PROGRAMMING LANGUAGE FOR MATHEMATICAL MODELS

LABORATORY MANUAL
B.TECH
(III YEAR – II SEM)
(2017-18)

Prepared by:
Prof A K RAI
Dr S Suganya Devi Associate. Professor
Mr. J Sandeep, Assistant Professor

DEPARTMENT OF AERONAUTICAL ENGINEERING

MALLA REDDY COLLEGE
OF ENGINEERING & TECHNOLOGY
(Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12 (B) of UGC ACT 1956
Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified)
Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, India

CERTIFICATE

This is to certify that laboratory manual on **"Programming Language For Mathematical Models LAB"** has been prepared as per syllabus B.Tech (ANE) R-15. It covers all the experiments listed in the syllabus**.**

A K Rai
Prof

Dr Suganya Devi
Prof

J Sandeep
Asst Prof

MNVS Swetha Bala
HOD(ANE)

Dr M Murali Krishna
Dean Academics

Dr VSK Reddy
Principal

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

III Year B. Tech, ANE-II Sem

L	T/P/D	C
0	-/3/-	2

(R15A2185)PROGRAMMING LANGUAGE FOR MATHEMATICAL MODELS LAB

Objectives: -

The course should enable the students to:

- Learn the mathematical programming language.
- Learn the problem-solving techniques
- Develop skills in programming language

LIST OF EXPERIMENTS: -

1. Introduction to modeling software.
2. Programs using mathematical functions and plotting functions.
3. Program to solve differential equations.
4. Program to solve system of equations using numerical methods.
5. Program to generate airfoil coordinates.
6. Program to find critical Mach number of an airfoil and to generate drag polar graph.
7. Program to find flow characteristics across shock waves.
8. Program to calculate the performance of turbofan.
9. Program to find the flow characteristics of a CD nozzle.
10. Program to calculate the deflection, bending moment, shear force in a beam.
11. Determine the buckling load of a column with different end conditions.
12. Find out displacements of a uniform bar/stepped bar subjected to mechanical/thermal loads.

Note: Any 10 Experiments can be conducted.

Equipment Needed:

1. **Computers**: Core 2 duo processor with 1 GB RAM
2. **Software:** MATLAB or scilab or equivalent software

Reference Books:

1. MATLAB an Introduction with Applications Fifth Edition AMOS GILAT by WILEY Publications
2. Programming in SCI lab by VINU V DAS NEW AGE INTERNATIONAL PUBLICATIONS

Outcomes:

The student should be able to

- Model aerospace problems into mathematical models.
- Revise computational strategies for developing applications.
- Develop applications (Simple to Complex) using programming language.

DEPARTMENT OF AERONAUTICAL ENGINEERING

VISION

Department of Aeronautical Engineering aims to be indispensable source in Aeronautical Engineering which has a zeal to provide the value driven platform for the students to acquire knowledge and empower themselves to shoulder higher responsibility in building a strong nation.

MISSION

a) The primary mission of the department is to promote engineering education and research.

(b) To strive consistently to provide quality education, keeping in pace with time and technology.

(c) Department passions to integrate the intellectual, spiritual, ethical and social development of the students for shaping them into dynamic engineers.

PROGRAMME EDUCATIONAL OBJECTIVES (PEO'S)

PEO1: PROFESSIONALISM & CITIZENSHIP To create and sustain a community of learning in which students acquire knowledge and learn to apply it professionally with due consideration for ethical, ecological and economic issues.
PEO2: TECHNICAL ACCOMPLISHMENTS To provide knowledge based services to satisfy the needs of society and the industry by providing hands on experience in various technologies in core field.
PEO3: INVENTION, INNOVATION AND CREATIVITY To make the students to design, experiment, analyze, interpret in the core field with the help of other multi disciplinary concepts wherever applicable.
PEO4: PROFESSIONAL DEVELOPMENT To educate the students to disseminate research findings with good soft skills and become a successful entrepreneur.
PEO5: HUMAN RESOURCE DEVELOPMENT To graduate the students in building national capabilities in technology, education and research.

PROGRAM SPECIFIC OBJECTIVES (PSO's)

1. To mould students to become a professional with all necessary skills, personality and sound knowledge in basic and advance technological areas.
2. To promote understanding of concepts and develop ability in design manufacture and maintenance of aircraft, aerospace vehicles and associated equipment and develop application capability of the concepts sciences to engineering design and processes.
3. Understanding the current scenario in the field of aeronautics and acquire ability to apply knowledge of engineering, science and mathematics to design and conduct experiments in the field of Aeronautical Engineering.
4. To develop leadership skills in our students necessary to shape the social, intellectual, business and technical worlds.

PROGRAM OBJECTIVES (PO'S)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **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.
3. **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.
4. **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.
5. **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.
6. **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.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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.
11. **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 multi disciplinary environments.
12. **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.

CONTENTS

CODE OF CONDUCT FOR THE LABORATORIES

- All students must observe the Dress Code while in the laboratory.
- Sandals or open-toed shoes are NOT allowed.
- Foods, drinks and smoking are NOT allowed.
- All bags must be left at the indicated place.
- The lab timetable must be strictly followed.
- Be PUNCTUAL for your laboratory session.
- Program must be executed within the given time.
- Noise must be kept to a minimum.
- Workspace must be kept clean and tidy at all time.
- Handle the systems and interfacing kits with care.
- All students are liable for any damage to the accessories due to their own negligence.
- All interfacing kits connecting cables must be RETURNED if you taken from the lab supervisor.
- Students are strictly PROHIBITED from taking out any items from the laboratory.
- Students are NOT allowed to work alone in the laboratory without the Lab Supervisor
- USB Ports have been disabled if you want to use USB drive consult lab supervisor.
- Report immediately to the Lab Supervisor if any malfunction of the accessories, is there.

Before leaving the lab

- Place the chairs properly.
- Turn off the system properly
- Turn off the monitor.
- Please check the laboratory notice board regularly for updates.

EXPERIMENT -1

INTRODUCTION TO MATLAB SOFTWARE

Aim: To study and solve following types of problems using MATLAB

(a) Simple arithmetic operations using command window.

(b) Creating arrays and mathematical operations with arrays(dot and cross product, inverse, transpose, Eigen values, solutions of linear equations).

(c) Creating simple script file.

Equipment and material needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher. In addition, thorough understanding of chapter 1, 2 and 3 of Reference book 1(MATLAB: An Introduction with Applications by Amos Gilat).

Brief introduction about MATLAB and algorithm: The MATLAB® is a very popular language for technical computing by students, engineers, and scientists in universities, research institutes, and industries all over the world. The software is popular and easy to use. A brief on few features are explained below. For complete coverage you should go through all the chapters of Reference 1. These will also be explained to you during the LAB.

It is utmost important that you bring this reference book during the LAB for ease of understanding and quick reference. It will be of help if the concerned faculty keeps the copy of the book in the Lab for their reference and demonstration.

Command Window: Command Window is used to enter variables and to run functions and M-file scripts.

Command History: Statements entered in the Command Window are logged in the Command History. From the Command History, one can view and search for previously run statements, as well as copy and execute selected statements. One can also create an M-file from selected statements.

Current Directory: A quick way to view or change the current directory is by using the current directory field in the desktop toolbar

Workspace: The MATLAB® workspace consists of the set of variables built up during a MATLAB session and stored in memory. Variables are added to the workspace by using functions, running M-files, and loading saved workspaces.

Editor: Editor is used to create and debug M-files, which are programs we write to run MATLAB® functions. The Editor provides a graphical user interface for text editing, as well as for M-file debugging. To create or edit an M-file one use File > New or File > Open, or use the edit function.

Keep the following points in mind when working with MATLAB.

- Upper and lower-case characters are not equivalent (MATLAB is case sensitive).
- Typing the name of a variable will cause MATLAB to display its current value.
- A semicolon at the end of a command suppresses the screen output.
- MATLAB uses both parentheses, (), and square brackets, [], and these are not interchangeable.
- The up arrow and down arrow keys can be used to scroll through previous commands. Also, an old command can be recalled by typing the first few characters followed by up arrow.
- One can type help topic to access online help on the command, function or symbol topic.
- You can quit MATLAB by typing exit or quit.
- First character of the saved M-file should be a letter and not a number. So do not save file using your roll no in the beginning of the file name.
- Having entered MATLAB, you should work through this tutorial by typing in the text that appears after the MATLAB prompt, >>, in the Command Window. After showing you what to type, we display the output that is produced. Faculty will demonstrate and help you in thefollowing tutorial.

  Defining Scalar Variable: Do not use reserved key words as variables (e.g. break,else, while etc.).

```
>> x=15;
X=15
>> x=3*x-12
X=
33
>>E=sin(x)^2+cos(x)^2
E=
1
>>
```

Creating Arrays and Mathematical Operations with Arrays: The array is a fundamental form that MATLAB uses to store and manipulate data. It is a list of numbers arranged in rows and/or columns. The simplest array is a row or a column of numbers(one-dimensional). A more complex array (2-D) is a collection of numbers arranged in rows and columns. In science and engineering 1-D arrays frequently represents vectors, and two-dimensional arrays often represent **matrices**. Faculty will demonstrate to you how to create and perform mathematical operations on arrays.

-Create an array from given population data in Table 2-1: (Referene-1 text book chapter-2).

Table 1: Population data

Year	1984	1986	1988	1990	1992	1994	1996
Population(millions)	127	130	136	145	158	178	211

>>yr= [1984,1986,1988,1990,1992,1994,1996]

yr=

1984 1986 1988 1990 1992 1994 1996

pop=[127,130,136,145,158,178,211]

-create a vector by specifying first term,the spacing and the last term.

Example:

> X=[1:2:13]

X

1 3 5 7 9 11 13

-create a vector using linear spacing by specifying the first and last terms, and the number of terms:

Va= linspace(0,8,6)

Va=

0 1.6000 3.2000 4.8000 6.4000 8.0000

Some of the useful commands for matrix manipulations are tabulated below with examples.

