**Aim : To study various commands of MATLAB**

**Software required : Matlab 7.0**

**Theory:**

**MATLAB FUNDAMENTALS**

MATLAB is a numeric computation software for engineering and scientific calculations. The name MATLAB stands for MATRIX LABORATORY.

MATLAB is primarily a tool for matrix computations. It was developed by John Little and Cleve Moler of MathWorks.

MATLAB is a high-level language whose basic data type is a matrix that does not require dimensioning. There is no compilation and linking as is done in high-level languages, such as C or FORTRAN.

MATLAB has a rich set of plotting capabilities. The graphics are integrated in MATLAB. Since MATLAB is also a programming environment, a user can extend the functional capabilities of MATLAB by writing new modules.

MATLAB has a large collection of toolboxes in a variety of domains. Some examples of MATLAB toolboxes are control system, signal processing, neural network, image processing, and system identification. The toolboxes consist of functions that can be used to perform computations in a specific domain.

**BASIC MATLAB COMMANDS**

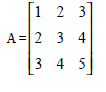
1. **Exit command:-** Exit is used to quit MATLAB.
2. **Help command**:- The help command followed by a function name is used to obtain information on a specific MATLAB function
3. **Length :-**

Matrix formation:-The basic data object in MATLAB is a matrix with real or complex elements. Scalars are thought of as a 1-by-1 matrix. Vectors are considered as matrices with a row or column.

MATLAB statements are normally of the form:

*variable = expression*

A matrix



Can be obtained by typing A = [1 2 3; 2 3 4; 3 4 5];

Expressions typed by the user are interpreted and immediately evaluated by the MATLAB system. If a MATLAB statement ends with a semicolon, MATLAB evaluates the statement but suppresses the display of the results.

The row vector B can be turned into a column vector by **transposition** which is obtained by typing

B=[2 3 4 5 6];

C = B’

1. **Casesen off**

MATLAB is case sensitive in naming variables, commands and functions.Thus b and B are not the same variable. If you do not want MATLAB to be case sensitive, you can use this command.

1. **Size** : It is used to obtain the size of a specific variable.

size(A)

The result will be a row vector with two entries. The first is the number of rows in A, the second the number of columns in A.

1. **% (**Comments). Everything appearing after % command is not executed.
2. **Length:** Length of a matrix(max(size))
3. **Clear:** Clears the variables or functions from workspace

**(9) Clc :** Clears the command window during a work session

**(10) Matrix operations**

The basic matrix operations are addition(+), subtraction(-),and multiplication (\*) of matrices. In addition to the above basic operations,MATLAB has two forms of matrix division: the left inverse operator \ or the right inverse operator /.

Matrices of the same dimension may be subtracted or added. Thus if E and F are entered in MATLAB as

E = [7 2 3; 4 3 6; 8 1 5];

F = [1 4 2; 6 7 5; 1 9 1];

We can perform following operations

G = E - F

H = E + F

I = E\*F

A scalar (1-by-1 matrix) may be added to or subtracted from a matrix.

J = H + 1

Any matrix can be multiplied by a scalar. For example,

2\*I

Matrix division can either be the left division operator \ or the right division operator /.The right division a/b, for instance, is algebraically equivalent to *a/b* while the left division a\b is algebraically equivalent to *b/a*

.

**(11) Ones(n,m)** Produces n-by-m matrix with all the elements being unity.

**(12) Eye(n)** gives n-by-n identity matrix

**(13) Zeros(n,m)** Produces n-by-m matrix of zeros

**(14) Array operations**

Array operations refer to element-by-element arithmetic operations. Preceding the linear algebraic matrix operations, \* / \ ‘ , by a period (**.**) indicates an array or element-by-element operation.

If A1 and B1 are matrices of the same dimensions, then A1.\*B1 denotes an array whose elements are products of the corresponding elements of A1 and B1.

Thus, if

A = [2 7 6 ; 8 9 3]; B = [6 4 3;2 3 4]; then we can perform

C = A.\*B

D = A./B

E = A.\B

**(15) Complex nunbers:-**

MATLAB allows operations involving complex numbers. Complex numbers are entered using function i or j. For example, a number *z* 2 *j*2 may be entered in MATLAB as z = 2+2\*i

**(16) Colon symbol (:):-**

The colon symbol (**:**) is one of the most important operators in MATLAB. It can be used

(1) to create vectors and matrices,

(2) to specify sub-matrices and vectors

(3) to perform iterations. The statement

t1 = 1:6

will generate a row vector containing the numbers from 1 to 6 with unit increment..MATLAB produces the result

t1 =1 2 3 4 5 6

Non-unity, positive or negative increments, may be specified.

