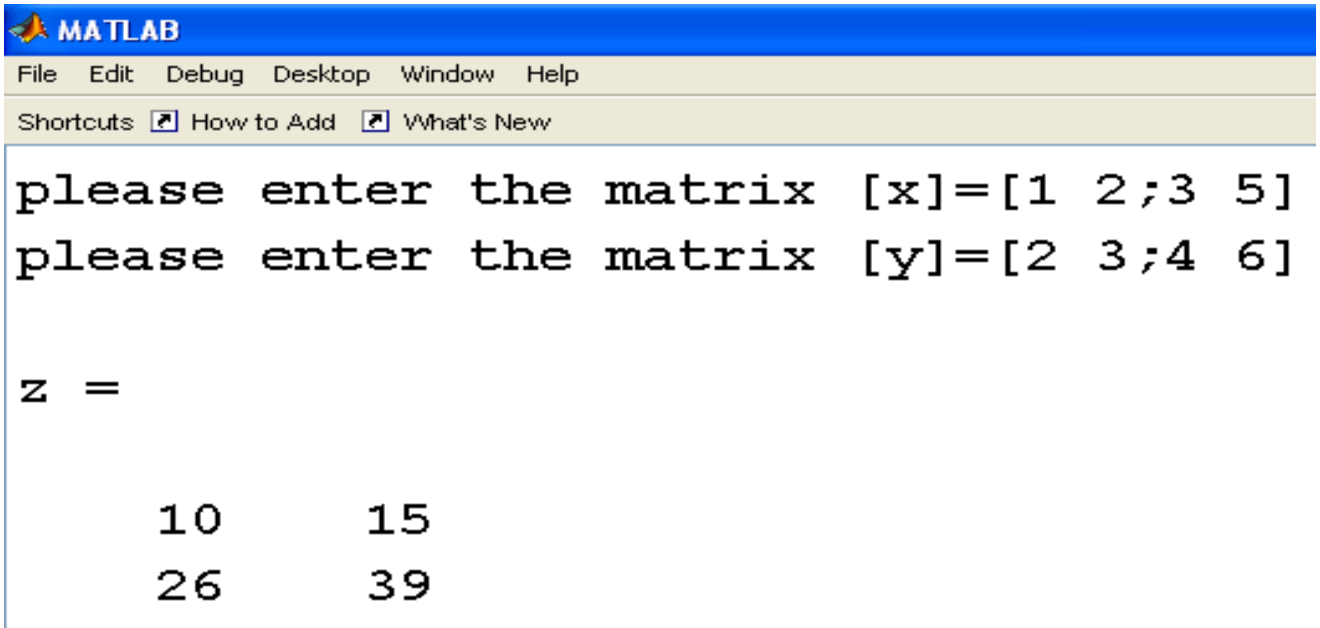
➢Opening M-Files





Structure of MATLAB programming





➢ Programming

MATLAB has several flow control constructs:

- ❑ "if"
- ❑ "switch and case"
- ❑ "for"
- ❑ "while"
- ❑ "continue"
- ❑ "break"
- ❑ "return"

We look at the most important ones!

if

C:\MATLAB7\work\as.m

File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

```
1  x=[];y=[];
2  x=input('please enter the matrix [x]=');
3  y=input('please enter the matrix [y]=');
4  if x > y
5      'greater'
6  elseif x < y
7      'less'
8  elseif x == y
9      'equal'
10 else
11     error('Unexpected situation')
12 end
```

MATLAB

File Edit Debug Desktop Window Help

Shortcuts How to Add What's New

```
please enter the matrix [x]=[1 2;3 5]
please enter the matrix [y]=[2 3;4 6]

ans =

less
```

Calculating worker's pay: if he works more than 40 hours, he is paid 50% extra.

C:\MATLAB7\work\as1.m*

File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

```
1 - t=input('please enter the number of hours worked=');
2 - h=input('please enter the hourly wage in rupees=');
3 - pay=t*h;
4 - if t>40
5 - pay=pay+(t-40)*0.5*h;
6 - end
7 - fprintf('the worker"s pay is Rs. %5.2f',pay)
```

MATLAB

File Edit Debug Desktop Window Help

Shortcuts How to Add What's New

```
please enter the number of hours worked=50
please enter the hourly wage in rupees=50
the worker"s pay is Rs.  2750.00>>
```

➢for

C:\MATLAB7\work\as2.m

File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

```
1 - x=[];
2 - m=input('please enter the no. of rows in matrix [x]=');
3 - n=input('please enter the no. of columns in matrix [x]=');
4 - for i = 1:m
5 -     for j = 1:n
6 -         x(i,j) = 1/(i+j);
7 -     end
8 - end
9 - x
```