Table - 2: Useful functions on Matrix operations

S.No	**Command**	**Description**
1	A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]	Matrix
2	A'	Transpose of Matrix
3	diag(A)	Diagonal elements of matrix
4	A(4,2)	the number in the fourth row and second column
5	100:-7:50	a row vector containing the integers from 100 to 50 with decrement of 7
6	sum(A(1:4,4))	computes the sum of the fourth column
7	Z=zeros(2,4)	2x4 matrix with all zeros
8	C=ones(1,3)	1x3 matrix with all ones
9	A(:,2) = []	Deleting second row
10	E = A([1,1,1],:)	copies the first row of *A three* times to create a new matrix
11	det(A)	Determinant of matrix

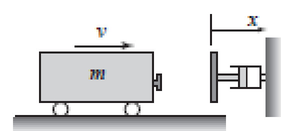
12	X = inv(A)	Inverse of a matrix
13	e = eig(A)	Eigen values of a matrix
14	eye(3x3)	Creates 3x3 Identity matrix
15	linspace(a,b,n)	creates a row vector of *n* regularly spaced elements between *a* and *b*
16	function [out1, out2, ...] = funname(in1, in2, ...)	*out1, out2, ...,* are the function outputs, *in1, in2, ...* are its inputs and *funname* is the function name then the function can be called in the command window or in other m-files.

Creating script file (M-file): A script file is a list of MATLAB commands, called a program that is saved in a file. When the script file is executed(run), MATLAB executes the commands. You will be demonstrated by the faculty, how to create, save and run a simple script file in which commands are executed in order in which they are listed, and in which all the variables are defined within the script file. On similar line we can write functions for various applications.

Result: -Students should be able to formulate and solve simple problems using various functions including Arrays and Matrices available in MATLAB.

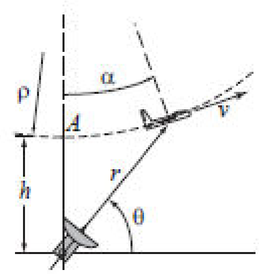
EXERCISES

1. A railroad bumper is designed to slow down a rapidly moving railroad car. After a 20,000 kg railroad car traveling at 20 m/s engages the bumper, its displacement *x* (in meters) and velocity *v* (in m/s) as a function of time *t* (in seconds) is given by:



$x(t) = 4.219(e^{-1.58t} - e^{-6.32t})$ and $v(t) = 26.67e^{-6.32t} - 6.67e^{-1.58t}$
Determine *x* and *v* for every two hundredth of a second for the first half second after impact. Display the results in a three-column table in which the first column is time (s), the second is displacement (m), and the third is velocity (m/s).

2. The airplane shown is flying at a constant speed of v = 50 m/s in a circular path of radius ρ = 2000 m and is being tracked by a radar station positioned a distance h = 500 m below the bottom of the plane path (point *A*). The airplane is at point *A* at t = 0, and the angle α as a function of time is given (in radians) by $\alpha = \frac{v}{\rho}t$. Write a MATLAB program that calculates θ and *r* as functions of time. The program should first determine the time at which $\alpha = 90^0$. Then construct a vector *t* having 15 elements over the interval $0 \le t \le t_{90}{}^0$, and calculate and *r* at each time. The program should print the values of , *h*, and *v*, followed by a15x3 table where the first column is *t*, the second is the angle θ in degrees, and the third is the corresponding value of *r*.



3. The balance of a loan, *B*, after *n* monthly payments is given by

$$B = A\left(1 - \frac{r}{1200}\right)^n - \frac{P}{r/1200}\left[\left(1 + \frac{r}{1200}\right)^n - 1\right]$$

where *A* is the loan amount, *P* is the amount of a monthly payment, and *r* is the yearly interest rate entered in % (e.g., 7.5% entered as 7.5). Consider a 5-year, $20,000 car loan with 6.5% yearly interest that has a monthly payment of $391.32. Calculate the balance of the loan after every 6 months (i.e., at *n* = 6, 12, 18, 24, ... , 54, 60). Each time calculate the percent of the loan that is already paid. Display the results in a three-column table, where the first column displays the month, and the second and third columns display the corresponding value of *B* and percentage of the loan that is already paid, respectively.

4. The surface of many airfoils can be described with an equation of the form

$$y = \mp\frac{tc}{0.2}\left[a_0\sqrt{x/c} + a_1\left(\frac{x}{c}\right) + a_2(x/c)^2 + a_3(x/c)^3 + a_4(x/c)^4\right]$$

Where t is the maximum thickness as a fraction of the chord length c. Given tat c = 1m and t = 0.2 m, the following values for y have been measured for a particular airfoil:

x(m)	0.15	0.35	0.5	0.7	0.85
Y(m)	0.08909	0.09914	0.08823	0.06107	0.03421

Determine the constants a_0, a_1, a_2, a_3 and a_4. (Write a system of five equations and five unknowns and use MATLAB to solve the equations)

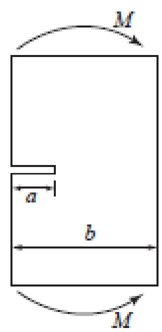
5. The wind chill temperature, , is the air temperature felt on exposed skin due to wind. In U.S. customary units it is calculated by:

$$T_{wc} = 35.74 + 0.6215T - 35.75v^{0.16} + 0.4275Tv^{0.16}$$

where *T* is the temperature in degrees F, and *v* is the wind speed in mi/h. Write a MATLAB program in a script file that displays the following chart of wind chill temperature for given air temperature and wind speed in the Command Window:

	Temperature (F)								
Speed (mi/h)	40	30	20	10	0	-10	-20	-30	-40
10	34	21	9	-4	-16	-28	-41	-53	-66
20	30	17	4	-9	-22	-35	-48	-61	-74
30	28	15	1	-12	-26	-39	-53	-67	-80
40	27	13	-1	-15	-29	-43	-57	-71	-84
50	26	12	-3	-17	-31	-45	-60	-74	-88
60	25	10	-4	-19	-33	-48	-62	-76	-91

6. The stress intensity factor due to the crack shown depends upon a geometrical parameter C_l given by: $C_l = \sqrt{\frac{2}{\pi\alpha}\tan\frac{\pi\alpha}{2}}\left[\frac{0.923+0.199\left(1-\sin\frac{\pi\alpha}{2}\right)}{\cos\frac{\pi\alpha}{2}}\right]$ where α = a/b. Calculate C_l for α between 0.05 and 0.95 at 0.05 increments and display the results in a two column table with the first column showing α and the second C_l



7. Compute the array and matrix product of $A = \begin{bmatrix} 8 & 7 & 11 \\ 6 & 5 & -1 \\ 0 & 2 & -8 \end{bmatrix}$, $B = \begin{bmatrix} 2 & 1 & 2 \\ -1 & 6 & 4 \\ 2 & 2 & 2 \end{bmatrix}$

8. Find a solution to the following set of equations:

$x + 2y + 3z = 12$
$-4x + y + 2z = 13$
$9y - 8z = -1$

What is the determinant of the coefficient matrix?

9. Write a function 'altitude' which takes static pressure in millibar as input argument and computes the pressure altitude in meters using standard atmosphere. (Kindly refer to any book on aerodynamics for relation between pressure and altitude in standard atmosphere.

EXPERIMENT -2

PROGRAMS FOR TWO-DIMENSIONAL (2-D) AND THREE-DIMENSIONAL (3-D)PLOTTING

Aim: To study the programs for creating 2D & 3D Plots.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher

Theory:

Plotting: Plots are very useful tool for presenting information. This is true in any field, but specially in science and engineering, where MATLAB is mostly used. MATLAB has many commands that can be used for creating different types of plots. These include standard plots with linear axes, plots with logarithmic and semi-logarithmic axes, polar plots, 3-D contour surfaces and mesh plots, and many more. In this experiment you will learn how MATLAB can be used to create and format many types of 2-D and 3-D plots.

2-D Plot of a Function. In many situations there is a need to plot a given function. This can be done by using the 'plot' or the 'fplot' command. In order to plot a function y =f(x) with the plot command, the user needs to first create a vector of x for the domain over which the function will be plotted. Then a vector y is created with the corresponding values of f(x) by using element-by-element calculations as explained in Experiment number 1. Once the two vectors are defined, they can be used in the plot command. The fplot command plots a function with the form y=f(x) between specified limits. The command has the form

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

'function' can be typed directly as a string inside the command. For example if the function that is being plotted is f(x) = 8x^2+5 cos(x),it is typed as:'8*x^2+5*cos(x)'. the function can include MATLAB built-in functions and functions that are created by the users.

Plotting multiple graphs in the same plot. In many situations there is a need to make several graphs in the same plot. There are three methods to plot multiple graphs in one figure. One is by using the plot command, the second by using the hold on and hold off and the third is by using the line command.

Three-Dimensional Graphics: MATLAB provides a variety of functions to display 3-D data. Some functions plot lines in 3-D, while others draw surfaces and wire frames.In addition, color can be used to represent a fourth dimension. When color is used in this manner, it is called pseudo color, since color is is not inherent or natural property of the underlying data in the way that color in a photograph is natural characteristic of the image.

(a) **LINE PLOTS**. General format of the command is plot3(x1,y1,z1,S1,x2,y2,z2,S2,…) where xn,yn and zn are vectors or matrices and Sn are optional character strings specifying color, marker symbol,and or line style.

(b) **MESH PLOT**. MATLAB defines a mesh surface by the z-coordinates of points above a rectangular grid in the x-y plane. It formats a mesh plot by joining adjacent points with straight lines. The result looks like a fishing net with knots at the data points.

(c) **SURFACE PLOTS**. A surface plot is like a mesh plot, except that the spaces between the lines called patches are filled in. Plots of this type are generated using the surf function.

All the above features will be demonstrated to you by the faculty with examples discussed below. Some important commands for plotting are tabulated below for quick reference.