For example, the statement

t2 = 3:-0.5:1

will result in t2 = 3.0000 2.5000 2.0000 1.5000 1.0000

The statement

t3 = [(0:2:10);(5:-0.2:4)]

will result in a 2-by-4 matrix

t3 =

0 2.0000 4.0000 6.0000 8.0000 10.0000

5.0000 4.8000 4.6000 4.4000 4.2000 4.0000

.

**(17) Linspace:-** generates linearly evenly spaced vectors, while **logspace** generates logarithmically evenly spaced vectors. The usage of these functions is of the form:

linspace(i\_value, f\_value, np)

logspace(i\_value, f\_value, np)

where

i\_value is the initial value

f\_value is the final value

np is the total number of elements in the vector.

(18) **Plotting Functions:**

MATLAB has built-in functions that allow one to generate bar charts, x-y, polar,and bar charts. MATLAB also allows one to give titles to graphs, label the x- and y-axes, and add a grid to graphs. In addition, there are commands for controlling the screen and scaling.

Some graph functions are

* **histogram** gives histogram bar graph
* **plot** plots along x-y axis

plot(x,y)

plots the elements of x (x-axis) versus the elements of y (y-axis).

* **plot(a1, b1, ’\*’, a2, b2, ’+’)**
* **polar** performs polar plot

**polar**(theta, rho)

where,

theta and rho are vectors, with the theta being an angle in radians and rho being the magnitude.

* **loglog** does log versus log plot
* **semilogx** does semilog x-y plot (x-axis logarithmic)
* **semilogy** does semilog x-y plot (y-axis logarithmic)
* **text** positions text at a specified location on graph.general format of text is

text(x, y, ’text’)

* **title** used to put title on graph

title('Response of an RC circuit')

* **xlabel** labels x-axis
* **ylabel** labels y-axis
* **grid** adds grid to a plot
* **subplot:**

The graph window can be partitioned into multiple windows. The **subplot** command allows one to split the graph window into two subdivisions or four subdivisions.

**subplot(i j k),plot(x,y)**

The digits *i* and *j* specify that the graph window is to be split into an *i-*by- *j* grid of smaller windows. The digit *k* specifies the *k th* window for the current plot.

**(19 ) Control ststements**

* **FOR LOOPS:**

“**FOR**” loops allow a statement or group of statements to be repeated a fixed number of times. The general form of a for loop is

for index = expression

statement group X

end

* **IF STATEMENTS:**

**IF** statements use relational or logical operations to determine what steps to perform in the solution of a problem.

there are several variations of the IF statement:

* + simple if statement
  + nested if statement
  + if-else statement
  + if-elseif statement
  + if-elseif-else statement.

The general form of the **simple if statement** is

if logical expression 1

statement group 1

end

* **WHILE LOOP**

A **WHILE** loop allows one to repeat a group of statements as long as a specified condition is satisfied. The general form of the WHILE loop is

while expression 1

statement group 1

end

statement group 2

**(20) Relational operator**

< less than

<= less than or equal

> greater than

>= greater than or equal

== equal

~= not equal

**(21) Logical Operators**

& and

! or

~ not

**(21) Input/Output commands**

MATLAB has commands for inputting and outputting information in the command window

1. ***Disp***

The **disp** command displays a matrix without printing its name. It can also be used to display a text string. The general form of the disp command is

disp(x)

disp(‘text string’)

1. ***fprintf***

**T**he **fprintf** can be used to print both text and matrix values. The format for printing the matrix can be specified, and line feed can also be specified. The general form of this command is

fprintf(‘text with format specification’, matrices)

Other **format specifiers** are

% d – unsigned decimal number

%f - floating point

%g - signed decimal number in either %e or %f format,

1. ***Input***

The **input** command displays a user-written text string on the screen, waits for an input from the keyboard, and assigns the number entered on the keyboard as the value of a variable. For example, if one types the command

r = input(‘Please enter the four resistor values’);

when the above command is executed, the text string ‘Please, enter the four resistor values’ will be displayed on the terminal screen. The user can then type an expression such as [10 15 30 25];

***(22) Break:*** The **break** command may be used to terminate the execution of ***for*** and ***while*** loops.

**(23)**  **M-files**

Normally, when single line commands are entered, MATLAB processes the commands immediately and displays the results. MATLAB is also capable of processing a sequence of commands that are stored in files with extension m.MATLAB files with extension m are called m-files.

**(24) Function Files**

Function files are m-files that are used to create new MATLAB functions.Variables defined and manipulated inside a function file are local to the function,and they do not operate globally on the workspace. However, arguments may be passed into and out of a function file. The general form of a function file is

function variable(s) = function\_name (arguments)

**Result:-**

**Conclusion:-**