BASIC SIMULATION WITH MAT LABORATORY
LAB MANUAL

Academic Year	**:**	**2019-2020**
Course Code	**:**	**AAEB01**
Regulation	**:**	**R18**
Class	**:**	**II Sem**
Branch	**:**	**AE**

Prepared
By

Dr. P K Mohanta, Professor

AERONAUTICAL ENGINEERING

INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal, Hyderabad - 500 043

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

AERONAUTICALENGINEERING

	Program Outcomes
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineeringproblems.
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainabledevelopment.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

Program Specific Outcomes	
PSO1	**Professional skills:** Able to utilize the knowledge of aeronautical/aerospace engineering in innovative, dynamic and challenging environment for design and development of new products
PSO2	**Problem solving skills**: imparted through simulation language skills and general purpose CAE packages to solve practical, design and analysis problems of components to complete the challenge of airworthiness for flightvehicles
PSO3	**Practical implementation and testing skills**: Providing different types of in house and training and industry practice to fabricate and test and develop the products with more innovative technologies.
PSO4	**Successful career and entrepreneurship**: To prepare the students with broad aerospace knowledge to design and develop systems and subsystems of aerospace and allied systems and become technocrats

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

Certificate

This is to certify that it is a bonafied record of practical work done by Sri/Kum._ ____bearing the Roll No. ____________________________of_ _____________ class ____ branch in the ______________________________________laboratory during the academic year___ ____________________ under oursupervision.

Head oftheDepartment **LecturerIn-Charge**

ExternalExaminer **InternalExaminer**

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES

Expt. No.	Program Outcomes Attained	Program Specific Outcomes Attained
I	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
II	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
III	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
IV	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
V	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
VI	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
VII	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
VIII	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
IX	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
X	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
XI	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
XII	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2

BASIC SIMULATION WITH MAT LABORATORY

II Semester: AE									
Course Code	Category	Hours / Week			Credits	Maximum Marks		
AAEB01	Foundation	L	T	P	C	CIE	SEE	Total
		0	0	3	1.5	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 24				Total Classes: 24		

OBJECTIVES:

The course should enable the students to:

I. Understand the procedures, algorithms, and concepts require to solve specificproblems.
II. Analyze the concepts of algebra, calculus and numerical solutions using MATLABsoftware.
III. Enrich the knowledge in MATLAB and can apply for projectworks.
IV. Interpret and visualize simple mathematical functions and operations thereon usingplots/display.

LIST OF EXPERIMENTS

Week-l — BASIC FEATURES

a. Features anduses.
b. Local environmentsetup.

Week-2 — ALGEBRA

a. Solving basic algebraicequations.
b. Solving system ofequations.
c. Two dimensionalplots.

Week-3 — CONTROL STRUCTURES

a. For Loop.
b. WhileLoop.
c. If- elseif- else controlstructure.

Week-4 — MATRICES

a. Addition, subtraction and multiplication ofmatrices.
b. Transpose of amatrix.
c. Inverse of amatrix.

Week-5 — SYSTEM OF LINEAR EQUATIONS

a. Rank of amatrix.
b. Gauss Jordanmethod.
c. LU decompositionmethod.

Week-6 — LINEAR TRANSFORMATION

a. Characteristicequation.
b. Eigenvalues.
c. Eigen vectors.

Week-7 — DIFFERENTIATION AND INTEGRATION

a. Higher order differentialequations.
b. Doubleintegrals.
c. Tripleintegrals.

Week-8 — NUMERICAL DIFFERENTION AND INTEGRATION

a. Trapezoidal, Simpson’smethod.
b. Eulermethod.
c. RungeKutta method

Week-9 — 3D PLOTTING

a. Lineplotting.
b. Surfaceplotting.
c. Volumeplotting.

Week-10	DEFLECTION OF SIMPLY SUPPORTED BEAM
a. Calculating vertical displacement with pointload. b. Calculating vertical displacement with uniformly distributedload. c. Calculating vertical displacement with uniformly varyingload.	
Week-11	**DEFLECTION OF CANTILEVER BEAM**
b. Calculating vertical displacement with pointload. c. Calculating vertical displacement with uniformly distributedload. c. Calculating vertical displacement with uniformly varying load	
Week-12	**FORMULATION OF IDEAL AND REAL GAS EQUATIONS**
a. Calculating the pressure, temperature, density for Earth's atmospheric conditions atdifferent altitudes. b. Calculating the pressure, temperature, density for other planets at differentaltitudes.	

Reference Books:

1. Cleve Moler, "Numerical Computing with MATLAB", SIAM, Philadelphia, 2nd Edition, 2008.
2. Dean G. Duffy, "Advanced Engineering Mathematics with MATLAB", CRC Press, Taylor& Francis Group, 6th Edition,2015.
3. Delores M. Etter, David C. Kuncicky, Holly Moore, "Introduction to MATLAB 7", Pearson Education Inc, 1st Edition,2009.
4. Rao. V. Dukkipati , "MATLAB for ME Engineers", New Age Science, 1st Edition,2008.

Web Reference:

1. http://www.tutorialspoint.com/matlab/
2. http://www.iare.ac.in

SOFTWARE AND HARDWARE REQUIREMENTS FOR A BATCH OF 30 STUDENTS:

SOFTWARE: Microsoft Windows 7 and MATLAB – V 8.5, which is also R2015a
HARDWARE: 30 numbers of Intel Desktop Computers with 2 GB RAM

EXPERIMENT 1

BASIC FEATURES

OBJECTIVES:

a. To know the history, features and uses ofMATLAB
b. To know the local environment ofMATLAB

CONTENT:

Introduction:

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyses data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spread sheets or traditional programming languages, such as C/C++ or Java. You can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

History:

1. Developed primarily by Cleve Moler in the 1970'sDerived from FORTRAN subroutines LINPACK and EISPACK, linear and eigenvaluesystems.
2. Developed primarily as an interactive system to access LINPACK andEISPACK.
3. Gained its popularity through word of mouth, because it was not sociallydistributed.
4. Rewritten in C in the 1980's with more functionality, which include plottingroutines.
5. The Math Works Inc. was created (1984) to market and continue development ofMATLAB.

Strengths:

1. MATLAB may behave as a calculator or as a programminglanguage
2. MATLAB combine nicely calculation and graphicplotting.
3. MATLAB is relatively easy tolearn
4. MATLAB is interpreted (not compiled), errors are easy tofix
5. MATLAB is optimized to be relatively fast when performing matrixoperations
6. MATLAB does have some object-orientedelements

Weaknesses:

1. MATLAB is not a general purpose programming language such as C, C++, orFORTRAN
2. MATLAB is designed for scientific computing, and is not well suitable for otherapplications
3. MATLAB is an interpreted language, slower than a compiled language such asC++
4. MATLAB commands are specific for MATLAB usage. Most of them do not have a direct equivalent with other programming languagecommands

Competition:

One of MATLAB's competitors is Mathematica the symbolic computation program. MATLAB is more convenient for numerical analysis and linear algebra. It is frequently used in engineering community. Mathematica has superior symbolic manipulation, making it popular among physicists. There are other competitors: Scilab, GNU Octave, and Rlab.

Key Features:

1. It is a high-level language for numerical computation, visualization and applicationdevelopment.
2. It also provides an interactive environment for iterative exploration, design and problemsolving.
3. It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differentialequations.
4. It provides built-in graphics for visualizing data and tools for creating customplots.
5. MATLAB's programming interface gives development tools for improving code quality, maintainability, and maximizingperformance.
6. It provides tools for building applications with custom graphicalinterfaces.

7. It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and MicrosoftExcel.

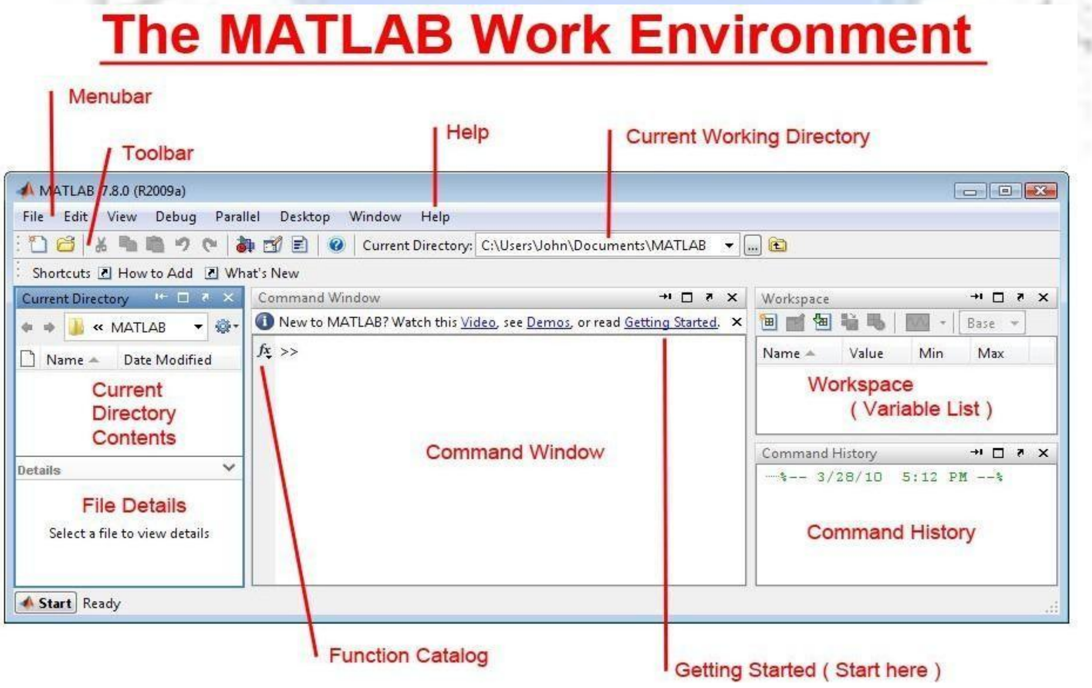
MATLAB's Power of Computational Mathematics:
MATLAB is used in every facet of computational mathematics. Following are some commonly used mathematical calculations where it is used most commonly:

