COURSE CONTENT

NUMERICAL METHODS USING MATLAB								
III Semester: AE								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
AAED04	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	-	2	1	40	60	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 45**				**Total Classes: 45**		
Prerequisite: Nil								

I. COURSE OVERVIEW:

Numerical Methods using MATLAB involves the application of mathematical techniques to solve complex problems through numerical approximation. MATLAB, a powerful programming language and environment, is widely used for implementing and analyzing numerical methods. This approach is essential in fields such as engineering, physics, and finance, where analytical solutions may be challenging or impossible to obtain. The use of MATLAB facilitates efficient algorithm implementation, visualization of results, and quick prototyping of numerical solutions. Students and professionals benefit from learning and utilizing numerical methods in MATLAB to address real-world problems in a computational context.

II. COURSE OBJECTIVES:

The students will try to learn:

I. The procedures, algorithms, and concepts require to solve specific problems.
II. The concepts of algebra, calculus and numerical solutions using MATLAB software.
III. The knowledge in MATLAB and can apply for project works.
IV. The simple mathematical functions and operations thereon using plots/display.

III. COURSE OUTCOMES:

After successful completion of the course, students will be able to:

CO1 Understand the numerical methods in MATLAB use for the accurate of solutions to mathematical problems that may lack analytical solutions or have complex expressions.
CO2 Make use of MATLAB a user-friendly platform for implementing numerical algorithms efficiently, enabling the quick and reliable solution of mathematical problems.
CO3 Utilize MATLAB’s built-in plotting and visualization tools facilitate the interpretation and presentation of numerical results, aiding in a better understanding of the solution behavior.
CO4 Apply Numerical methods often involve iterative processes, and MATLAB's programming capabilities make it easy to refine and optimize algorithms for improved convergence and accuracy.
CO5 Make use of MATLAB's matrix-oriented approach is well-suited for handling large datasets, making it advantageous for numerical methods dealing with extensive data or complex systems of equations.
CO6 Apply Numerical methods in MATLAB find applications in various disciplines, including engineering, physics, finance, and more, providing a versatile toolset for solving diverse computational challenges.

IV. COURSE CONTENT:

Exercises for Numerical Methods Using MATLAB

Note: Students are encouraged to bring their own laptops for laboratory practice sessions.

1. Getting Started with MATLAB Student Version

1.1 MATLAB Student Version Installation procedure

System requirement

Supported Platforms and Operating Systems:

Microsoft Windows 10, 64-bit

Minimum Hardware Requirements for MATLAB Student Product:

Processor(s): Workstation class

4 GB RAM

25 GB hard drive space

Computer must have a physical C:/" drive present

Graphics card and driver: Professional workstation class 3-D

OpenGL-capable

Installation Procedure

1. Extract (unzip) the downloaded installation files.
2. Right-click on setup.exe and select Run as Administrator. (This will run setup.exe from the extracted files.)
3. Read and accept the clickwrap to continue.
4. Click the right arrow button to accept the default values throughout the installation.
5. Click the exit button to close the installer.
6. The MATLAB Student software is now installed.
7. Reboot your machine and then run the MATLAB. Student product from your Start menu by selecting Workbench.

Problem size limits

- No Geometry Export

Limits for MATLAB Student and Discovery (Refine Mode)

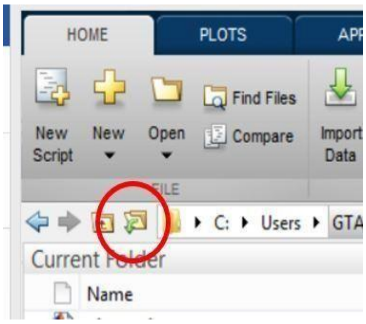
- Aerospace Tool Box: 250K nodes/elements
- No.of Licensed units: 60

1.2 Getting Started with MATLABworkbench

Open MATLAB: Windows Start Menu button → MATLAB → Workbench • Under the Toolbox Analysis Systems category, click and drag analysis system Static Structural onto the Project Schematic, drop it on target Create standalone system (this will be the only target available

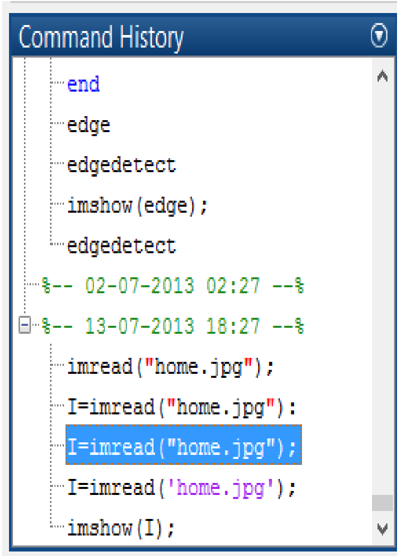
The desktop includes these panels:

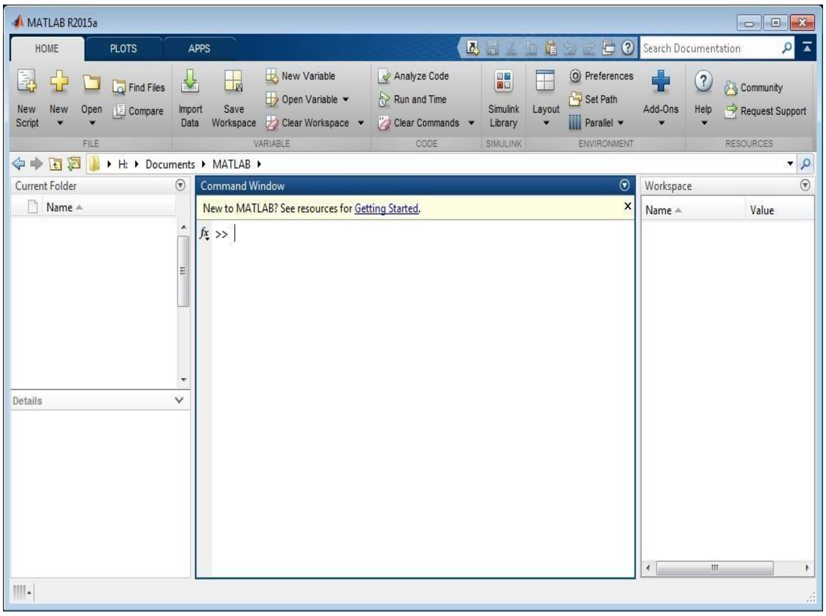
Current Folder - This panel allows you to access the project folders and files.



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

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 educational purpose.

