

ELG3125 Signal and System Analysis

Introduction to MATLAB

TA: Hiva Shahoei

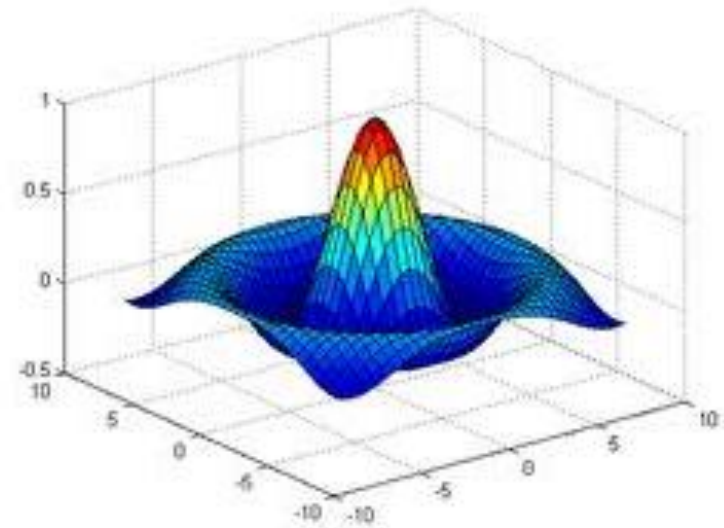
(hshah060@uottawa.ca)

Fall, 2011



Outline

1. MATLAB Overview
2. Desktop and Interfaces
3. Basics
4. Help



MATLab Overview



MATLab (MATrix LABoratory) is a high-performance language for technical computing. Typical uses include:

- Math and computation,
- Algorithm development
- Data acquisition,
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics

MATLab Overview



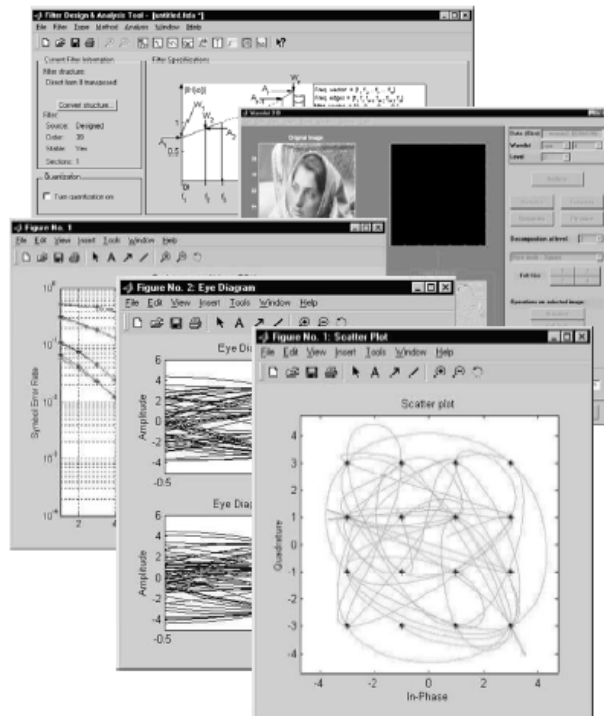
A high-level matrix/array language with

- control flow statements,
- functions,
- data structures,
- input/output,
- and object-oriented programming features.

MATLab Overview



Toolboxes

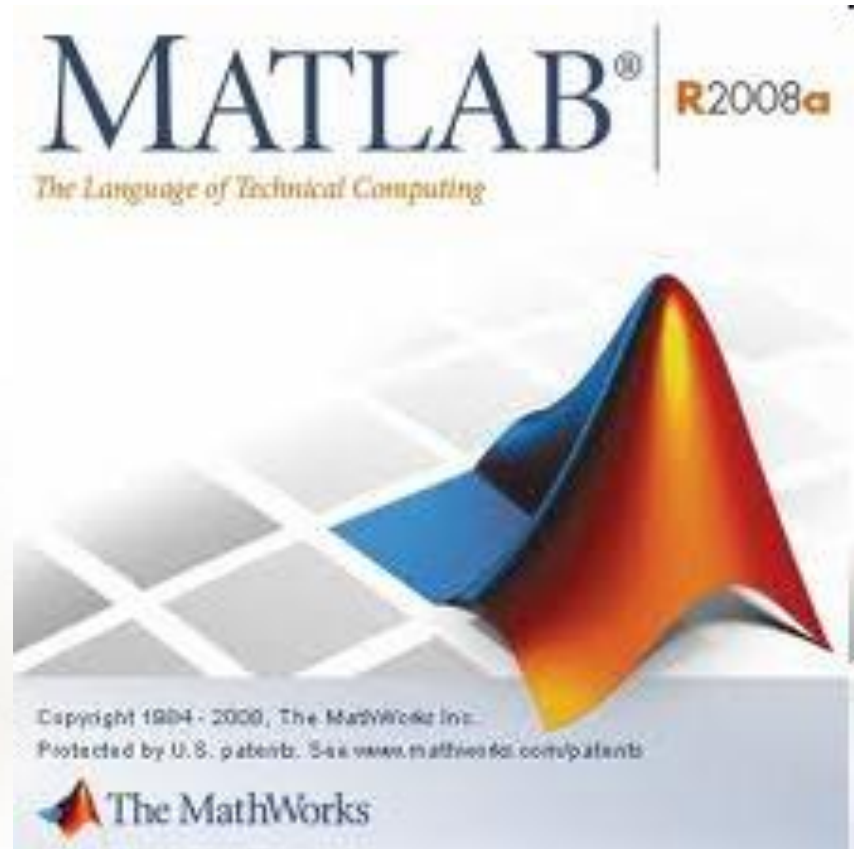


- Signal Processing
- Communications
- Filter Design
- Wavelet Analysis
- Statistics
- Optimization
- Image Processing
- Others...



