INTRODUCTION TO SCIENTIFIC & ENGINEERING COMPUTING BIL 108E, CRN24023

Dr. S. Gökhan Karaman

Technical University of Istanbul

February 22, 2010

ARRAYS

ARRAYS

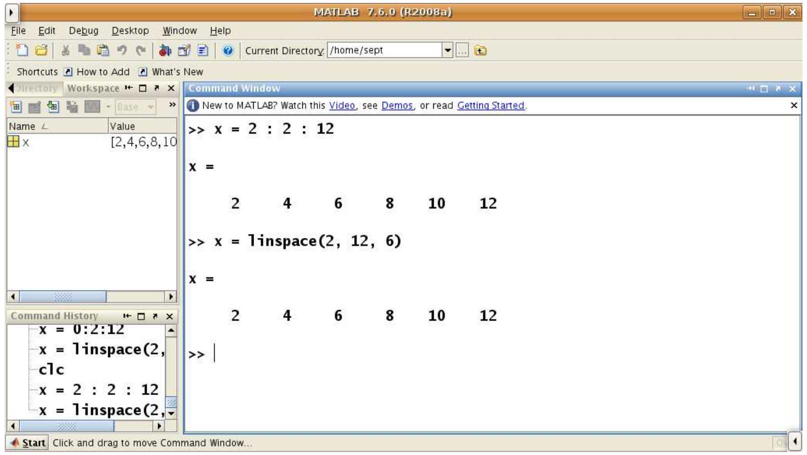
- Matlab stores all types of variables as in the form of an array.
- A single element array is called a scalar.
- An array with one column or row is called a vector.
- An array with m rows and n columns, where $m, n \neq 1$ is called a matrix.

ARRAYS

ARRAYS

- `x = start : increment : end`
- `x = linspace(start, end, size_of_vector)`
  `logspace`
- `X =` $[a_{11}, a_{12}, a_{13}; a_{21}, a_{22}, a_{23}]$
- `x = x'` (transpose of x)

ARRAYS

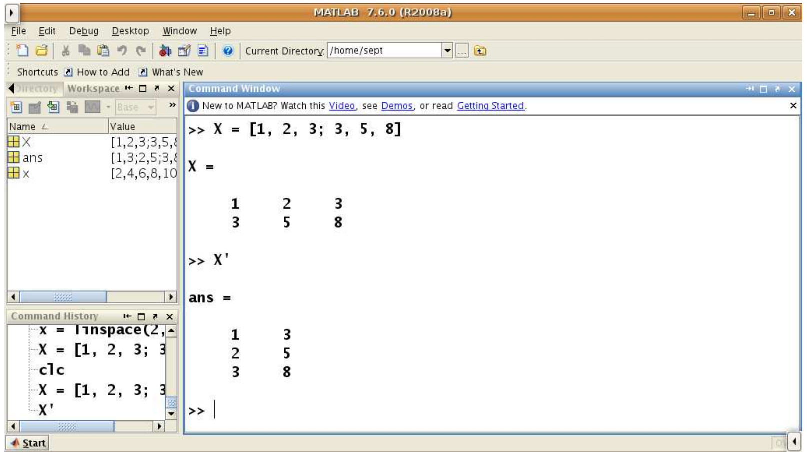

MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Shortcuts How to Add What's New
Directory Workspace
Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start Click and drag to move Command Window...

Name	Value
x	[2,4,6,8,10

```
>> x = 2 : 2 : 12

x =

     2     4     6     8    10    12

>> x = linspace(2, 12, 6)

x =

     2     4     6     8    10    12

>> 
```

```
x = 0:2:12
x = linspace(2,
clc
x = 2 : 2 : 12
x = linspace(2,
```

ARRAYS


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Shortcuts How to Add What's New
Directory Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
X	[1,2,3;3,5,8
ans	[1,3;2,5;3,8
x	[2,4,6,8,10

```
>> X = [1, 2, 3; 3, 5, 8]

X =

     1     2     3
     3     5     8

>> X'

ans =

     1     3
     2     5
     3     8

>> 
```

```
x = linspace(2,
X = [1, 2, 3; 3
clc
X = [1, 2, 3; 3
X'
```

VECTORS

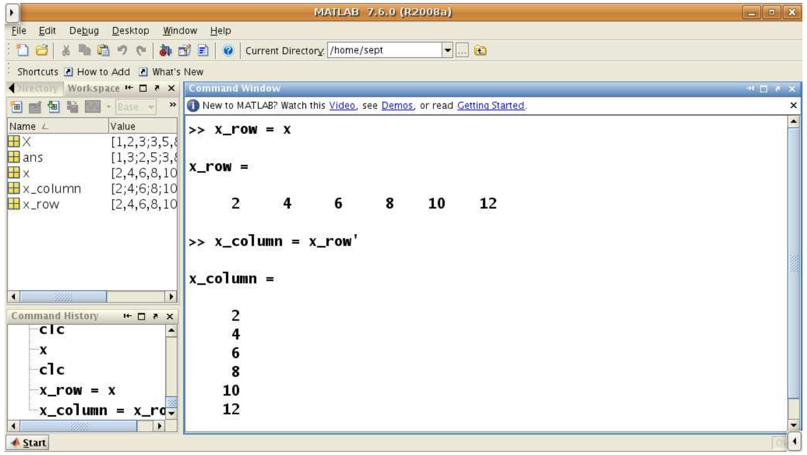
ROW VECTOR

$$\mathbf{a} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix}$$

COLUMN VECTOR

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

VECTORS


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Shortcuts How to Add What's New
Directory Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
X	[1,2,3;3,5,8
ans	[1,3;2,5;3,8
x	[2,4,6,8,10
x_column	[2;4;6;8;10
x_row	[2,4,6,8,10

```
>> x_row = x

x_row =

     2     4     6     8    10    12

>> x_column = x_row'

x_column =

     2
     4
     6
     8
    10
    12
```

```
clc
x
clc
x_row = x
x_column = x_ro
```

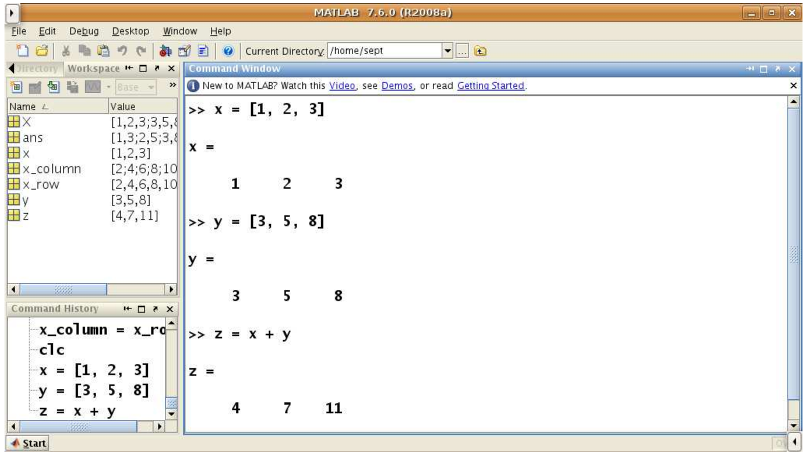
VECTOR ADDITION AND SUBTRACTION

Addition and subtraction are element-by element operations

- $c = a + b,\ c_i = a_i + b_i,\ i = 1, 2, ..., n$
- $d = a - b,\ d_i = a_i - b_i,\ i = 1, 2, ..., n$

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \pm \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_1 \pm b_1 \\ a_2 \pm b_2 \\ \vdots \\ a_m \pm b_m \end{pmatrix}$$

VECTOR ADDITION AND SUBTRACTION

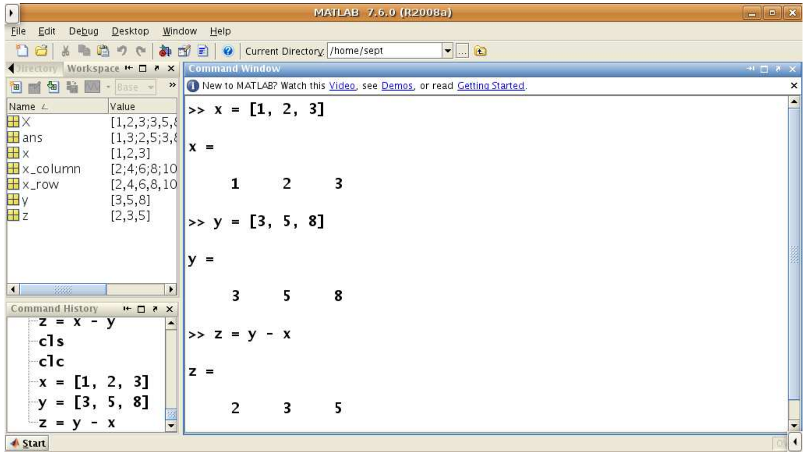

MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
X	[1,2,3;3,5,8
ans	[1,3;2,5;3,8
x	[1,2,3]
x_column	[2;4;6;8;10
x_row	[2,4,6,8,10
y	[3,5,8]
z	[4,7,11]

```
x_column = x_ro
clc
x = [1, 2, 3]
y = [3, 5, 8]
z = x + y
```

```
>> x = [1, 2, 3]

x =

     1     2     3

>> y = [3, 5, 8]

y =

     3     5     8

>> z = x + y

z =

     4     7    11
```