2. MATLAB adds variable to the workspace and displays the result in the Command Window.

2. Exercises on Algebra

Algebra is a branch of mathematics that deals with symbols and the rules for manipulating those symbols. It typically involves solving equations and working with variables to analyze and understand mathematical relationships. Here are some key aspects of algebra. At an advanced level, algebra becomes more abstract, studying structures like groups, rings, fields, and vector spaces. Abstract algebra generalizes and formalizes the fundamental algebraic concepts and techniques encountered in earlier study.

2.1 Roots of the equation

Calculate the a) Find the roots of the equations $6x^5 -41x^4 +97x^3 -97x^2 +41x-6$

Hints

```
1. Roots of Equation 6x5 -41x4 +97x3 -97x2 +41x-6
```

```
• v = [6, -41, 97, -97, 41,-6]; % writing the coefficients
s = roots(v);
```

```
2. General Post Processing
```

```
• 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));
```

2.2 Soving the linear equations

A linear equation is an equation that involves only constants and variables raised to the first power, with no exponents or other operations.
Find the values of x, y, z of the equations x+y+z=3,x+2y+3z=4,x+4y+9z=6

Hints

```
1. Identify the Variables: Determine which letters represent the variables in the
   equation. Often, variables are represented by letters like x, y, and z.


2. Define the Coefficients Matrix and the Constants Vector: You need to represent
   your linear equations in matrix form Ax=b, where A is the coefficients matrix,
   xx is the variables vector, and bb is the constants vector.
```

```
3. Choose a Suitable Solver: MATLAB provides several solvers for different types
   of linear systems. The choice of solver depends on the properties of your matrix
   AA and the specific problem you're solving.
```

```
% Define the coefficients matrix A and the constants vector b
A = [2, 1; -1, 3];
b = [5; 10];

% Solve the linear system Ax = b using linsolve
x = linsolve(A, b);
```

```
% Display the solution
disp('The solution to the linear system is:');
disp(x);
```

Try

1. Solve a system of linear equations using MATLAB, that can represent the system in matrix form Ax=b, where A is the matrix of coefficients, x is the vector of unknowns, and b is the vector of constants
2. Solve a system of linear equations using MATLAB, with different forms by ODE forms.

3. Exercises on control structures

Control structures in MATLAB allow you to control the flow of execution of your code. They enable you to make decisions, repeat operations, and perform tasks conditionally. They are fundamental to writing effective and efficient MATLAB code, allowing you to implement complex logic and handle various scenarios dynamically.

3.1 Conditional statements allow you to execute specific blocks of code based on certain conditions (if, elseif, else Statements).

```
x = 10;
if x < 5
disp('x is less than 5');
elseif x == 5
disp('x is equal to 5');
else
disp('x is greater than 5');
end
```

Hints

```
% Example 1: Temperature evaluation using ‘if’ statement
temperature = 25;
if temperature > 30
disp('It is hot');
elseif …………………………………….
```

```
disp(………………………………..);
else
disp(……………………………………………..);
end
```

```
% Example 2: if-else statement
y = 3;

if y > 5
disp('y is greater than 5');
else
disp('y is not greater than 5');
```

```
end

% Example 3: if-else statement
y = 3;
if y > 5
disp(………………………………);
else
disp(…………………………….);
end
```

3.2Switch Case

The **switch** statement in MATLAB is used for executing one of several groups of statements, depending on the value of a variable or expression

The basic syntax of the switch statement in MATLAB is as follows:

```
switch expression
case value1
	% Code to execute if expression is equal to value1
case value2
	% Code to execute if expression is equal to value2
    % Add more cases as needed
otherwise
	% Code to execute if expression does not match any case
end
```

Hints

```
% Example 1: Simple switch Statement
day = 'Monday';

switch day
case 'Monday'
disp('Start of the work week.');
case…………………………………..
disp(…………………………………………);
case……………………………………….
disp('Weekend has started!');
case 'Sunday'
```

```
disp(…………………………………..);
otherwise
disp('Middle of the work week.');
end
```

```
% Example 2: Switch with Numeric Values

number = 3;

switch number
case 1
disp('One');
case ………………………………
disp('Two');
```

```
case ………………………………….
disp('……………………….');
case ………………………….
disp('……………………………..');
otherwise
disp('Number is not between 1 and 4');
end

% Example 3: Nested Switch Statements

shape = 'rectangle';
color = 'blue';

switch shape
case…………………………………………………………….
disp('Shape is a circle.');
case 'rectangle'
disp('Shape is a rectangle.');
switch………………………………………….
case…………………………………………..
disp('Color is red.');
case 'blue'
disp('Color is blue.');
otherwise
disp………………………………………
end
otherwise
disp('Shape is unknown.');
end
```

Try

1. Change the location of point load and calculate the shear force, bending moment and stress distribution in the beam

2. Change the location of point loads in tapered beam and find the deflection and the displacement variation along thebeam length

3.3 FOR loop MATLAB

In MATLAB, a for loop is used to execute a sequence of statements multiple times, iterating over a range of values. The loop variable increments or decrements according to a specified range, and the block of code within the loop is executed once for each value in that range.

The basic syntax of a for loop in MATLAB is:

```
for index = startValue:endValue
   % Code to execute
end
```

Hints

```
% Example 1: A simple for loop that iterates from 1 to 5 and displays the iteration variable

for i = 1:5
disp…………………………………………….
end
```

```
% Example 2: Loop Through a Vector

vec = [10, 20, 30, 40, 50];

for i = ……………………………..
disp(['Element ', num2str(i), ' is ', num2str(vec(i))]);
end


% Example 3: Sum of First N Natural Numbers

N = 10;
sum = 0;

for i ……………………………………………..
sum……………………………………………………….
end

disp……………………………………………….
```

4. Exercises on Matrices

4.1 Addition, Subtraction, and Multiplication of Matrices

Matrices are a fundamental aspect of MATLAB, which stands for "Matrix Laboratory". MATLAB is designed for matrix operations, making it an ideal environment for working with linear algebra, numerical analysis, and related fields.

Matrix addition is done element-wise, and the matrices must be of the same size.

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

B = [9 8 7; 6 5 4; 3 2 1];
C = A + B;
The result is:

C =

```
   10    10    10

   10    10    10

   10    10    10
```

Hints

```
% Example 1: Matrix subtraction is done element-wise, and the matrices must
be of the same size.

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

```
D = A - B;

The result is:

D =
    -8    -6    -4
    -2     0     2
     4     6     8

% Example 2: For matrix multiplication, the number of columns in the first
matrix must equal the number of rows in the second matrix.

A = [1 2 3; 4 5 6];
B = [7 8; 9 10; 11 12];
E = A * B;

The result is:

E =
    58    64
   139   154

% Example 3: Element-wise Multiplication.