Start to run MATLAB.....

- Starting MATLAB and a screen similar to the right one appears, followed by the desktop interface.





Desktop and Interfaces (Default)

- Command Window
- Command History
- Current Directory
- Workspace

Annotations for the MATLAB interface:

- Menus change, depending on the tool you are using.
- Enter MATLAB statements at the prompt.
- View or change the current directory.
- Move or resize the Command Window.
- Contents in Current Directory
- Command History

The interface shows the MATLAB 7.5.0 (R2007b) desktop with the following components:

- Current Directory:** A file browser showing folders like 'mydemos', 'profile_results', 'sea_temp', 'temp' and an M-file 'collatz.m'.
- Command History:** A list of executed commands including 'clc', 'a= rand(5)', 'format long', 'openvar a', and 'b=(50.44, 0.1234567)'.
- Command Window:** A text area for entering MATLAB code, currently showing '>>'.

Desktop and Interfaces



- Rearrange the desktop, for example one layout like the left one. It gives more space for each window.

The image shows the MATLAB 7.4.0 (R2007a) desktop environment. The Command Window is active, displaying the command `>> magic(5)` and its output, a 5x5 magic square matrix. The desktop layout is clean, with the Command Window, Command History, Workspace, and Current Directory windows visible. The Start button is visible in the bottom left corner, and the OVR button is in the bottom right corner.

```
>> magic(5)

ans =

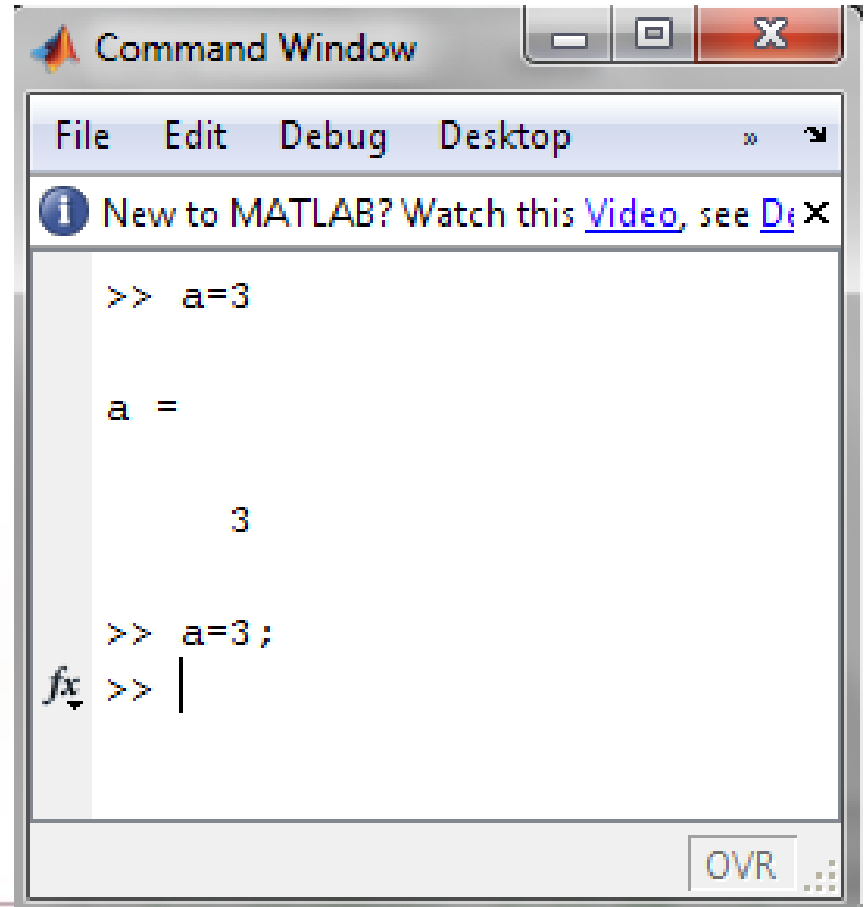
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

>>
```




Command Window

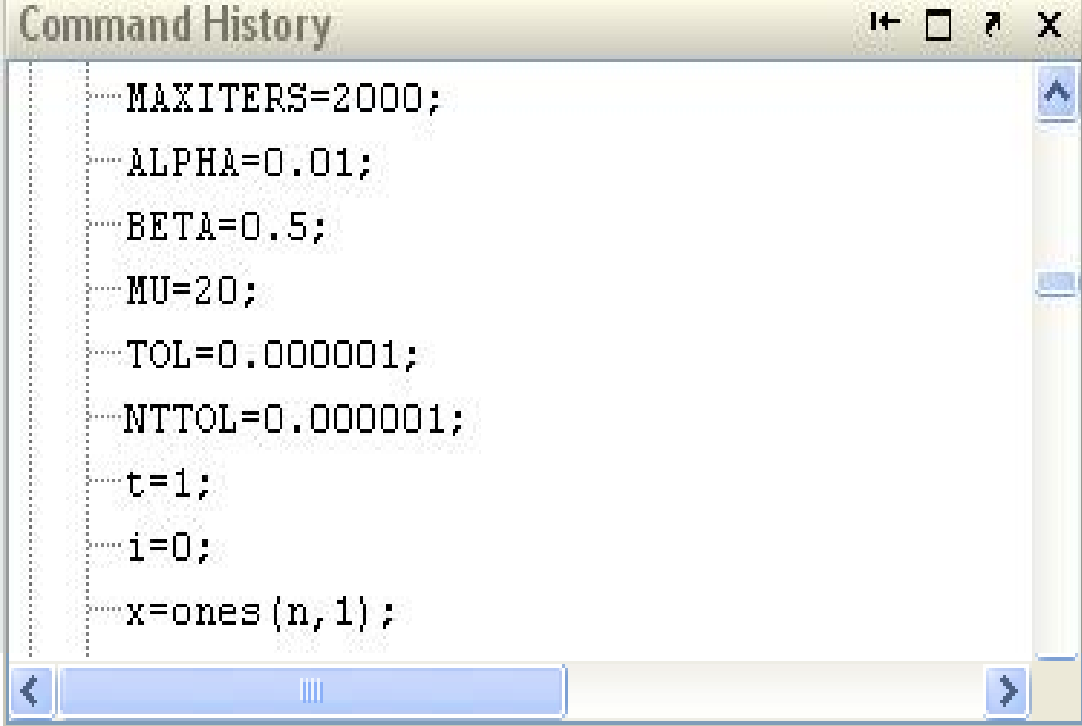
- Prompt `>>` appears and Matlab command can be entered.
- Press `<ENTER>` to execute the Matlab command and next prompt `>>` appears.
- If a command is terminated with a semicolon (`;`), no results will be displayed.
- Otherwise results will be displayed before next prompt appears.