MATLAB

File Edit Debug Desktop Window Help

Shortcuts How to Add What's New

```
please enter the no. of rows in matrix [x]=3
please enter the no. of columns in matrix [x]=3

x =

    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
    0.2500    0.2000    0.1667
```

➢ Vectorization

Vectorization of algorithms makes MATLAB programs run faster than for loops.

```
>> x = .01;
 >> for k = 1:1001
     y(k) = log10(x);
     x = x + .01;
    end
>>y
```

A vectorized version of the same code is

```
>> x = .01:.01:10;
>>y = log10(x)
```

➢while

C:\MATLAB7\work\as3.m*

File Edit Text Cell Tools Debug Desktop Window Help

```
1  y1=[];y2=[];
2  x=5;
3   while  x>1
4          y1=[y1,x];
5          x=x-1;
6          y2=[y2,x];
7   end
8   y1
9   y2
```

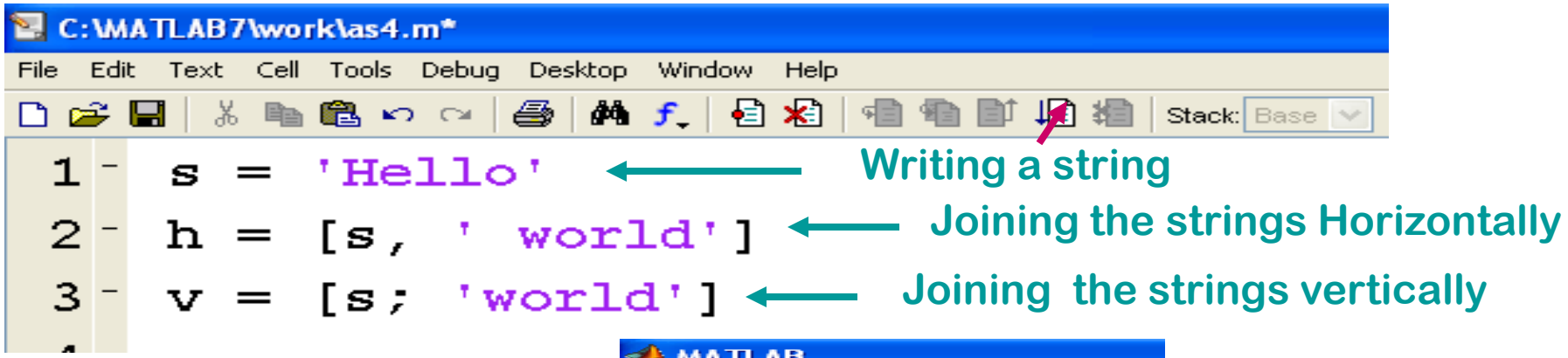
MATLAB

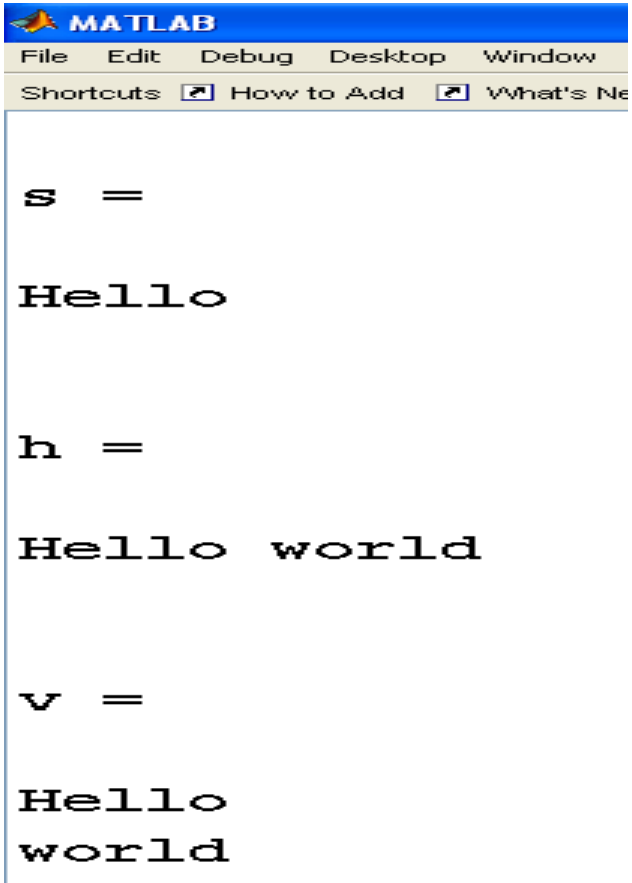
File Edit Debug Desktop Window Help

Shortcuts How to Add What's New

```
y1 =

     5     4     3     2


y2 =

     4     3     2     1
```