A = [1 2 3; 4 5 6];
B = [7 8 9; 10 11 12];

D = ………………………………………;

disp('………………………………………');
disp(elementWiseProd);

The result is:

D ………………………
```

Try

1. Change the number of elements, matrix size and find the element wise multiplication and division.
2. Change the number of dimensions, matrix size and apply it to solve ODE problems.

5. Exercises on system of linear equations

5.1 Rank of the matrix

In MATLAB, the rank of a matrix is determined by the number of linearly independent rows or columns. The rank function is used to compute this value, which is essential in understanding the properties of the matrix, such as its solvability in linear systems and its inevitability.

Example 1: Rank of a Full Rank Matrix

A = [1 2 3; 4 5 6; 7 8 10];

r = rank(A);

disp(['The rank of the matrix A is ', num2str(r)]);

The output will be:

The rank of the matrix A is 3

Hints

```
% Example 1: Rank of a Matrix with Linearly Dependent Rows
B = [1 2 3; 2 4 6; 3 6 9];
r = rank(B);
```

```
disp(['The rank of the matrix B is ', num2str(r)]);

Output:

The rank of the matrix B is 1
```

```
% Example 2: Rank of a Rectangular Matrix

C = [1 2 3; 4 5 6; 7 8 9; 10 11 12];
r = rank(C);
disp(['The rank of the matrix C is ', num2str(r)]);

Output:…………………………………….

The rank of the matrix C is 2

% Example 3: Full Rank Matrix
A = [1 2 3; 4 5 6; 7 8 9];
r = rank(A);
disp…………………………………………………………..

Output:…………………………………….


% Example 4: Linearly Dependent Rows
B = [1 2 3; 2 4 6; 3 6 9];
r = rank(B);
disp…………………………………………………………..

Output:…………………………………….

% Example 5: Rectangular Matrix
C = [1 2 3; 4 5 6; 7 8 9; 10 11 12];
r = rank(C);
disp…………………………………………………………..

Output:…………………………………….

% Example 6: Identity Matrix
I = eye(4);
r = rank(I);
disp…………………………………………………………..

Output:…………………………………….
```

```
% Example 7: Sparse Matrix
D = [1 0 0; 0 0 0; 0 0 2];
r = rank(D);
disp……………………………………………………………..

Output:…………………………………….

% Example 8: Random Matrix
E = rand(5, 5);
r = rank(E);
disp……………………………………………………………..

Output:…………………………………….

% Example 9: Singular Matrix
F = [1 2 3; 4 5 6; 7 8 9];
r = rank(F);
disp……………………………………………………………..

Output:…………………………………….

% Example 10: Zero Matrix
G = zeros(3, 3);
r = rank(G);
disp……………………………………………………………..

Output:…………………………………….
```

Try

1. Use the matrix operations for rectangular matrices and find the rank of resultant matrix.
2. Matrix operations element wise for square matrices and find the rank of resultant matrix.

5.2 Row Echelon Form

To transform a matrix into its Row Echelon Form (REF) in MATLAB, you can use the built-in function rref which stands for "reduced row echelon form". However, if you want to implement the algorithm manually, here's a step-by-step guide and corresponding MATLAB code to achieve this.

Steps to Transform a Matrix to Row Echelon Form

1. Start with the leftmost nonzero column. This is a pivot column. The pivot position is at the top of this column.
2. Select the pivot row by finding the row with the largest absolute value in the pivot column. Swap this row with the top row if necessary.
3. Normalize the pivot row by dividing the entire row by the pivot element.
4. Eliminate all entries below the pivot position by subtracting an appropriate multiple of the pivot row from each row below.
5. Cover the row containing the pivot position and repeat the process on the submatrix that remains. Continue until there are no more pivot positions to cover

Syntax:

```
function REF = rowEchelonForm(A)

    [rows, cols] = size(A);

    REF = A;  % Initialize REF with the matrix A

    % Iterate over each column

for col = 1:cols

        % Find the pivot row

pivotRow = col;

whilepivotRow<= rows && REF(pivotRow, col) == 0

pivotRow = pivotRow + 1;

end

        % If no pivot row is found, move to the next column

ifpivotRow> rows

continue;

end

 % Swap the current row with the pivot row

ifpivotRow ~= col

temp = REF(col, :);

REF(col, :) = REF(pivotRow, :);

REF(pivotRow, :) = temp;

end

                % Normalize the pivot row

REF(col, :) = REF(col, :) / REF(col, col);

                % Eliminate all entries below the pivot

for row = col + 1:rows

REF(row, :) = REF(row, :) - REF(row, col) * REF(col, :);

end

end
end
```

Sample Problem:
A = [1 2 3; 4 5 6; 7 8 9];

RREF = rref(A);

disp('Reduced Row Echelon Form:');

disp(RREF);

Hints

```
% Example 1: For matrix A = [1 2 3; 4 5 6; 7 8 9], modify it in the rowEchelonForm
```

```
Row Echelon Form:
    1.0000    2.0000    3.0000
         0   -3.0000   -6.0000
         0         0         0

Using rref(A) will produce:

Reduced Row Echelon Form:
    1.0000         0   -1.0000
         0    1.0000    2.0000
         0         0         0
```

```
% Example 2: Simple Matrix
A = [1 2 3; 4 5 6; 7 8 9];
REF_A = rowEchelonForm(A);
disp('Row Echelon Form (Custom Function) for A:');
disp(REF_A);

RREF_A = rref(A);
disp('Reduced Row Echelon Form (Built-in Function) for A:');
disp(RREF_A);

% Example 3: Rectangular Matrix
B = [1 2 3; 4 5 6; 7 8 9; 10 11 12];
REF_B = rowEchelonForm(B);
disp('Row Echelon Form (Custom Function) for B:');
disp(REF_B);

RREF_B = rref(B);
disp('Reduced Row Echelon Form (Built-in Function) for B:');
disp(RREF_B);

% Example 4: Matrix with Linearly Dependent Rows
C = [1 2 3; 2 4 6; 3 6 9];
REF_C = rowEchelonForm(C);
disp('Row Echelon Form (Custom Function) for C:');
disp(REF_C);

RREF_C = rref(C);
disp('Reduced Row Echelon Form (Built-in Function) for C:');
disp(RREF_C);

% Example 5: Sparse Matrix
D = [1 0 0; 0 0 0; 0 0 2];
REF_D = rowEchelonForm(D);
```

```
disp('Row Echelon Form (Custom Function) for D:');
disp(REF_D);

RREF_D = rref(D);
disp('Reduced Row Echelon Form (Built-in Function) for D:');
disp(RREF_D);
```

5.3 LU Decomposition

LU decomposition, also known as LU factorization, is a method of decomposing a matrix into the product of a lower triangular matrix LL and an upper triangular matrix UU. This is particularly useful for solving linear systems of equations, inverting matrices, and calculating determinants.

Synatax

```
function [L, U] = luDecomposition(A)

  % Get the size of the matrix

  [n, m] = size(A);

    % Initialize L and U

  L = eye(n);