VECTOR ADDITION AND SUBTRACTION


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
X	[1,2,3;3,5,8
ans	[1,3;2,5;3,8
x	[1,2,3]
x_column	[2;4;6;8;10
x_row	[2,4,6,8,10
y	[3,5,8]
z	[2,3,5]

```
z = x - y
cls
clc
x = [1, 2, 3]
y = [3, 5, 8]
z = y - x
```

```
>> x = [1, 2, 3]

x =

     1     2     3

>> y = [3, 5, 8]

y =

     3     5     8

>> z = y - x

z =

     2     3     5
```

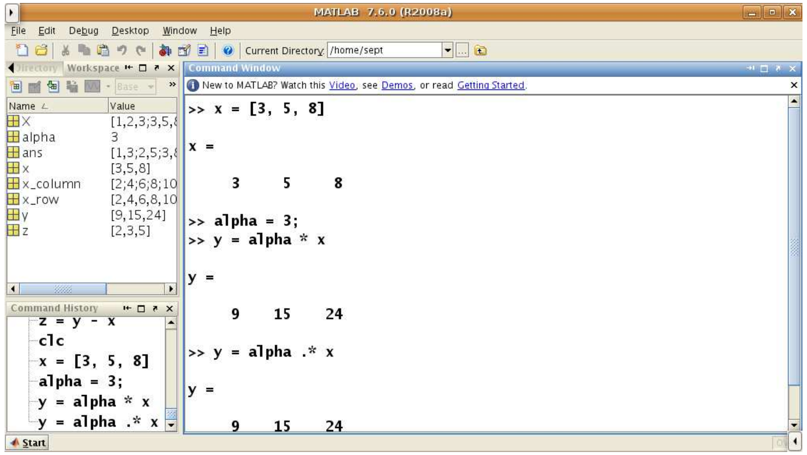
MULTIPLICATION BY SCALAR

Multiplication by a scalar involves multiplying each element in the vector by the scalar:

- $b = \alpha \times (a_i) = (\alpha \times a_i)$

$$\mathbf{b} = \alpha \times \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \alpha \times a_1 \\ \alpha \times a_2 \\ \vdots \\ \alpha \times a_n \end{pmatrix}$$

MULTIPLICATION BY SCALAR


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
X	[1,2,3;3,5,8
alpha	3
ans	[1,3;2,5;3,8
x	[3,5,8]
x_column	[2;4;6;8;10
x_row	[2,4,6,8,10
y	[9,15,24]
z	[2,3,5]

```
z = y - x
clc
x = [3, 5, 8]
alpha = 3;
y = alpha * x
y = alpha .* x
```

```
>> x = [3, 5, 8]

x =

     3     5     8

>> alpha = 3;
>> y = alpha * x

y =

     9    15    24

>> y = alpha .* x

y =

     9    15    24
```

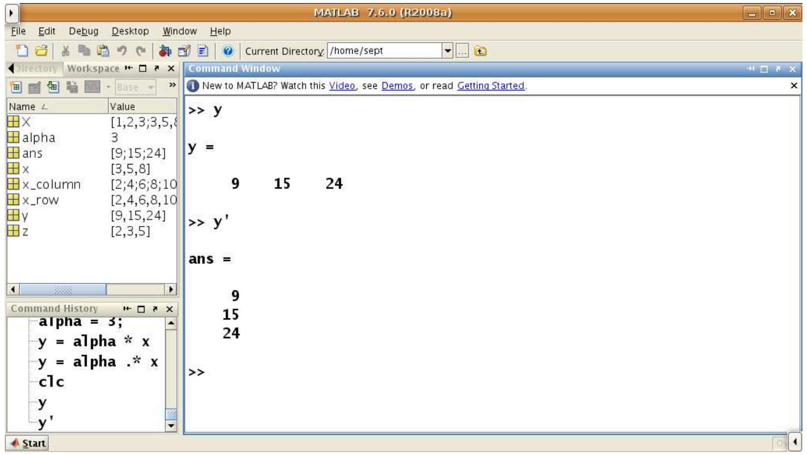
TRANSPOSE OF A VECTOR

TRANSPOSE OF A VECTOR

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

$$\mathbf{a}^{\mathbf{T}} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix}$$

TRANSPOSE OF A VECTOR


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Directory Workspace
Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
X	[1,2,3;3,5,8
alpha	3
ans	[9;15;24]
x	[3,5,8]
x_column	[2;4;6;8;10
x_row	[2,4,6,8,10
y	[9,15,24]
z	[2,3,5]

```
alpha = 3;
y = alpha * x
y = alpha .* x
clc
y
y'
```

```
>> y

y =

     9    15    24

>> y'

ans =

     9
    15
    24

>>
```

OPERATIONS ON ARRAYS

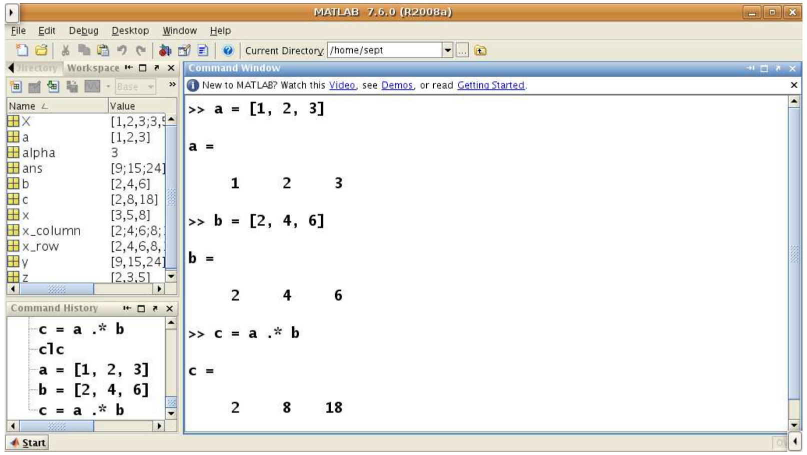
Multiplication and Division

DOT PRODUCT

$$C = A\ .*\ \ B = (a_1 \times b_1, a_2 \times b_2, \dots, a_n \times b_n)$$

$$C = A\ ./\ \ B = (a_1/b_1, a_2/b_2, \dots, a_n/b_n)$$

OPERATIONS ON ARRAYS

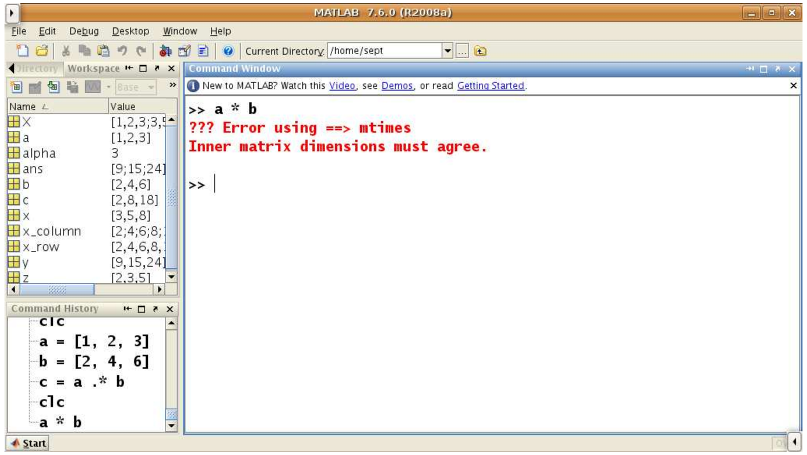

MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Directory Workspace
Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
X	[1,2,3;3,5
a	[1,2,3]
alpha	3
ans	[9;15;24]
b	[2,4,6]
c	[2,8,18]
x	[3,5,8]
x_column	[2;4;6;8;
x_row	[2,4,6,8,
y	[9,15,24]
z	[2,3,5]

```
c = a .* b
clc
a = [1, 2, 3]
b = [2, 4, 6]
c = a .* b
```

```
>> a = [1, 2, 3]

a =

     1     2     3

>> b = [2, 4, 6]

b =

     2     4     6

>> c = a .* b

c =

     2     8    18
```

OPERATIONS ON ARRAYS


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
X	[1,2,3;3,5
a	[1,2,3]
alpha	3
ans	[9;15;24]
b	[2,4,6]
c	[2,8,18]
x	[3,5,8]
x_column	[2;4;6;8;
x_row	[2,4,6,8,
y	[9,15,24]
z	[2,3,5]

```
clc
a = [1, 2, 3]
b = [2, 4, 6]
c = a .* b
clc
a * b
```

```
>> a * b
??? Error using ==> mtimes
Inner matrix dimensions must agree.

>> 
```