➢Concatenation of strings





Basic Graphics

- Creating a Plot
- Axis Labels and Titles
- Multiple Data Sets in One Graph
- Specifying Colors, Line Styles and Markers
- Imaginary and Complex Data
- Adding Plots to an Existing Graph
- Multiple Plots in One Figure
- Saving Figures

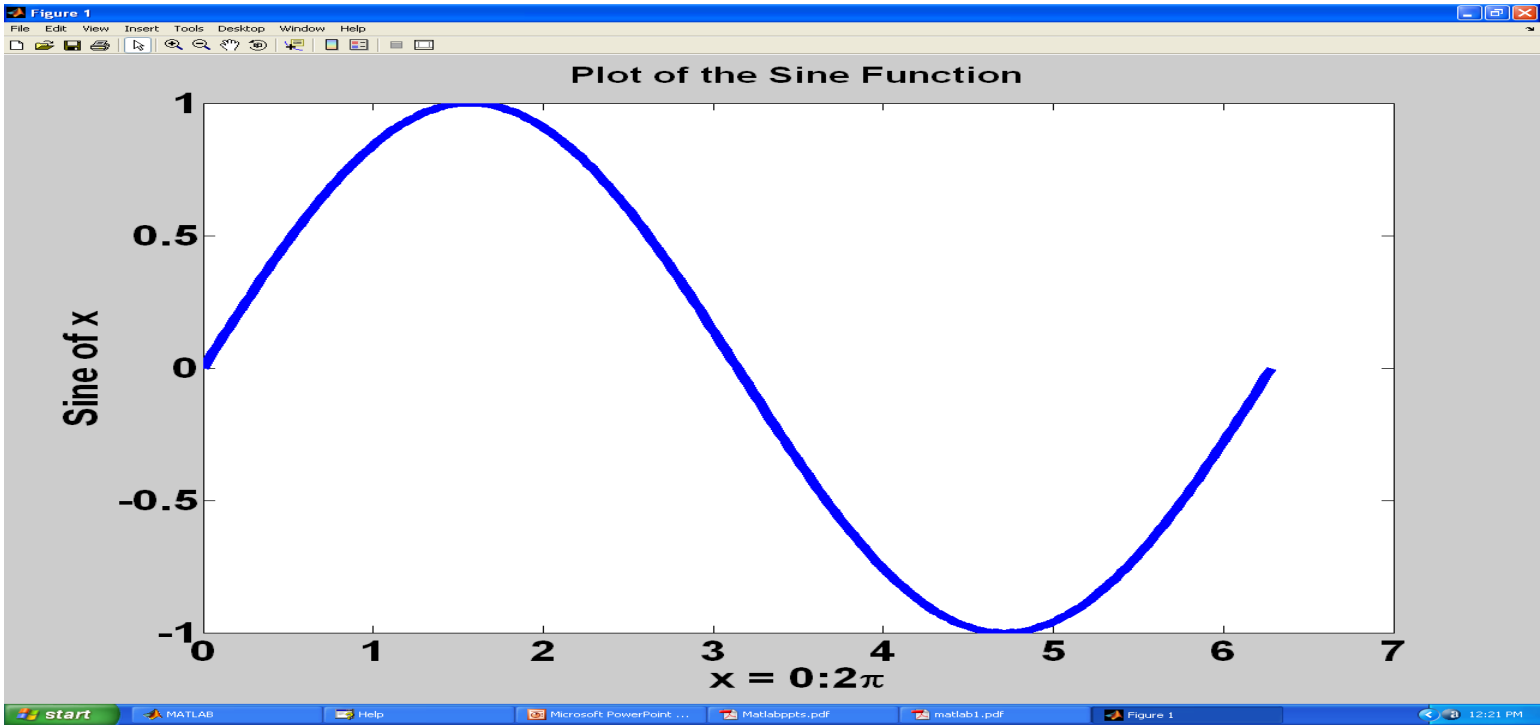
➢ Creating a Plot, Labeling the axes and add a title:

C:\MATLAB7\work\as5.m*

File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

```
1 - x=0:pi/100:2*pi;
2 - y=sin(x);
3 - plot(y)
4 - xlabel('x = 0:2\pi')
5 - ylabel('Sine of x')
6 - title('Plot of the Sine Function','FontSize',12)
```