Command History

- Command History store the commands entered previously.
- Provide an easy way to repeat a command later on.

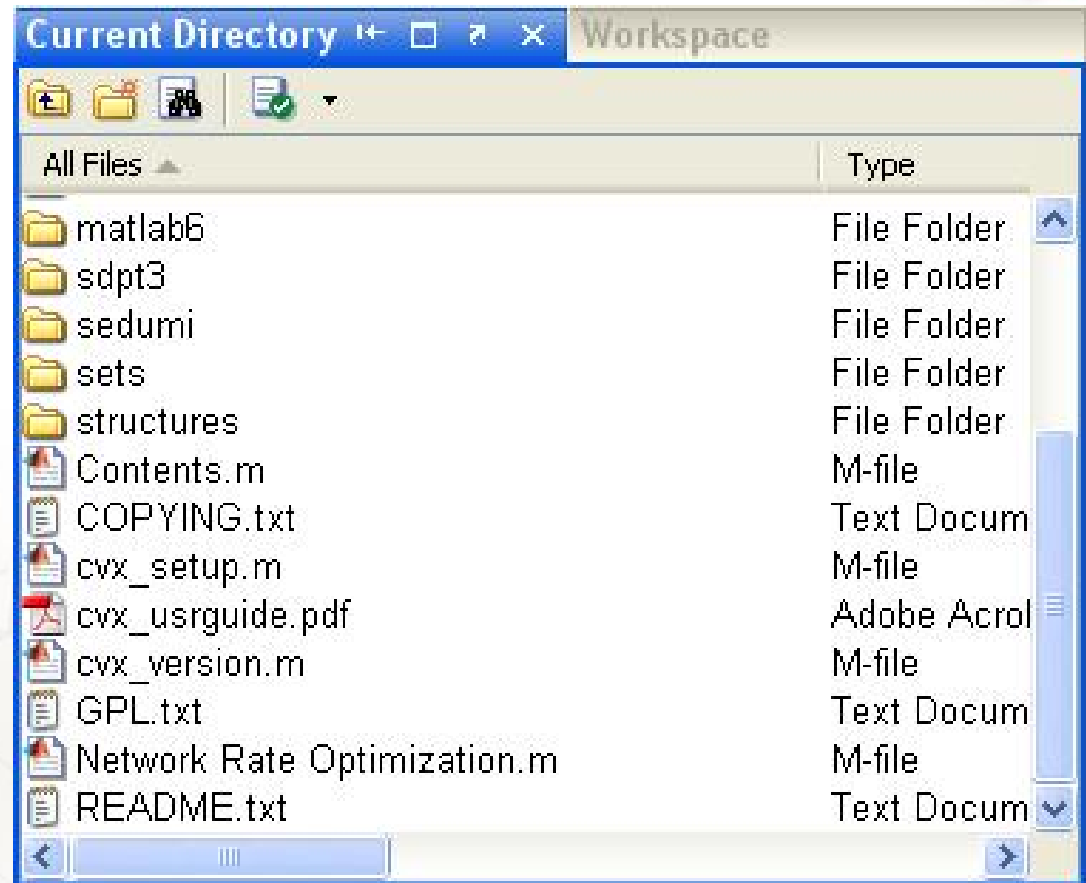


```
Command History
-----
MAXITERS=2000;
ALPHA=0.01;
BETA=0.5;
MU=20;
TOL=0.000001;
NTTOL=0.000001;
t=1;
i=0;
x=ones(n,1);
```



Current Directory

- Show all the folders and files in the current directory





Workspace

- The Workspace browser shows the name of each variable, its value, and the Min and Max calculations
- The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session

The screenshot shows the MATLAB Workspace browser window. The title bar reads 'Current Directory' and 'Workspace'. Below the title bar is a toolbar with icons for workspace operations and a 'Stack: Base' dropdown menu. The main area contains a table with the following data:

Name	Value	Min	Max
MU	20	20	20
NTTOL	1.0000e-06	1.00...	1.00...
TOL	1.0000e-06	1.00...	1.00...
i	0	0	0



Matlab Symbols

- `>>` Command prompt
- `...` Continue statement in next line
- `,` Separate statements and data, e.g.,
`A = [5.92, 8.13, 3.53]`
- `%` Start comment which ends at the end of line, e.g.,
`% The purpose of this routine is to compute`
`% the value of ...`
- `;` Suppress output or used as row separator in a matrix
- `:` Specify a range and generates a sequence of numbers that you can use in creating or indexing into arrays. For example, `N = 6:17.`

MATLab Variables



Variable name

- CASE SENSITIVE. e.g., x and X are two different variables
- up to 63 characters (as of MATLAB 6.5 and newer)
- must **start with a letter** and can be followed by letters, digits, or underscores. e.g., x3_2, but 2_x3 is not correct.