1. Dealing with Matrices andArrays
2. 2-D and 3-D Plotting andgraphics
3. LinearAlgebra
4. Algebraic Equations
5. Non-linear Functions
6. Statistics
7. Data Analysis
8. Calculus and DifferentialEquations
9. NumericalCalculations
10. Integration
11. Transforms
12. CurveFitting
13. Various other specialfunctions

Uses of MATLAB:
MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including:

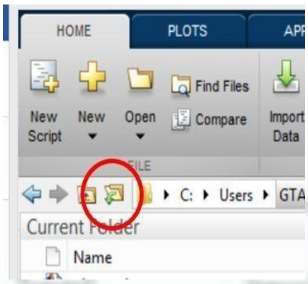
1. Signal processing andCommunications
2. Image and videoProcessing
3. Controlsystems
4. Test andmeasurement
5. Computationalfinance
6. Computationalbiology

Understanding the MATLAB Environment:
MATLAB development IDE can be launched from the icon created on the desktop. The main working window in MATLAB is called the desktop. When MATLAB is started, the desktop appears in its default layout:



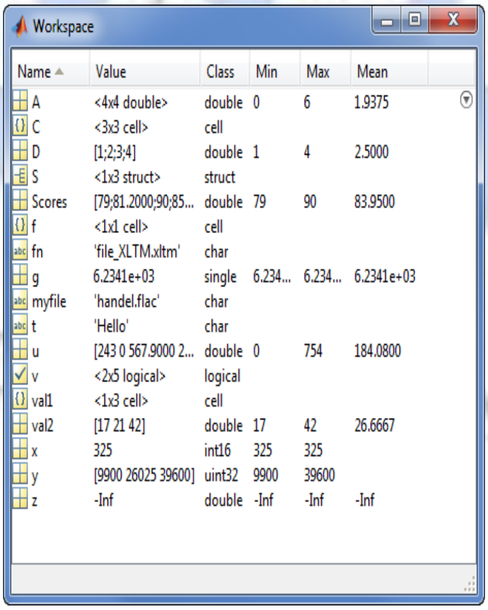
The desktop includes these panels:

Current Folder - This panel allows you to access the project folders and files.

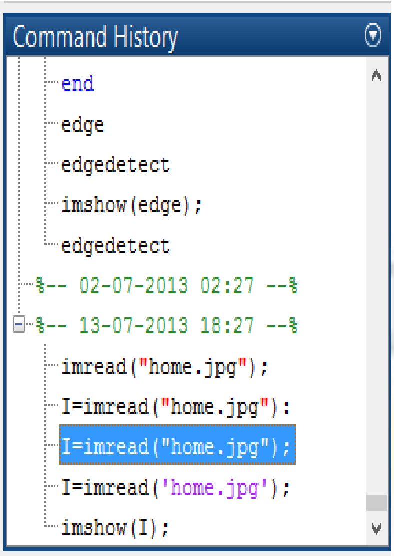


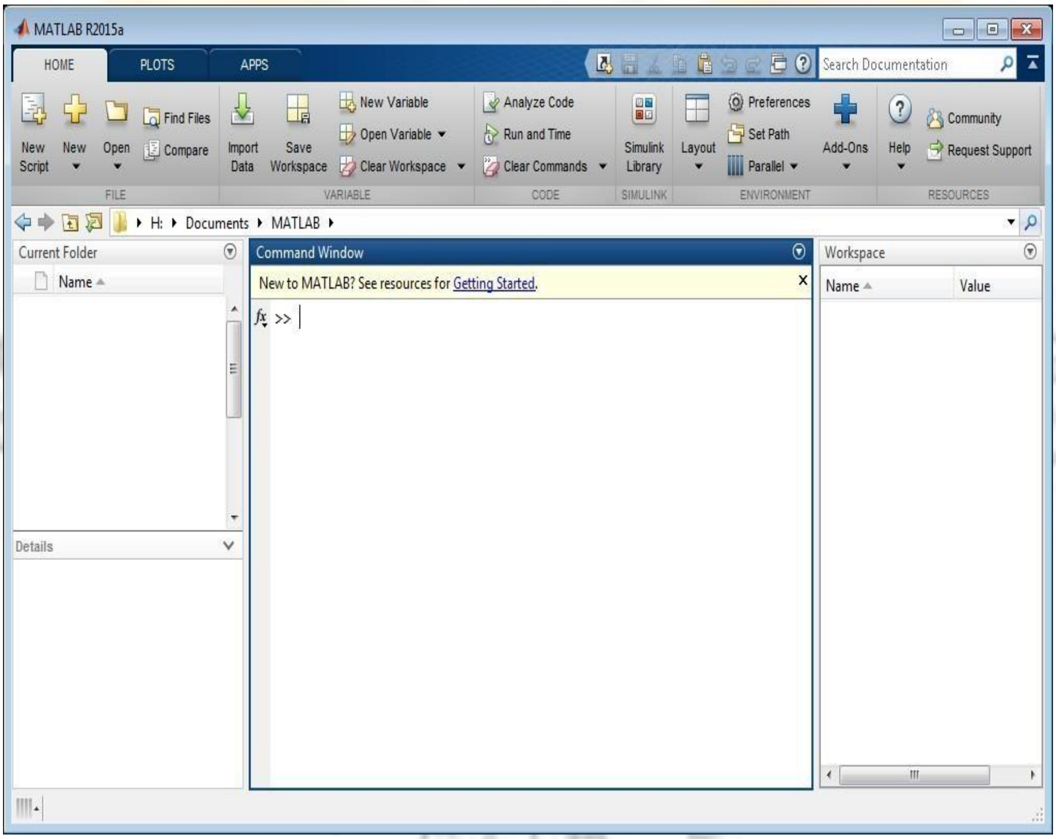
Command Window - This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).

Workspace - The workspace shows all the variables created and/or imported from files.



Command History - This panel shows or rerun commands that are entered at the command line.





You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window. Usually, there are 2 types of prompt:

>>For full version

EDU> for educational version

Note:

1. To simplify the notation, we will use this prompt, >>, as a standard prompt sign, though our MATLAB version is for educationalpurpose.
2. MATLAB adds variable to the workspace and displays the result in the CommandWindow.

Managing workspace and file commands:

Command	Description
cd	Change current directory
clc	Clear the Command Window
clear (all)	Removes all variables from the workspace
clear x	Remove x from the workspace
copy file	Copy file or directory
delete	Delete files
dir	Display directory listing
exist	Check if variables or functions are defined
help	Display help for MATLAB functions
look for	Search for specified word in all help entries
mkdir	Make new directory
move file	Move file or directory
pwd	Identify current directory
rmdir	Remove directory
type	Display contents of file
what	List MATLAB files in current directory
which	Locate functions and files
who	Display variables currently in the workspace
whos	Display information on variables in the workspace

Commonly used Operators and Special Characters:
MATLAB supports the following commonly used operators and special characters:

Operator	Purpose
+	Plus; addition operator.
-	Minus; subtraction operator.
*	Scalar and matrix multiplication operator.
.*	Array multiplication operator.
^	Scalar and matrix exponentiation operator.
.^	Array exponentiation operator.
\	Left-division operator.
/	Right-division operator.
.\	Array left-division operator.
./	Array right-division operator.
:	Colon; generates regularly spaced elements and represents an entire row or column.

( )	Parentheses; encloses function arguments and array indices; overrides precedence.
[ ]	Brackets; enclosures array elements.
.	Decimal point.
…	Ellipsis; line-continuation operator
,	Comma; separates statements and elements in a row
;	Semicolon; separates columns and suppresses display.
%	Percent sign; designates a comment and specifies formatting.
_	Quote sign and transpose operator.
._	Non-conjugated transpose operator.
=	Assignment operator.

Note:
If you end a statement with a semicolon, MATLAB performs the computation, butsuppresses the display of output in the Command Window.

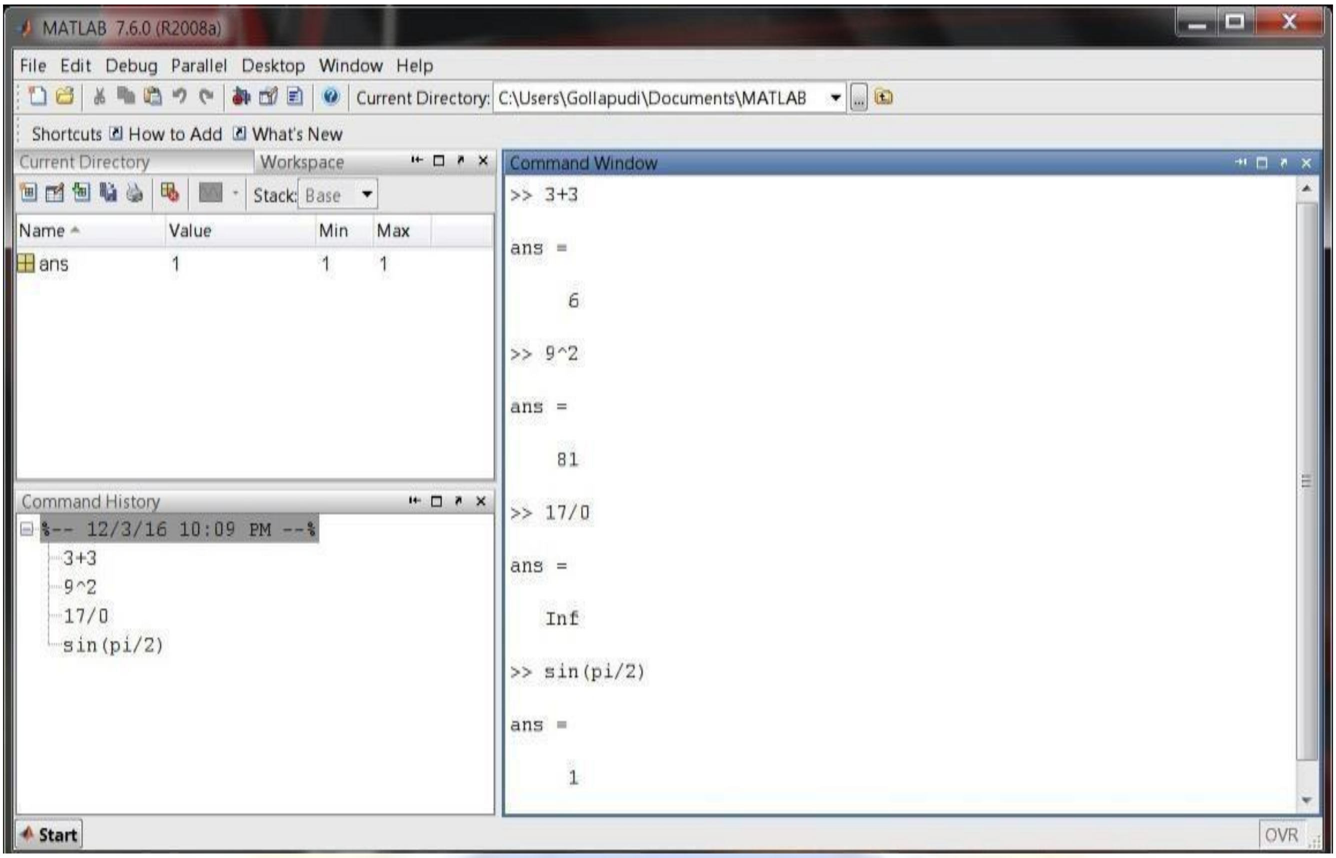
Special Variables and Constants:
MATLAB supports the following special variables and constants:

Name	**Meaning**
ans	Most recent answer.
eps	Accuracy of floating-point precision.
i,j	The imaginary unit √-1.
Inf	Infinity.
NaN	Undefined numerical result (not a number).
pi	The number π

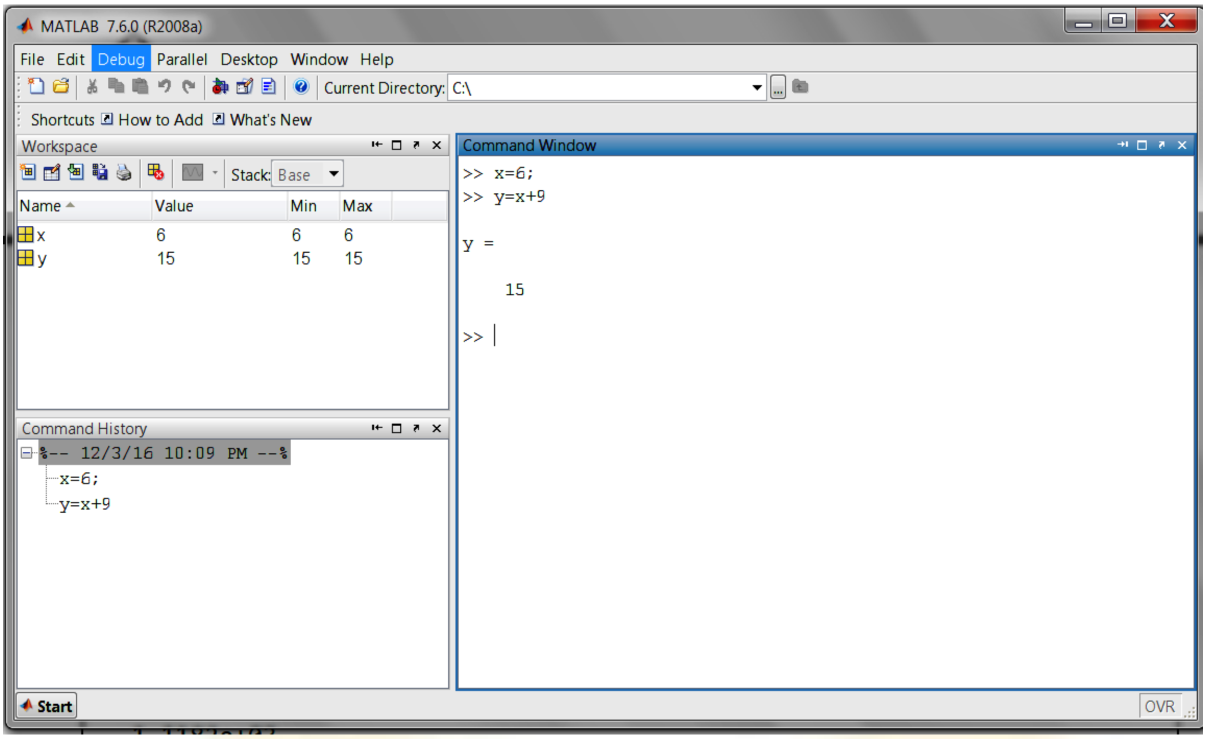
Naming Variables:
Variable names consist of a letter followed by any number of letters, digits or underscore. MATLAB is **case-sensitive**. Variable names can be of any length; however, MATLAB uses only first N characters, where N is given by the function **namelengthmax**.

Saving Your Work:
The **save** command is used for saving all the variables in the workspace, as a file with .mat extension, in the current directory.
For example, save myfile
You can reload the file anytime later using the **load** command. load myfile

Example 1:


MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\Gollapudi\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Stack: Base
Name Value Min Max
ans 1 1 1
Command History
%-- 12/3/16 10:09 PM --%
3+3
9^2
17/0
sin(pi/2)
Command Window
>> 3+3
ans =
6
>> 9^2
ans =
81
>> 17/0
ans =
Inf
>> sin(pi/2)
ans =
1
Start
OVR

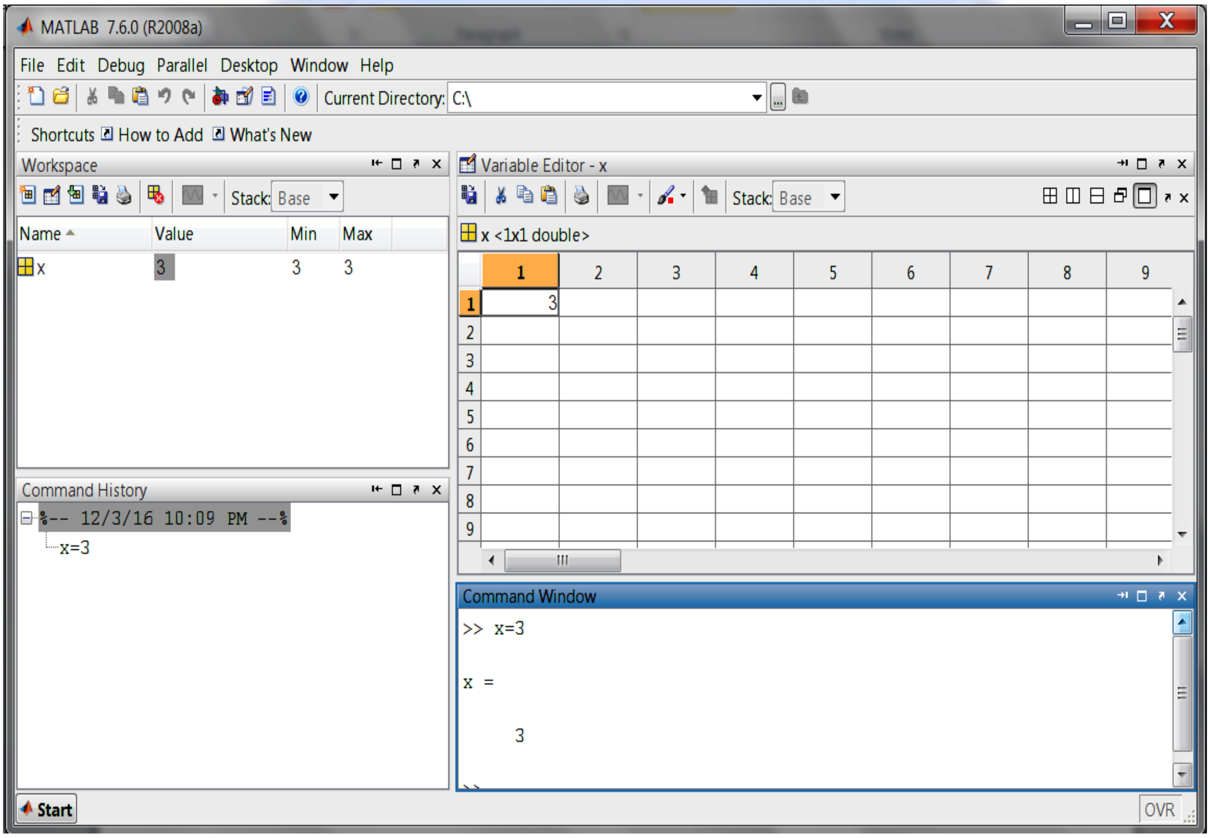
Example 2:



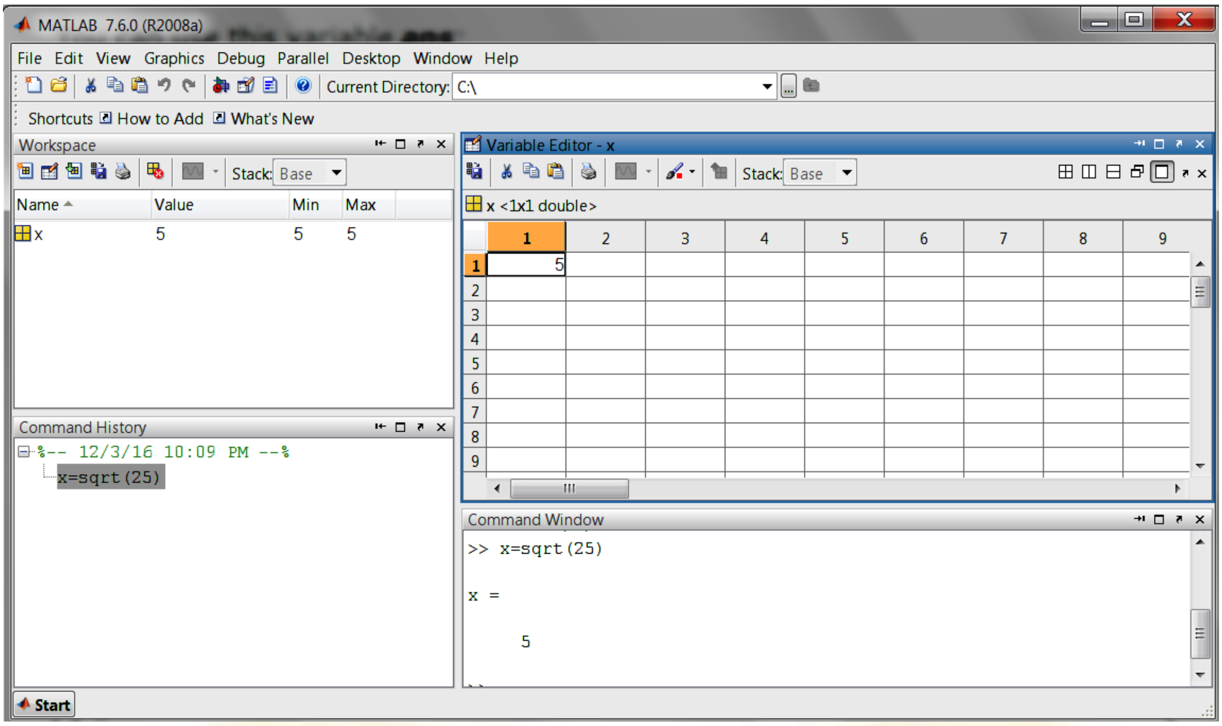
In MATLAB environment, every variable is an array or matrix.