➢ Multiple Data Sets in One Graph

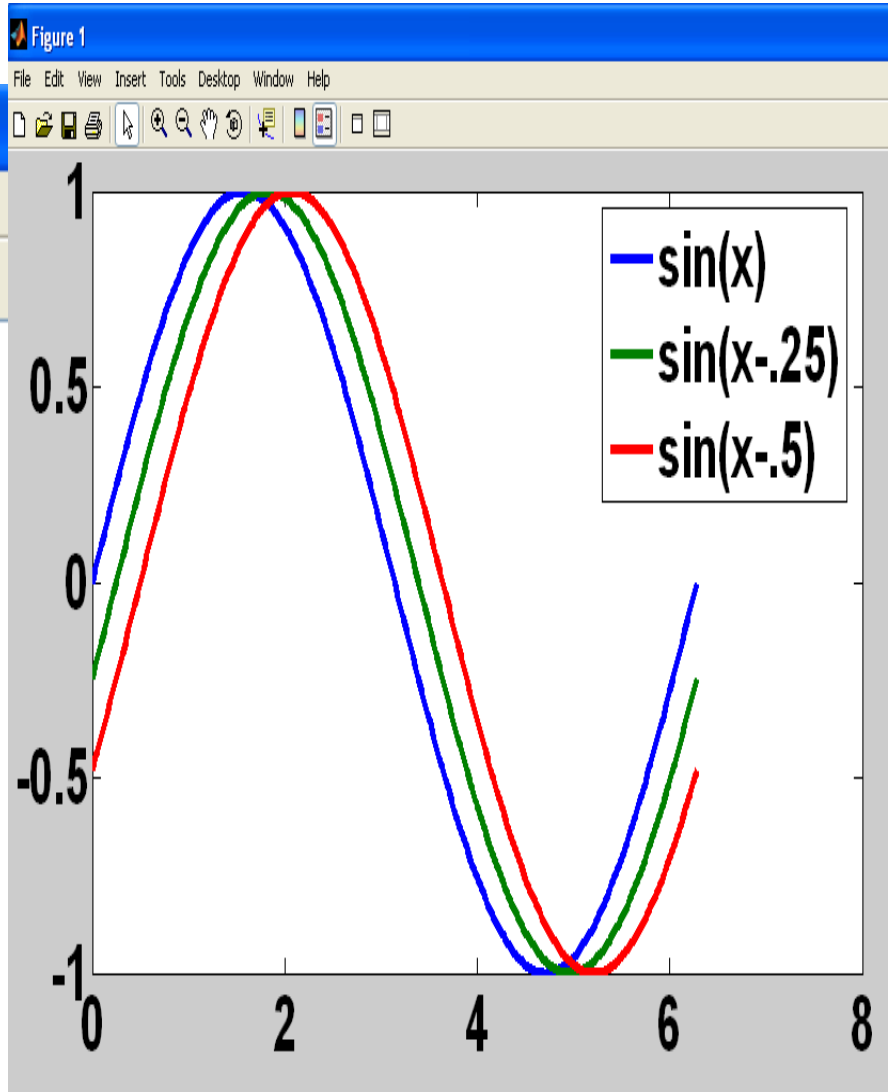
Multiple x-y pair arguments create multiple graphs with a single call to plot.

C:\MATLAB7\work\as5.m

File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

```
1 - x = 0:pi/100:2*pi;
2 - y = sin(x);
3 - y2 = sin(x-.25);
4 - y3 = sin(x-.5);
5 - plot(x,y,x,y2,x,y3)
6 - legend('sin(x)','sin(x-.25)','sin(x-.5)')
```



➢ Specifying Colors, Line Styles and Markers:

```
plot(x,y,'color_style_marker')
```

Specifier	Line Color	Specifier	Marker Style
b	blue	.	point
g	green	o	circle
r	red	x	x-mark
c	cyan	+	Plus
m	magenta	*	Star
y	yellow	s	Square
k	black	d	Diamond
		v	triangle down
Specifier	**Line Style**	^	triangle up
-	solid	<	triangle left
:	dotted	>	triangle right
-.	dash-dot	p	pentagram
--	dashed	h	hexagram