  U = zeros(n);

    % Perform the LU Decomposition

for j = 1:n

    % Upper Triangular Matrix U

for i = 1:j

      U(i,j) = A(i,j) - L(i,1:i-1) * U(1:i-1,j);

end

        % Lower Triangular Matrix L

for i = j+1:n

      L(i,j) = (A(i,j) - L(i,1:j-1) * U(1:j-1,j)) / U(j,j);

end

end
end
```

Sample Problems

A = [4 3; 6 3];

% LU Decomposition using built-in function

[L, U] = lu(A);

disp('Matrix A:');

disp(A);

disp('Lower Triangular Matrix L:');

disp(L);

disp('Upper Triangular Matrix U:');

disp(U);

% Example Matrix

A = [4 3; 6 3];

% LU Decomposition using custom function

[L, U] = luDecomposition(A);

disp('Matrix A:');

disp(A);

disp('Lower Triangular Matrix L:');

disp(L);

disp('Upper Triangular Matrix U:');

disp(U);

Hints

```
% Define the matrix
A = [4 3; 6 3];

% Perform LU Decomposition using built-in function
[L, U] = lu(A);

% Display the results
disp('Matrix A:');
disp(A);
disp('Lower Triangular Matrix L:');
disp(L);
disp('Upper Triangular Matrix U:');
disp(U);

Output:
```

```
Matrix A:
     4     3
     6     3

Lower Triangular Matrix L:
     1     0
     1.5   1

Upper Triangular Matrix U:
     4     3
     0    -1.5


% Define the matrix
B = [1 2 3; 4 5 6; 7 8 9];

% Perform LU Decomposition using built-in function
[L, U] = lu(B);

% Display the results
disp('Matrix B:');
disp(B);
disp('Lower Triangular Matrix L:');
disp(L);
disp('Upper Triangular Matrix U:');
disp(U);


Output:

Matrix B:
     1     2     3
     4     5     6
     7     8     9

Lower Triangular Matrix L:
     1     0     0
     0.1429    1     0
0.5714  0.5000  1

Upper Triangular Matrix U:
     7     8     9
     0    0.8571    1.7143
     0          0         0


% Define the matrix
C = [4 3 2 1; 3 2 1 4; 2 1 4 3; 1 4 3 2];

% Perform LU Decomposition using built-in function
[L, U] = lu(C);

% Display the results
disp('Matrix C:');
disp(C);
disp('Lower Triangular Matrix L:');
disp(L);
disp('Upper Triangular Matrix U:');
disp(U);
```

```
Output:

Matrix C:
     4     3     2     1
     3     2     1     4
     2     1     4     3
     1     4     3     2

Lower Triangular Matrix L:
     1     0     0     0
     0.7500    1     0     0
0.5000  0.6000    1     0
     0.2500 -1.2000  0.2118 1

Upper Triangular Matrix U:
     4     3     2     1
0  -0.2500  -0.5000 3.2500
     0          0 3.6000  1.4000
     0          0          0  -2.0471
```

6. Exercises on Linear Transformation

6.1. Characteristic Equation

In the context of linear systems or differential equations, the characteristic equation typically refers to an equation that involves the eigenvalues of a matrix or the roots of a differential equation.

Matrix Form:

- For a matrix AAA, the characteristic equation is given by det(A−λI)=0\det(A - \lambda I) = 0det(A−λI)=0, where λ\lambdaλ are the eigen values of AAA.

MATLAB CODE:

```
1. MATLAB CODE:
```

```
% Define a matrix A
A = [1 2; 3 4];
% Calculate eigenvalues using built-in function eig()
eigenvalues = eig(A);
 disp('Eigenvalues of matrix A:');
disp(eigenvalues);
% Alternatively, you can solve the characteristic equation manually % characteristic
equation is det(A - lambda * I) = 0
% For our matrix A = [1 2; 3 4],
it becomes (1-lambda)*(4-lambda) - 6 = 0
 % Define coefficients of the characteristic polynomial a = 1; b = -(A(1,1) +
A(2,2)); c = det(A);
```

```
% Solve the quadratic characteristic equation ax^2 + bx + c = 0 lambda = roots([a
b c]); disp('Eigenvalues calculated using characteristic equation:'); disp(lambda);
```

Try

1. Instead of temperature apply heat flux on the sides of square and find the temperature distribution.
2. Introduce a hole at the center of square and find the temperature distribution

6.2. Eigen Values

Given a square matrix AAA, an eigenvalue λ\lambdaλ and its corresponding eigenvector v\mathbf{v}v satisfy the equation: Av=λvA \mathbf{v} = \lambda \mathbf{v}Av=λv

Hints

```
1.
Preferences: Structural Preprocessor:
```

```
% Example matrix
A = [4 -2 1; -2 4 -2; 1 -2 3];

% Compute eigenvalues
eigenvalues = eig(A);

% Display the eigenvalues
disp('Eigenvalues of A:');
disp(eigenvalues);• Solve → Current LS → Ok
General Post Proc:
• Plot results → Temperature distribution
```

Try

1. Instead of temperature apply heat flux on the sides of square and find the temperature distribution.
2. Introduce a hole at the center of square and find the temperature distribution

6.3. Eigen Vectors

Given a square matrix AAA, an eigenvalue λ\lambdaλ and its corresponding eigenvector v\mathbf{v}v satisfy the equation:

Av=λvA \mathbf{v} = \lambda \mathbf{v}Av=λv

This equation states that when matrix AAA is applied to eigenvector v\mathbf{v}v, the result is a scaled version of v\mathbf{v}v by λ\lambdaλ. Eigenvectors are thus directions in the vector space that are only scaled by the matrix AAA.

Hints

```
1. Preferences: Structural Preprocessor:
% Visualize eigenvectors
figure;
hold on;
plot([0 V(1,1)], [0 V(2,1)], 'b', 'LineWidth', 2); % Eigenvector 1
plot([0 V(1,2)], [0 V(2,2)], 'r', 'LineWidth', 2); % Eigenvector 2
axis equal;
legend('Eigenvector 1', 'Eigenvector 2');
title('Eigenvectors of Matrix A');
xlabel('x');
ylabel('y');
grid on;• Solve → Current LS → Ok
General Post Proc:
• Plot results → Temperature distribution
```

Try

1. Instead of temperature apply heat flux on the sides of square and find the temperature distribution.
2. Introduce a hole at the center of square and find the temperature distribution

7. Exercises on Differentiation and Integration

7.1 Higher order differential equations

A higher order differential equation generally takes the form:

F(x,y,dydx,d2ydx2,…,dnydxn)=0F\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \ldots, \frac{d^ny}{dx^n} \right) = 0F(x,y,dxdy,dx2d2y,…,dxndny)=0

where y=y(x)y = y(x)y=y(x) is the unknown function, xxx is the independent variable, and nnn denotes the order of the highest derivative present.

Hints