Example 3:

In the above example it creates a 1-by-1 matrix named ‘x’and stores the value 3 in its element.



Example 4:

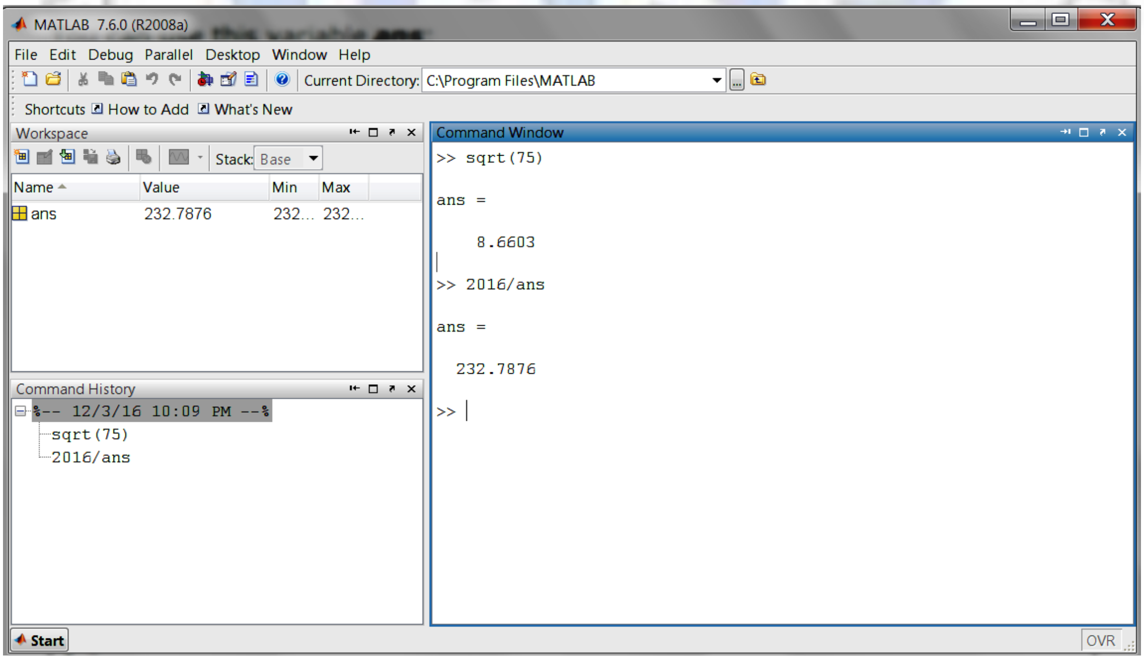


In this example x is to find the square root of 25 it creates a 1-by-1 matrix named 'x'and stores the value 5 in its element.

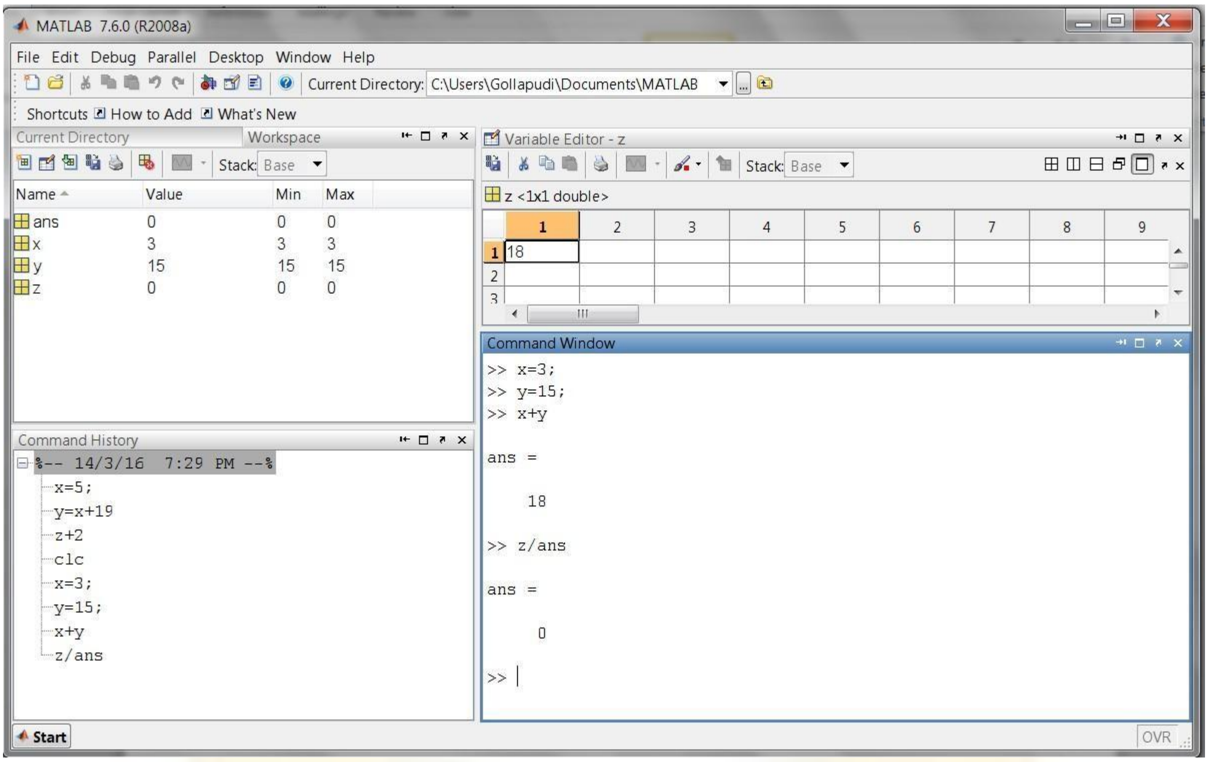
Note:

1. Once a variable is entered into the system, you can refer to itlater.
2. Variables must have values before they areused.
3. When you do not specify an output variable, MATLAB uses the variable ans, short for *answer*, to store the results of yourcalculation.

Example 5:



Example 6:



In the above example we have multiple assignments

EXPERIMENT 2

ALGEBRA

OBJECTIVES:

a) Find the roots of the equations $6x^5 -41x^4 +97x^3 -97x^2+41x-6$
b) Find the values of x, y, z of the equationsx+y+z=3,x+2y+3z=4,x+4y+9z=6
c) For $f(x)=8x^8 -7x^7 +12x^6 -5 x^5 +8 x^4 +13 x^3 -12x+9$ compute f(2),roots of f(x) and plot for $0 \le x \le 20$

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in currentdirectory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

Roots of the equations $6x^5 -41x^4 +97x^3 -97x^2 +41x-6$

```
v = [6, -41, 97, -97, 41,-6]; % writing the coefficients
s = roots(v);
disp('The first root is: '), disp(s(1));
disp('The second root is: '), disp(s(2));
disp('The third root is: '), disp(s(3));
disp('The fourth root is: '), disp(s(4));
disp('The fifth root is: '), disp(s(5));
```

Values of x,y,z of the equations x+y+z=3,x+2y+3z=4,x+4y+9z=6

```
A=[1,1,1;1,2,3;1,4,9];
b=[3;4;6];
inv(A)
inv(A)*b
```

f(2), roots and plot of f(x)

```
p=[8 -7 12 -5 8 13 0 -12 9];
polyval(p,2)
roots(p)
x=0:0.1:20;
y=polyval(p,x);
plot(x,y)
```

OUTPUT:

Roots of the equations $6x^5 -41x^4 +97x^3 -97x^2 +41x-6$

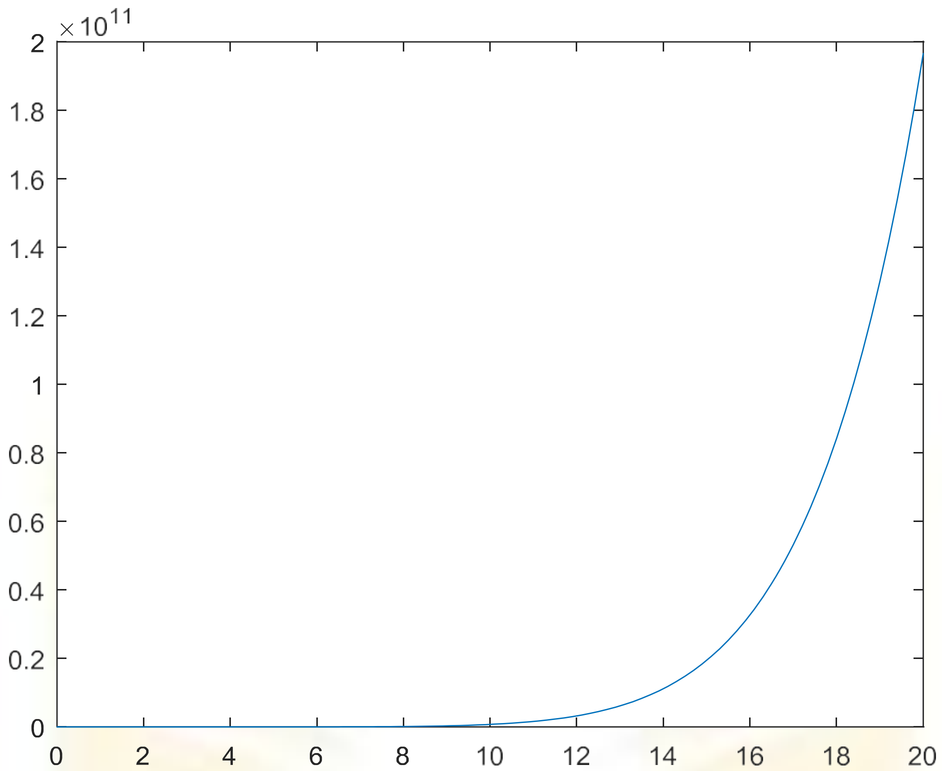
```
The first root is:
    3.0000

The second root is:
    2.0000

The third root is:
    1.0000

The fourth root is:
    0.5000

The fifth root is:
    0.3333
```

Values of x,y,z of the equations x+y+z=3,x+2y+3z=4,x+4y+9z=6

```
ans =

    3.0000  -2.5000    0.5000
   -3.0000   4.0000  -1.0000
    1.0000  -1.5000    0.5000


ans =

    2.0000
    1.0000
    0.0000
```

f(2),roots and plot of f(x)

```
ans =

        1977


ans =

  -0.2079 + 1.3091i
  -0.2079 - 1.3091i
  -0.8053 + 0.4306i
  -0.8053 - 0.4306i
   0.8878 +0.9318i
   0.8878 - 0.9318i
   0.5629 + 0.3828i
   0.5629 - 0.3828i
```


×10^11
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
14
16
18
20

EXPERIMENT 3

CONTROL STRUCTURES

OBJECTIVES:

a. To know how to use For Loop withexamples.
b. To know how to use While Loop withexamples.
c. To know how to use If- else if- else control structure withexamples.

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in currentdirectory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

while loop

The while loop repeatedly executes statements while a specified condition is true
The syntax of a while loop in MATLAB is as following:

Syntax

```
while <expression>
 <statements>
end
```

Example

```
a =10;
% while loop execution
while( a < 20 )
 fprintf('value of a: %d\n', a);
 a = a + 1;
end
```

for loop

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.
The syntax of a for loop in MATLAB is as following:

Syntax

```
for index = values
 <program statements>
 ...
end
```

Example

```
for a = 10:20
 fprintf('value of a: %d\n', a);
end
```

if loop

if expression, statements, end evaluates an expression, and executes a group of statements when the expression is true. An expression is true when its result is nonempty and contains only nonzero elements (logical or real numeric). Otherwise, the expression is false.

The elseif and else blocks are optional. The statements execute only if previous expressions in the if...end block are false. An if block can include multiple elseifblocks.

Syntax

```
if expression
    statements
elseif expression
    statements
else
    statements
end
```

Example

```
nrows =4
ncols =6
A = ones(nrows,ncols)
for c = 1:ncols
forr = 1:nrows
if r == c
            A(r,c) = 2
        elseifabs(r-c) == 1
            A(r,c) = -1
        else
            A(r,c) = 0
end
end
end
```

OUTPUT:
While loopvalue of a: 10 value of a: 11 value of a: 12 value of a: 13 value of a: 14 value of a: 15 value of a: 16 value of a: 17 value of a: 18 value of a:19

for loopvalue of a: 10 value of a: 11 value of a: 12 value of a: 13 value of a: 14 value of a: 15 value of a: 16 value of a: 17 value of a: 18 value of a: 19 value of a:20

if loop

```
nrows =

     4


ncols =

     6


A =

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


A =

     2     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
```

A =

2	1	1	1	1	1
-1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

A =

2	1	1	1	1	1
-1	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1

A =

2	1	1	1	1	1
-1	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

A =

2	-1	1	1	1	1
-1	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

A =

2	-1	1	1	1	1
-1	2	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

A =

2	-1	1	1	1	1
-1	2	1	1	1	1
0	-1	1	1	1	1
0	1	1	1	1	1

A =

2	-1	1	1	1	1
-1	2	1	1	1	1
0	-1	1	1	1	1
0	0	1	1	1	1

```
A =

     2    -1     0     1     1     1
    -1     2     1     1     1     1
     0    -1     1     1     1     1
     0     0     1     1     1     1


A =

     2    -1     0     1     1     1
    -1     2    -1     1     1     1
     0    -1     1     1     1     1
     0     0     1     1     1     1


A =

     2    -1     0     1     1     1
    -1     2    -1     1     1     1
     0    -1     2     1     1     1
     0     0     1     1     1     1


A =

     2    -1     0     1     1     1
    -1     2    -1     1     1     1
     0    -1     2     1     1     1
     0     0    -1     1     1     1


A =

     2    -1     0     0     1     1
    -1     2    -1     1     1     1
     0    -1     2     1     1     1
     0     0    -1     1     1     1


A =

     2    -1     0     0     1     1
    -1     2    -1     0     1     1
     0    -1     2     1     1     1
     0     0    -1     1     1     1


A =

     2    -1     0     0     1     1
    -1     2    -1     0     1     1
     0    -1     2    -1     1     1
     0     0    -1     1     1     1
```

A =

2	-1	0	0	1	1
-1	2	-1	0	1	1
0	-1	2	-1	1	1
0	0	-1	2	1	1

A =

2	-1	0	0	0	1
-1	2	-1	0	1	1
0	-1	2	-1	1	1
0	0	-1	2	1	1

A =

2	-1	0	0	0	1
-1	2	-1	0	0	1
0	-1	2	-1	1	1
0	0	-1	2	1	1

A =

2	-1	0	0	0	1
-1	2	-1	0	0	1
0	-1	2	-1	0	1
0	0	-1	2	1	1

A =

2	-1	0	0	0	1
-1	2	-1	0	0	1
0	-1	2	-1	0	1
0	0	-1	2	-1	1

A =

2	-1	0	0	0	0
-1	2	-1	0	0	1
0	-1	2	-1	0	1
0	0	-1	2	-1	1

A =

2	-1	0	0	0	0
-1	2	-1	0	0	0
0	-1	2	-1	0	1
0	0	-1	2	-1	1

A =

2	-1	0	0	0	0
-1	2	-1	0	0	0
0	-1	2	-1	0	0
0	0	-1	2	-1	1

A =

2	-1	0	0	0	0
-1	2	-1	0	0	0
0	-1	2	-1	0	0
0	0	-1	2	-1	0

EXPERIMENT 4

MATRICES

OBJECTIVES:

a) Find the addition, subtraction and multiplication ofmatrix

$$A=\begin{bmatrix} 1 & 2 & -9 \\ 2 & -1 & 2 \\ 3 & -4 & 3 \end{bmatrix}$$

$$B=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

b) Find the transpose ofmatrix

$$A=\begin{bmatrix} 1 & 2 & -9 \\ 2 & -1 & 2 \\ 3 & -4 & 3 \end{bmatrix}$$

c) Find the inverse ofmatrix

$$A=\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 1 & 2 & 5 \end{bmatrix}$$

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in currentdirectory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

Addition, Subtraction and Multiplication of matrix

```
a=[1 2 -9 ; 2 -1 2; 3 -4 3];
b=[1 2 3; 4 5 6; 7 89];
disp('The matrix a= ');a
disp('The matrix b=');b
% to find sum of a and b
 c=a+b;
disp('The sum of a and b is ');c
% to find difference of a and b
d=a-b;
disp('The difference of a and b is ');d
%to find multiplication of a and b
e=a*b;
disp('The product of a and b is ');e
% to find element-by-element multiplication
```

Transpose of matrix

```
A=[1,2,-9;2,-1,2;3,-4,3]
B = A.'
```

Inverse of matrix

```
a = [1 2 3; 2 3 4; 1 2 5]
inv (a)
```

OUTPUT:

Addition, Subtraction and Multiplication of matrix

```
The matrix a=

a =

     1     2    -9
     2    -1     2
     3    -4     3

The matrix b=

b =

     1     2     3
     4     5     6
     7     8     9

The sum of a and b is

c =

     2     4    -6
     6     4     8
    10     4    12

The difference of a and b is

d =

     0     0-12
    -2    -6    -4
    -4   -12    -6

The product of a and b is

e =

   -54   -60   -66
    12    15    18
     8    10    12
```

Transpose of matrix

```
A =

     1     2    -9
     2    -1     2
     3    -4     3


B =

     1     2     3
     2    -1    -4
    -9     2     3
```

Inverse of matrix

```
a =

     1     2     3
     2     3     4
     1     2     5


ans =

   -3.5000    2.0000    0.5000
    3.0000   -1.0000   -1.0000
   -0.5000         0    0.5000
```

EXPERIMENT 5

SYSTEM OF LINEAREQUATIONS

OBJECTIVES:

a) Find the rank ofmatrix

$$A=\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 9 & 10 & 11 \\ 13 & 14 & 15 \end{bmatrix}$$

b) Find the row echelonform

$$A=\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 9 & 10 & 11 \\ 13 & 14 & 15 \end{bmatrix}$$

c) Find the LU decomposition of thematrix

$$\begin{bmatrix} 2 & -3 & -1 \\ 1/2 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in currentdirectory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

Rank of matrix

```
A= [1, 2, 3; 5, 6, 7;9, 10, 11;13, 14, 15]
rank (A)
```

Row echelon form

```
A = [1, 2, 3;5, 6, 7;9, 10, 11;13, 14, 15]
R = rref(A)
```

LU decomposition

```
A = [2 -3 -1; 1/2 1 -1; 0 1 -1]
[L, U] = lu(A)
```

OUTPUT:

Rank of matrix

```
A =

     1     2     3
     5     6     7
     9    10    11
    13    14    15


ans =

     2
```

Row echelon form

```
A =

     1     2     3
     5     6     7
     9    10    11
    13    14    15


R =

     1     0    -1
     0     1     2
     0     0     0
     0     0     0
```

LU decomposition

```
A =

    2.0000   -3.0000   -1.0000
     0.5000    1.0000 -1.0000
          0    1.0000-1.0000


L =

    1.0000         0         0
    0.2500    1.0000         0
         0    0.5714    1.0000


U =

      2.0000 -3.0000 -1.0000
          0    1.7500-0.7500
          0         0-0.5714
```

EXPERIMENT 6

LINEAR TRANSFORMATION

OBJECTIVES:

a) Find the characteristic equation of thematrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

b) Find the eigen values of thematrix

$$\begin{bmatrix} 1 & 8 & -10 \\ -4 & 2 & 4 \\ -5 & 2 & 8 \end{bmatrix}$$

c) Find the eigen vector of thematrix

$$\begin{bmatrix} 3 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}$$

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in currentdirectory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

Characteristics equation

```
A = [1 2 3; 4 5 6; 7 8 0]
p = poly(A)
```

Eigen values

```
A = [1 8 -10; -4 2 4; -5 2 8]
e =eig(A)
```

Eigenvector

```
A=[3,1,1;1,0,2;1,2,0];
[eigenvector, eigenvalue] = eig(A)
```

OUTPUT:

Characteristics equation

```
A =

     1     2     3
     4     5     6
     7     8     0


p =

    1.0000 -6.0000 -72.0000 -27.0000
```

Eigen values

```
A =

     1     8   -10
    -4     2     4
    -5     2     8


e =

  11.6219 +0.0000i
  -0.3110 +2.6704i
  -0.3110 - 2.6704i
```

Eigenvector

```
eigenvector=

    0.0000    0.5774   -0.8165
    0.7071   -0.5774   -0.4082
   -0.7071   -0.5774   -0.4082


eigenvalue =

   -2.0000         0         0
         0    1.0000         0
         0         0    4.0000
```

EXPERIMENT 7

DIFFERENTIATION AND INTEGRATION

OBJECTIVES:

a) Solve

$$(D^2 + 5D + 6)y = e^x$$

b) Solve

$$\int_0^5 \int_0^{x^2} x(x^2 + y^2)dxdy$$

c) Solve

$$\int_0^3 \int_0^{3-x} \int_0^{3-x-y} xyzdxdydz$$

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in currentdirectory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

Higher order differential

```
dsolve('D2y+5*Dy+6*y=exp(x)','x')
```

Double Integration

```
symsx y
firstint=int(x*(x^2+y^2),y,0,x^2)
answer=int(firstint,x,0,5)
```

Triple Integration

```
symsx y z
firstans=int(int(int(x*y*z,z,0,3-x-y),y,0,3-x),x,0,3)
```

OUTPUT:

Higher order differential

ans =

exp(x)/12 + C1*exp(-2*x) + C2*exp(-3*x)

Double Integration

firstint =

(x^5*(x^2 + 3))/3

answer =

453125/24

Triple Integration

firstans =

81/80

EXPERIMENT 8

NUMERICAL DIFFERENTION AND INTEGRATION

OBJECTIVES:

a) Evaluate by using trapezoidal and simpsonsmethod

$$\int_{0}^{1.2} e^{x}$$

b) Evaluate

$y^1=x+y,\ y(0)=1$

of size h=0.2 by using eulers and range kutta method

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in currentdirectory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

Trapezoidal Method:

```
x=0:0.2:1.2;
y=exp(x);
trapz(x,y)
```

Simpson's Method:

```
quad('exp(x)',0,1.2)
```

Euler's and Runge-Kutta method:

```
f=@(x,y) (x+y);
[x,y]=ode23(f,[0:0.2:1],1)
[x,y]=ode45(f,[0:0.2:1],1)
```

OUTPUT:

Trapezoidal Method:

```
ans =

    2.3278
```

Simpson's Method:

```
ans =

    2.3201
```

Euler's method

```
x =

         0
    0.2000
    0.4000
    0.6000
    0.8000
    1.0000


y =

    1.0000
    1.2428
    1.5836
    2.0442
    2.6510
    3.4364
```

Runge-Kutta method

```
x =

         0
    0.2000
    0.4000
    0.6000
    0.8000
    1.0000


y =

    1.0000
    1.2428
    1.5836
    2.0442
    2.6511
    3.4366
```

EXPERIMENT 9

3D PLOTTING

OBJECTIVES:

a) Evaluate

$$\oint (3x^2 - 8y^2)dx + (4y - 6xy)dy$$

Where the region is bounded by

$$y = x^2, y = \sqrt{x}$$

b) Plot the surfacefor

$$f = (2 - \cos \pi x)e^y$$

c) Plot the surfacefor

2+cost

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in current directory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

Line Integral
```
clear all clc
symsx y
f=[3*x.^2-8*y.^2 4*y-6*x*y];
disp('Along the curve y=x.^2')
a=subs(f,y,x.^2);
b=diff(x.^2,x);
c=b*a(2);
d=int(a(1),x,0,1);
e=int(c,x,0,1);
u=d+e
disp('Along the curve y=sqrt(x)')
p=subs(f,y,sqrt(x));
q=diff(sqrt(x),x);
r=q*p(2);
s=int(p(1),x,1,0);
t=int(r,x,1,0);
v=s+t
I=u+v
x=-2:0.5:2;
y2=sqrt(x);
y1=x.^2;
plot(x,y1,'r', x,y2,'g');
grid on
```

Surface

```
x=-1:.1:1;
y=0:.1:1.5;
[X,Y]=meshgrid(x,y);
F=(2-cos(pi*X)).*exp(Y);
surf(X,Y,F);
xlabel('x');
ylabel('y');
```

Volume

```
t = 0:pi/10:2*pi;
figure
[X,Y,Z] = cylinder(2+cos(t));
surf(X,Y,Z)
axissquare
```

OUTPUT:

Line Integral

Along the curve y=x.^2

u =

-1

Along the curve y=sqrt(x)

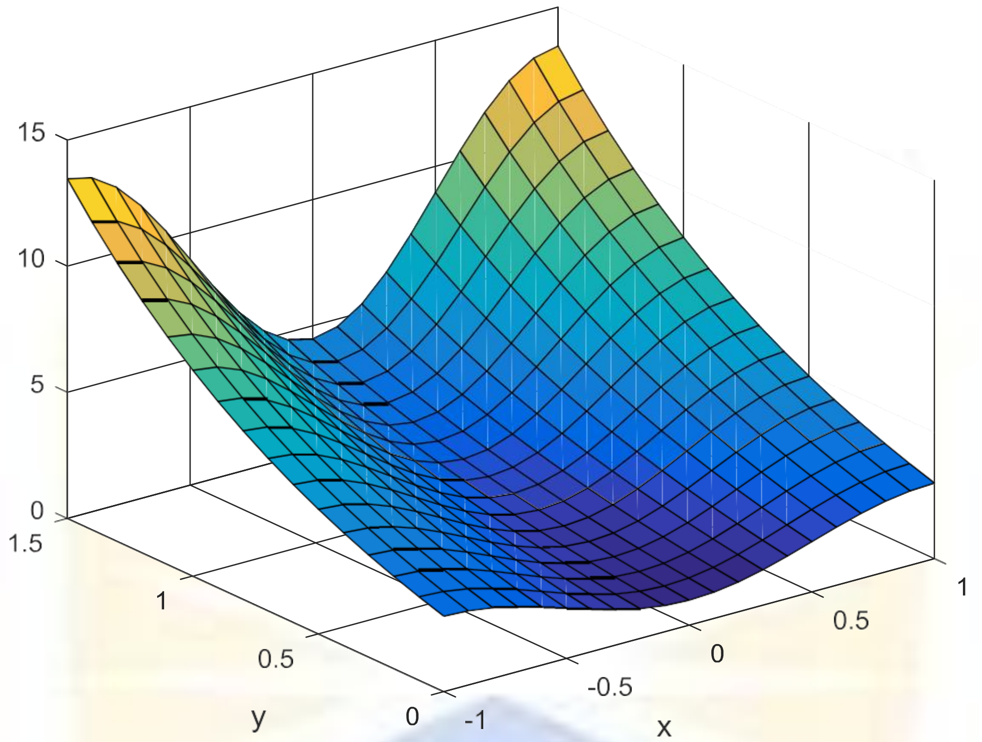
v=

5/2

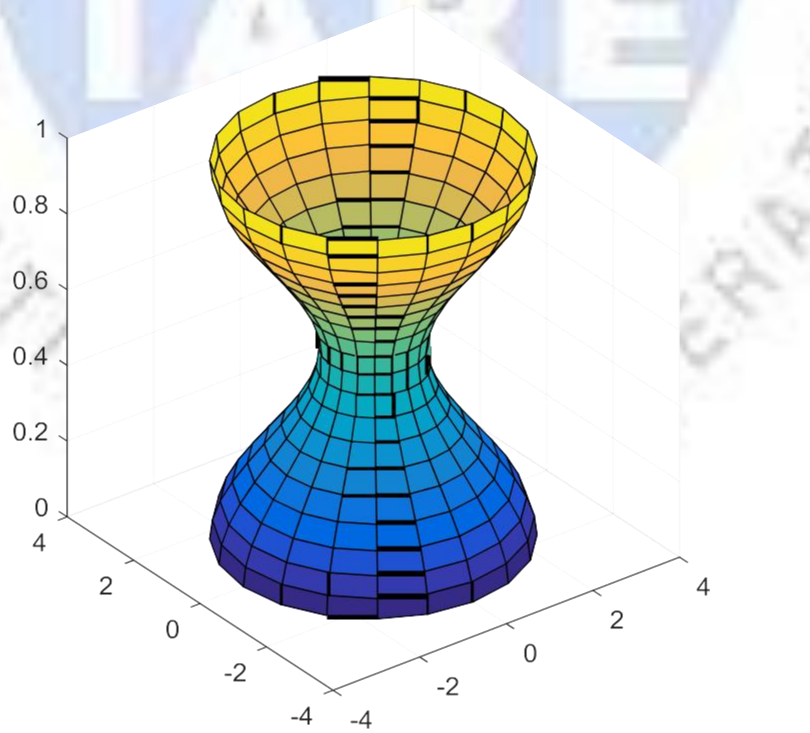
I=

3/2

Surface



Volume



EXPERIMENT 10

DEFLECTION OF SIMPLY SUPPORTED BEAM

OBJECTIVES:

a. Calculating vertical displacement with pointload.
b. Calculating vertical displacement with uniformly distributedload.
c. Calculating vertical displacement with uniformly varyingload.