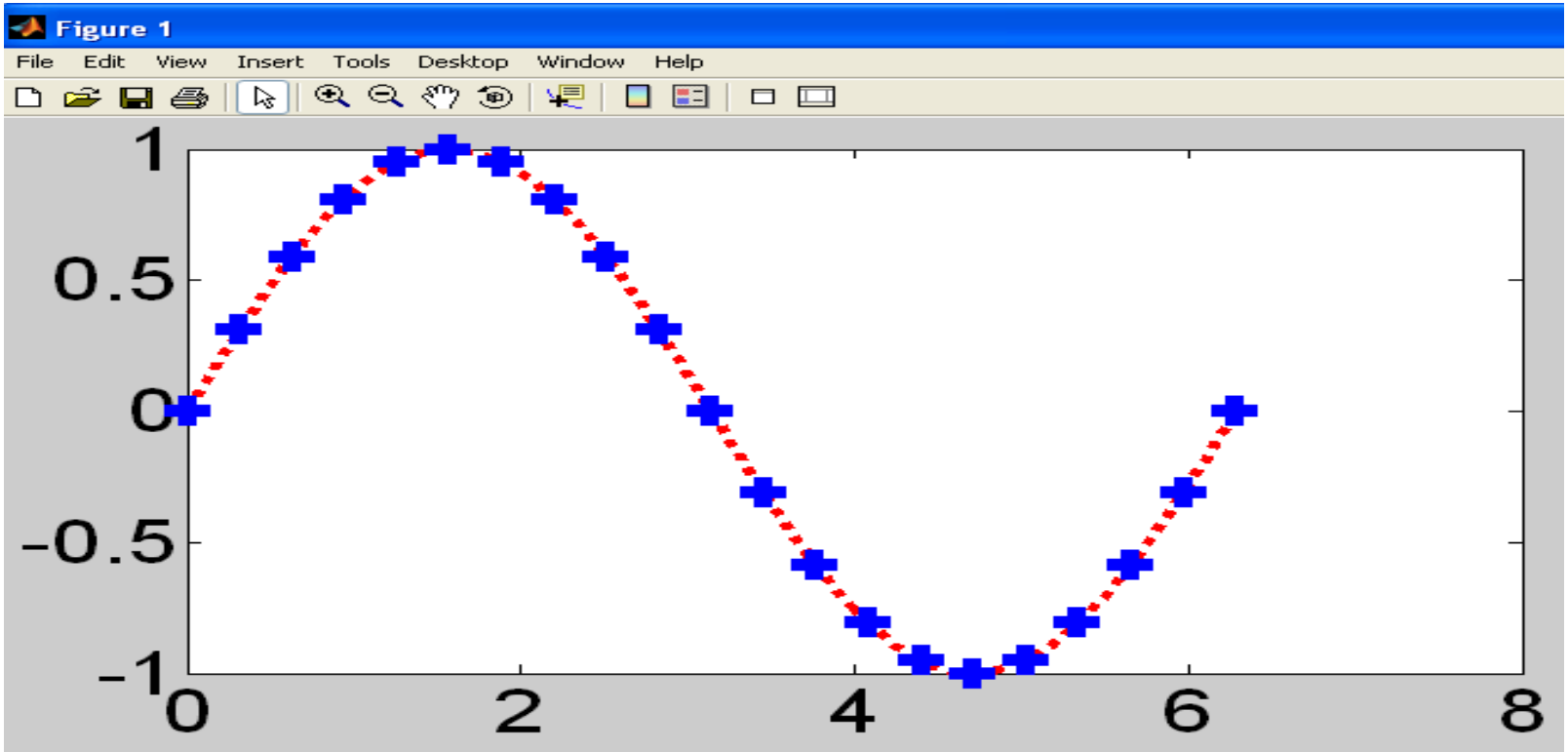
Example:

C:\MATLAB7\work\as5.m

File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

```
1 - x1 = 0:pi/100:2*pi;
2 - x2 = 0:pi/10:2*pi;
3 - plot(x1,sin(x1),'r:',x2,sin(x2),'b+')
```



➢ Imaginary and Complex Data (Z)

plot(Z)