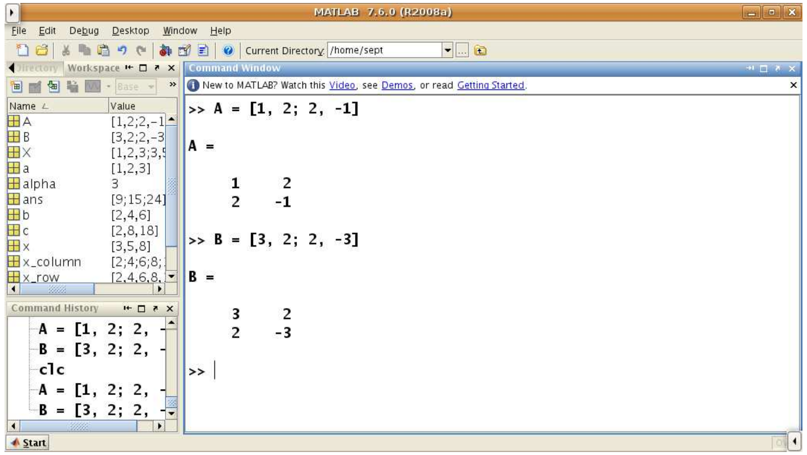
MATRIX MULTIPLICATION

MATRIX MULTIPLICATION

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

$$i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, n$$

MATRIX MULTIPLICATION

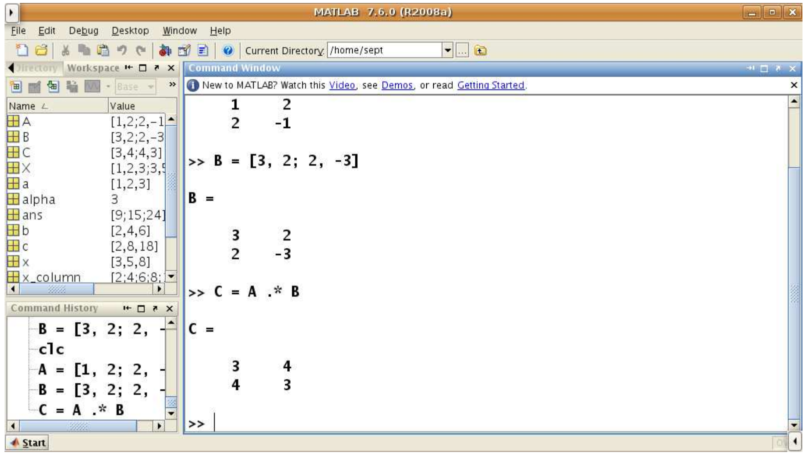

MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
A	[1,2;2,-1
B	[3,2;2,-3
X	[1,2,3;3,5
a	[1,2,3]
alpha	3
ans	[9;15;24]
b	[2,4,6]
c	[2,8,18]
x	[3,5,8]
x_column	[2;4;6;8;
x_row	[2,4,6,8,

```
A = [1, 2; 2, -
B = [3, 2; 2, -
clc
A = [1, 2; 2, -
B = [3, 2; 2, -
```

```
>> A = [1, 2; 2, -1]

A =

     1     2
     2    -1

>> B = [3, 2; 2, -3]

B =

     3     2
     2    -3

>> 
```

MATRIX MULTIPLICATION

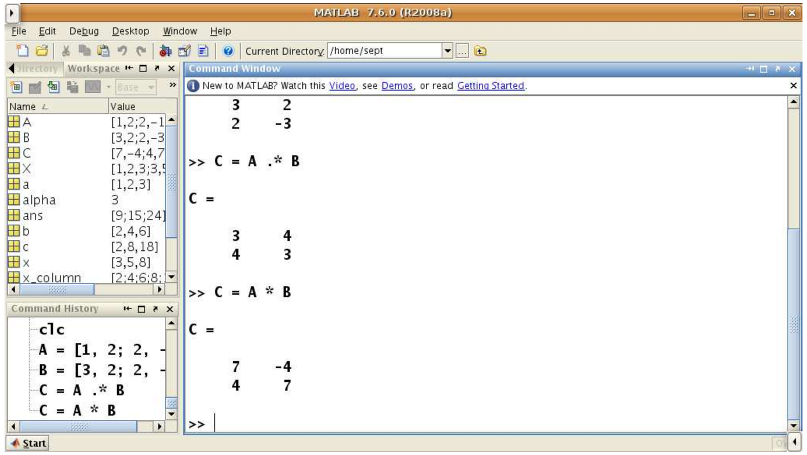

MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Workspace
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
A	[1,2;2,-1
B	[3,2;2,-3
C	[3,4;4,3]
X	[1,2,3;3,5
a	[1,2,3]
alpha	3
ans	[9;15;24]
b	[2,4,6]
c	[2,8,18]
x	[3,5,8]
x_column	[2;4;6;8;

```
B = [3, 2; 2, -
clc
A = [1, 2; 2, -
B = [3, 2; 2, -
C = A .* B
```

```
     1     2
     2    -1

>> B = [3, 2; 2, -3]

B =

     3     2
     2    -3

>> C = A .* B

C =

     3     4
     4     3

>> 
```

MATRIX MULTIPLICATION


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Directory Workspace
Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
A	[1,2;2,-1
B	[3,2;2,-3
C	[7,-4;4,7
X	[1,2,3;3,5
a	[1,2,3]
alpha	3
ans	[9;15;24]
b	[2,4,6]
c	[2,8,18]
x	[3,5,8]
x_column	[2;4;6;8;

```
clc
A = [1, 2; 2, -
B = [3, 2; 2, -
C = A .* B
C = A * B
```

```
     3     2
     2    -3

>> C = A .* B

C =

     3     4
     4     3

>> C = A * B

C =

     7    -4
     4     7

>> 
```

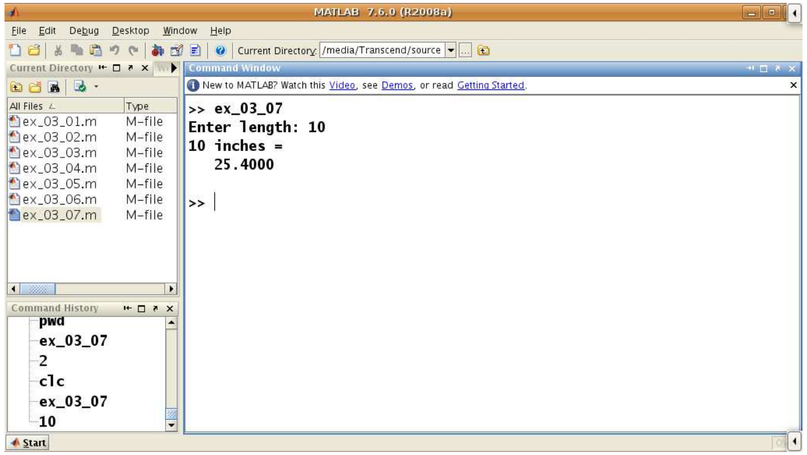
INPUT AND OUTPUT

The input statement

variable = input(' *prompt* ')

```
inch = input('Enter length: ');
centimeter = inch * 2.54;
disp([num2str(inch), ' inches = ']);
disp(centimeter);
```

INPUT AND OUTPUT

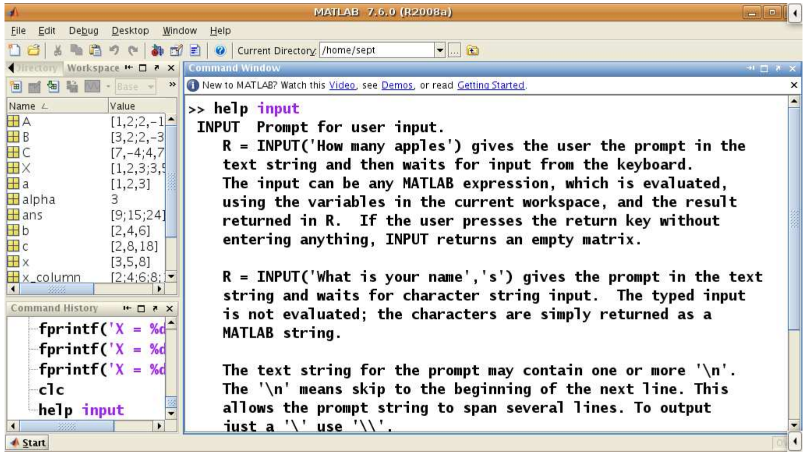

MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /media/Transcend/source
Current Directory
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

All Files	Type
ex_03_01.m	M-file
ex_03_02.m	M-file
ex_03_03.m	M-file
ex_03_04.m	M-file
ex_03_05.m	M-file
ex_03_06.m	M-file
ex_03_07.m	M-file

```
pwd
ex_03_07
2
clc
ex_03_07
10
```

```
>> ex_03_07
Enter length: 10
10 inches =
   25.4000

>> 
```

INPUT AND OUTPUT


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /home/sept
Directory Workspace
Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

Name	Value
A	[1,2;2,-1
B	[3,2;2,-3
C	[7,-4;4,7
X	[1,2,3;3,5
a	[1,2,3]
alpha	3
ans	[9;15;24]
b	[2,4,6]
c	[2,8,18]
x	[3,5,8]
x_column	[2;4;6;8;

```
fprintf('X = %d
fprintf('X = %d
fprintf('X = %d
clc
help input
```

```
>> help input
 INPUT  Prompt for user input.
    R = INPUT('How many apples') gives the user the prompt in the
    text string and then waits for input from the keyboard.
    The input can be any MATLAB expression, which is evaluated,
    using the variables in the current workspace, and the result
    returned in R.  If the user presses the return key without
    entering anything, INPUT returns an empty matrix.

    R = INPUT('What is your name','s') gives the prompt in the text
    string and waits for character string input.  The typed input
    is not evaluated; the characters are simply returned as a
    MATLAB string.

    The text string for the prompt may contain one or more '\n'.
    The '\n' means skip to the beginning of the next line. This
    allows the prompt string to span several lines. To output
    just a '\' use '\\'.
```

input

- The prompt message prompts for the value(s) to be entered. It must be enclosed in apostrophes (single quotes).
- A semicolon at the end of the input statement will prevent the value entered from being immediately echoed on the screen.
- You normally do not use input from the command line, since you shouldn't need to prompt yourself in command – line mode.
- Vectors and matrices may also be entered with input, as long as you remember to enclose the elements in square brackets.
- You can enter an expression in response to the prompt – for example, $a + b$ (as long as a and b have been defined) or rand(5). When entering an expression in this way, don't include a semicolon (it is not part of the expression).