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in current directory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

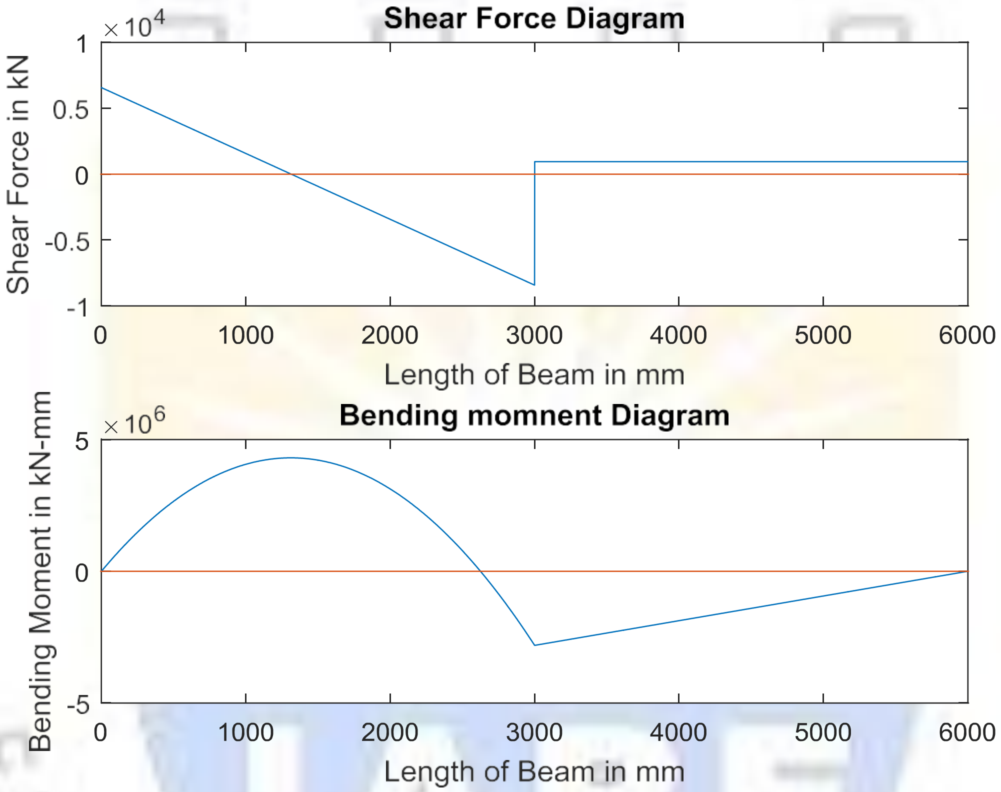
PROGRAM:

```
clear all
clc
 %----------
fprintf (' | SSB with UDL and at Half span | \n')
fprintf(' UDL \n')
fprintf(' ||||||||||||||||||||||||||\n')
fprintf('_____________________________________________________\n')
fprintf(' | | |\n')
fprintf(' |------------L------------|------------L------------| \n')
fprintf(' |---> x \n')
%Dimensions of Beam
L = 3000; %in mm
B =300;
D =300;
% Grade of Concrete
GC =25;
% Loading Conditions
UDL= 5;
%Calculations
E = 5000 * sqrt(GC);
I = (B*(D^3))/12 ;
i=0;
for x=0:0.2:(2*L)
i=i+1;
if x<=L
sf(i) = (((7*UDL*L)/16)-(UDL*x));
sf1(i)=0;
bm(i) = ((UDL*x)*((7*L)-(8*x))/16);
bm1(i)=0;
else sf(i) = (UDL*L)/16;
sf1(i)=0;
bm(i) = ((UDL*L)*(x-(2*L))/16);
bm1(i)=0;
end
end
%Plotting the Results
x=0:0.2:(2*L);
subplot(2,1,1)
plot(x,sf,x,sf1)
```

```
title('Shear Force Diagram')
xlabel('Length of Beam in mm')
ylabel('Shear Force in kN')
subplot(2,1,2)
plot(x,bm,x,bm1)
title('Bending momnent Diagram')
xlabel('Length of Beam in mm')
ylabel('Bending Moment in kN-mm')
```

OUTPUT:
| SSB with UDL and at Half span |
UDL
||||||||||||||||||||||||||||||

_______________________________________ _______________________________________
| | |
|------------L------------|------------L----------- |
|-- >x



EXPERIMENT 11

DEFLECTION OF CANTILEVER BEAM

OBJECTIVES:

a. Calculating vertical displacement with point load.
b. Calculating vertical displacement with uniformly distributedload.
c. Calculating vertical displacement with uniformly varyingload\

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in current directory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

```
clear all
clc
fprintf('Fixed beam with point load at the quarter span \n')
fprintf(' PL \n')
fprintf('| |\n')
fprintf('|__________|___________________________________________| \n')
fprintf('| | |\n')
fprintf('|----L/2---|------L/2------|------------L------------| \n')
fprintf(' |---> x \n')
 %Dimensions of Beam
L = 3000% in mm;
B = 300% in mm;
D = 300% in mm;
 % Grade of Concrete
GC = 25% in N/mm^2;
% Loading Conditions
PL= 5;
% Fixed end moments
Ma=((PL*L)/6);
Mb=((PL*L)/24);
Mc=-((PL*L)/48);
%Reactions at support
Ra=(PL/2)+((Ma+Mb)/L);
Rc=((Ma-Mb)/(2*L));
Rb=PL-Ra-Rc;
 i=0;
for x=0:0.1:(3*L)
i=i+1;
if x<(L/2);
sf(i)=Ra;
sf1(i)=0;
bm(i)=((Ra*x)-Ma);
bm1(i)=0;
else if(L/2)<x<L;
sf(i)=(Ra-PL);
sf1(i)=0;
bm(i)=(Ra*x)+(PL*(x-(L/2)))-Ma;
bm1(i)=0;
    else L<x<(3*L);
```

```
sf(i)=(Ra-PL+Rb);
sf1(i)=0;
bm(i)=((Ra*x)+(PL*(x-(L/2)))-Ma+Rb(x-L));
bm1(i)=0;
end
end
end
x=0:0.1:(3*L);
subplot(2,1,1)
plot(x,sf,x,sf1)
title('Shear ForceDiagram')
xlabel('Length of Beam in mm')
ylabel('Shear Force in kN')
subplot(2,1,2)
plot(x,bm,x,bm1)
title('Bending momnent Diagram')
xlabel('Length of Beam in mm')
ylabel('Bending Moment in kN-mm')
```

OUTPUT:

Fixed beam with point load at the quarter span

```
 PL
| |
|___________|_____________________________________________|
| | |
|----L/2---|------L/2------|------------L -----------|
 |-- >x
```

L =

3000

B =

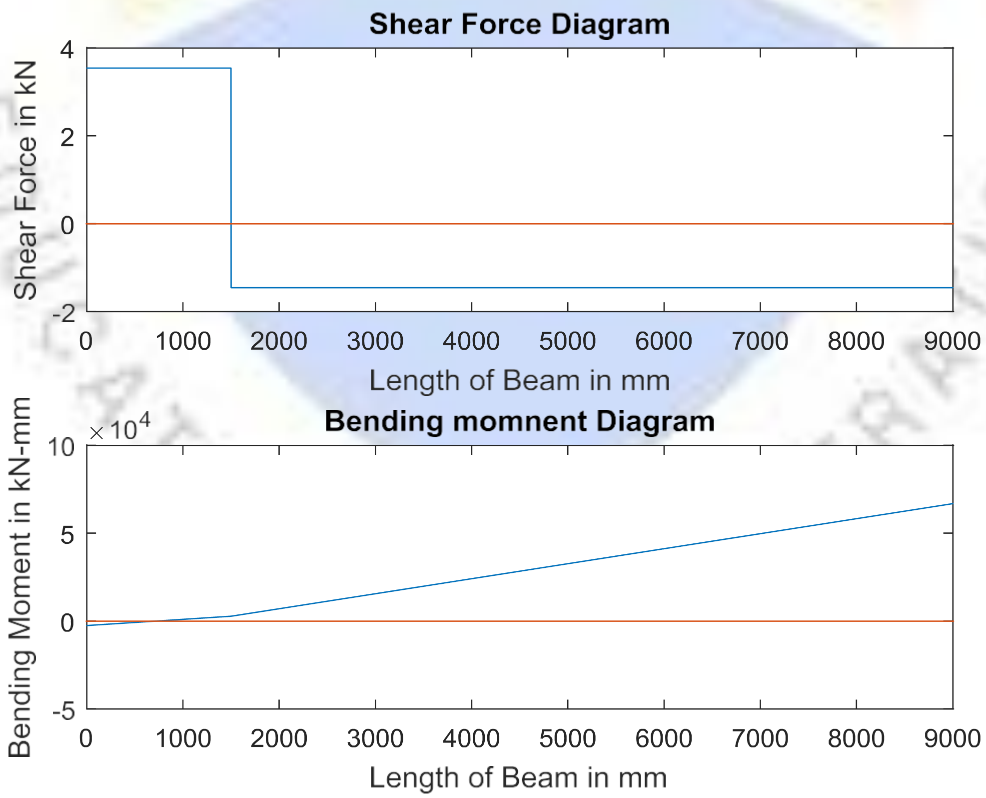
300

D =

300

GC =

25



EXPERIMENT 12

FORMULATION OF IDEAL AND REAL GAS EQUATIONS

OBJECTIVES:

a. Calculating the pressure, temperature, density for Earth's atmospheric conditions at differentaltitudes.
b. Calculating the pressure, temperature, density for other planets at differentaltitudes.