where Z is a complex vector or matrix, is equivalent to

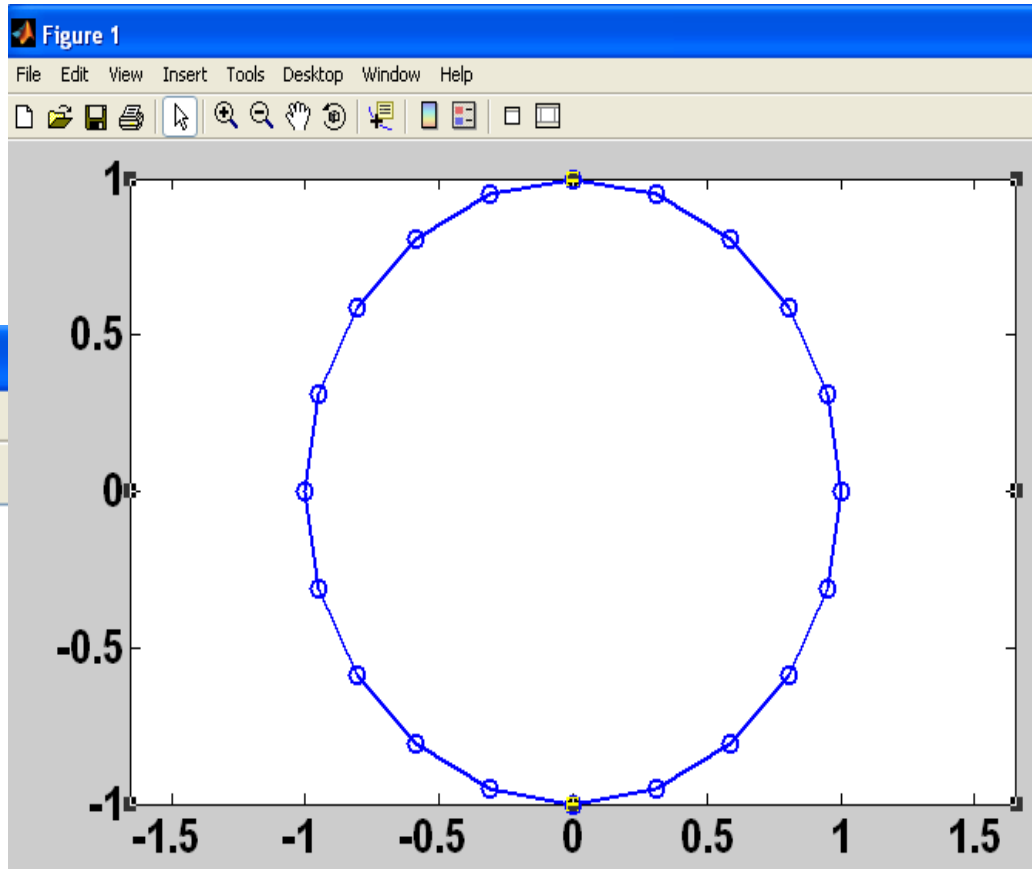
plot(real(Z),imag(Z))

Example:

C:\MATLAB7\work\as5.m

File Edit Text Cell Tools Debug Desktop Window Help

```
1 - t = 0:pi/10:2*pi;
2 - plot(exp(i*t),'-o')
3 - axis equal
```