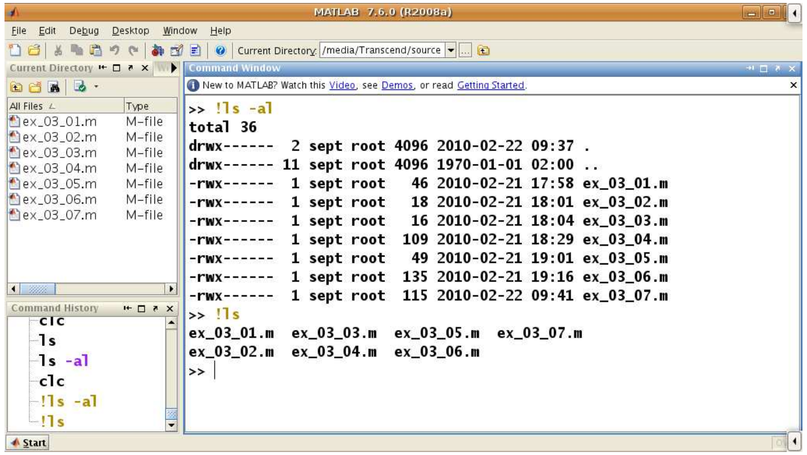
USING SYSTEM COMMANDS

Executing operating system commands

Example;

`!time`

USING SYSTEM COMMANDS


MATLAB 7.6.0 (R2008a)
File
Edit
Debug
Desktop
Window
Help
Current Directory: /media/Transcend/source
Current Directory
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

All Files	Type
ex_03_01.m	M-file
ex_03_02.m	M-file
ex_03_03.m	M-file
ex_03_04.m	M-file
ex_03_05.m	M-file
ex_03_06.m	M-file
ex_03_07.m	M-file

```
>> !ls -al
total 36
drwx------  2 sept root 4096 2010-02-22 09:37 .
drwx------ 11 sept root 4096 1970-01-01 02:00 ..
-rwx------  1 sept root   46 2010-02-21 17:58 ex_03_01.m
-rwx------  1 sept root   18 2010-02-21 18:01 ex_03_02.m
-rwx------  1 sept root   16 2010-02-21 18:04 ex_03_03.m
-rwx------  1 sept root  109 2010-02-21 18:29 ex_03_04.m
-rwx------  1 sept root   49 2010-02-21 19:01 ex_03_05.m
-rwx------  1 sept root  135 2010-02-21 19:16 ex_03_06.m
-rwx------  1 sept root  115 2010-02-22 09:41 ex_03_07.m
>> !ls
ex_03_01.m  ex_03_03.m  ex_03_05.m  ex_03_07.m
ex_03_02.m  ex_03_04.m  ex_03_06.m
>> 
```

```
clc
ls
ls -al
clc
!ls -al
!ls
```

fprintf

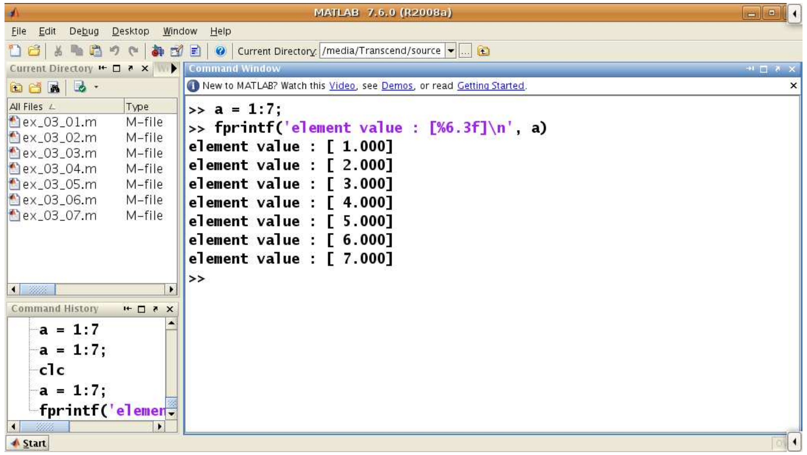
- `fprintf`

- `fprintf(`'*formatstring*'`,` *listofvariables*`)`
- `fprintf(`'*filename*'`,` '*formatstring*'`,` *listofvariables*`)`

Example;

`fprintf( 'myfile', '%f', x )`

fprintf


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /media/Transcend/source
Current Directory
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start

All Files	Type
ex_03_01.m	M-file
ex_03_02.m	M-file
ex_03_03.m	M-file
ex_03_04.m	M-file
ex_03_05.m	M-file
ex_03_06.m	M-file
ex_03_07.m	M-file

```
>> a = 1:7;
>> fprintf('element value : [%6.3f]\n', a)
element value : [ 1.000]
element value : [ 2.000]
element value : [ 3.000]
element value : [ 4.000]
element value : [ 5.000]
element value : [ 6.000]
element value : [ 7.000]
>>
```

```
a = 1:7
a = 1:7;
clc
a = 1:7;
fprintf('elemen
```

GENERAL FILE I/O

- `fopen`
- `fclose`
- `fread`
- `fwrite`
- `fseek`

GOOD PROGRAMMING STYLE

Here are some hints on how to improve your programming style:

- You should make liberal use of comments, both at the beginning of a script to describe briefly what it does and any special methods that may have been used, and throughout the coding to introduce different logical sections.
- The meaning of each variable should be described briefly in a comment when it is initialized. You should describe variables systematically, for example, in alphabetical order.
- Blank lines should be freely used to separate sections of coding (e.g., before and after loop structures).

GOOD PROGRAMMING STYLE

cont'd,

- Coding (i.e., statements) inside structures (fors, ifs, whiles) should be indented (tabulated) a few columns to make them stand out.
- Blank spaces should be used in expressions to make them more readable – for example, on either side of operators and equal signs. However, blanks may be omitted in places in complicated expressions where this may make the logic clearer.

INTRODUCTION TO GRAPHICS

PRESENTING AND VISUALIZING GRAPHICAL DATA

A picture is worth a thousand words.

- 2D plotting
- 3D plotting

SIMPLE 2*D* GRAPHICS

`plot` statement

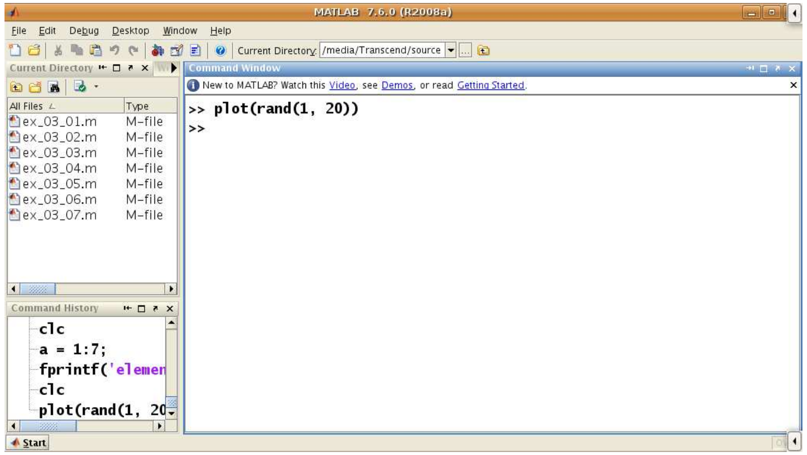
In its sipmplest form `plot` takes a single vector argument.

Example;

```
plot(rand(1, 20))
```

Plots 20 random numbers against the integers 1-20.