MATLab Variables

- Variables are stored in Matlab in the form of matrices
which are generally of size $M \times N$.
- Elements of matrix can be real or complex numbers.
- A scalar is a 1×1 matrix.
- A row vector is a $1 \times N$ matrix.
- A column vector is a $M \times 1$ matrix.



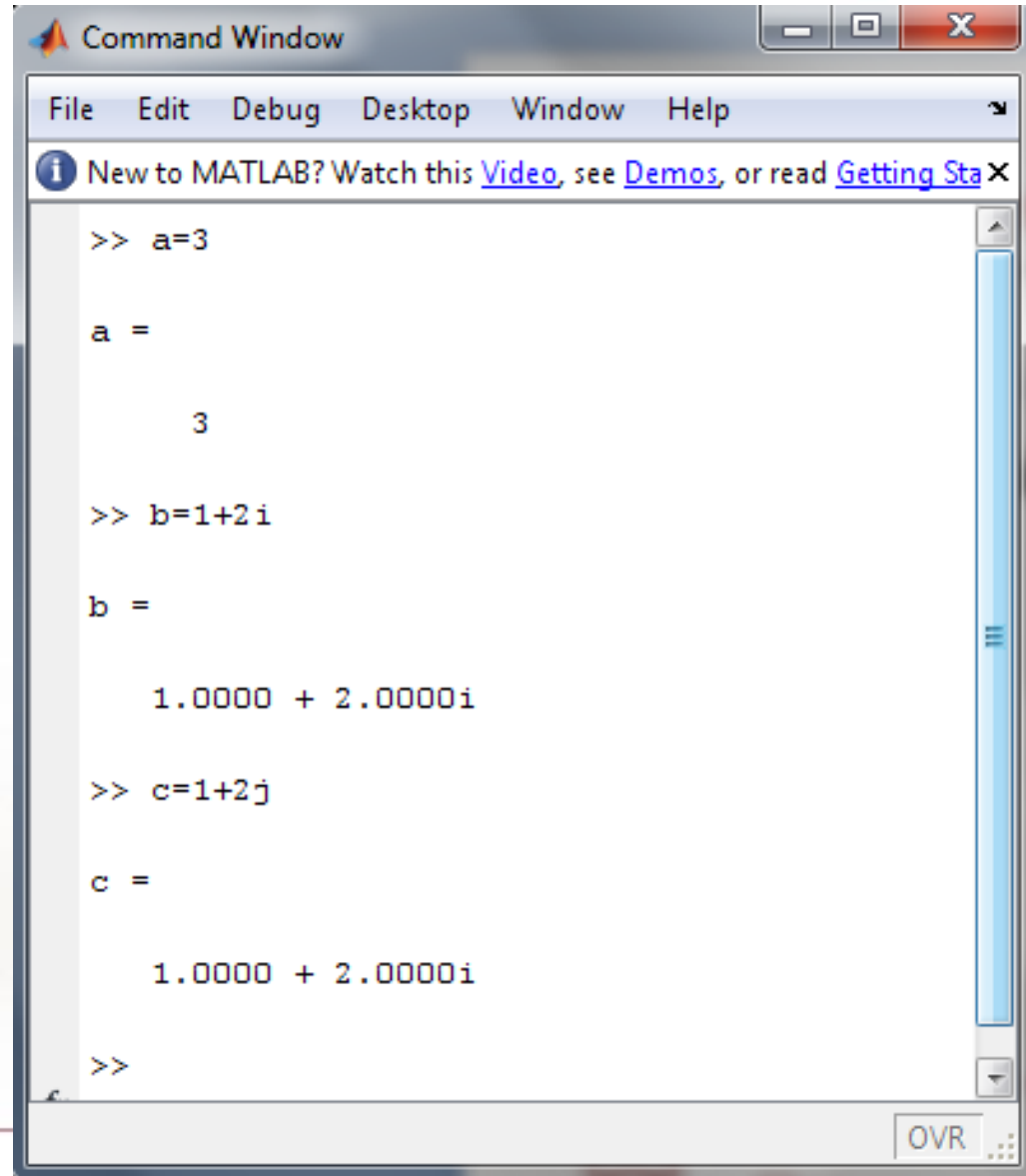
MATLab Variables

Special variables

- **ans** Default variable name for results
- **pi** Value of π
- **eps** Smallest incremental number
- **inf** Infinity
- **NaN** Not a number e.g. 0/0
- **i and j** $i = j = \text{square root of } -1$
- **realmin** The smallest usable positive real number
- **realmax** The largest usable positive real number

Scalars examples

- Real numbers
- Complex numbers



```
Command Window
File Edit Debug Desktop Window Help
New to MATLAB? Watch this Video, see Demos, or read Getting Sta X
>> a=3
a =
    3
>> b=1+2i
b =
    1.0000 + 2.0000i
>> c=1+2j
c =
    1.0000 + 2.0000i
>>
```

Vectors examples

- Row vector

$x=[1\ 2\ 3]$

- Column vector

$x=[1;2;3]$

```
Command Window
File Edit Debug Desktop Window Help
>> x=[1 2 3]
x =
     1     2     3
>> x=[1,2,3]
x =
     1     2     3
>> x=[1;2;3]
x =
     1
     2
     3
>> x=[1 2 3]'
x =
     1
     2
     3
>>
```



Matrix example

$x=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 9]$

- Matrix is input row by row.
- Row elements are separated by space or comma
- Different rows are separated by semicolon.

```
Command Window
File Edit Debug Desktop Window Help
>> x=[1 2 3;4 5 6;7 8 9]
x =
     1     2     3
     4     5     6
     7     8     9
>> |
```