Table 2: Important plot commands in MATLAB

S.No	Command	Description
1	plot(*x*, *y*)	Plots the variation of y with respect to x
2	xlabel('x')	Gives the label for x axis
3	ylabel('cos(x)')	Gives the label for y axis
	title('plot name')	Gives the name of plot
4	fplot ('function string,' [*x*start, *x*end])	The function fplot gets around our choice of interval used to generate the plot, and instead decides the number of plotting points to use for us.
5	plot(t,f,t,g,'--')	To plot multiple functions, call the plot(*x, y*) command with multiple pairs *x, y* defining the independent and dependent variables used in the plot in pairs. This is followed by a character string enclosed in single quotes to tell us what kindof line to use to generate the second curve
6	'Linewidth'2	Increases thickness of curve
7	legend('sinh(x)','cosh(x)')	To name the curves
8	plot(x,y,'r',x,z,'b')	Differentiates the curves with colors r-red, b-blue, g-green, k-black, w-white, y-yellow, m-magenta, c-cyan.
9	axis([xmin xmax ymin ymax])	Plot range
10	subplot(1,2,1)	Creates the plot with 2 panes and 1 row, and that this particular plot will appear in the first pane
11	polar (theta, r)	Creates Polar plots
12	bar(x,y)	Creates bar chart
13	[x,y] = meshgrid(–5:0.1:5,–3:0.1:3);	Generate a matrix of elements that give the range over *x* and *y* we want to use along with the specification of increment in each case.
14	mesh(x,y,z)	3D plots
15	surf(x,y,z)	Shaded surface plots

Example 1: Plotting multiple graphs in the same plot using fplot and hold command.

Plot three sine waves with different phases. For the first, use a line width of 2 points. For the second, specify a dashed red line style with circle markers. For the third, specify a cyan, dash-dotted line style with asterisk markers.

Code:

```
fplot(@(x) sin(x+pi/5),'Linewidth',2);
holdon
fplot(@(x) sin(x-pi/5),'--or');
fplot(@(x) sin(x),'-.*c')
holdoff
```

Result: -The plot is shown in Fig 2.1**.** The default limit for x : -5,5

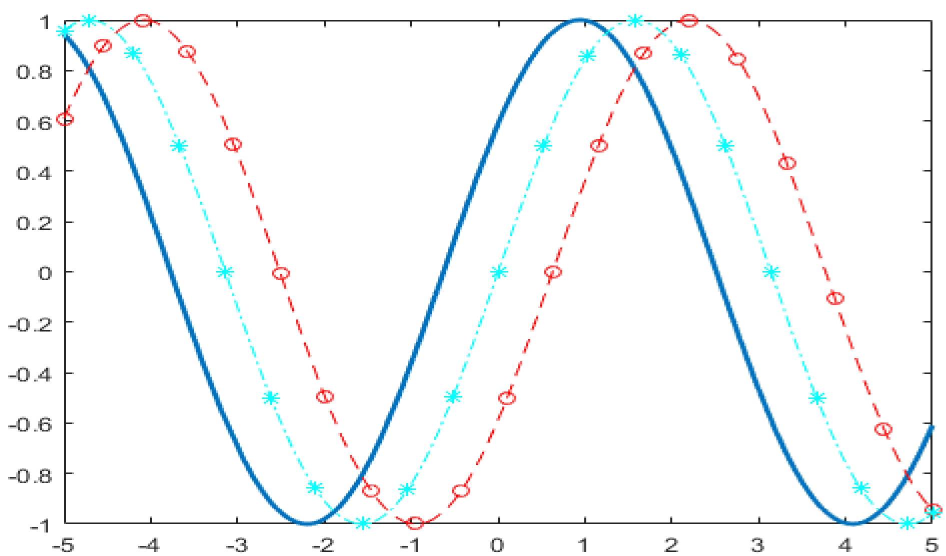


Fig 2.1

Example 2: Polar plot of a function r=3 cos^2(0.5θ) + θ for 0<= θ<= 2*pi
t= linspace (0,2*pi,200);
r=3*cos(0.5*t).^2+t;
polar(t, r)

Result: polar plot is shown in Fig 2.2

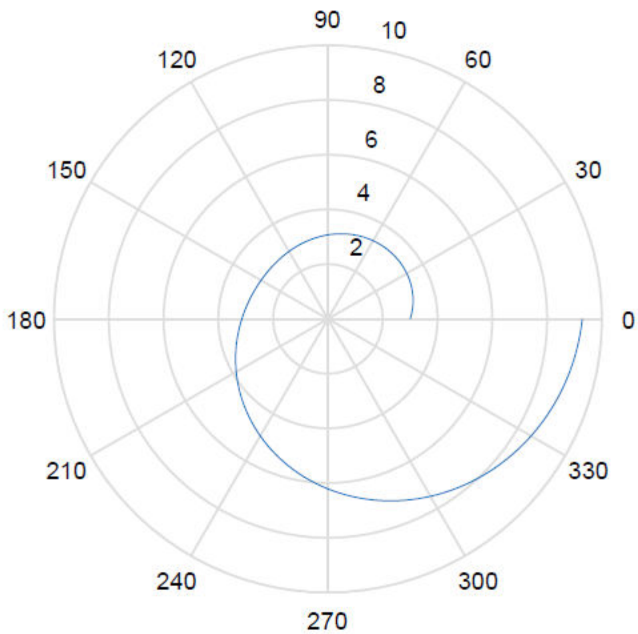
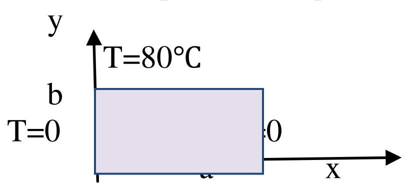


Fig 2.2 Polar plot

Example 3 for 3-D plot. Three sides of a rectangular plate (a=5m, b=4m) are kept at temperature of 0 deg C and one side is kept at a temperature T1=80 deg C, as shown in the figure. Determine and plot the temperature distribution T(x,y) in the plate.



T=0

Solution. The temperature distribution, T(x,y) in the plate can be determined by solving the 2-D heat equation. For given boundary conditions T9x,y) can be expressed analytically by a Fourier series:

$$T(x,y)=\frac{4T_1}{\pi}\sum_{n=1}^{\infty}\frac{\sin\left[(2n-1)\frac{\pi x}{a}\right]}{(2n-1)}\frac{\sinh\left[(2n-1)\frac{\pi y}{a}\right]}{\sinh\left[(2n-1)\frac{\pi b}{a}\right]}$$

A program in a script file that solves the problem is listed below. The program follows these steps:

(a) Create an X,Y grid in the domain $0\leq x \leq a$and $0\leq y \leq b$. The length of the plate, a, is divided into 20 segments, and the width of the plate, b, is divided into 16 segments.

(b) Calculate the temperature at each point of the mesh. The calculation are done point by point using a double loop. At each point the temperature is determined by adding k terms of the Fourier series.

(c) Make a surface plot of T.

```
% 3-D plot for heat equation PLMM lab
%
% script file
 a=5;
 b=4;na=20;nb=16;T0=80;k=5;
x=linspace(0,a,na);
y=linspace(0,b,nb);
[X,Y]=meshgrid(x,y);
for i=1:nb
for j=1:na
   T(i,j)=0;
for n=1:k
   ns=2*n-1;      % third loop, n, is the
% term of the Fourier series,% k is the number of  %terms
T(i,j)=T(i,j)+sin(ns*pi*X(i,j)/a).*sinh(ns*pi*Y(i,j)/a)/(sinh(ns*pi*b/a)...
*ns);
end
  T(i,j)=T(i,j)*4*T0/pi;
end
end
mesh(X,Y,T)
xlabel( 'x  (m)'); ylabel('y  (m) ');
zlabel(' T  ( ^0C) ')
```

Program was executed with two different values of k (5 &50). The mesh plots are shown in each case in the figure 2.3 and 2.4. the temperature should be uniformly at y=4 m. Note the effect of k on the accuracy at y=4m.

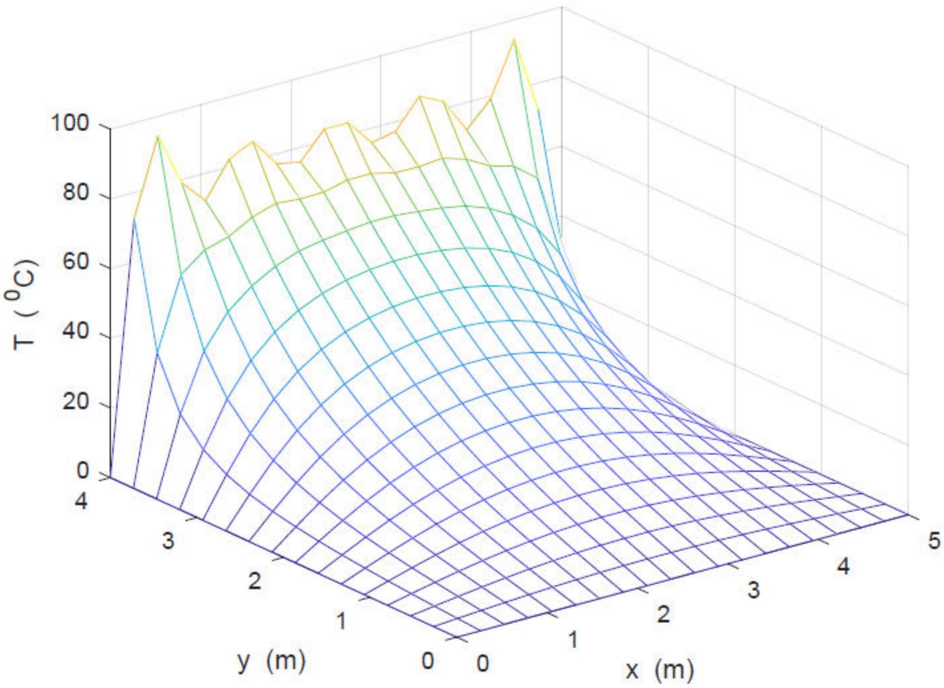


Fig 2.3: for k=5;