SIMPLE 2*D* GRAPHICS



Current Directory: /media/Transcend/source

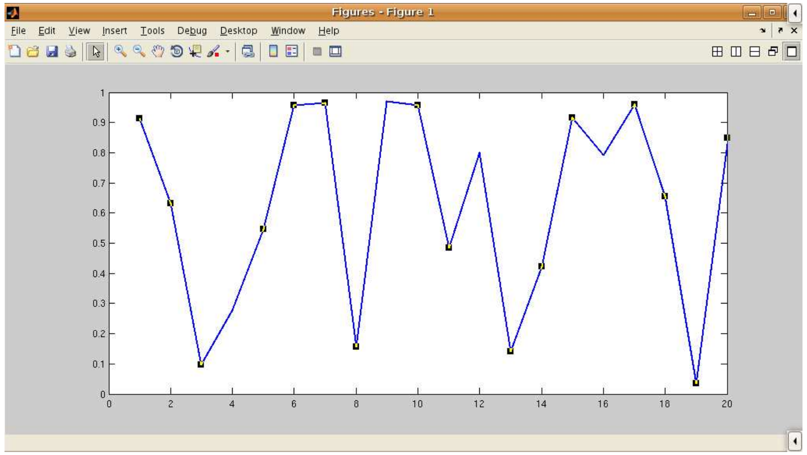
All Files	Type
ex_03_01.m	M-file
ex_03_02.m	M-file
ex_03_03.m	M-file
ex_03_04.m	M-file
ex_03_05.m	M-file
ex_03_06.m	M-file
ex_03_07.m	M-file

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> plot(rand(1, 20))
>>
```

```
clc
a = 1:7;
fprintf('elemen
clc
plot(rand(1, 20
```

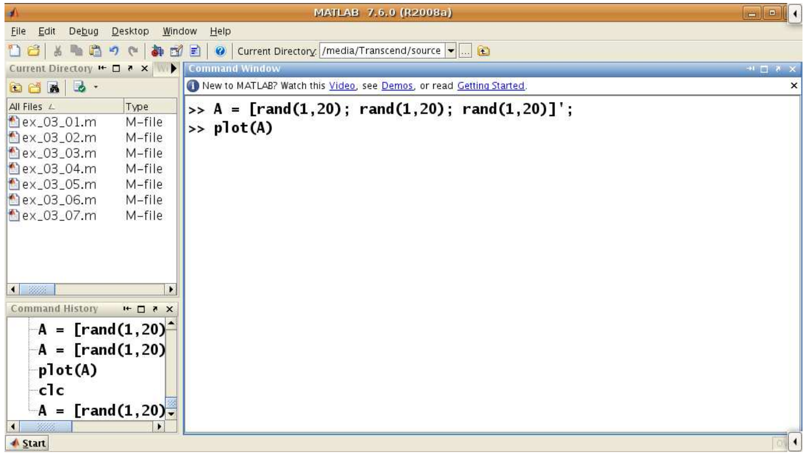
SIMPLE 2*D* GRAPHICS



SIMPLE 2*D* GRAPHICS

- If the argument is a matrix , its columns are plotted against element indexes.
- Axes are automatically scaled and drawn to include the minimum and maximum data points.

SIMPLE 2*D* GRAPHICS


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /media/Transcend/source
Current Directory
Command Window
Command History
Start

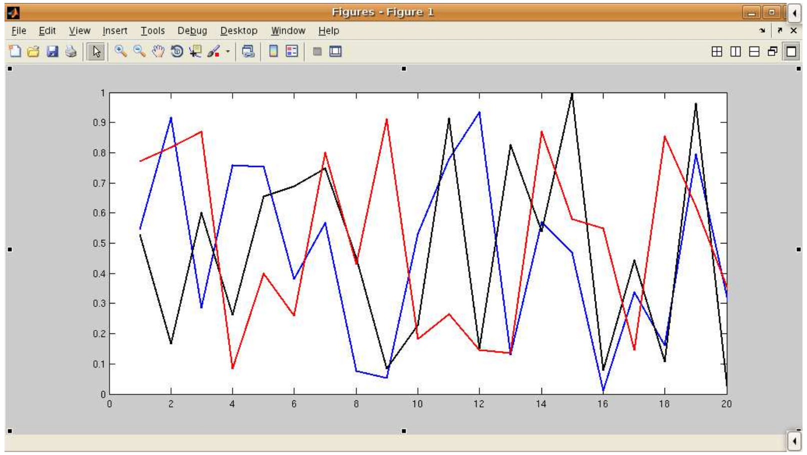
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

All Files	Type
ex_03_01.m	M-file
ex_03_02.m	M-file
ex_03_03.m	M-file
ex_03_04.m	M-file
ex_03_05.m	M-file
ex_03_06.m	M-file
ex_03_07.m	M-file

```
>> A = [rand(1,20); rand(1,20); rand(1,20)]';
>> plot(A)
```

```
A = [rand(1,20)
A = [rand(1,20)
plot(A)
clc
A = [rand(1,20)
```

SIMPLE 2*D* GRAPHICS



SIMPLE 2*D* GRAPHICS

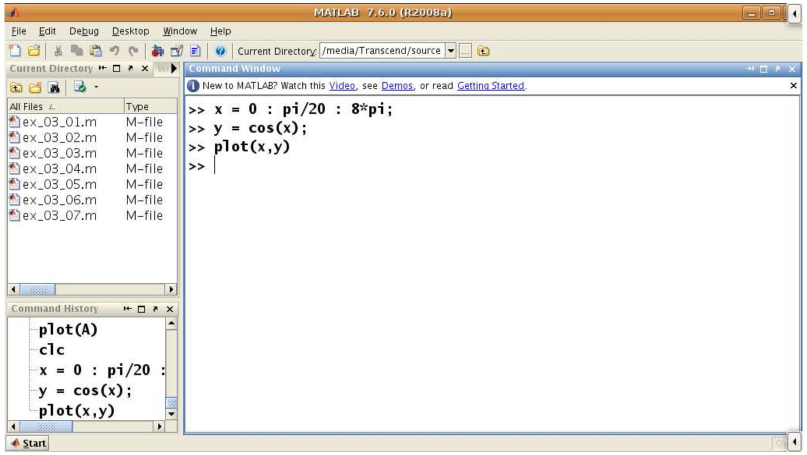
```
plot(x, y)
```

x and y are vectors of the same length. Example;

```
x = 0 : pi/20 : 8*pi;
y = cos(x);
plot(x, y)
```

i th point coordinates are x_i, y_i

SIMPLE 2*D* GRAPHICS


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /media/Transcend/source
Current Directory
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
All Files
Type
Command History
Start

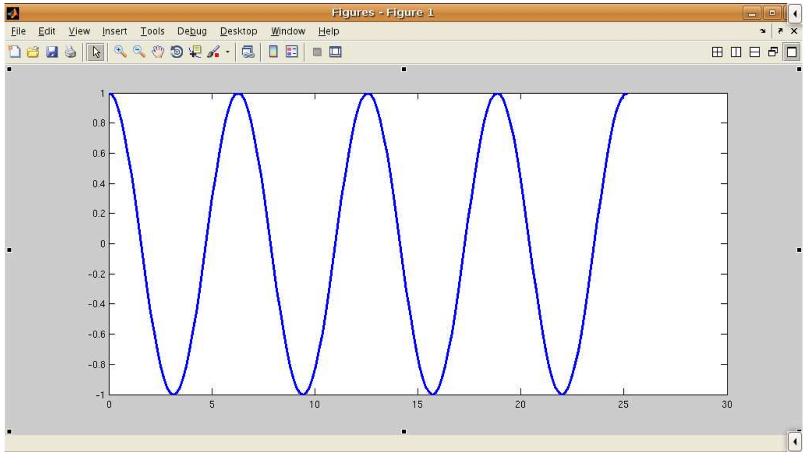
All Files	Type
ex_03_01.m	M-file
ex_03_02.m	M-file
ex_03_03.m	M-file
ex_03_04.m	M-file
ex_03_05.m	M-file
ex_03_06.m	M-file
ex_03_07.m	M-file

```
>> x = 0 : pi/20 : 8*pi;
>> y = cos(x);
>> plot(x,y)
>>
```

Command History:

```
plot(A)
clc
x = 0 : pi/20 :
y = cos(x);
plot(x,y)
```

SIMPLE 2*D* GRAPHICS



DRAWING STRAIGHT LINES

DRAWING STRAIGHT LINES

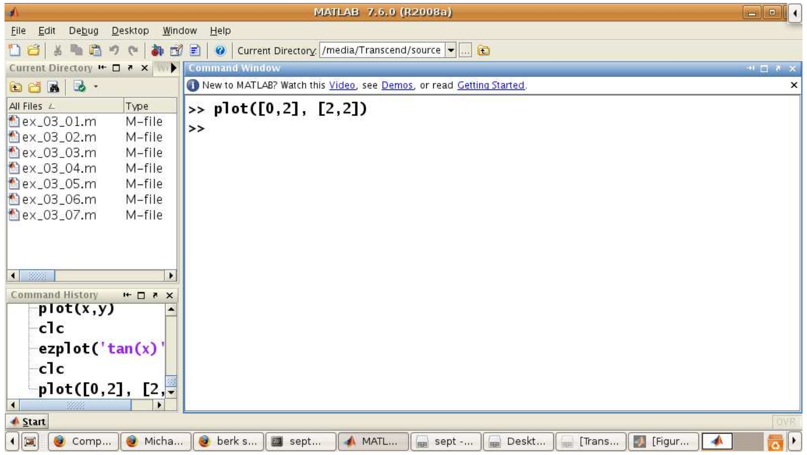
Straight-line graphs are drawn by giving the *x* and *y* coordinates of the end points by two vectors.