```
1. MATLAB CODE:
```

```
% Define the differential equation function
function dydx = myODE(x, y)
dydx = zeros(2,1);
dydx(1) = y(2); % This is dy/dx
dydx(2) = -y(1); % This is d^2y/dx^2 + y
end
% Define the range of x values and initial conditions
xspan = [0 10];
y0 = [0 1]; % y(0) = 0, dy/dx(0) = 1
% Solve the differential equation
[x, y] = ode45(@myODE, xspan, y0);
% Plot the solution
plot(x, y(:,1)); % y(:,1) contains the values of y(x)
xlabel('x');
```

```
ylabel('y(x)');
```

```
2. Solution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name
cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all
DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.
2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

7.2 Double Integrals

A double integral is an integral where the integrand is a function of two variables, typically denoted as f(x,y)f(x, y)f(x,y), over a region in the xyxyxy-plane. The notation for a double integral over a region RRR is:

∬Rf(x,y) dx dy\iint_R f(x, y) \, dx \, dy∬Rf(x,y)dxdy

Hints

```
1. Preferences: Structural Preprocessor:
```

```
% Define the function to be integrated
fun = @(x, y) x^2 + y;

% Define the limits of integration
xmin = 0;
xmax = 1;
ymin = 0;
ymax = 2;

% Compute the double integral
Q = integral2(fun, xmin, xmax, ymin, ymax);

disp(['The value of the double integral is: ', num2str(Q)]);
Solution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name
cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all
```

```
DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.
2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

7.3 Triple Integrals

The triple integral of a function f(x,y,z)f(x, y, z)f(x,y,z) over a region DDD in space is denoted as:

∭Df(x,y,z) dV\iiint_D f(x, y, z) \, dV∭Df(x,y,z)dV

where dVdVdV represents the differential volume element. The region DDD is typically described using appropriate bounds for xxx, yyy, and zzz.

Hints

```
1. Preferences: Structural Preprocessor:

% Define the function to be integrated
f = @(x, y, z) x.^2 + y.^2 + z.^2;
% Define the limits for the variables
x_min = 0; x_max = 1;
y_min = 0; y_max = 2;
z_min = 0; z_max = 3;
% Evaluate the triple integral using 'integral3' function
integral_result = integral3(f, x_min, x_max, y_min, y_max, z_min, z_max);
disp(['The value of the triple integral is: ', num2str(integral_result)]);
```

```
Solution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name
cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all
DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.

2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

8. Exercises on Numerical Differention and Integration

8.1 Trapezoidal rule

Numerical integration involves approximating the definite integral of a function over a specified interval. Popular methods include the Trapezoidal rule and Simpson's rule.

The Trapezoidal rule approximates the integral by approximating the area under the curve with trapezoids.

Hints

```
function I = trapezoidal_rule(f, a, b, n)
h = (b - a) / n;
x = a:h:b;
y = f(x);
I = h * (sum(y) - (y(1) + y(end)) / 2);
End
% Example usage:
% Define your function:
f = @(x) sin(x);
a = 0;
b = pi;
n = 100;
integral_approx = trapezoidal_rule(f, a, b, n);
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(m^3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok →
Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
1. Solution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name
cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all
DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.
2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

8.2 Euler method

Given a first-order differential equation dydx=f(x,y)\frac{dy}{dx} = f(x, y)dxdy=f(x,y) with initial condition y(x0)=y0y(x_0) = y_0y(x0)=y0, the Euler method proceeds as follows:

1. **Discretize the Interval:** Choose a step size hhh.
2. **Iterate:** Use the formula yn+1=yn+h·f(xn,yn)y_{n+1} = y_n + h \cdot f(x_n, y_n)yn+1=yn+h·f(xn,yn), where yny_nyn is the approximation of yyy at xnx_nxn.

Hints

```
Preferences: Structural Preprocessor:
% Euler Method Example
% Solving dy/dx = x + y, y(0) = 1
% Define the function f(x, y)
f = @(x, y) x + y;
% Initial condition
x0 = 0;
y0 = 1;
% Step size
h = 0.1;
% Number of steps
N = 10;
% Arrays to store results
x = zeros(1, N+1);
y = zeros(1, N+1);
% Initial values
x(1) = x0;
y(1) = y0;
% Euler method iteration
for n = 1:N
    x(n+1) = x(n) + h;
    y(n+1) = y(n) + h * f(x(n), y(n));
end
% Plotting the results
plot(x, y, '-o');
```

```
xlabel('x');

ylabel('y');

title('Euler Method Solution: dy/dx = x + y');

grid on;Solution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name
cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all
DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.
2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

8.3 RungeKutta method

The Runge-Kutta methods are a family of numerical techniques used for solving ordinary differential equations (ODEs). The most commonly used variant is the fourth-order Runge-Kutta method (RK4), which is known for its balance between accuracy and computational efficiency. Here's a brief outline of RK4:

Hints

```
Preferences: Structural Preprocessor:
```

```
• function [x, y] = RK4(f, x0, y0, h, xmax)
    x = x0:h:xmax;
    y = zeros(size(x));
    y(1) = y0;

    for n = 1:length(x)-1
        k1 = h * f(x(n), y(n));
        k2 = h * f(x(n) + h/2, y(n) + k1/2);
        k3 = h * f(x(n) + h/2, y(n) + k2/2);
        k4 = h * f(x(n) + h, y(n) + k3);

        y(n+1) = y(n) + (k1 + 2*k2 + 2*k3 + k4) / 6;
    end
endSolution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name
cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all
DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.
2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

9. Exercises on 3D Plotting

9.1 Line plotting

A cantilever beam with length of550cm and cross section area of 100 mm2 subjected to nonlinear load.. Find the deflection and nonlinear behavior.

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(m^3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok →
Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
2. Solution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name
cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all
DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.
2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

9.2 Surface plotting

A cantilever beam with length of550cm and cross section area of 100 mm2 subjected to nonlinear load.. Find the deflection and nonlinear behavior.

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
```

```
• Material Properties → Material Models → Structural → Linear → Elastic → Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(m^3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok → Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
2. Solution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.
2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

9.3 Volume plotting

A cantilever beam with length of550cm and cross section area of 100 mm2 subjected to nonlinear load.. Find the deflection and nonlinear behavior.

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic → Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(m^3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok → Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
2. Solution:
```

```
• Analysis Type → New analysis →Transient → Ok
• Parameters →functions→ define/edit→ type in result
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all DOF→ keypoint 1→ ok
```

Try

1. Use a cantilever beam with same length as stated in above problem but with circular cross section and find its nonlinear behavior.
2. Use the same cantilever beam subjected to torque and find its nonlinear behavior.