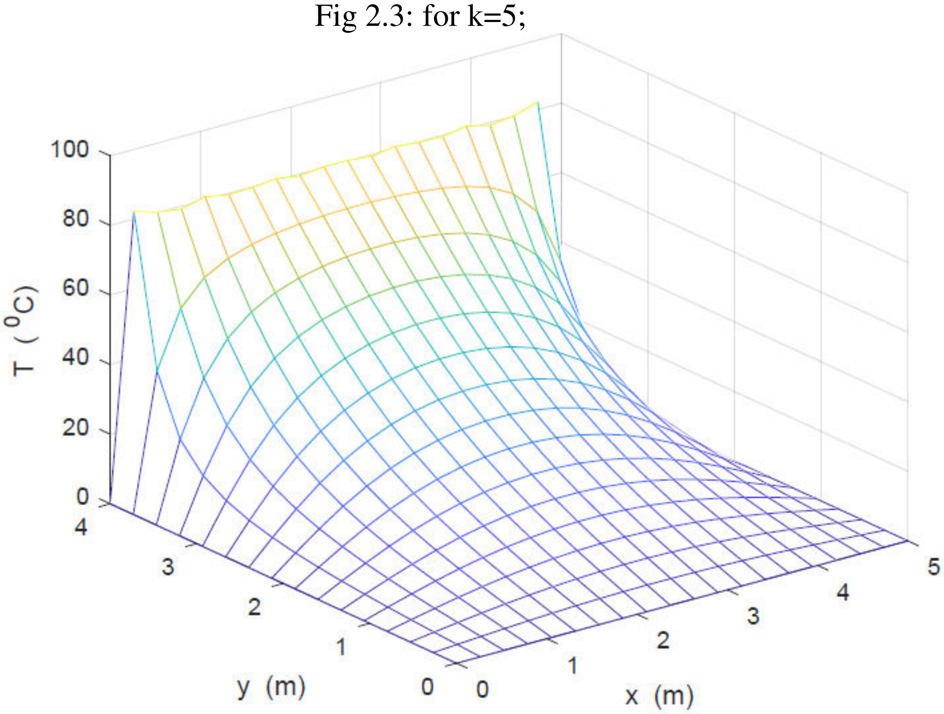


Fig 2.4: For k=50

EXERCISES

1. Acycloid is a curve traced by a point on a circle that rolls along a line. The parametric equation of a cycloid is given by

x=r(t-sint) and y=r(1-cost)

Plot a cycloid with r=1.5 and $0 \leq t \leq 4\pi$

2. Solve the problem No 24 relating to NACA airfoil given in chapter 5 of reference book 1.

3. Solve the problem no 31 relating to simply supported beam given in Chapter 5 of reference 1

4. Solve the problem no 26 relating to vibrations of chapter 5 of reference 1.

5. Solve the problem no 19 of chapter 5 of reference 1 relating to tensile strength**.**

6. Solve the problem no 14 of chapter 10 of reference 1 relating to defect in crystal lattice.

7. Solve problem No 19 of chapter 10 of reference book 1.

8. In a study of the effect of various factors on the growth performance of activated sludge, the oxygen uptake rate was measured at various temperatures.

Temperature °C	Oxygen uptake rate grams oxygen per gram dry Weight
5	0.01
10	0.04
15	0.10
20	0.20
25	0.25
30	0.28
35	0.30
40	0.25
45	0.02

Write a script M-file that generates a plot of these data, including title, labels, and grid. Print the generated graph.

EXPERIMENT 3

PROGRAM TO SOLVE DIFFERENTIAL EQUATIONS

Aim: To learn various functions available in MATLAB to solve initial value problems(IVPs) and Boundary value problems(BVPs).

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher.

Theory:

MATLAB has the capability to solve a wide variety of problems involving differential equations. In this exercise we will learn how to solve IVPs because they appear most often in applications. The IVP solvers in MATLAB compute the time history of a set of coupled first-order differential equations with known initial conditions. In mathematical terms, these problems have the form

$$\dot{\boldsymbol{y}}=\mathbf{f}(\boldsymbol{t},\boldsymbol{y}) \qquad \mathbf{y}(\boldsymbol{t_0})=\boldsymbol{y_0}$$

Which is vector notation for the set of differential equations

$\dot{y_1}= f_1(t, y_1, y_2 \dots y_n)\ \ y_1(t_0)=y_{10}$

$\dot{y_2}= f_2(t, y_1, y_2 \dots y_n)\ ,\ y_2(t_0) =y_{20}$

$\dot{y_n}= f_n(t, y_1, y_2 \dots y_n)\ \ y_n(t_0) =y_{n0}$

Where $\dot{y_i} =\frac{dy_i}{dt}$, n is the number of first order differential equations and y_{i0} is the initial conditions associated with the ith equations. When an initial value problem is not specified as a set of first order differential equations, it must be rewritten as one. For example, consider the classic **van der Pol equation**

$\ddot{x}$ -μ$(1 - x^2)\dot{x}$ + x=0;

Where μis a parameter greater than zero. If we choose y_1=x and y_2 =dx/dt, the van der Pol equation becomes

$\dot{y_1}= y_2$

$\dot{y_2}= \mu(1 - {y_1}^2)y_2 - y_1$

ODE Suite Solvers:

The MATLAB ODE suite offers various initial value problems solvers. Each has characteristics appropriate for different initial value problems. The calling syntax for each solver is identical, making it relatively easy to change solvers for a given problem. Some of them are given below

(a) **Ode23**: an explicit one-step Runge-Kutta low-order (2^{nd}-to 3^{rd}-order) solver. Suitable for problems that exhibit mild stiffness, problems where lower accuracy is acceptable, or problems where f(t,y) is not smooth(e.g. discontinuities)

(b) **Ode23s**: an implicit one-step modified Rosen Brock solver of order two. Suitable for stiff problems where lower accuracy is acceptable or where f(t,y) is discontinuous. **Stiff are generally described as problems in which the underlying time constants vary by several orders of magnitude or more**.

(c) **Ode23tb**: An implicit trapezoidal rule followed by a backward differentiation of order two. Similar to ode23s. it can be more efficient than ode15s for crude tolerance.

(d) Ode45: An explicit one-step Runge Kutta medium-order (4^{th}- to 5^{th}-order) solver. Suitable for non-stiff problems that require moderate accuracy. **This is typically the first solver to try on a new problem.**

Algorithm: Before a set of differential equations can be solved, they must be coded in a function M-file as ydot = odefile(t,y).That is, the file must accept a time t and a solution y and return values for the derivatives.

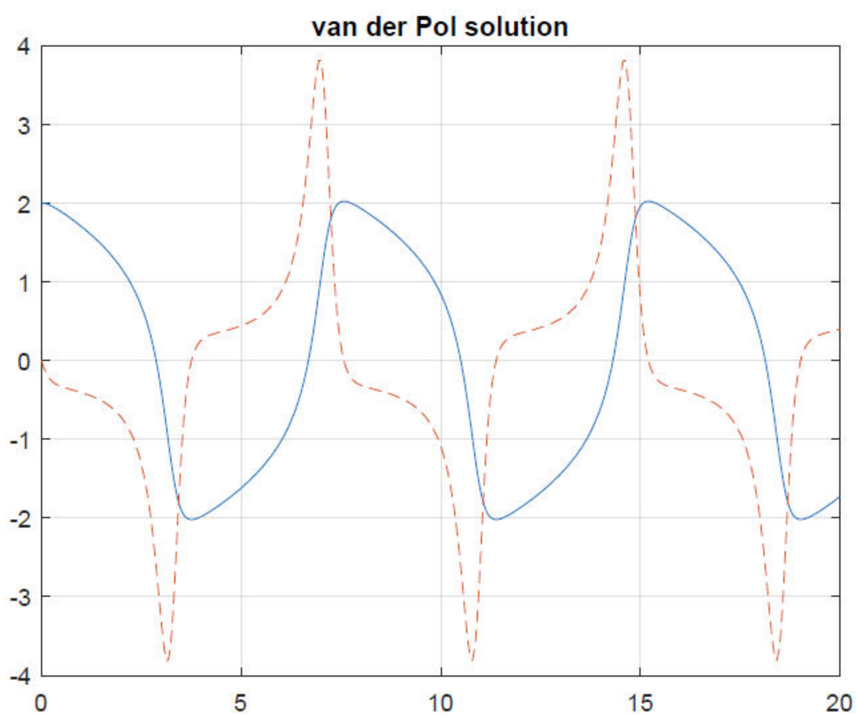
Example: Solve the van der Pol equation using ode45 function. For the van der Pol equation, create ODE function file as written below:

```
function ydot= vdpol(t,y)

%VDPOL van der pol equation.

% Ydot= VDPOL(t ,Y)

% Ydot(1) =Y(2)

% Ydot(2) =mu*(1-Y(1)^2)*Y(2)-Y(1)

% mu=2

Mu=2;

Ydot= [ y(2) ; mu*(1-y(1)^2)*y(2)-y(1)];

end
```

Note that the input arguments are t and y, but that this particular function does not use t. Note also that the output ydot must be a column vector.
Given the preceding ODE file, this set of ODEs is solved by using the following commands**:**

```
>>tspan=        [0 20];            % time span to integrate over
>>y0=[2;0];                          % initial conditions(must be column)
>>[t,y]=ode(@vdpol,tspan,y0);
>>size(t)                    % number of time points
Ans=
333      1
>>size(y)
Ans=
333                 2
>>plot(t,y(:,1),t,y(:,2),'—')
>>)
>>title(' van der Pol Solutions')
```

Plot is shown on the next page.



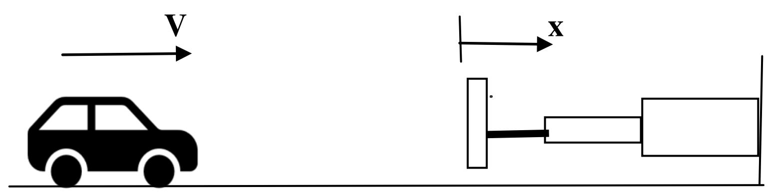
You can try with μ=0.8 and different initial conditions (e.g 0.1, 0 .1, and 2,3). Plot y(1) on x-axis and y(2) on y-axis. What do you conclude about van der Pol equation?