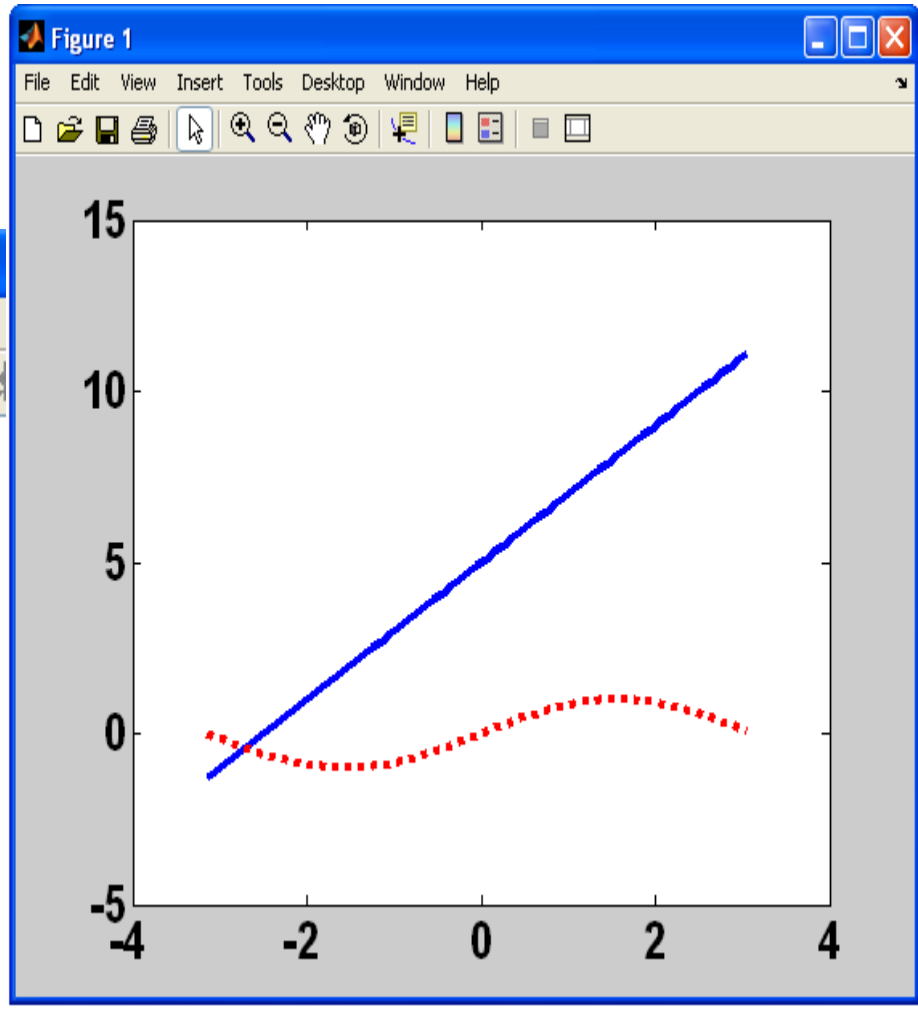
➢ Adding Plots to an Existing Graph

hold on command adds the new data to the current graph

C:\MATLAB7\work\as5.m

File Edit Text Cell Tools Debug Desktop Window Help

```
1 -  x=-pi:0.1:pi;
2 -  y=2*x+5;
3 -  plot(x,y)
4 -  hold on
5 -  y=sin(x);
6 -  plot(x,y,'r:')
```



➢ Multiple Plots in One Figure

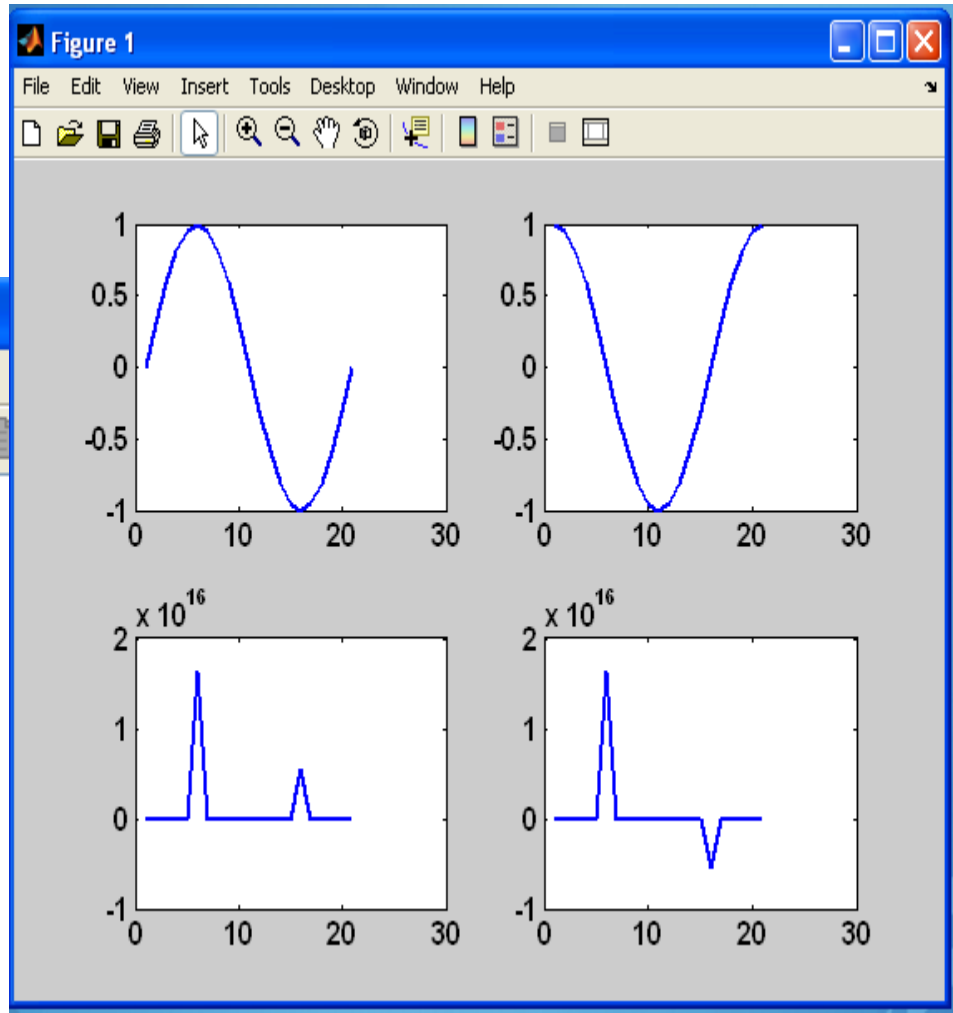
The **subplot(m,n,p)** command enables to display multiple plots in the same window