```
plot([0 2], [2 2])
```

Matlab has a set of easy-to-use plotting commands.

```
ezplot('tan(x)')
```

SIMPLE 2D GRAPHICS



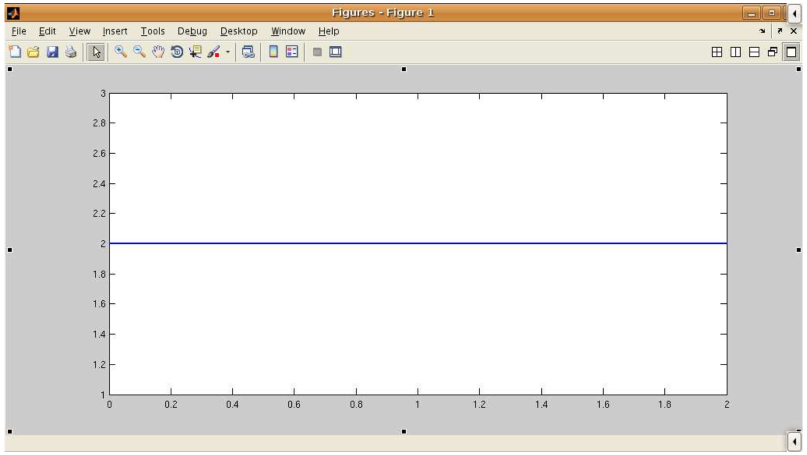
All Files	Type
ex_03_01.m	M-file
ex_03_02.m	M-file
ex_03_03.m	M-file
ex_03_04.m	M-file
ex_03_05.m	M-file
ex_03_06.m	M-file
ex_03_07.m	M-file

```
>> plot([0,2], [2,2])
>>
```

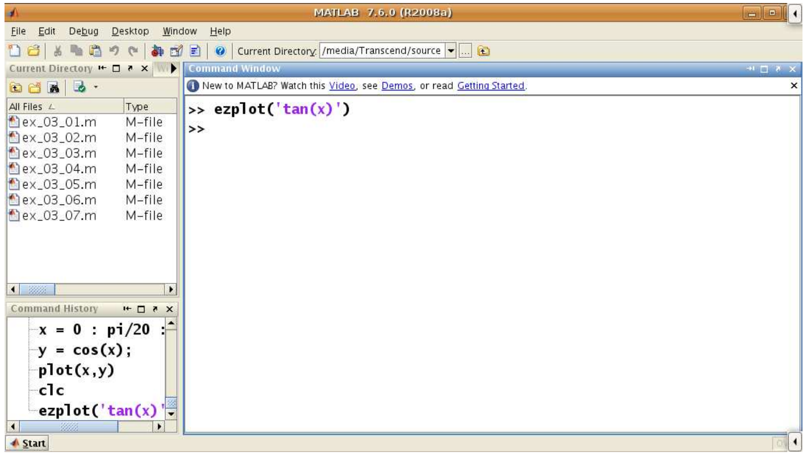
Command History:

```
plot(x,y)
clc
ezplot('tan(x)'
clc
plot([0,2], [2,
```

SIMPLE 2D GRAPHICS



SIMPLE 2D GRAPHICS


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: /media/Transcend/source
Current Directory
Command Window
Command History
Start

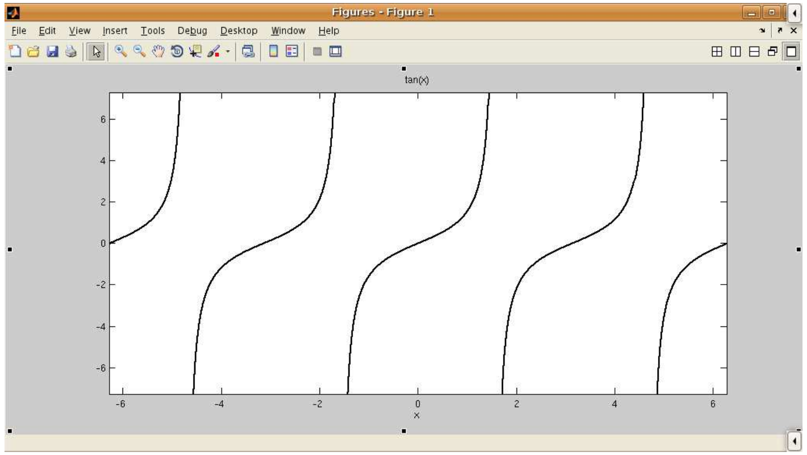
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

All Files	Type
ex_03_01.m	M-file
ex_03_02.m	M-file
ex_03_03.m	M-file
ex_03_04.m	M-file
ex_03_05.m	M-file
ex_03_06.m	M-file
ex_03_07.m	M-file

```
>> ezplot('tan(x)')
>>
```

```
x = 0 : pi/20 :
y = cos(x);
plot(x,y)
clc
ezplot('tan(x)'
```

SIMPLE 2D GRAPHICS



LABEL SETTINGS

Graphs may be labeled with the following statements:

- `gtext('text')` writes a string in the graph window. Text may be placed also with **Tools–Edit Plot** from the figure window.
- `grid` add/removes grid lines.
- `text(x, y, 'text')` writes text at the point specified by *x* and *y*.
- `title('text')` writes the text as a title at the top of the graph.
- `xlabel('horizontal')` labels the *x*-axis.
- `ylabel('vertical')` labels the *y*-axis.

MULTIPLE PLOTS

Multiple plots on the same axis

- Use `hold` to keep the current plot on the axes. Released with either `hold off` or `hold`.
- Use `plot` with multiple arguments.
  plot(x_1, y_1, x_2, y_2)
- Use `plotyy` to have independent *y*-axis on the left and on the right
  plotyy(x, y_1, x, y_2)
- Use the form `plot(x, y)`, where *x* and *y* may be both matrices or one may be a vector and one a matrix.

LINE STYLES, MARKERS AND COLOR

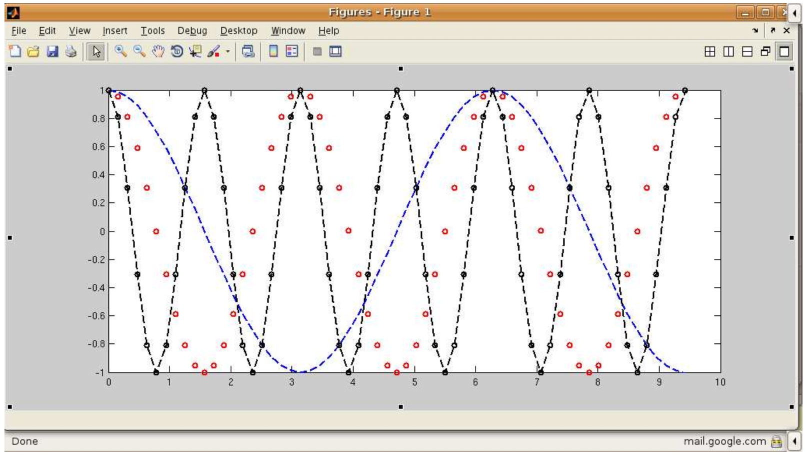
Line style, markers and color for a graph may be selected with a string argument to plot.

Example;

```
x = 0 : pi/20 : 3*pi;
y = cos(x);
plot(x, y, '--')
hold
plot(x, cos(2*x), 'o')
plot(x, cos(4*x), 'om--')
```

Available color symbols **c, m, y, k, r, g, b, w**

MULTIPLE PLOTS



AXIS SETTINGS

Axis limits can be overriden with

`axis([xmin, xmax, ymin, ymax])`

- Sets the scaling on the current plot.
- Use `Inf` or `-Inf` for the autoscaled limit.
- Use `axis auto` to return to the automatic axis scaling.
- `v = axis` returns the current axis scaling in the vector v.
- Use `axis manual` to freeze current scaling, so subsequent plots use the same limits.
- Use `axis equal` to make equal unit length on both axis. The effect is undone with `axis normal`.
- Turn axis labeling and tick marks with `axis off` and `axis on`.

MULTIPLE PLOTS - subplot

To show a number of plots in the same figure window use `subplot` function.

`subplot(m, n, p)`

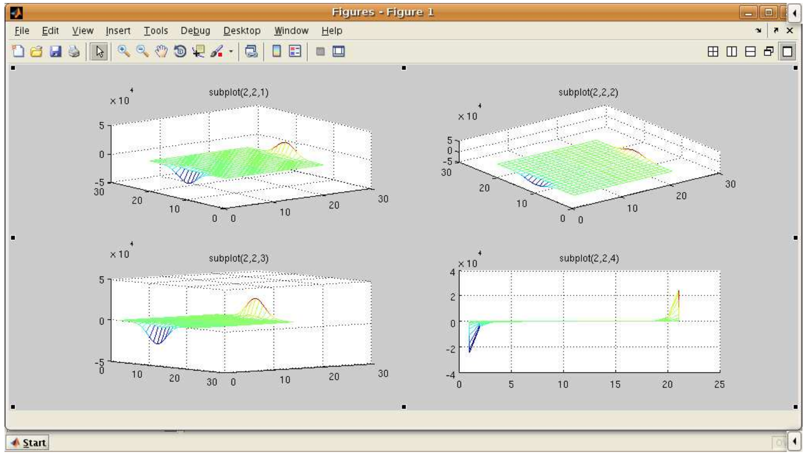
divides the figure window into *mxn* small sets and selects the *p* th set for the current plot.

The command `subplot(1,1,1)` returns to a single set of axes.

MULTIPLE PLOTS - subplot