Zero matrix

- `>> x=zeros(3)`

- $x =$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- `>> x=zeros(2,3)`

- $x =$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



One Matrix

- `>> x=ones(3)`

`x =`

1	1	1
1	1	1
1	1	1

- `>> x=ones(4,3)`

`x =`

1	1	1
1	1	1
1	1	1
1	1	1



Identity Matrix (Unit Matrix)

```
>> x=eye(3)
```

x =

1	0	0
0	1	0
0	0	1

```
>> y=eye(5)
```

y =

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Matrix Operation



```
>> A=[10 20 30;11 21 31]
```

```
A =
```

```
10 20 30  
11 21 31
```

```
>> a=A(1,2)
```

```
a =
```

```
20
```

```
>> a1=A(1,:)
```

```
a1 =
```

```
10 20 30
```

```
>> a2=A(:,2)
```

```
a2 =
```

```
20  
21
```

```
>> A1=A(1:2,1:2)
```

```
A1 =
```

```
10 20  
11 21
```



Arithmetic Operator

Matrix Operation:

+ Addition

- Subtraction

* Multiplication

/ Right division

\ Left division

^ Power

' Conjugate transpose

A=[10 20 30;11 21 31]

B=[1 1; 1 1; 1 1]

>> A*B

ans =

60 60

63 63



Arithmetic Operator

Element-by-element operations.

- . * Multiplication
- . / Right division
- . \ Left division
- . ^ Power
- . ' Transpose (unconjugated)

```
A=[10 20 30;11 21 31]
```

```
C=[1 1 1;1 1 1]
```

```
>> A.*C
```

```
ans =
```

```
10 20 30
```

```
11 21 31
```



Relational operators

- $==$ equal
- \neq not equal
- $<$ less than
- \leq less than or equal
- $>$ greater than
- \geq greater than or equal



Flow control

- **if-elseif-else** statements
- **switch-case** statements
- **for** loops
- **while** loops





Basic Math Functions

- $\text{abs}(x)$ absolute value
- $\text{exp}(x)$ exponential
- $\text{sin}(x), \text{cos}(x)$ sine, cosine
- $\text{log}(x), \text{log}_{10}(x)$ natural logarithm, common logarithm
- $\text{sqrt}(x)$ square root
- $\text{sign}(x)$ signum
- $\text{round}(x), \text{fix}(x)$ round towards nearest integer,
round towards zero
- $\text{floor}(x), \text{ceil}(x)$ round towards negative infinity,
round towards plus infinity
- $\text{size}(x), \text{length}(x)$ size of array,
length of vector



Two types of M-Files

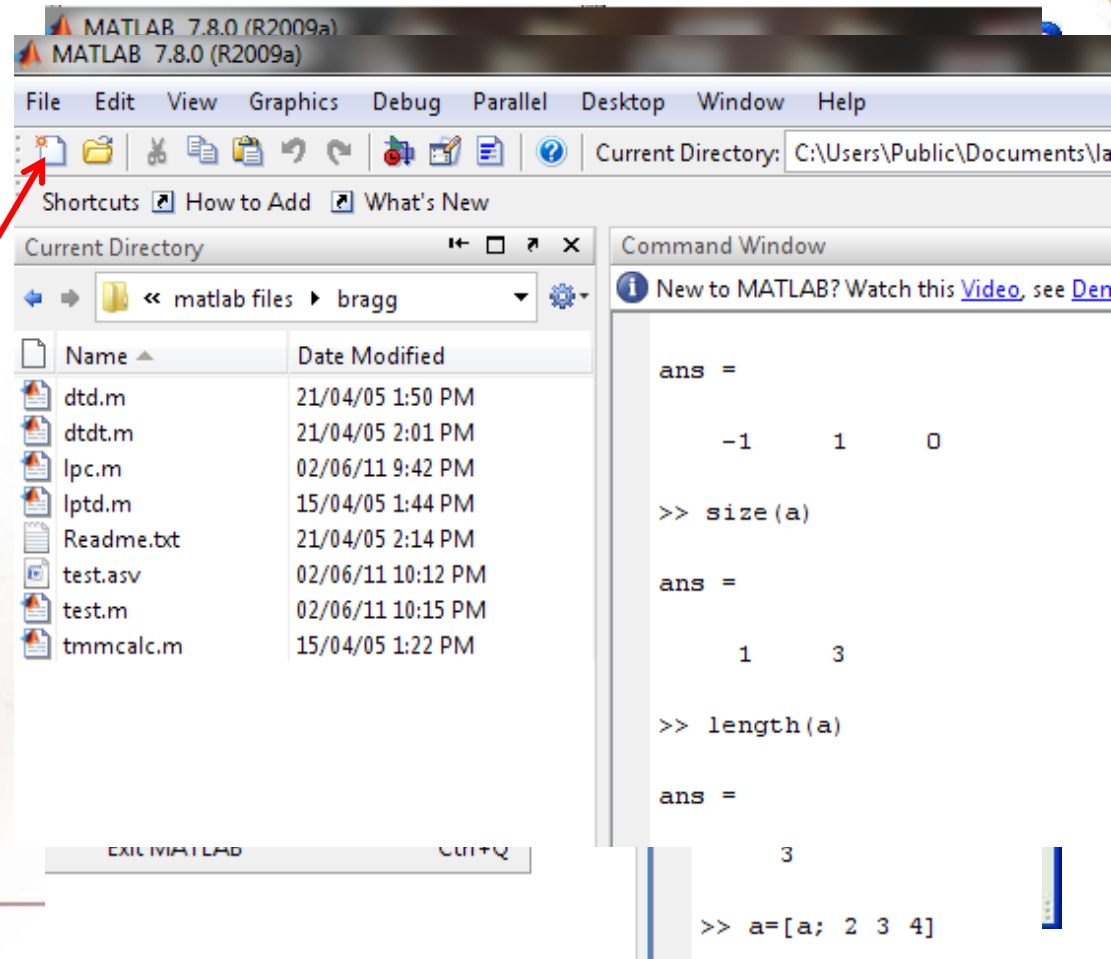
M-files are used to execute a series of commands.