EXERCISES

1. A safety bumper is placed at the end of a race track to stop out-of-control cars. The bumper is designed such that the force that the bumper applies to the car is a function of the velocity v and the displacement x of the front edge of the bumper according to the equation:

$$F = K v^3 (x+1)^3$$

Where K=30 (s kg)/m ^{5} is a constant.A car with a mass m of 1,500 kg hits the bumper at a speed of 90 km/h. Determine and plot the velocity of the car as a function of its position for $0 \leq x \leq 3$ m.



2. Problem No 35(pertaining to airplane parachute) of chapter 9 of Reference No 1.
3. Problem No 30 of reference No 1 chapter 9.
4. Problem No 31 of Reference No 1 chapter 9.

EXPERIMENT 4

SYSTEMS OF EQUATIONS USING NUMERICAL METHODS

Aim: To learn various methods for solving systems of equations using numerical methods.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher.

Theory: Numerical methods are commonly used for solving mathematical problems that are formulated in science and engineering where it is difficult or impossible to obtain exact solutions. MATLAB has a large library of functions for numerically solving a wide variety of mathematical problems. Faculty will demonstrate methods of solving integration problems, finding minimum or maximum of a function

Some useful functions for numerical methods are

Command	**Description**
integral	integrate a function
feval	evaluate the value of a math function
fzero	solves an equation with one variable
fminbnd	determines the minimum of a function
trapz	used for integrating functions that is given as data points.

Example: Use numerical integration to calculate the following integral

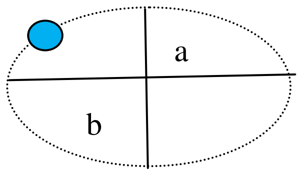
$$\int_0^8 (xe^{-.8x} + 0.2)\,dx$$

```
>> f=@(x) x.exp(-x.^0.8)+0.2;

>> integral(f,0,8)
Ans=
  3.11604
```

EXERCISES

1. The orbit of Pluto is elliptical in shape, with a=5.9065 $\times 10^9$ km and b=5.7208 $\times 10^9$ km. The perimeter of an ellipse can be calculated by

   P=4a$\int_0^{\frac{\pi}{2}} \sqrt{1 - k^2 \sin\theta^2}\,d\theta$



2. Solve problem no 19 of chapter 9 of Reference 1.

3. Solve the problem no 20 of Reference No 1.

EXPERIMENT -5

PROGRAM TO GENERATE AIRFOIL COORDINATES

Aim: Write a code for generating airfoil coordinates.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher or Sci-lab.

Theory:

NACA 4- digit airfoil specification

This NACA airfoil series is controlled by 4 digits e.g. NACA 2412, which designate the camber, position of the maximum camber and thickness. If an airfoil number is NACA MPXX e.g.NACA 2412
then:

- M is the maximum camber divided by 100. In the example M=2 so the camber is 0.02 or 2% of the chord
- P is the position of the maximum camber divided by 10. In the example P=4 so the maximum camber is at 0.4 or 40% of the chord.
- XX is the thickness divided by 100. In the example XX=12 so the thickness is 0.12 or 12% of the chord.

The NACA airfoil section is created from a camber line and a thickness distribution plotted perpendicular to the camber line.

The equation for the camber line is split into sections either side of the point of maximum camber position (P). In order to calculate the position of the final airfoil envelope later the gradient of the camber line is also required. The equations are:

	Front ($0 \le x < p$)	Back ($p \le x \le 1$)
Camber	$y_c = \frac{M}{P^2}\left(2Px - x^2\right)$	$y_c = \frac{M}{(1-P)^2}\left(1 - 2P + 2Px - x^2\right)$
Gradient	$\frac{dy_c}{dx} = \frac{2M}{P^2}(P - x)$	$\frac{dy_c}{dx} = \frac{2M}{(1-P)^2}(P - x)$

The thickness distribution is given by the equation:

$$y_t = \frac{T}{0.2}\left(a_0 x^{0.5} + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4\right)$$

Where:

$a_0 = 0.2969$ $a_1 = -0.126$ $a_2 = -0.3516$ $a_3 = 0.2843$

$a_4 = -0.1015$ or -0.1036 for a closed trailing edge

- The constants a0 to a4 are for a 20% thick airfoil. The expression T/0.2 adjusts the constants to the required thickness.
- At the trailing edge (x=1) there is a finite thickness of 0.0021 chord width for a 20% airfoil. If a closed trailing edge is required the value of a4 can be adjusted.
- The value of yt is a half thickness and needs to be applied both sides of the camber line.

Using the equations above, for a given value of x it is possible to calculate the camber line position Yc, the gradient of the camber line and the thickness. The position of the upper and lower surface can then be calculated perpendicular to the camber line.

$$\theta = \operatorname{atan}\left(\frac{dy_c}{dx}\right)$$

Upper Surface $x_u = x_c - y_t \sin(\theta)$ $y_u = y_c + y_t \cos(\theta)$

Lower Surface $x_l = x_c + y_t \sin(\theta)$ $y_l = y_c - y_t \cos(\theta)$

The most obvious way to to plot the airfoil is to iterate through equally spaced values of x calculating the upper and lower surface coordinates. While this works, the points are more widely spaced around the leading edge where the curvature is greatest and flat sections can be seen on the plots. To group the points at the ends of the airfoil sections a cosine spacing is used with uniform increments of β

$$x = \frac{(1-\cos(\beta))}{2} \quad \text{where: } 0 \le \beta \le \pi$$

Algorithm:

i. Give the specifications of airfoil.
ii. Give the number of points required.
iii. Write code for camber and gradient equations
iv. Equations for theta and thickness distribution
v. Equations to generate upper curve and lower curve coordinates separately

MATLAB code for generating 4-digit airfoil: Airfoil number is required to be entered as vector as [x x x x]. Then program asks for chord length. Co-ordinates are computed and plotted.

```
% code for 4-digit NACA airfoil
% Notes:
% 1) That this code Plots NACA 4 Digit Series ONLY.

% Airfoil Equation
    AirfoilAsk = 'Enter The Airfoil Number in Row Vector as [x x x x]:  ';

Airfoil = input(AirfoilAsk);

NACA = Airfoil;

ChordAsk = 'Enter The Airfoil Chord Length:  ';    % Airfoil Chord

Chord = input(ChordAsk);

x = 0:0.0001:Chord;

if length(NACA) == 4

  disp(['NACA 4 Digit Series: NACA ', num2str(NACA(1)) num2str(NACA(2)) num2str(NACA(3)) num2str(NACA(4))])

  if NACA(1) == 0 && NACA(2) == 0

    Symm = 1;

    disp('Symmetric Airfoil')

  else

    Symm = 0;

    disp('Cambered Airfoil')

  end

end

if Symm == 1

  t = str2num([num2str(NACA(3)),num2str(NACA(4))])/100;

  y_upper = 5*t*Chord*(0.2969*sqrt(x/Chord)-0.126*(x/Chord)-0.3516*(x/Chord).^2+0.2843*(x/Chord).^3-0.1015*(x/Chord).^4);

  y_lower = -y_upper;
```

```
    x_upper = x;
    x_lower = x;
else
    m = NACA(1)/100;
    p = NACA(2)*Chord/10;
    t = str2num([num2str(NACA(3)),num2str(NACA(4))])/100;
    for i = 1:length(x)
        if x(i)/Chord<=p
            y_camber(i) = m*x(i)/p^2*(2*p-x(i)/Chord);
            dy_camber(i) = 2*m/p^2*(p-x(i)/Chord);
        else
            y_camber(i) = m*(Chord-x(i))/(1-p)^2*(1+x(i)/Chord-2*p);
            dy_camber(i) = 2*m/(1-p)^2*(p-x(i)/Chord);
        end
    end
    y_t = 5*t*Chord*(0.2969*sqrt(x/Chord)-0.126*(x/Chord)-
0.3516*(x/Chord).^2+0.2843*(x/Chord).^3-0.1015*(x/Chord).^4);
    theta = atan(dy_camber);
    x_upper = x-y_t.*sin(theta);
    x_lower = x+y_t.*sin(theta);
    y_upper = y_camber+y_t.*cos(theta);
    y_lower = y_camber-y_t.*cos(theta);
end
% Plots
figure
```

hold on

grid on

axis equal

plot(x_upper,y_upper,x_lower,y_lower,'LineWidth',1.5,'color','b')

plot(x,y_camber,'--','LineWidth',1,'color','r')

title(['NACA 4 Digit Series: NACA ', num2str(NACA(1)) num2str(NACA(2)) num2str(NACA(3)) num2str(NACA(4))])

xlabel('x')

ylabel('y')

Result: The output is shown in figure 5.1

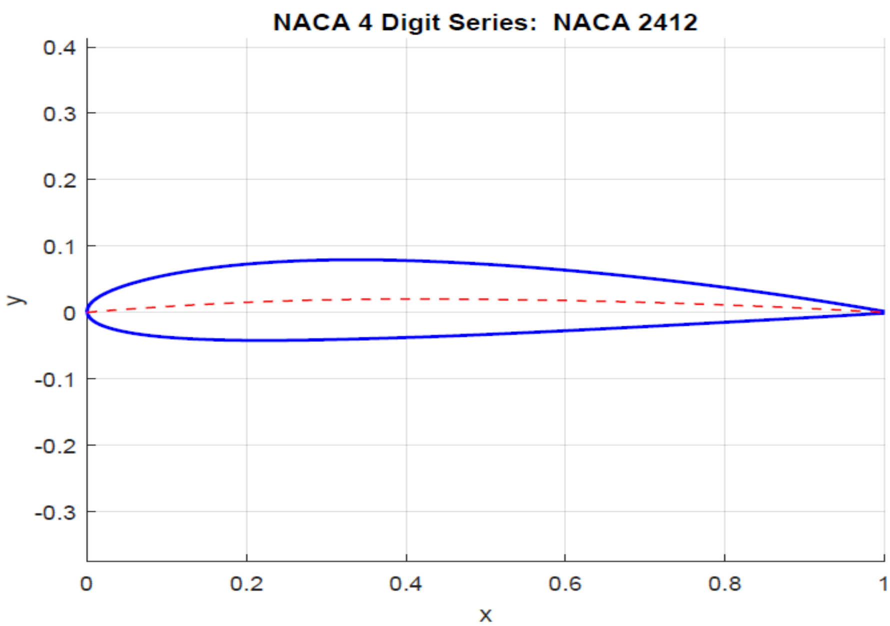


Fig 5.1

EXERCISES

1. Draw coordinates of NACA 0006 .
2. Draw coordinates of NACA 2415.

EXPERIMENT 6

PROGRAM TO FIND CRITICAL MACH NUMBER AND DRAG POLAR

Aim: Write a code to find critical Mach number and to draw drag polar of an airfoil.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher or Sci-lab.

