LAB 3

Spring 2011, BESE- 15 A&B

Fundamental Concepts

Objective

The aim of this introductory lab is to introduce you to the basic functions in the Matlab and Numerical Methods with Matlab toolbox. By the end of today's lab, you should be able to understand the Arrays operations, Element by element on Arrays.

Submission Requirements

You are expected to complete the assigned tasks within the lab session and show them to the lab engineer/instructor. Some of these tasks are for practice purposes only while others (marked as '*Exercise*' or '*Question*') have to be answered in the form of a lab report that you need to prepare. Following guidelines will be helpful to you in carrying out the tasks and preparing the lab report.

Guidelines

- In the exercises, you have to put the output in your Lab report. You may add screen print to the report by using the 'Print Screen' command on your keyboard to get a snapshot of the displayed output. This point will become clear to you once you actually carry out the assigned tasks**.**

- Name your reports using the following convention:

  Lab#_Rank_YourFullName

  - '#' replaces the lab number
  - '*Rank*' replaces Maj/Capt/TC/NC/PC
  - '*YourFullName*' replaces your complete name.
- You need to submit the report even if you have demonstrated the exercises to the lab engineer/instructor or shown them the lab report during the lab session.

Addition and Subtraction

- The operations addition and subtraction can be used with array of identical (Same order) size.
- The sum or difference of two array is obtained by adding or subtracting their corresponding elements.
- When a scalar is added to or subtracted from an array the number is add or subtracted from all the elements of the array.

Practice 1

```
>> VectA=[8  5  4];

>> VectB=[10  2  7];

>> VectC= VectA+VectB
VectC= 18  7  11
>> A=[5  -3  8; 9  2  10];

>> B=[10  7  4;  -11  15  1];

>> C=A+B

C=      15      4      12
        -2      17     11

>> D=A-B

D=      -5      -10    4
        20      -13    9


>> C-8

ans=    7       -4     4
        -10     9      3
```

Array Multiplication

- Multiplication operation is executed by Matlab according to the rules of linear algebra.
- If A and B are two matrices, the operation A*B can be carried out only if the number of columns in matrix A is equal to the number of rows in matrix B.
- If A and B are both nxn then A*B not equal to B*A
- Power operation can only be executed with a square matrix
- Two vectors can be multiplied if both have the same number of elements, and one is a row vector and the other is column vector.
- An array is multiplied by a number then each element in the array is multiplied by the number

Practice 2

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

>> B=[6 1; 2 5; 7 3];

>> C= A*B
C=      28  27
        65  49
        98  32
        84  38

>> D=B*A;
???Error using ➔*
Inner matrix dimensions must agree.

>> F=[1 3; 5 7];

>> G=[4 2; 1 6];

>> F*G

ans=    7       20
        27      52


>> G*F

ans=    14      26
        31      45

>> A=[2 5 7 0; 10 1 3 4; 6 2 11 5];

>>b=3;

>>C=b*A
C=      6       15      21      0
        30      3       9       12
        18      6       33      15
```

Array Division

- The division operation can be done with the help of the identity matrix and the inverse operation.
- eye() is used to create Identity matrix, if a matrix A is square, it can be multiplied by the identity matrix I, from the left or from the right. (A*I = I*A = I)
- inv() is used to find inverse of matrix. The matrix B is the inverse of the matrix A if when the two matrices are multiplied the product is the identity matrix.(B*A = A*B = I, $B=A^{-1}$)
- det() is used to find determinant of matrix. Not every matrix has an inverse. A matrix has an inverse only if it is square and its determinant is not equal to zero.

Practice 3

```
>> A=[2 1 4; 4 1 8; 2 -1 3];

>>det(A);                          %|A| is not equal to zero

>> B=inv(A);                        %Use the inv() to find the inverse of A.

>> A*B                             %Multiplication of A and B gives the identity matrix.
ans=    1       0       0
        0       1       0
        0       0       1

>> A*A^-1                          %Use of power of -1 to find the inverse of A
ans=    1       0       0
        0       1       0
        0       0       1
```

Element by element operation

- When operation applied on each element is said to be element by element. Addition and subtraction are by definition already element by element operations but multiplication, division and exponential are done by using . (dot) operator.
- This operation is applied on two arrays then must be the same size of both arrays but one of them can be a scalar.
- If two vectors a and b are: a=[a_1 a_2 a_3] and b=[b_1 b_2 b_3] then element by element of the two vectors as follows

a.*b = [a_1*b_1 a_2*b_2 a_3*b_3] a./b = [a_1/b_1 a_2/b_2 a_3/b_3] a.^b = [a_1^b_1 a_2^b_2 a_3^b_3]

Symbol	**Description**	**Symbol**	**Description**
.*	Multiplication	./	Right division
.^	Exponential	.\	Left division

Practice 4

```
>> A=[2 6 3; 5 8 4];

>>B=[1 4 10; 3 2 7];

>> A.*B                            %Element by element multiplication of array.
ans=    2       24      30
        15      16      28