- Script: Running with No Input Arguments;
 - Function: Running with Input Arguments;
- ✓ Both have a .m extension.
 - ✓ Created in Editor Window.



Creating a New Script M-File

- click the New M-file button on the MATLAB desktop toolbar;
- or select File > New > M-File from the MATLAB desktop.





Open an Existing M-File

- File>Open

The * after the filename indicates the file has not been saved since it was last modified.

Set the right-hand text limit (red line) using **Preferences -> Display**.

Toolbar for cell features.

Set breakpoints in the alley.

Colors highlight syntax elements.

```
1 function sequence=collatz(n)
2 % Collatz problem. Generate a sequence of integers resolving to 1
3 % For any positive integer, n:
4 % Divide n by 2 if n is even
5 % Multiply n by 3 and add 1 if n is odd
6 % Repeat for the result
7 % Continue until the result is 1
8
9 sequence = n;
10 next_value = n;
11 while next_value > 1
12     if rem(next_value,2)==0
13         next_value = next_value/2;
14     else
```

Status bar information includes cursor line and column numbers.

Example of a Script M-File



```
a=pi/3; %60 degrees  
x=sin(a);  
b=pi/4; %90 degrees  
y=cos(b);  
c=x*y
```

```
c =
```

```
0.6124
```




Example of a function M-File

The screenshot shows a MATLAB editor window titled "Editor - Untitled3*" with a menu bar (File, Edit, Text, Go, Cell, Tools, Debug, Desktop) and a toolbar. The code in the editor is as follows:

```
1 function [ output ] = Untitled3 ( input )
2 %UNTITLED3 Summary of this function goes here
3 % Detailed explanation goes here
4
5
6 end
```

Below the editor, the status bar shows "Untitled3" and "Ln 1 Col 40". To the right, a portion of another window titled "ettin..." is visible, showing a "Debug" toolbar and some text including "name (a)" and "grees".



MATLab Graphing

- **plot** Plot in linear coordinates as a continuous function
- **stem** Plot in linear coordinates as discrete samples
- **loglog** Logarithmic x and y axes
- **semilogx** Linear y and logarithmic x axes
- **semilogy** Linear x and logarithmic y axes
- **xlabel** Label x-axis
- **ylabel** Label y-axis
- **title** Put a title on the plot

MATLab Graphing (cont.)

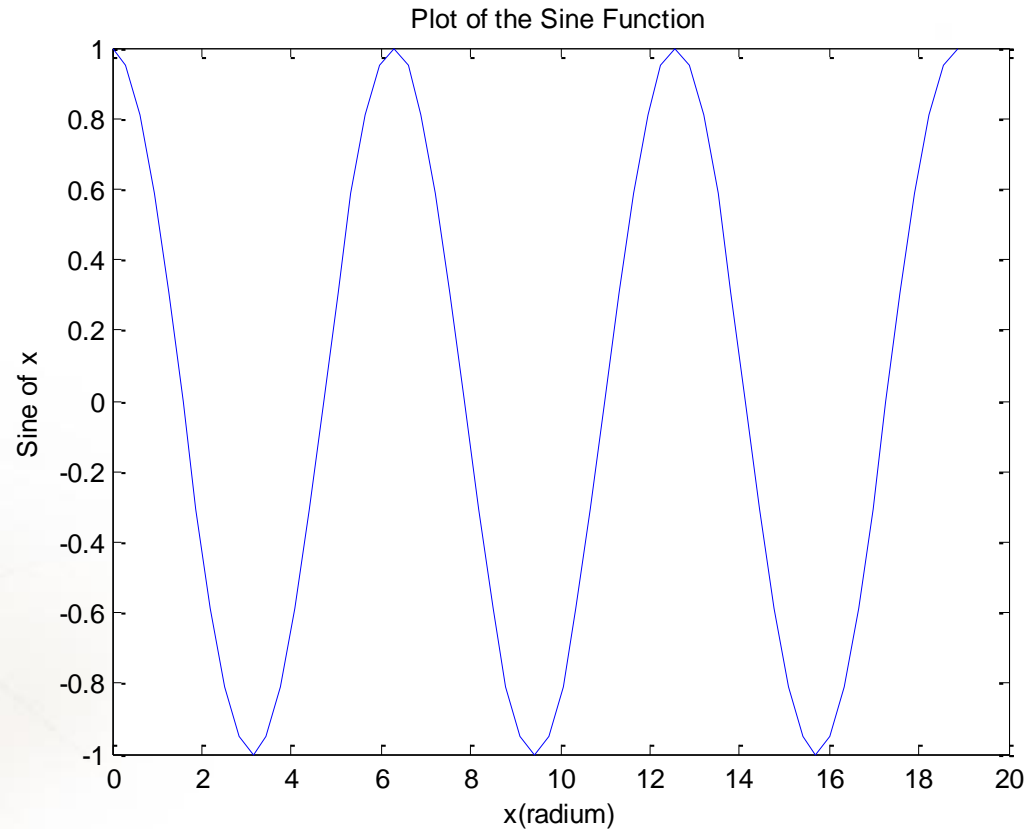


- **grid** Add a grid to the plot
- **text** Allow placing text at specified coordinates of the plot
- **axis** Control axis scaling and appearance
- **figure** Create a figure for plotting
- **hold** Hold current graph
- **subplot** Divide figure window and select figure position