Theory :

Critical Mach number: In aerodynamics the critical Mach number (Mcr or M*) of an airfoil is the lowest Mac Number at which the airflow over some point of the airfoil reaches the speed of sound but does not exceed it. This creates a weak shock wave as aircraft exceeds the critical Mach number, its drag coefficient increases suddenly, causing dramatically increased drag.

Plot the variation of Coefficient of Pressure with respect to Mach number using Prandtl-Gluaret compressibility correction for given airfoil

$$Cp = \frac{Cp_0}{\sqrt{M_\infty^2 - 1}} \text{ ------- (1)}$$

Where Cp_0 is incompressible pressure coefficient of given airfoil and M_∞ is freestream Mach number.

Similarly plot the variation of Critical Coefficient of Pressure with respect to Mach number for given airfoil using equation

$$Cp_{cr} = \frac{2}{\gamma M_\infty^2}\left[\left(\frac{1+[(\gamma-1)/2]M_\infty^2}{1+(\gamma-1)/2}\right)^{\gamma/(\gamma-1)} - 1\right] \text{ --------- (2)}$$

The intersection of these two curves will give critical Mach number of respective airfoil.

Drag Polar: The Drag Polar is the relationship between the lift on an airfoil and its drag, expressed in terms of coefficients. It may be described by an equation or displayed in a diagram called a polar plot.

Algorithm

Critical Mach number

i. Give the incompressible pressure coefficient of airfoil.
ii. Give the range of Mach number.

iii. Calculate coefficient of pressure using equation (1).
iv. Calculate Critical coefficient of pressure using equation (2)
v. Plot the graph between coefficient of pressure and Mach number using two equations.

Drag Polar:

i. Load pressure data for various angle of attack
ii. Calculate lift and drag coefficient of airfoil

Result: The intersection of curve 1 and curve 2 gives critical Mach number (Refer fig 6).

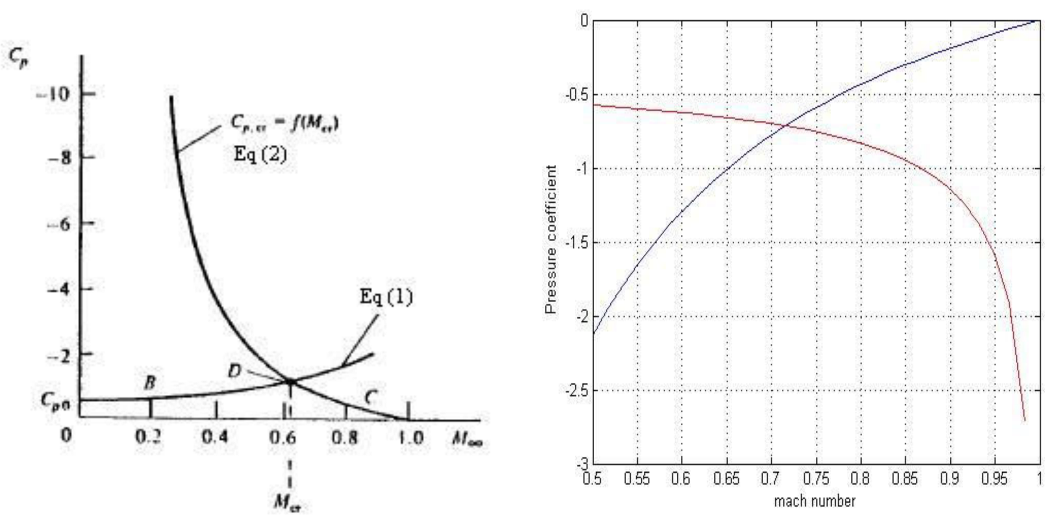


Fig 6: Critical Mach Number

EXERCISES

1. The minimum pressure coefficient for an NACA 0009 airfoil in low-speed flow is -0.25. Calculate the critical Mach number for this airfoil using Prandtl - Glauert rule and Karman-Tsien rule using Matlab code.

2. Plot drag polar plots of symmetric and cambered airfoils using Matlab code.

EXPERIMENT 7

FLOW CHARACTERISTICS ACROSS SHOCK WAVES

Aim: Write a code to find flow characteristics across shock waves

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher or Sci-lab.

Theory: Shock waves are formed when a pressure front moves at supersonic speeds and pushes on the surrounding air. At the region where this occurs, sound waves travelling against the flow reach a point where they cannot travel any further upstream and the pressure progressively builds in that region; a high-pressure shock wave rapidly forms.

Normal Shock: If the shock is perpendicular to flow direction then it is known as normal shock. As the flow passes shock wave Mach number decreases with the relation

$$M_2^2 = \frac{1 + [(\gamma - 1)/2]M_1^2}{\gamma M_1^2 - (\gamma - 1)/2}$$

The rise in pressure, density, and temperature after normal shock can be calculated as follows:

$$\frac{\rho_2}{\rho_1} = \left[\frac{(\gamma + 1)M_1^2}{2 + (\gamma - 1)M_1^2}\right]$$

$$\frac{P_2}{P_1} = \left[1 + \frac{2\gamma}{\gamma + 1}(M_1^2 - 1)\right]$$

$$\frac{T_2}{T_1} = \frac{P_1}{P_2}\frac{\rho_2}{\rho_1}$$

Oblique Shock: When analyzing shock waves in a flow field, which are still attached to the body, the shock wave which is deviating at some arbitrary angle from the flow direction is termed oblique shock. For a given Mach number, M_1, and corner angle, θ, the oblique shock angle, β, and the downstream Mach number, M_2 based on M_{n2}, can be calculated.

$$tan\theta = 2cot\beta\left[\frac{sin^2\beta - 1}{M_1^2(\gamma + cos2\beta) + 2}\right]$$

$$M_{n1} = M_1 sin\beta$$

$$M_2 = \frac{M_{n2}}{\sin(\beta - \theta)}$$

Algorithm:

i. Enter the range of Mach number.

ii. Enter the value of specific heat ratio.

iii. Write the equations for pressure, density and temperature variation across shocks.

iv. Plot the variation

Result: Shown in Fig 7.

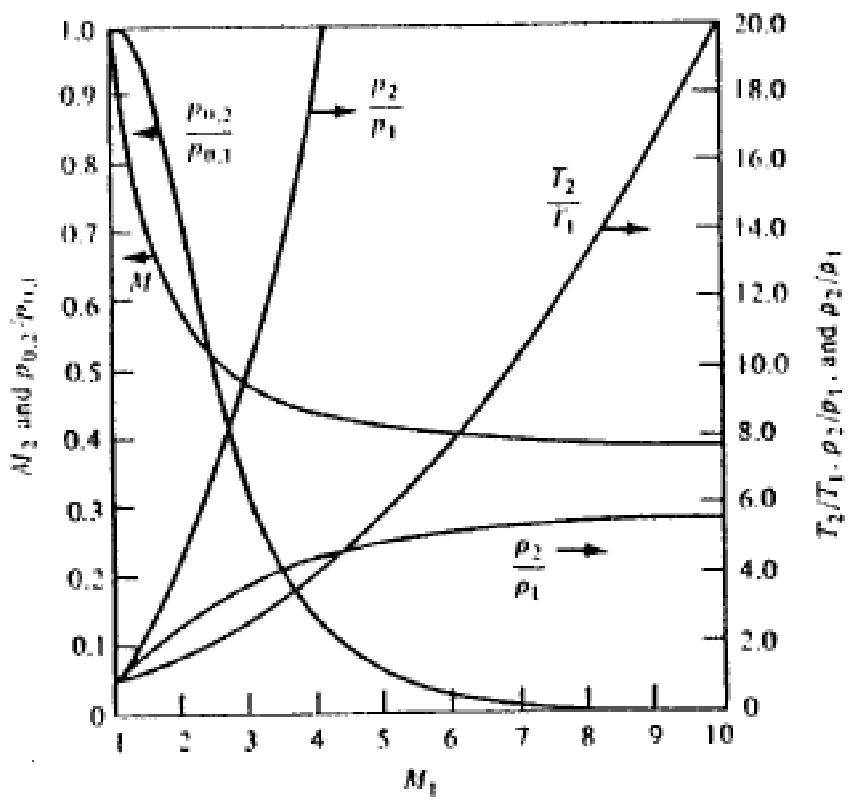


Fig 7: Variation of flow properties across normal shock wave

EXERCISES

1. Write a Matlab code to solve oblique shock problems.
2. Write a Matlab code to solve expansion wave problems.

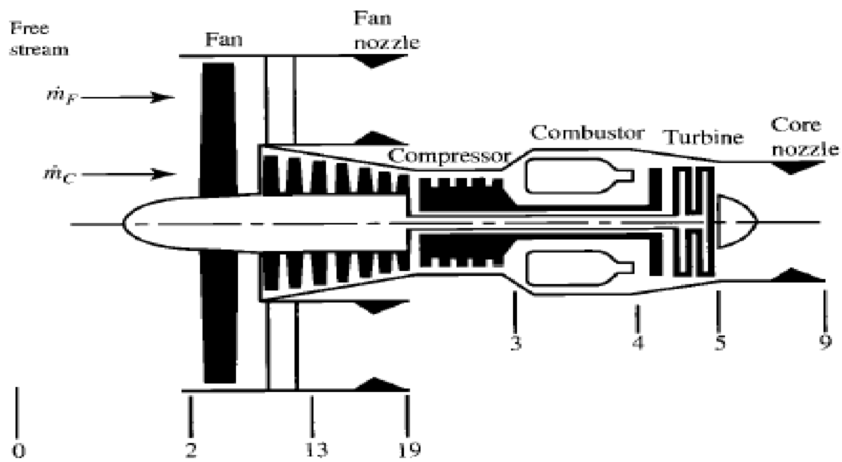
Experiment 8

PERFORMANCE OF TURBOFAN ENGINE

Aim: Write a Matlab code to find the performance of turbofan engine.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher or Sci-lab.