Example:

C:\MATLAB7\work\as5.m

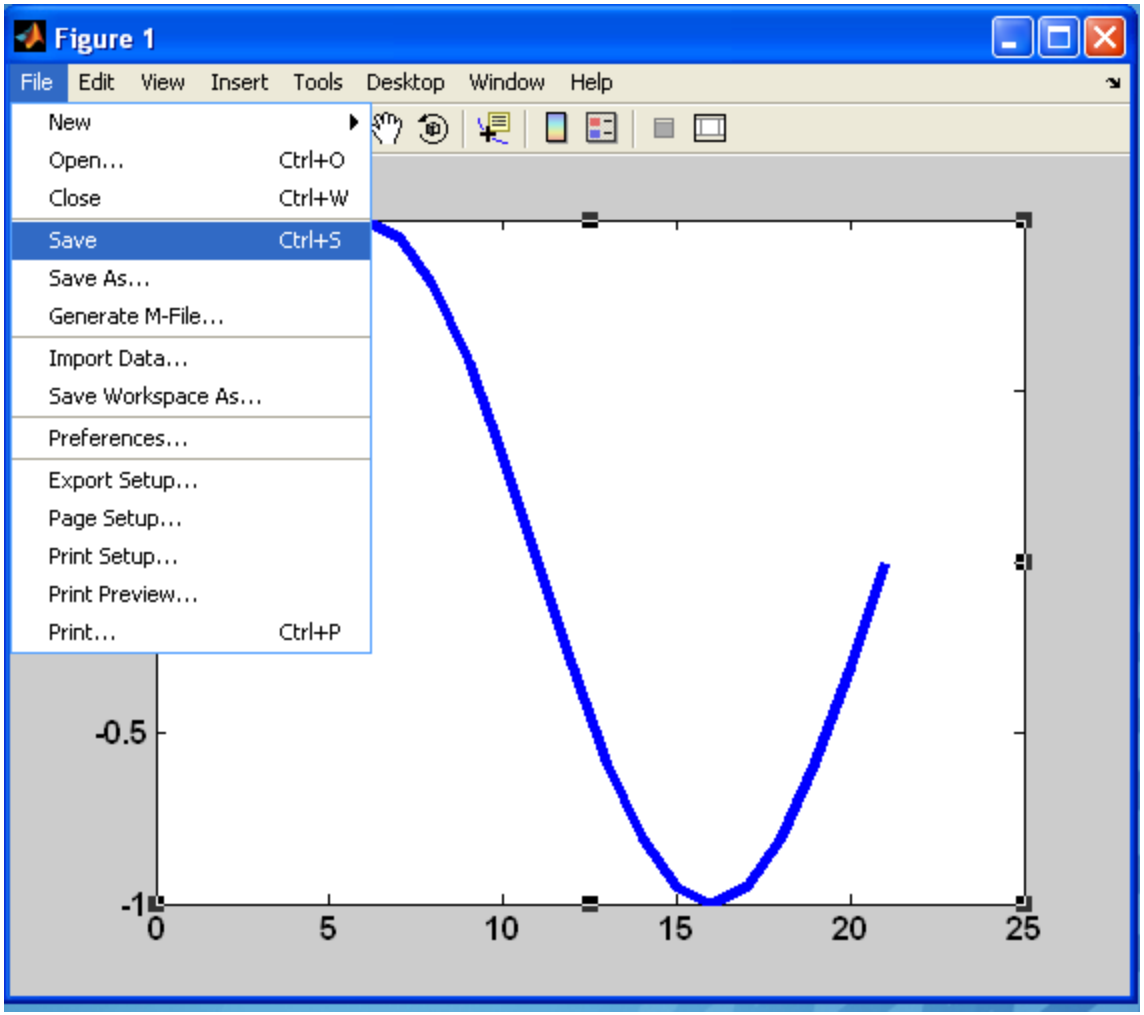
File Edit Text Cell Tools Debug Desktop Window Help

```
1 - t = 0:pi/10:2*pi;
2 - subplot(2,2,1),plot(sin(t))
3 - subplot(2,2,2),plot(cos(t))
4 - subplot(2,2,3),plot(tan(t))
5 - subplot(2,2,4),plot(sec(t))
```



➢ Saving Figures

Save a figure by selecting Save from the File menu



THANK YOU

dastushar74@gmail.com

Questions?