Plot



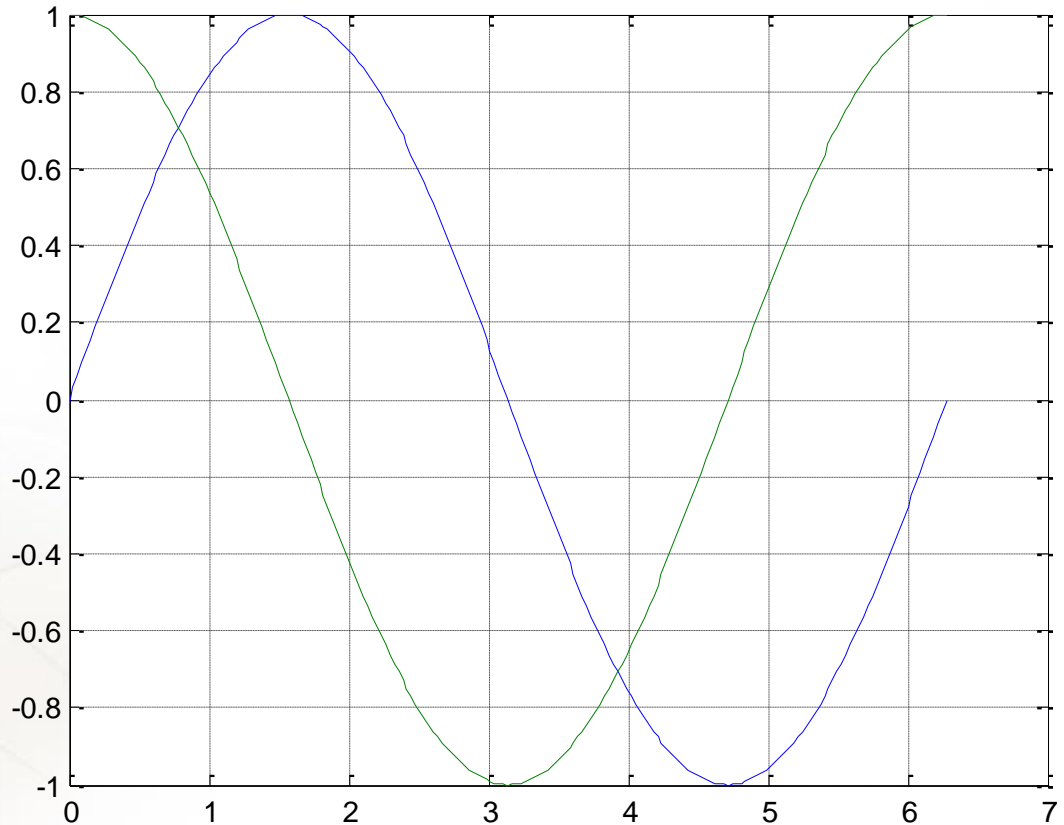
- $x = 0:\pi/10:6*\pi;$
- $y = \cos(x);$
- `plot(x,y)`
- `xlabel('x(radium)')`
- `ylabel('Sine of x')`
- `title('Plot of the Sine Function')`





Usage of hold

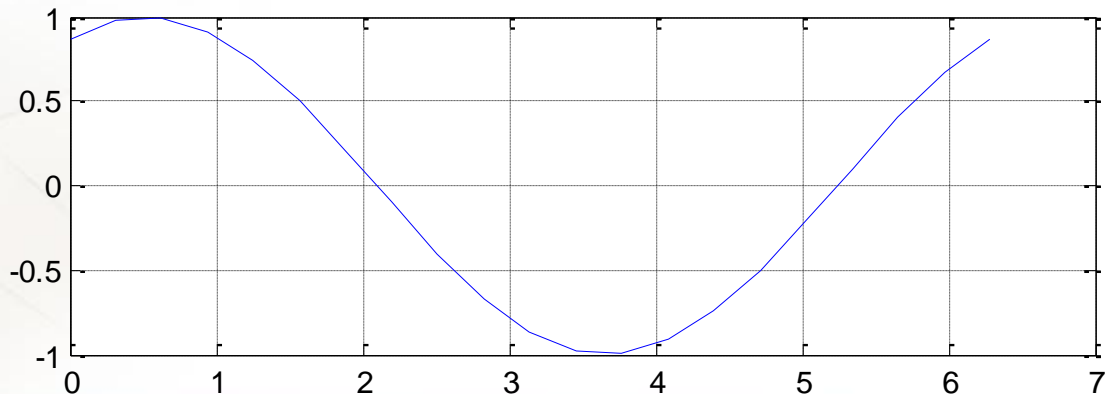
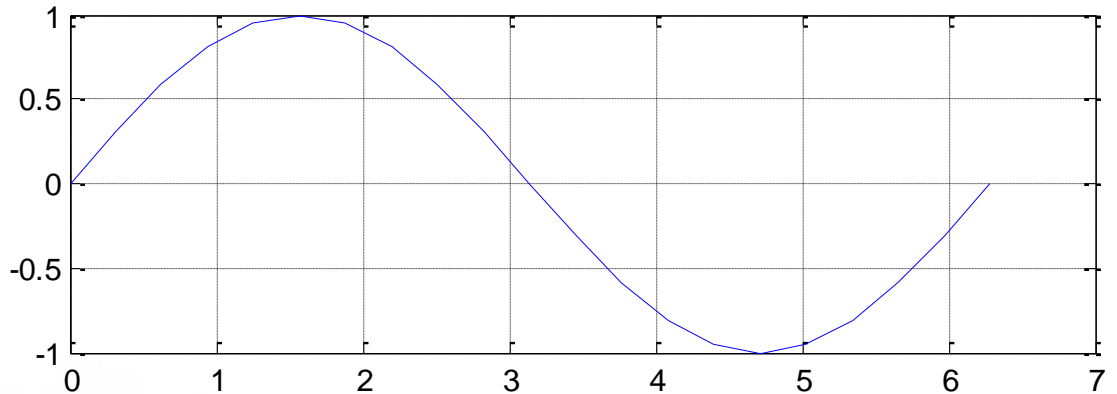
```
x=0:pi/100:2*pi;  
y1=sin(x);  
y2=sin(x+pi/2);  
plot(x,y1,'b');  
hold on;  
plot(x,y2,'g'),grid
```





