MPSTME NMIMS- B.Tech. CSBS Year IV sem VII

IT Workshop/MATLAB:-Lab Assignment 01

Varun Khadayate A016

NOTE:

- Explore MATLAB Desktop environment before starting these experiments.
- All the variables provided are case sensitive.

1. Launch MATLAB, create list of following variables in command window:

a) m=10, n=25, p=43

```
>> m=10, n=25, p=43

m =

    10


n =

    25


p =

    43
```

b) $A=m^2$, $B=n^3$, $C=(A+B)*p$

```
>> A=m^2,B=n^3, C=(A+B)*p

A =

   100


B =

       15625


C =

      676175
```

c) t=0.1, f=0.5, a=5, x=a*sin(2πft)

```
>> t=0.1, f=0.5, a=5, x=a*sin(2*pi*f*t)

t =

    0.1000


f =

    0.5000


a =

     5


x =

    1.5451
```

d) y=mx+C

```
>> y=m*x+C

y =
```

e) $k=(t^2+1)(t^2-1)$

```
>> k=(t^2+1)*(t^2-1)

k =

   -0.9999
```

2. (a) From Question1 make a new variable '*v'*, overwriting part (c), i.e.,

$$x=a*sin(2\pi ft)$$

by adding *cosh(t)*

```
>> v = x + cosh(t)

v =

    2.5501
```

(b) Create variable '*r'* , store value to it to find the area of circle :

$$A=\pi r^2$$

where '*r'* is the radius of circle. Further, using the built in function **namelengthmax ,** find the maximum number of character in "A".

[Hint: store value of '*r'* as 10]

```
>> r = 10, A= pi*r^2

r =

    10


A =

  314.1593
```

3. Explore the **solve** command using MATLAB help and find the solution for the problem given :

X + 1 = 2, find x.

```
>> syms x
>> solve(x + 1 == 2, x)

ans =

1
```

4. Complete the table using **help** command:

Command name	**Purpose**
whos	whos List current variables, long form. whos is a long form of WHO. It lists all the variables in the current workspace, together with information about their size, bytes, class, etc. In a nested function, variables are grouped into those in the nested function and those in each of the containing functions, each group separated by a line of dashes. In nested functions and in functions containing nested functions, even uninitialized variables are listed. whos GLOBAL lists the variables in the global workspace. whos -FILE FILENAME lists the variables in the specified .MAT file. whos ... VAR1 VAR2 restricts the display to the variables specified. The wildcard character '*' can be used to display variables that match a pattern. For instance, whos A* finds all variables in the

			current workspace that start with A. whos -REGEXP PAT1 PAT2 can be used to display all variables matching the specified patterns using regular expressions. For more information on using regular expressions, type "doc regexp" at the command prompt. Use the functional form of whos, such as whos('-file',FILE,V1,V2), when the filename or variable names are stored in strings. S = whos(...) returns a structure with the fields: name -- variable name size -- variable size bytes -- number of bytes allocated for the array class -- class of variable global -- logical indicating whether variable is global sparse -- logical indicating whether value is sparse complex -- logical indicating whether value is complex nesting -- struct with the following two fields: function -- name of function where variable is defined level -- nesting level of the function persistent -- logical indicating whether variable is persistent You must use the functional form of whos when there is an output argument. Examples for pattern matching: whos a* % Show variable names starting with "a" whos -regexp ^b\d{3}$ % Show variable names starting with "b" % and followed by 3 digits whos -file fname -regexp \d % Show variable names containing any % digits that exist in MAT-file fname See also who, clear, clearvars, save, load. Overloaded methods: Simulink.whos Reference page in Help browser doc whos
		clear	clear Clear variables and functions from memory. clear removes all variables from the workspace. clear VARIABLES does the same thing. clear GLOBAL removes all global variables. clear FUNCTIONS removes all compiled MATLAB and MEX-functions.

			clear ALL removes all variables, globals, functions and MEX links. clear ALL at the command prompt also clears the base import list. clear IMPORT clears the base import list. It can only be issued at the command prompt. It cannot be used in a function. clear CLASSES is the same as clear ALL except that class definitions are also cleared. If any objects exist outside the workspace (say in userdata or persistent in a locked program file) a warning will be issued and the class definition will not be cleared. clear CLASSES must be used if the number or names of fields in a class are changed. clear JAVA is the same as clear ALL except that java classes on the dynamic java path (defined using JAVACLASSPATH) are also cleared. clear VAR1 VAR2 ... clears the variables specified. The wildcard character '*' can be used to clear variables that match a pattern. For instance, clear X* clears all the variables in the current workspace that start with X. clear -REGEXP PAT1 PAT2 can be used to match all patterns using regular expressions. This option only clears variables. For more information on using regular expressions, type "doc regexp" at the command prompt. If X is global, clear X removes X from the current workspace, but leaves it accessible to any functions declaring it global. clear GLOBAL X completely removes the global variable X. clear GLOBAL -REGEXP PAT removes global variables that match regular expression patterns. Note that to clear specific global variables, the GLOBAL option must come first. Otherwise, all global variables will be cleared. clear FUN clears the function specified. If FUN has been

			locked by MLOCK it will remain in memory. Use a partial path (see PARTIALPATH) to distinguish between different overloaded versions of FUN. For instance, 'clear inline/display' clears only the INLINE method for DISPLAY, leaving any other implementations in memory. clear ALL, clear FUN, or clear FUNCTIONS also have the side effect of removing debugging breakpoints and reinitializing persistent variables since the breakpoints for a function and persistent variables are cleared whenever the program file changes or is cleared. Use the functional form of clear, such as clear('name'), when the variable name or function name is stored in a string. Examples for pattern matching: clear a* % Clear variables starting with "a" clear -regexp ^b\d{3}$ % Clear variables starting with "b" and % followed by 3 digits clear -regexp \d % Clear variables containing any digits See also clearvars, who, whos, mlock, munlock, persistent, import. Reference page in Help browser doc clear
		pwd	pwd Show (print) current working directory. pwd displays the current working directory. S = pwd returns the current directory in the string S. See also cd. Reference page in Help browser doc pwd
		diary	diary Save text of MATLAB session. diary FILENAME causes a copy of all subsequent command window input and most of the resulting command window output to be appended to the named file. If no file is specified, the file 'diary' is used. diary OFF suspends it. diary ON turns it back on. diary, by itself, toggles the diary state.

			Use the functional form of diary, such as diary('file'), when the file name is stored in a string. See also save. Reference page in Help browser doc diary

5.	Given *theta = 145 dgrees.* A vector can be represented by its rectangular coordinates *x* and *y* or by its polar coordinates *r* and *theta*. Theta is measured in radians. The relationship between them is given by the equations:
	x = r * cos (theta)
	y = r * sin (theta)
	
	Assign values for the polar coordinates to variables *r* and *theta*. Then, using these values, assign the corresponding rectangular coordinates to variables *x* and *y*.
	`>> theta = 145`
	
	`theta =`
	
	`145`
	
	`>> x = r * cos (theta)`
	`y = r * sin (theta)`
	
	`x =`
	
	`8.8386`
	
	`y =`
	
	`4.6775`

6.	The combined resistance R_r of three resistors R1, R2, and R3 in parallel is given by $$R_r = \frac{1}{\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3}}$$ Create variables for the three resistors and store values in each, and then calculate the combined resistance. (Hint: consider R1=50Ω, R2=25Ω and R3=60Ω)

```
>> R1=50, R2=25, R3=60

R1 =

    50


R2 =

    25


R3 =

    60

>> R=1/((1/R1)+(1/R2)+(1/R3))

R =

   13.0435
```