10. Exercises on Deflection of Simply Supported Beam

10.1 Calculating vertical displacement with point load

A cantilever beam made of mild steel as following specifications L=150 cm, b=10cm, h=10cm shown in Fig 8.1.is subjected to a periodic force, which is mathematically represented below. The amplitude of the force is 1000 N.P = 1000k sin $(\frac{\pi t}{4})$. Find the deflection of beam.

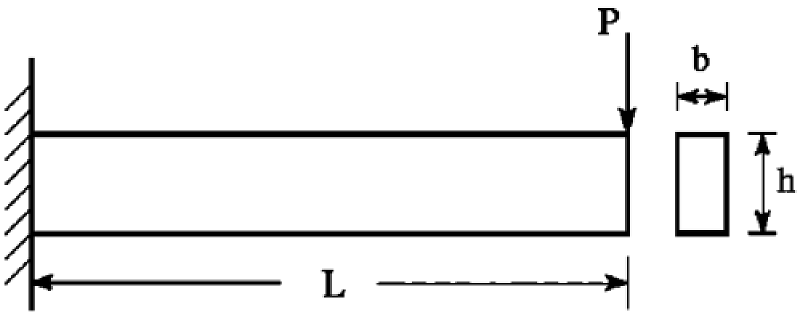


Fig 8.1. Cantilever beam with periodic force

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic → Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(m^3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok → Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
3. Solution:
```

```
• Analysis Type → New analysis → → Ok
• Parameters →functions→ define/edit→ type in result 1000k sin ( πt/4)
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter name cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→ all DOF→ keypoint 1→ ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for simply supported beam.
2. Change the magnitude of periodic load and repeat the same analysis.

10.2Calculating vertical displacement with uniformly distributed load.

A cantilever beam made of mild steel as following specifications L=150 cm, b=10cm, h=10cm shown in Fig 8.1.is subjected to a periodic force, which is mathematically represented below. The amplitude of the force is 1000 N. P = 1000k sin ($\frac{\pi t}{4}$). Find the deflection of beam.

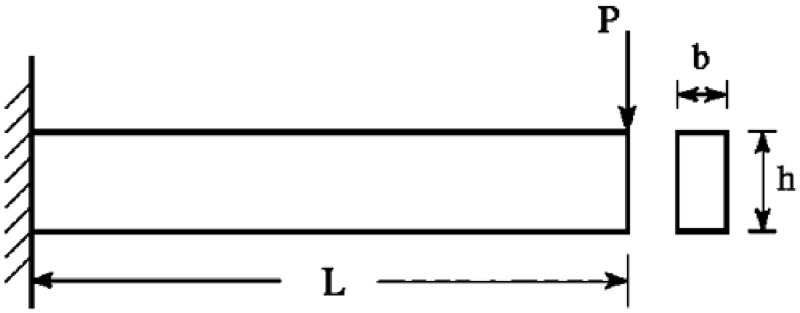


Fig 8.1. Cantilever beam with periodic force

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(m^3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok →
Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
2. Solution:
```

```
• Analysis Type → New analysis → → Ok
• Parameters →functions→ define/edit→ type in result 1000k sin ( πt/4)
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter
name cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→
all DOF→ keypoint 1→ ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for simply supported beam.
2. Change the magnitude of periodic load and repeat the same analysis.

10.3 Calculating vertical displacement with uniformly varying load

A cantilever beam made of mild steel as following specifications L=150 cm, b=10cm, h=10cm shown in Fig 8.1.is subjected to a periodic force, which is mathematically represented below. The amplitude of the force is 1000 N. P = 1000k sin ($\frac{\pi t}{4}$). Find the deflection of beam.

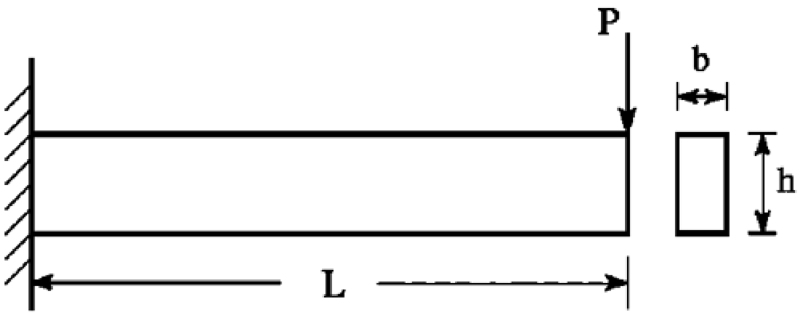


Fig 8.1. Cantilever beam with periodic force

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(mˆ3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok →
Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
2. Solution:
```

```
• Analysis Type → New analysis → → Ok
• Parameters →functions→ define/edit→ type in result 1000k sin ( πt/4)
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter
name cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→
all DOF→ keypoint 1→ ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for simply supported beam.
2. Change the magnitude of periodic load and repeat the same analysis.

11. Exercises on Deflection of Cantilever Beam

11.1 Calculating vertical displacement with point load

A cantilever beam made of mild steel as following specifications L=150 cm, b=10cm, h=10cm shown in Fig 8.1.is subjected to a periodic force, which is mathematically represented below. The amplitude of the force is 1000 N. P = 1000k sin ($\frac{\pi t}{4}$). Find the deflection of beam.

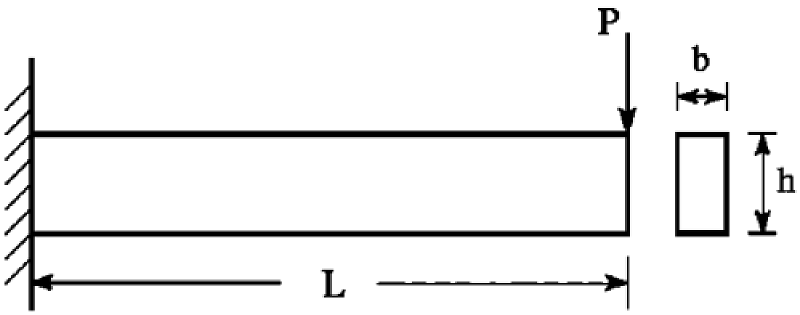


Fig 8.1. Cantilever beam with periodic force

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(mˆ3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok →
Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
2. Solution:
```

```
• Analysis Type → New analysis → → Ok
• Parameters →functions→ define/edit→ type in result 1000k sin ( πt/4)
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter
name cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→
all DOF→ keypoint 1→ ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for simply supported beam.
2. Change the magnitude of periodic load and repeat the same analysis.

11.2 Calculating vertical displacement with uniformly distributed load.

A cantilever beam made of mild steel as following specifications L=150 cm, b=10cm, h=10cm shown in Fig 8.1.is subjected to a periodic force, which is mathematically represented below. The amplitude of the force is 1000 N. P = 1000k sin ($\frac{\pi t}{4}$). Find the deflection of beam.