Usage of subplot

```
x=0:pi/10:2*pi;  
y1=sin(x);  
y2=sin(x+pi/3);  
subplot(2,1,1);  
plot(x,y1),grid;  
subplot(2,1,2);  
plot(x,y2),grid;
```

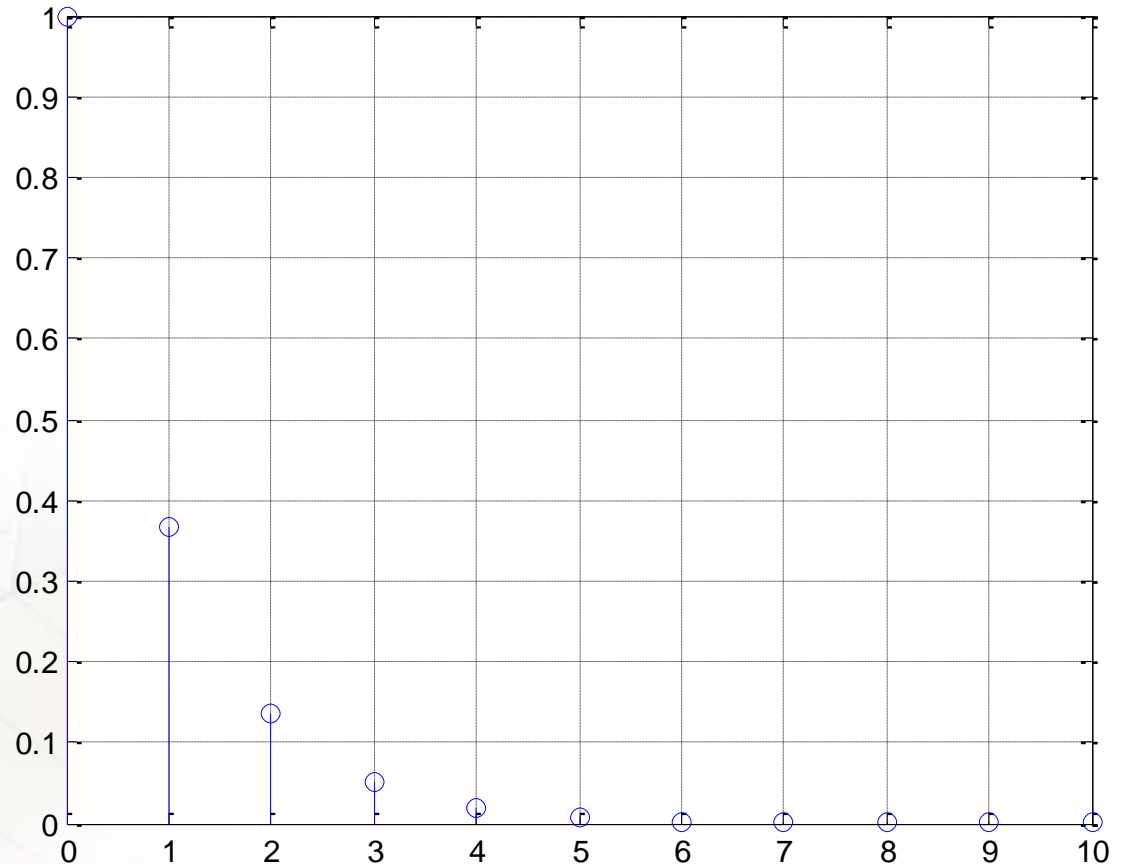




Stem

$$y[n] = e^{-n}$$

- `n=0:10;`
- `y=exp(-n);`
- `stem(n,y),grid;`



How to get help?

- help

Display help text
in Command Window.

e.g.,

>>help plot

```
Command Window
>> help plot
PLOT Linear plot.
PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
then the vector is plotted versus the rows or columns of the matrix,
whichever line up. If X is a scalar and Y is a vector, disconnected
line objects are created and plotted as discrete points vertically at
X.

PLOT(Y) plots the columns of Y versus their index.
If Y is complex, PLOT(Y) is equivalent to PLOT(real(Y),imag(Y)).
In all other uses of PLOT, the imaginary part is ignored.

Various line types, plot symbols and colors may be obtained with
PLOT(X,Y,S) where S is a character string made from one element
from any or all the following 3 columns:

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

For example, PLOT(X,Y,'c+:') plots a cyan dotted line with a plus
at each data point; PLOT(X,Y,'bd') plots blue diamond at each data
point but does not draw any line.
```




Exercise

- ✓ Set up matrix A of 4x5, matrix B of 5x3;
do multiplication $C=A*B$ with comma and without comma behind.
- ✓ Set up two matrices of 4x3,
do multiplication, addition, subtraction with element-by-element
operator;
- ✓ Plot $x=\sin(x)+\cos(x+\pi/3)$
specify the range of x with colon.
the plot needs x label, y label, title, green color, grid.



Enjoy signals.....



uOttawa

ELG3125 Signal and System Analysis Fall 2011
School of Electrical Engineering and Computer Science
(EECS)