>> C=A./B;                         % Element by element division of array

>> B.^3;                           %Element by element exponentional of array B.
```

Practice 5

Function $y=(x^2+1)^3 x^3$, calculate the value of y for the following values of x

-2.5 -2 -1.5 -1 -0.5 1 1.5 2 2.5 3

Built in function for analyzing Arrays

Function	**Description**	**Example**
mean(A)	If A is a vector, returns the mean value of the elements of the vector	>>A=[5 9 2 4]; >>mean(A) ans=5
C=max(A)	If A is a vector, C is the largest element in A. If A is a matrix, C is a row vector containg the largest element of each column of A.	>>A=[5 9 2 4 11 6 7 11 0 1]; >>C=max(A) ans=11
[d,n]=max(A)	If A is a vector, d is the largest element in A, n is the position of the element	>>[d,n]=max(A) d=11 n=5
C=min(A)	The same as max(A), but for the smallest element.	>>A=[5 9 2 4]; >>C=min(A) C=2
[d,n]=min(A)	The same as [d,n]=max(A), but for the smallest element.	
C=sum(A)	If A is a vector, returns the sum of the element of the vector	>>A=[5 9 2 4]; >>C=sum(A) C=2
sort(A)	If A is a vector, arranges the element of the vector in ascending order	>>A=[5 9 2 4]; >>sort(A) ans=2 4 5 9
C=median(A)	If A is a vector, returns the median value of the elements of the vector	>>A=[5 9 2 4]; >>C=median(A) C=4.5000
C=std(A)	If A is a vector, returns the standard deviation of the elements of the vector.	>>A=[5 9 2 4]; >>C=std(A) C=2.9439
rand	Generates a single random number between 0 and 1	>>rand Ans=0.2311
rand(1,n)	Generates an n elements row vector of random numbers between 0 and 1	>>a=rand(1,4);
rand(n)	Generates an n x n matrix with random numbers between 0 and 1	>>b=rand(3);
rand(m,n)	Generates an m x n matrix with random numbers between 0 and 1	>>c=rand(2,4)
Randperm(n)	Genereates a row vector with n elements that are random permutation of integers 1 to n	>>randperm(8) Ans= 8 2 7 4 3 6 5 1

Exercise 1

Use Matlab to show that the sum of the infinite series $sum(1/n^2)$ converges to $pi^2/6$. Do it by computing the sum for

a) n=100
b) n=1000
c) n=10000
d) n=100000

Exercise 2

Solve the following system of foure linear equations

$$5x+4y-2z+6w=4$$

$$3x+6y+6z+4.5w=13.5$$

$$6x+12y-2z+16w=20$$

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

Graphics

MATLAB produce two and three dimensional plots of curves and surface

Two Dimensional Plots

- Plots are very useful tool for presenting information.
- Plot title, Legend, X axis label, Y axis label, Text label and Markers.
- Graph is shown in figure window.

Plot Command

The plot command is used to create two dim plots.

The simplest from of the command is plot(x,y), Where x (horizental axis) and y (vertical axis) are vector.

Both vectors must have the same number of elements.

The curve is constructed of straight line segments that connect the points.

The figure that is created has axes with linear scale and default range.

Practice 5

```
>> x=[1  2  3  5  7  7.5  8  10];

>>y=[2  6.5  7  7  5.5  4  6  8];

>>plot(x,y)
```

plot(x,y,'line specifiers','PropertyName', PropertyValue)

Line Specifiers:

- Line specifiers are optional and can be used to define the style and coloar of the line and the type of markers.
- Within the string the specifiers can be typed in any order.
- Specifier is optional i.e. None, one, two, or all the three can be included in command.

Line Style	Specifier	Line Color	Specifier	Marker Type	Specifier
solid(default)	-	red	r	plus sign	+
dashed	--	green	g	circle	o
dotted	:	blue	b	asterisk	*
dash-dot	-.	cyan	c	point	.
		megenta	m	square	s
		yellow	y	diamond	d
		black	k		
		white	w		

Property Name and Property Value:
Properties are optional and can be used to specify the thickness of the line, the size of marker, and color of marker's edge line and fill.
The property name is typed as a string, followed by comma and value for property.

Property Name	Description	Possible Property Values
LineWidth or linewidth	Specifies the width of the line	A number in units of points (default 0.5).
MarkerSize or markersize	Specifies the size of marker	A number in units of points.
MarkerEdgeColor or markeredgecolor	Specifies the color of the marker, or the color of the edge line for filled markers.	Color specifiers from the table above, typed as a string
MarkerFaceColor or markerfacecolor	Specifies the color of the filling for filled markers.	Color specifiers from the table above, typed as a string

Practice 6

```
>>plot(x,y,'r');
>>plot(x,y,'--y');
>>plot(x,y,'*');
>>plot(x,y,'g:d');
>>plot(x,y,'-mo','LineWidth',2,'markersize',12,'MarkerEdgeColor','g','markerfacecolor','y');
```