```
[x, y] = meshgrid(-3:0.3:3);
z = x .* exp(x .^ 2 - y .^ 2);
subplot(2,2,1)
mesh(z),title('subplot(2,2,1)')
subplot(2,2,2)
mesh(z)
view(-37.5,70),title('subplot(2,2,2)')
subplot(2,2,3)
mesh(z)
view(37.5,-10),title('subplot(2,2,3)')
subplot(2,2,4)
mesh(z)
view(0,0),title('subplot(2,2,4)')
```

MULTIPLE PLOTS - subplot



figure

`figure(h)`, creates a new figure window or make the *h* th figure window current.

h is called figure handle

`clf` clears current figure.

`cla` deletes all plots and text from the current axes.

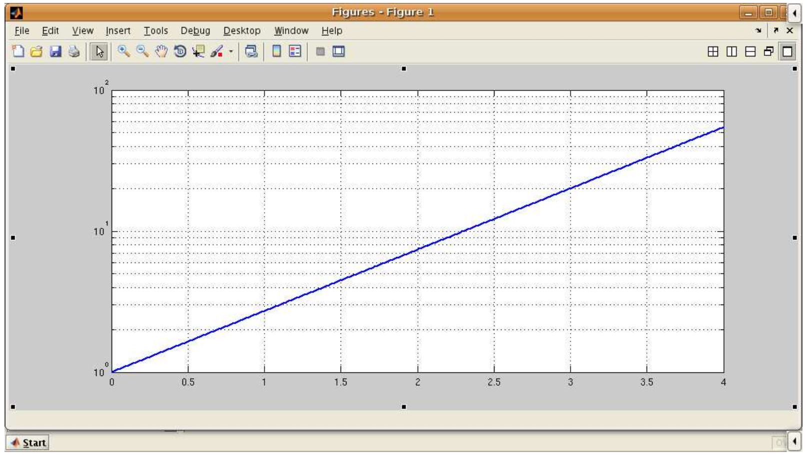
LOGARITHMIC PLOT

- `loglog` plots both axis in logarithmic scale
- `semilogx` plots only x axis in logarithmic scale
- `semilogy` plots only y axis in logarithmic scale

LOGARITHMIC PLOT

```
x = 0:0.01:4;
semilogy(x, exp(x)), grid
```

LOGARITHMIC PLOT



POLAR PLOT

The point (x, y) in cartesian coordinates represented by the point (θ, r) in *polar* coordinates

$x = rcos(\theta)$

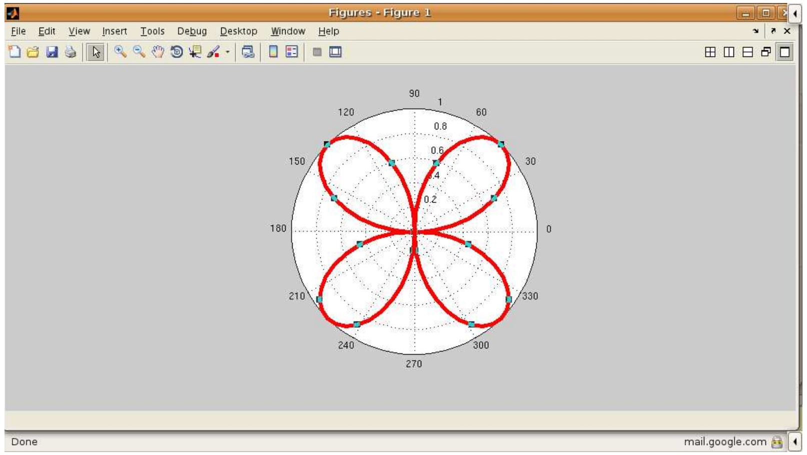
$y = rsin(\theta)$

`polar(theta, r)`

POLAR PLOT

```
x = 0 : pi/40 : 2 * pi;
polar(x, sin(2*x)), grid
```

POLAR PLOT



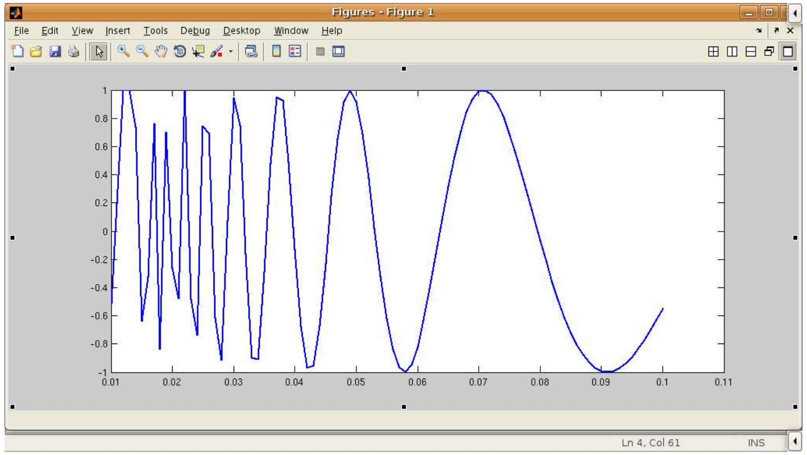
fplot

`fplot` plots the function given with the string.

- Use `fplot` if the function changes rapidly.
- Reduce the increment when using the `plot` command.

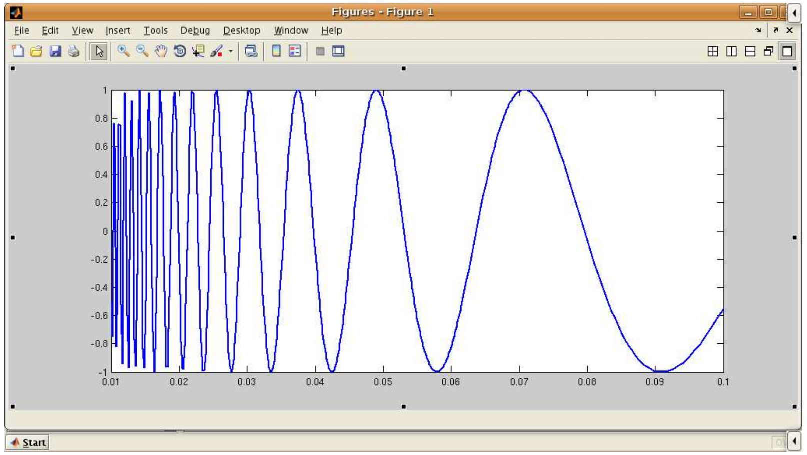
fplot

```
x = 0.01:0.001:0.1;
plot(x, sin(1./x))
```



fplot

```
fplot('sin(1/x)', [0.01 0.1])
```



SIMPLE 3D GRAPHICS

The `plot3` function

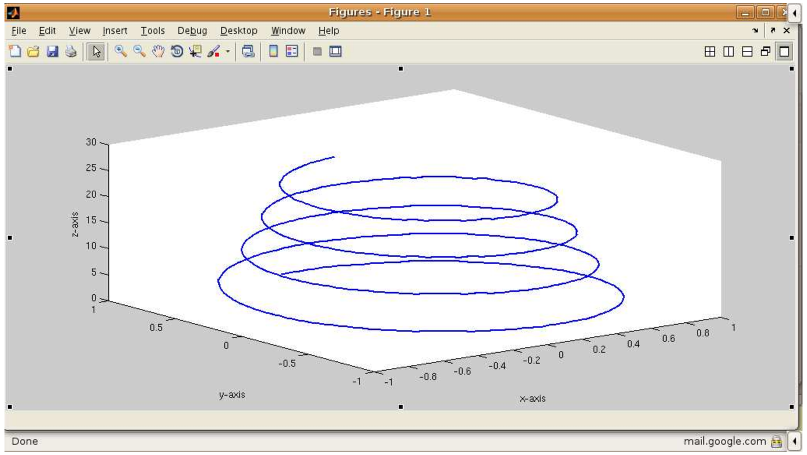
Usage;

```
plot3(x, y, z)
```

Example;