Theory: The turbofan is a type of airbreathing jet engine that is widely used in aircraft propulsion. The *turbo* portion refers to a gas turbine engine which achieves mechanical energy from combustion, and the *fan*, a ducted fan that uses the mechanical energy from the gas turbine to accelerate air rearwards.



For clarity we adopt the following notations

M_0 – Mach number
T_0 – Initial Temperature
γ – Ratio of specific heats
T_4 – Combustor entry temperature
a_0 – speed of sound
π_c – Compressor pressure ratio
π_f – Fan pressure ratio
F – Thrust
f – fuel to air ratio

The performance of Ideal turbofan engine is plotted in terms of fuel to air ratio vs compressor pressure ratio and Thrust vs compressor pressure ratio using following equations:

$$F = a_0 \frac{1}{1+\alpha}\left[\frac{V_9}{a_0} - M_0 + \alpha\left(\frac{V_{19}}{a_0} - M_0\right)\right]$$

$$\frac{V_9}{a_0} = \sqrt{\frac{2}{\gamma-1}\left\{\tau_\lambda - \tau_r\left[\tau_c - 1 + \alpha\left(\tau_f - 1\right)\right] - \frac{\tau_\lambda}{\tau_f \tau_c}\right\}}$$

$$\frac{V_{19}}{a_0} = \sqrt{\frac{2}{\gamma-1}\left(\tau_r \tau_f - 1\right)}$$

Ram temperature ratio $\tau_r = 1 + \frac{\gamma-1}{2} M_0^2$

Burner to Exit Temperature ratio$\tau_\lambda = \frac{T_{t4}}{T_0}$

Compressor temperature ratio $\tau_c = (\pi_c)^{(\gamma-1)/\gamma}$

Fan temperature ratio $\tau_f = \left(\pi_f\right)^{(\gamma-1)/\gamma}$

Fuel to air ratio $f = \frac{C_p T_0}{h_{pr}}(\tau_\lambda - \tau_r \tau_c)$

Specific Fuel Consumption $S = \frac{f}{(1+\alpha)F}$

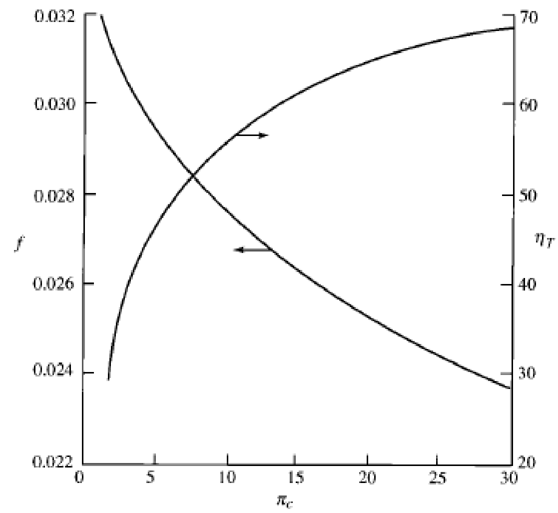
Thermal Efficiency $\eta_T = 1 - \frac{1}{\tau_r \tau_c}$

Algorithm:

i. Give the range of compressor pressure ratio.
ii. Enter all the required data like Mach number, Initial Temperature, Ratio of specific heats, Combustor entry temperature.
iii. Write code for above equations to find performance of turbofan engine.

Result: Shown in the figure 8.

i. Plot compressor pressure ratio vs Thrust,
ii. Compressor pressure ratio vs fuel to air ratio and
iii. Compressor pressure ratio vs Thermal efficiency.



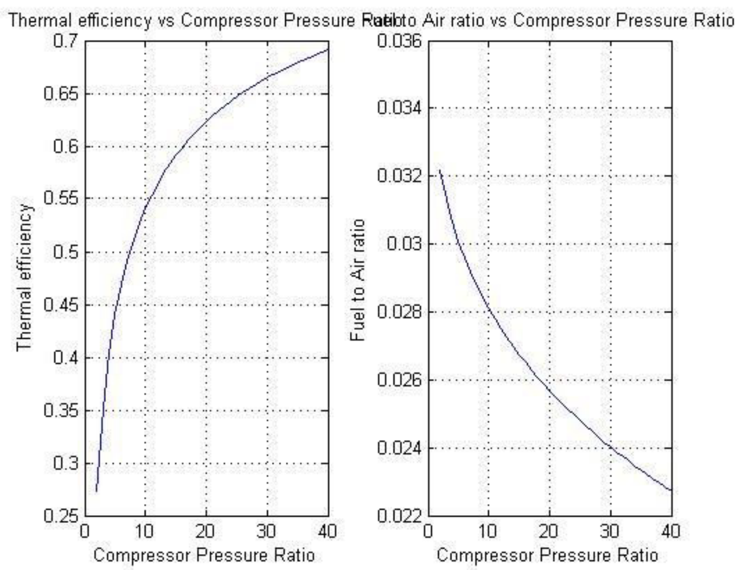


Fig 8: Compressor ratio vs fuel air ratio and Thermal efficiency

EXERCISES

1. Write Matlab code to analyze turbojet performance for varying compressor pressure ratio.

EXPERIMENT 9

FLOW CHARACTERISTICS OF A CD NOZZLE

Aim: Write a code to find the variation of flow characteristics in a CD nozzle.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher or Sci-lab.

Theory: For a supersonic flow to develop from a reservoir where the velocity is zero, the subsonic flow must first accelerate through a converging area to a throat, followed by continued acceleration through an enlarging area. The nozzles on a rocket designed to place satellites in orbit are constructed using such converging-diverging geometry.

The following relation is used to find the variation of Mach number with respect to area ratio.

$$\left(\frac{A}{A^*}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M^2\right)\right]^{(\gamma+1)/(\gamma-1)}$$

Then use isentropic relation to find pressure, density and temperature variation across nozzle.

Algorithm:

i. Give the range of Area ratio.
ii. Enter all the required data like Mach number, Ratio of specific heats
iii. Write code for above equations to find Mach number.
iv. Write code for isentropic relations.

Result: Plot the variation of flow characteristics along nozzle from inlet to outlet.

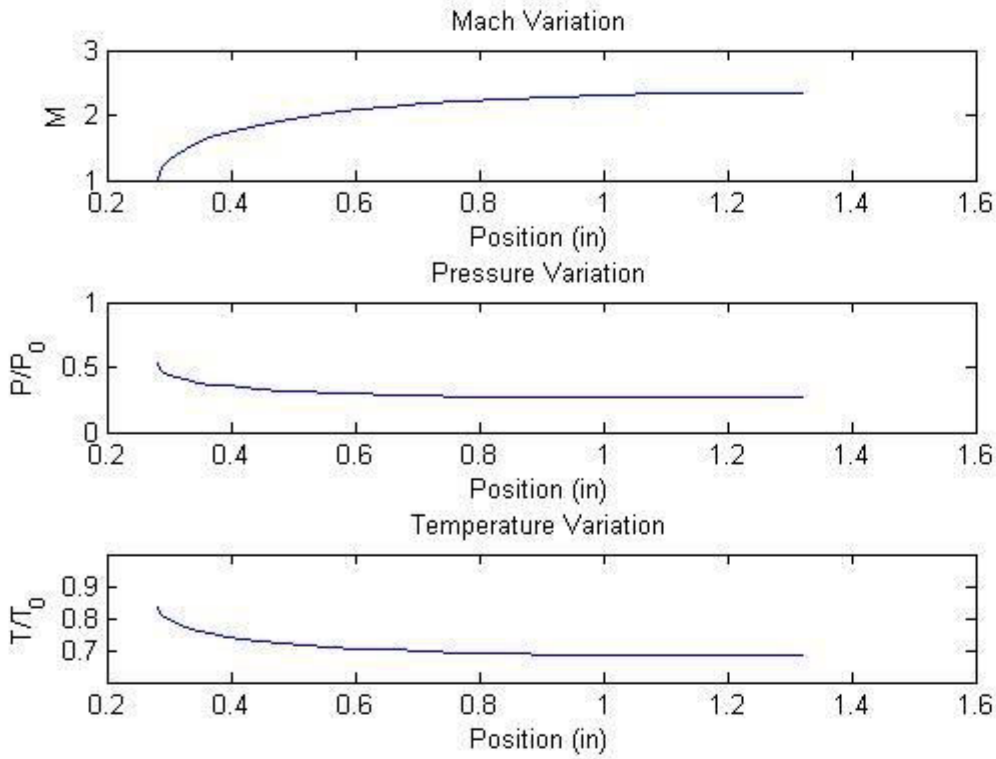


Fig 9. Variation of flow characteristics along nozzle

EXERCISES

1. Write a Matlab code for a supersonic wind tunnel to produce Mach 2.4 at standard sea level conditions to calculate exit to throat area ratio of nozzle and reservoir pressure and temperature.
2. The reservoir pressure of a supersonic wind tunnel is 5 atm. A static pressure probe measures 4 atm, 2.64 atm and 0.5 atm along centerline of nozzle. Write a matlab code to calculate local Mach number and area ratio.