SOFTWARE REQUIRED:

1. MATLABR2013a.
2. Windows 7/XPSP2.

PROCEDURE:

1. OpenMATLAB
2. Open newM-file
3. Type theprogram
4. Save in current directory
5. Compile and Run theprogram
6. For the output see command window\ Figurewindow

PROGRAM:

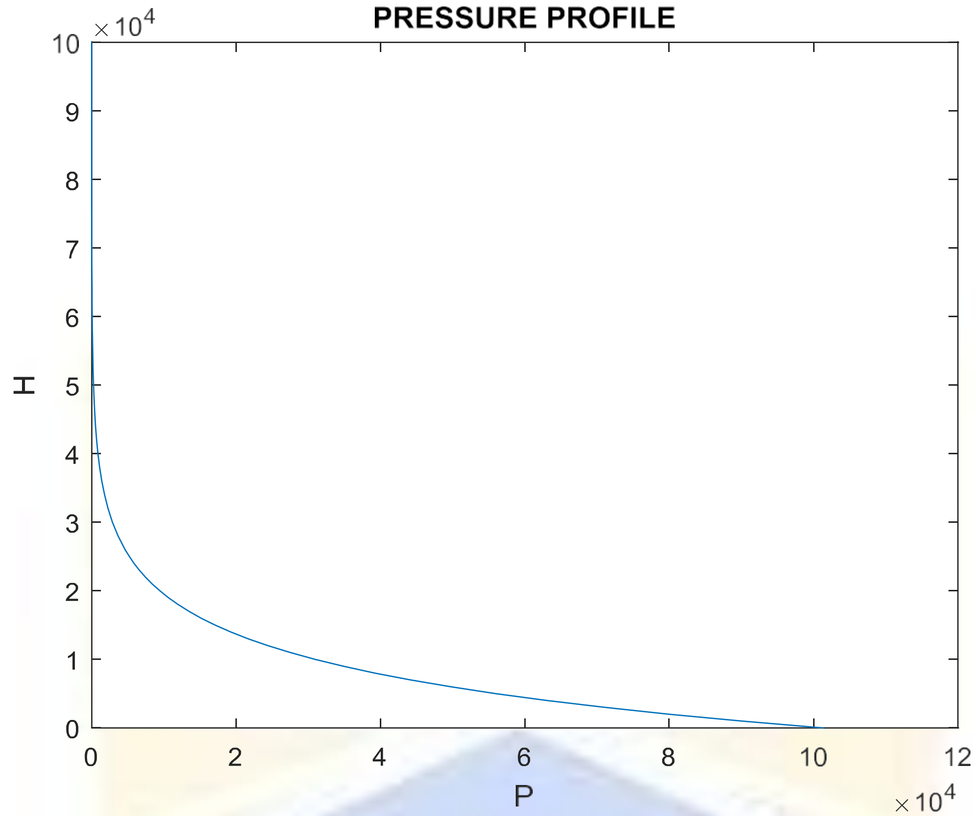
Pressure, temperature, density for Earth's atmospheric conditions at different altitudes

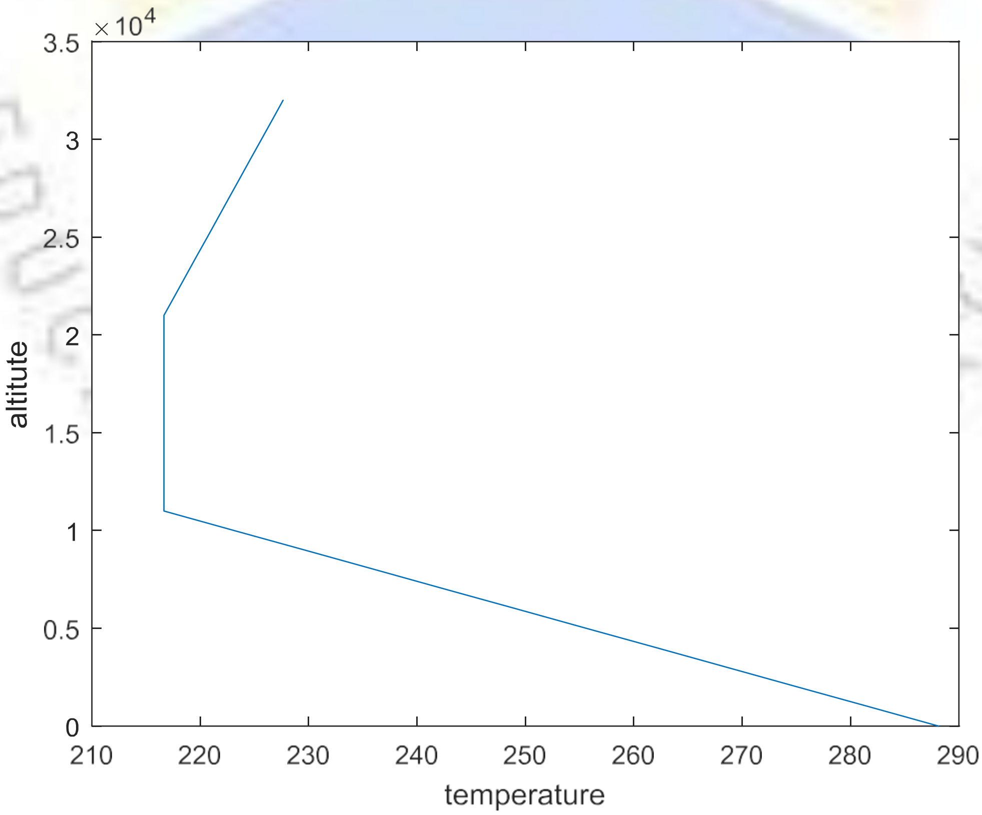
```
%PRESSURE PROFILE
clc;
clear all;
T=288.15;
G=9.81;
P1=101325;
R=287;
H=0:1000:100000;
P=P1*exp((-G*H)/(R*T));
plot(P,H)
title('PRESSURE PROFILE')
xlabel('P')
ylabel('H')

%ISA PROFILE
clc;
clearall;
t(1)=288.15;
x(1)=0;
i=1;
for h=1000:1000:32000
x(i+1)=h;
if x(i)>=0 && x(i)<11000
lp=-0.0065;
alt(i)=x(i+1)-x(i);
t(i+1)=t(i)+lp*alt(i);
elseifx(i)>=11000 && x(i)<=20000
lp=0;
alt(i)=x(i+1)-x(i);
t(i+1)=t(i)+lp*alt(i);
else
lp=0.001;
alt(i)=x(i+1)-x(i);
t(i+1)=t(i)+lp*alt(i);
end
i=i+1;
end
plot(t,x)
xlabel('temperature')
```

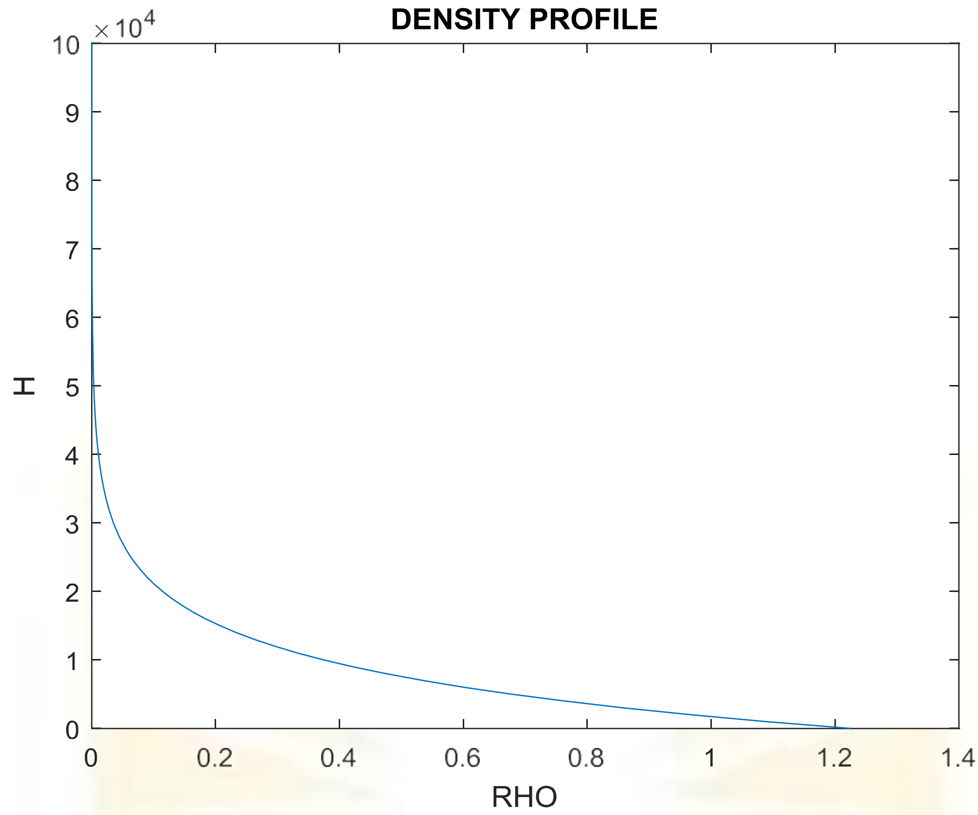
```
ylabel('altitude')
%DENSITY PROFILE
clc;
clear all;
T=288.15;
G=9.81;
RHO1=1.225;
R=287;
H=0:1000:100000;
RHO=RHO1*exp((-G*H)/(R*T));
plot(RHO,H)
title('DENSITY PROFILE')
xlabel('RHO')
ylabel('H')
```

OUTPUT:

Pressure, temperature, density for Earth's atmospheric conditions at different altitudes







DENSITY PROFILE
× 10^4
10
9
8
7
6
5
4
3
2
1
0
H
0
0.2
0.4
0.6
0.8
1
1.2
1.4
RHO