```
>>yr=[1988:1:1994];                    %Sales Years
>>sle=[8 12 20 22 18 24 27];           %Sales in millions
>>plot(yr,sle,'—r*','linewidth',2,'markersize',12);
```

Plot of function
Plot of given functgion can be done in Matlab by using the plot or the fplot commands.

Using plot: Plot a function y=f(x) with plot command, user first needs to create a vector of values of x for the domain that function will be plotted.
Then a vector y is created with the corresponding values of f(x) by using element by element calculation
Plot the x and y vectors by using plot command.

Using fplot: Plot a function y=f(x) with fplot command between specified limits.

fplot('function',limits, n, line specifiers)

function can be typed directly as a string and The function can not inclue prefiouly defined variables.
limits is a vector with two element that specify the domain of x [xmin,xmax] or [xmin,xmax,ymin,ymax].
n is the number of points between interval.
Line specifier are the same as in plot command.

Practice 7

Examples: $y=3.5^{-0.5x}\cos(6x)$ for -2<=x<=4 and y=cos(x), z=cos(x)2 for 0<=x<=pi

```
>>x=[-2:0.01:4];                      %Create vector x with domain of function
>>y=3.5.^(-0.5*x).*cos(6*x);          %Create vector y with funtion value at each x
>>plot(x,y);
```

%Two or more graphs can be created in the same plot by typing pairs of vectros.

```
>>x=0:0.1:pi;                         % specifies the domain
>>y=cos(x);                           % define 1st function
>>z=cos(x).^2;                        % define 2nd function
>>plot(x,y,x,z,'o')                   % display graph. (x, y 1st fun , x, y 2nd fun )
```

Examples: plot of function y=3x3-26x+10, and its first and second derivatives for -2<=x<=4

```
>>x=[-2:0.01:4];
>>y=3*x.^3-26*x+6;
>>yd=9*x.^2-26;
>>ydd=18*x;
>>plot(x,y,'-b',x,yd,'—r',x,ydd,':k');

>>fplot('x^2+4*sin(2*x)-1',[-3 3]);
>>fplot('tanh',[-2,2])                 % plots y=tanh(x) over [-2,2]
```

Using the hold on, hold off commands

Hold on command keeps the figure window with first plot open, including the axis properties and formatting if any was done.
Additional graphs can be added with plot command that are typed next.
Hold off command stops this process.

Practice 8

Examples: plot of function y=3x3-26x+10, and its first and second derivatives for -2<=x<=4

```
>>x=[-2:0.01:4];
>>y=3*x.^3-26*x+6;
>>yd=9*x.^2-26;
>>ydd=18*x;
>>plot(x,y,'-b');
>>hold on
>>plot(x,yd,'—r');
>>plot(x,ydd,':k');
>>hold off
```

Using the line command

With the line command additional graphs (lines) can be added to a plot that already exists.

line(x,y,'PropertyName',PropertyValue)

The line is almost same as plot except the line specifier, but the line style, color and marker can be specified as property name and property value features.

Practice 9

Examples: plot of function y=3x3-26x+10, and its first and second derivatives for -2<=x<=4

```
>>x=[-2:0.01:4];
>>y=3*x.^3-26*x+6;
>>yd=9*x.^2-26;
>>ydd=18*x;
>>plot(x,y,'LineStyle','-','color','b');
>>line(x,yd,'LineStyle','--','color','r');
>>line(x,ydd,'LineStyle',':','color','k');
%Example  Plot x and y on graph using linspace
>>x=linspace(0,1);                    % Row vector with 100 points
>>y=x.^n.*exp(x);                     % Gives row vector
>>plot(x,y,'o')                       %or     >>plot(x, x.^n.*exp(x))
%Example Plot ellipse c(t)=(2cos(t),3sin(t)), where 0<=t<=2pi
>>t=0:0.2:2*pi;
>>plot(2*cos(t),3*sin(t));
```

Formatting a Plot

The plot needs to be formatted to have a specific look and to display information in addition to the graph itself i.e. axis labels, plot title, legend, grid, range of custom axis, and text labels. Formatting can be done by using Commands and Editior

Formatting plot using Commands
It is useful when a plot command is a part of a computer program.
Formatting commands are entered after the plot or fplot commands.

xlabel('text as string')	%placed next to the x-axis
ylabel('text as string')	%placed next to the y-axis
title('text as string')	%placed at the top of figure as title
text(x,y,'text as string')	%placed in the figure such that first character is at point x,y
gtext('text as string')	%placed in figure at position specified by the user with mouse

The legend command
The leged shows a sample of line type of each graph that is plotted, and place a label, specified by the user.

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

where pos is an optional number that specifies where the figure placed.

pos=-1	places the ouside the axes boundries on the right side
pos=0	places the inside the axes boundries in a location that less interfers with graph
pos=1	places the upper-right corner of the plot(default)
pos=2	places the upper-left corner of the plot
pos=3	places the lower-left corner of the plot
pos=4	places the lower-right corner of the plot.

The axis command
Matlab creates axes with min and max limits values, but axis command can be used to change the range and the appearance of the axes.

axis([xmin,xmax])	%Sets the limits of the x axis
axis([xmin,xmax,ymin,ymax])	%Sets the limits of both x and y axes
axis equal	%Sets the same scale of both axes

The grid command

grid on	%adds grid lines to the plot
grid off	%removes grid lines from the plot