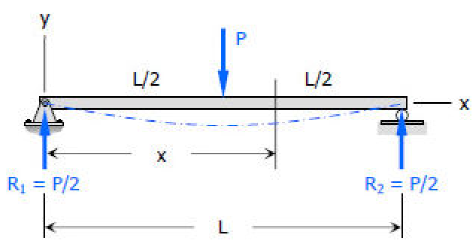
EXPERIMENT 10
DEFLECTION, SHEAR FORCE AND BENDING MOMENT OF BEAM

Aim: To study the programs for deflections, shear forces and bending moments of beams under various boundary and loading conditions.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher or Sci-lab.

Theory: Deflection of beams using double integration method

Determine the maximum deflection δ in a simply supported beam of length L carrying a concentrated load P at midspan.



Algorithm

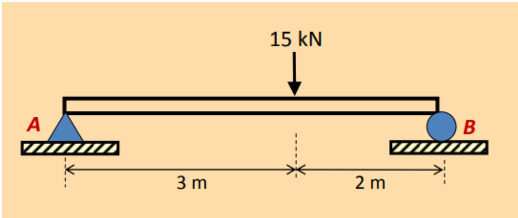
- Take moment at any end
- Performing double integration get slope and deflection equation interms of two integrated constant (C1 and C2)
- Consider both end x & y values, find out the constants C1 and C2
- Using C1 and C2 get deflection at mid span of the beam

Result:

$$\delta = \frac{PL^3}{48EI}$$

Shear force and Bending moments

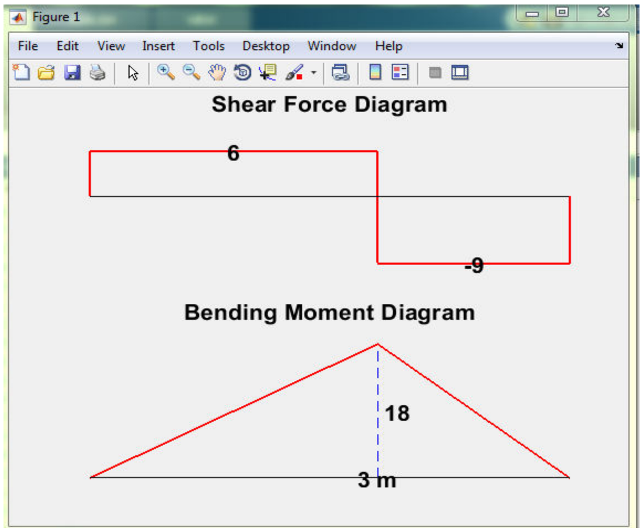
Calculate the shear force diagram and bending moment diagram for the beam subjected to a concentrated load.



Algorithm:-(Method-1)

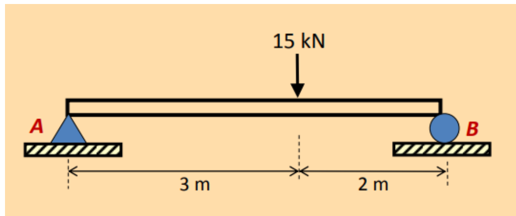
- Read the dimensions, loading and boundary conditions
- Based on loading and load position calculate both end reaction forces
- Discretize the x axis in to n points; accordingly calculate shear force V and bending moments M for each point.
- Draw shear force and bending moment diagram using the V and M values

Result:-



Shear force and Bending moments

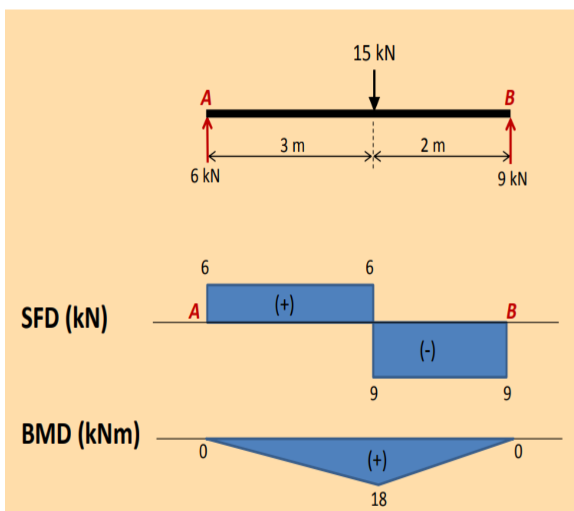
Calculate the shear force and bending moment for the beam subjected to a concentrated load as shown in the figure.



Algorithm:- (Method-2)

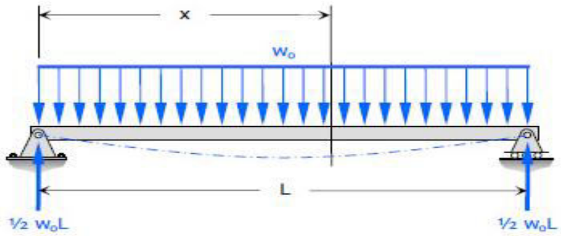
- Enter the dimensions, loading and boundary conditions
- Take moment at $\sum M_A = 0$ or $\sum M_B = 0$
- Find out $\sum F_Y = 0$ and $\sum F_X = 0$
- Calculate the unknown reaction forces

Result:-

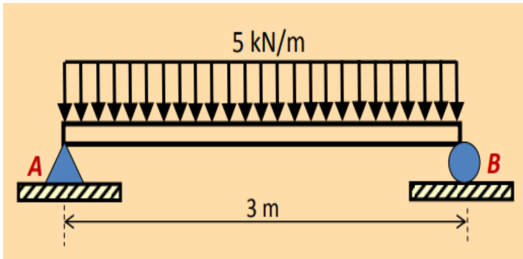


EXERCISES

1. Determine the maximum deflection δ in a simply supported beam of length L carrying a uniformly distributed load of intensity w_o applied over its entire length.



2. Calculate the shear force and bending moment for the beam subjected to an uniformly distributed load as shown in the figure.



EXPERIMENT 11

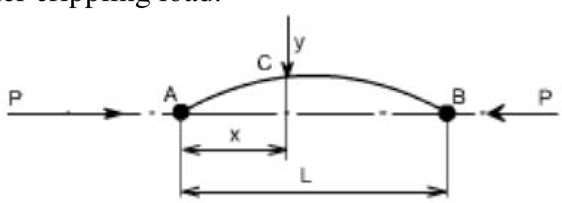
DETERMINE THE BUCKLING LOAD OF A COLUMN WITH DIFFERENT END CONDITIONS.

Aim: To study program for Euler Crippling Load of column under different boundary condition using integration method.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher or Sci-lab.

Theory:

Column with simply supported ends: **-** Consider an axially loaded strut, shown below, and is subjected to an axial load 'P' this load 'P' produces a deflection 'y' at a distance 'x' from one end. Calculate the Euler crippling load.



Algorithm:

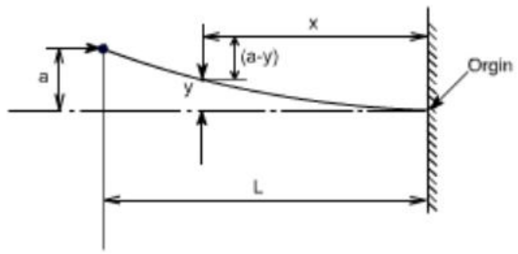
- Take bending moment at C
- Using bending moment equation obtain second order differential equation interms of y,P,EI
- Assume solution equation according to your differential equation interms of constants A and B
- Apply both end boundary conditions inorder to obtain constant value A and B
- Using A and B findout the Euler crippling load.

Result:

$$P_e = \frac{\pi^2 E\ I}{L^2}$$

EXERCISES

1. Consider an axially loaded strut, shown below, and is subjected to an axial load ‘P' this load ‘P' produces a deflection ‘y' at a distance ‘x' from one end. Calculate the Euler crippling load.



EXPERIMENT 12

DISPLACEMENTS OF A UNIFORM BAR/STEPPED BAR SUBJECTED TO MECHANICAL /THERMAL LOADS

Aim: To study the Mechanical behavior of a stepped bar under axial load using finite element method.

Equipment needed: Core 2 duo processor with 1GB RAM and MATLAB software Version 2008a or higher or Sci-lab.

Problem

Consider a bar as shown in Fig .1 An axial load of 200KN is applied at point p. Take $A_1 = 2400\ mm^2, E_1 = 70x10^9 N/m\text{^}2, A_2 = 600mm^2, E_2 = 200x10^9 N/m\text{^}2$

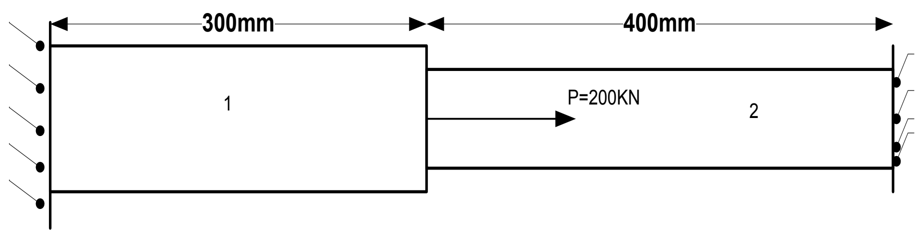


Fig (i)

Calculate the following:

- The nodal displacement at point p
- Stress in each material
- Reaction force

Algorithm:

- Read the dimensions, boundary condition and loading condition of a given beam.
- Read the material properties of the bar
- According to the number of element find out elemental stiffness matrix
- Get global stiffness matrix after assemblage of stiffness matrix
- Applying boundary conditions loading
- Find out the unknowns

Results:

$U_p = 2.3256e - 10$, R1 =-130.2326, R2 = -69.7674, Strees1 = 0.0543, Stress2 = -0.1163.

EXERCISES

1. Consider a bar as shown. An axial load of 200KN is applied at point p. Take $A_1 = 2400\ mm^2, E_1 = 70x10^9 N/m\text{^}2, A_2 = 600mm^2, E_2 = 200x10^9 N/m\text{^}2$