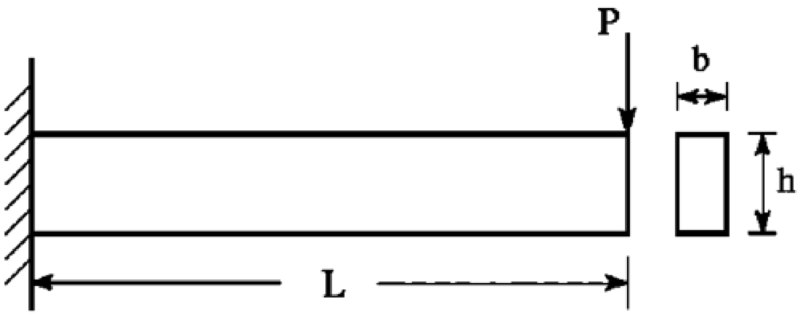


Fig 8.1. Cantilever beam with periodic force

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(mˆ3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok →
Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
2. Solution:
```

```
• Analysis Type → New analysis → → Ok
• Parameters →functions→ define/edit→ type in result 1000k sin ( πt/4)
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter
name cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→
all DOF→ keypoint 1→ ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for simply supported beam.
2. Change the magnitude of periodic load and repeat the same analysis.

11.3 Calculating vertical displacement with uniformly varying load

A cantilever beam made of mild steel as following specifications L=150 cm, b=10cm, h=10cm shown in Fig 8.1.is subjected to a periodic force, which is mathematically represented below. The amplitude of the force is 1000 N. P = 1000k sin $(\frac{\pi t}{4})$. Find the deflection of beam.

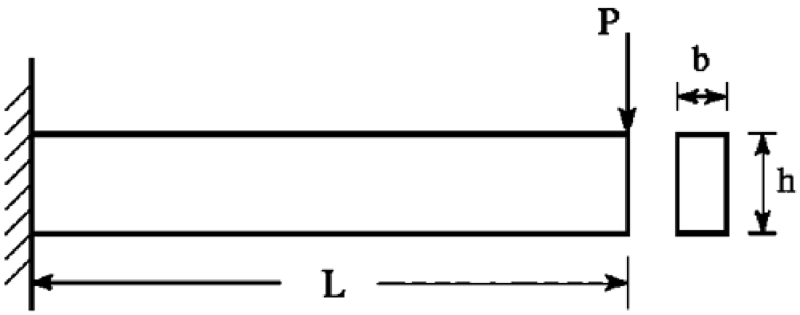


Fig 8.1. Cantilever beam with periodic force

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → Beam 3
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2e11 → PRXY = 0.33 → Density = 7850kg/(m^3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0);(100,0,0) → Ok →
Lines → Areas → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 20 → Ok → Mesh → Lines → Ok
• Select parameters → select functions define
```

```
2. Solution:
```

```
• Analysis Type → New analysis → → Ok
• Parameters →functions→ define/edit→ type in result 1000k sin ( πt/4)
• Select file→ file name= transient→ desktop→ save
• Parameters→ functions→ read from file→ open transient→ give table parameter
name cantilever
• Select loads→ define loads→ apply→ structural→ displacement→ on keypoints→
all DOF→ keypoint 1→ ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for simply supported beam.
2. Change the magnitude of periodic load and repeat the same analysis.

12. Exercises on Formulation of Ideal and Real Gas Equations

12.1 Calculating the pressure, temperature, density for Earth's atmospheric conditions at different altitudes.

Perform static structural analysis of an aircraft rectangular wing of a span 100cm and the chord length of wing is 10 cm and cross-sectional area is 100 cm2. Find the deflection and deformation of wing.

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → wing • Real constants → Add Set1 → Cross section area = 100
cm^2 → Izz = 833.33 cm^4 → Height = 10 cm → Ok → Close
```

```
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2.5e7 → PRXY = 0.28 → Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0) ;(100,0,0) ;(150,0,0)
→ Ok → Lines → Straight Lines → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 20 → Ok → Mesh → Lines → Ok
```

```
2. Solution:
```

```
• Analysis Type → New analysis → Static → Ok
• Define Loads → Apply → Structural → Displacement → On Key Points → Ok → All
DOF Constrained → Ok → Force/Moments → On Key Points → FY = - 10000 → Ok
• Solve → Current LS → Ok
```

```
3.General Post Proc:
```

```
• Plot results → Deformed Shapes → Deformed + Un deformed
• Contour Plots → Nodal Solutions → DOF Solution → Y- Component Displacement
Ok
```

Try

1. Repeat the above analysis for a tapered wing.
2. Repeat the above analysis for tapered wing with uniformly varying load

12.2 Calculating the pressure, temperature, density for other planets at different altitudes.

Perform model analysis of an aircraft rectangular wing of a span 100 cm and the chord length of wing is 10cm and cross-sectional area is 100 cm2. Calculate the natural frequency of wing.

Hints

```
1. Preferences: Structural Preprocessor:
```

```
• Element → Add → wing
• Real constants → Add Set1 → Cross section area = 100 cm^2 → Izz = 833.33
cm^4 → Height = 10 cm → Ok → Close
• Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic → Ex = 2.5e7 → PRXY = 0.28 → Density = 7800kg/(m^3) →Ok
• Modeling → Create → Key points → Inactive CS → (0,0,0) ;(100,0,0) ;(150,0,0)
→ Ok → Lines →Straight Lines → Ok
• Meshing → Size Controls → Manual Sizing → Lines → Picked Lines → No. of
elements = 10 → Ok → Mesh → Lines → Ok 6.4.2
```

```
2. Solution:
```

```
• Analysis Type → New analysis → Modal → Ok
• Analysis Option → Block Lancoz → No. of Modes to extract = 5 → No. of Modes
to expand = 5 → Ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for a tapered wing.
2. Repeat the above analysis for tapered wing with uniformly varying load