```
t = 0: pi/40 : 8*pi
plot3(exp(-0.02*t).*sin(t), exp(-0.02*t).*cos(t),t),
xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis')
```

SIMPLE 3D GRAPHICS



`comet3`

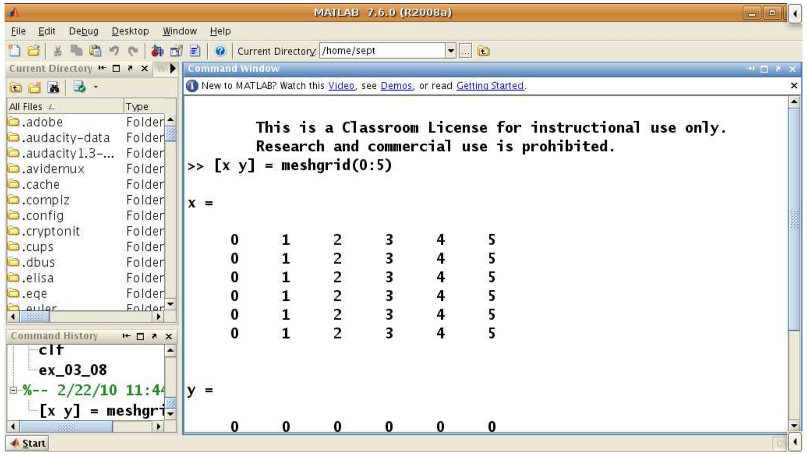
`comet3` is similar to `plot3`

except it draws an animated graphic.

MESH SURFACES

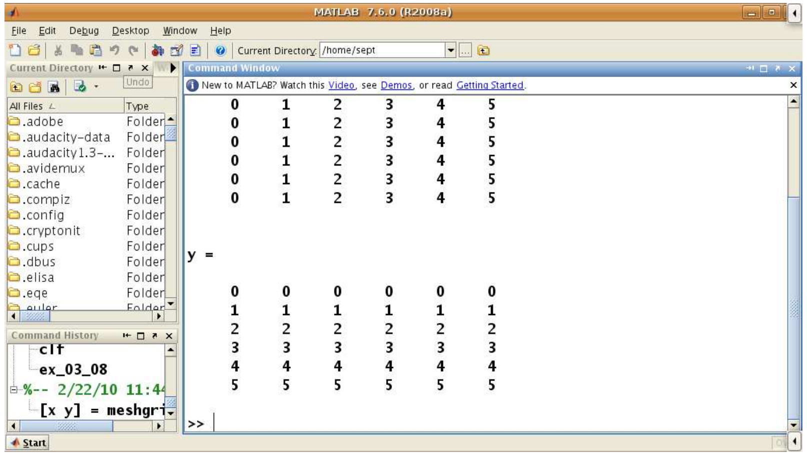
```
meshgrid

mesh(x, y, z)

surf(x, y, z)

contour

[x, y] = meshgrid(0:5)
```

MESH SURFACES



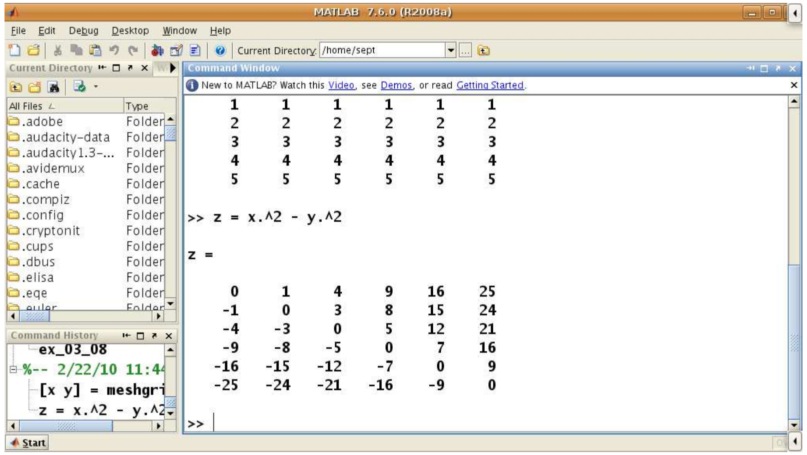
```
         This is a Classroom License for instructional use only.
         Research and commercial use is prohibited.
>> [x y] = meshgrid(0:5)

x =

     0     1     2     3     4     5
     0     1     2     3     4     5
     0     1     2     3     4     5
     0     1     2     3     4     5
     0     1     2     3     4     5
     0     1     2     3     4     5


y =

     0     0     0     0     0     0
```

MESH SURFACES



```
     0     1     2     3     4     5
     0     1     2     3     4     5
     0     1     2     3     4     5
     0     1     2     3     4     5
     0     1     2     3     4     5
     0     1     2     3     4     5


y =

     0     0     0     0     0     0
     1     1     1     1     1     1
     2     2     2     2     2     2
     3     3     3     3     3     3
     4     4     4     4     4     4
     5     5     5     5     5     5

>>
```

MESH SURFACES

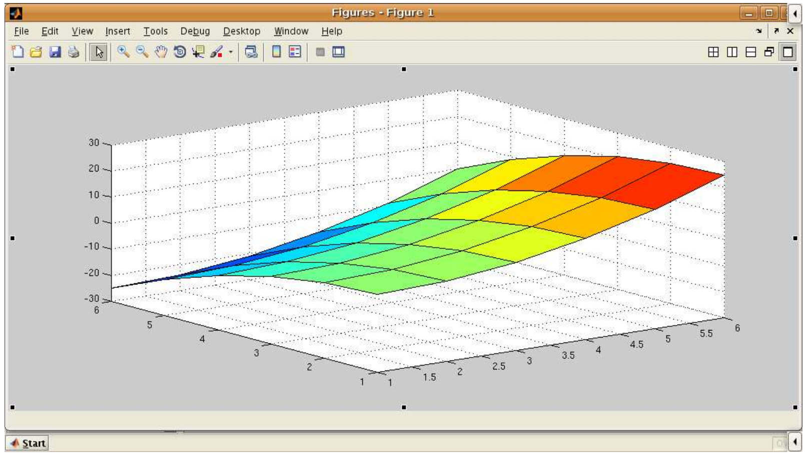


```
     1     1     1     1     1     1
     2     2     2     2     2     2
     3     3     3     3     3     3
     4     4     4     4     4     4
     5     5     5     5     5     5

>> z = x.^2 - y.^2

z =

     0     1     4     9    16    25
    -1     0     3     8    15    24
    -4    -3     0     5    12    21
    -9    -8    -5     0     7    16
   -16   -15   -12    -7     0     9
   -25   -24   -21   -16    -9     0

>>
```

MESH SURFACES

```
[x y] = meshgrid(0:5);
z = x.^2 - y.^2;
surf(z)
```

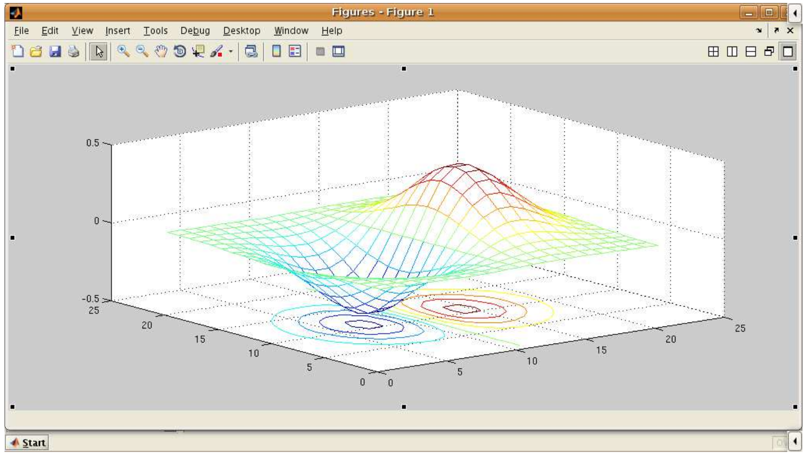
MESH SURFACES



MESH SURFACES

```
[x y] = meshgrid(-2:.2:2);
z = x .* exp(-x.^2 -y.^2);
meshc(z)
```

MESH SURFACES



MATRIX VISUALIZATION

```
mesh(A)
spy(A)
```

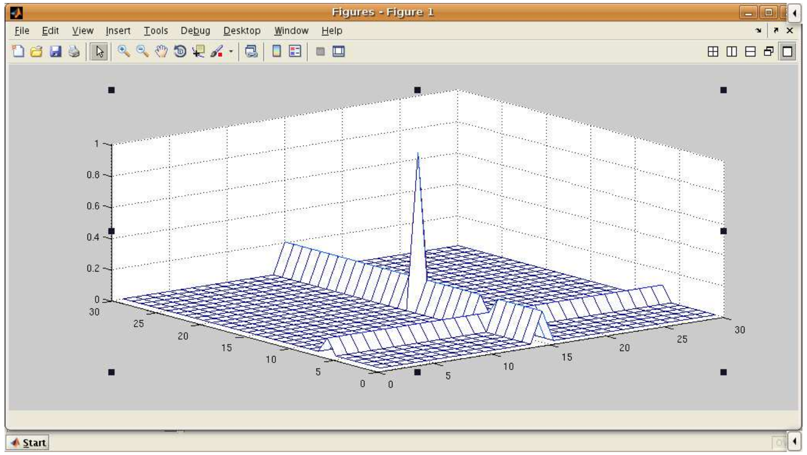
A complete list of graphics functions

MATLAB help

MATRIX VISUALIZATION

```
a = zeros(30,30);
a(:,15) = 0.2*ones(30,1);
a(7,:) = 0.1*ones(1,30);
a(15,15) = 1;
mesh(a)
```

MATRIX VISUALIZATION



References

References for Week 3

1. Brian Hahn, Daniel T.Valentine, Essential Matlab for Engineers and Scientists, Elsevier, 2010.
2. Misza Kalechman, Practical Matlab Basics for Engineers, CRC Press, 2009.