13. Exercises on Shear Force and Bending Moment Diagrams-Cantilever Beam

13.1 Calculating shear force and bending moment for point load.

Calculate the deformation of the aluminum fuselage section under the application of internal load of 100000 Pa.The radius of fuselage is 0.15m and thickness is 2 mm.

```
1. Preferences: Structural Preprocessor:
```

```
•Element type → Add / edit/Delete → Add → Solid – 10 node 92→ Apply Add → Beam 2 Node 188 → Apply → Add → Shell →Elastic 4 node 63 Real Constants → Add → Select shell → give thickness (I) = 1→ ok → close.
•Material properties → material models → Structural → Linear → Elastic → Isotropic EX =0.7e11; PRXY = 0.3; Density = 2700
•Pre-processor → modelling → Create → Areas → Circle → Annulus WP x = 0 ; WP y = 0; Rad – 1 = 2.5; Rad - 2 = 2.3 OK Pre-processor → Modelling → Create → Circle → Solid –WP x = 0; X = 2.25; Y = 0 Radius = 0.15 Apply WP x = 0; X = -2.25; Y = 0 Radius = 0.15 Apply WP x = 0; X =0; Y = 2.25; Radius = 0.15 Apply WP x = 0; X = 0; Y = -2.25 Radius = 0.15 OK
•Pre-processor → Modelling → Operate → Booleans → Add → Areas – Pick all OK
•Preprocessor → Modelling → Operate → Extrude → Areas → By XYZ offset X= 0; Y=0; Z = 5
•Pre-processor →Meshing → Size controls → Manual Size → All Areas → give element edge length as 0.15 → ok
• Meshing → Size controls → Manual Size → All lines → give element edge length as→ ok
•Meshing → Mesh → areas → free → select box type instead of single → select the total volume → ok
```

```
2. Solution:
```

```
•Loads → define loads → Apply → Structural → Displacement → On areas → select box type → select box (4 points at centre) → all DOF → ok Select → ALL DOF arrested Define loads → Apply → Structural → Pressure → on areas → select the internal surface of the fuselage and give value (100000) → ok
• Solve → Current LS → Ok
General Post Proc:
• Plot results → Deformed Shapes → Deformed + Un deformed
```

Try

1. Repeat the above analysis for a monocoquefuselage.
2. Repeat the above analysis for a monocoque fuselage with both internal and external load.

13.2 Calculating shear force and bending moment for uniformly distributed load.

Calculate the natural frequency of the aluminum fuselage section under the application of internal load of 100000 Pa. The radius of fuselage is 0.15m and thickness is 2 mm.

```
1. Preferences: Structural Preprocessor:
```

```
•Element type → Add / edit/Delete → Add → Solid – 10 node 92→ Apply Add →
Beam 2 Node 188 → Apply → Add → Shell →Elastic 4 node 63 Real Constants → Add
→ Select shell → give thickness (I) = 1→ ok → close.
•Material properties → material models → Structural → Linear → Elastic →
Isotropic EX =0.7e11; PRXY = 0.3; Density = 2700
•Pre-processor → modelling → Create → Areas → Circle → Annulus WP x = 0 ; WP
y = 0; Rad – 1 = 2.5; Rad - 2 = 2.3 OK Pre-processor → Modelling → Create →
Circle → Solid –WP x = 0; X = 2.25; Y = 0 Radius = 0.15 Apply WP x = 0; X =
-2.25; Y = 0 Radius = 0.15 Apply WP x = 0; X =0; Y = 2.25; Radius = 0.15
Apply WP x = 0; X = 0; Y = -2.25 Radius = 0.15 OK
•Pre-processor → Modelling → Operate → Booleans → Add → Areas – Pick all OK
•Preprocessor → Modelling → Operate → Extrude → Areas → By XYZ offset X= 0;
Y=0; Z = 5
•Pre-processor →Meshing → Size controls → Manual Size → All Areas → give
element edge length as 0.15 → ok
• Meshing → Size controls → Manual Size → All lines → give element edge length
as→ ok
•Meshing → Mesh → areas → free → select box type instead of single → select
the total volume → ok
```

```
2. Solution:
```

```
• Analysis Type → New analysis → Modal → Ok
• Analysis Option → Block Lancoz → No. of Modes to extract = 5 → No. of Modes
to expand = 5 → Ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for a monocoquefuselage.
2. Repeat the above analysis for a monocoque fuselage with both internal and external load.

13.3 Calculating shear force and bending moment for uniformly varying load.

Calculate the natural frequency of the aluminum fuselage section under the application of internal load of 100000 Pa. The radius of fuselage is 0.15m and thickness is 2 mm.

```
1. Preferences: Structural Preprocessor:
```

```
•Element type → Add / edit/Delete → Add → Solid – 10 node 92→ Apply Add →
Beam 2 Node 188 → Apply → Add → Shell →Elastic 4 node 63 Real Constants → Add
→ Select shell → give thickness (I) = 1→ ok → close.
•Material properties → material models → Structural → Linear → Elastic →
Isotropic EX =0.7e11; PRXY = 0.3; Density = 2700
•Pre-processor → modelling → Create → Areas → Circle → Annulus WP x = 0 ; WP
y = 0; Rad – 1 = 2.5; Rad - 2 = 2.3 OK Pre-processor → Modelling → Create →
Circle → Solid –WP x = 0; X = 2.25; Y = 0 Radius = 0.15 Apply WP x = 0; X =
-2.25; Y = 0 Radius = 0.15 Apply WP x = 0; X =0; Y = 2.25; Radius = 0.15
Apply WP x = 0; X = 0; Y = -2.25 Radius = 0.15 OK
```

```
•Pre-processor → Modelling → Operate → Booleans → Add → Areas – Pick all OK
•Preprocessor → Modelling → Operate → Extrude → Areas → By XYZ offset X= 0; Y=0; Z = 5
•Pre-processor →Meshing → Size controls → Manual Size → All Areas → give element edge length as 0.15 → ok
• Meshing → Size controls → Manual Size → All lines → give element edge length as→ ok
•Meshing → Mesh → areas → free → select box type instead of single → select the total volume → ok
```

```
2. Solution:
```

```
• Analysis Type → New analysis → Modal → Ok
• Analysis Option → Block Lanczos → No. of Modes to extract = 5 → No. of Modes to expand = 5 → Ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for a monocoquefuselage.
2. Repeat the above analysis for a monocoque fuselage with both internal and external load.

14.Exercises on Shear Force and Bending Moment Diagrams-Over Hanging Beam

14.1 Calculating shear force and bending moment for point load

A simple retractable landing gear subjected to a load of 10000 N. Find the deformation and stress developed in the landing gear.

```
1. Preferences: Structural Preprocessor:
```

```
•Preferences → Structural → H-Method → OK
•Preprocessor → Element Type → Add → Add → Select Link → 2D spar 1 → Apply
•Preprocessor → Element Type → Add → Add → Select Beam → 2 Node 188 → OK → Close
•RealConstants → Add → Add → Select Type Link 1 → Click OK
•Enter the cross-sectional area =1 → OK → Close
•Material Properties → Material Models → Structural → Linear → Elastic → Isotropic Enter
the Young’s Modulus (EXY) = 3e7
Poisson’s Ratio (PRXY) = 0.3
Sections →Beam →Common Sections →Subtype → Select Solid Circle R=0.5 N=20 T=0, Mesh view
•Preprocessor → Modeling → Create → Key points → In Active CS
•Create the key points according to the table
```

KP no	X	Y	Z
1	0	0	0
2	-12	0	0
3	12	0	0
4	0	-12	0
5	0	-12	0
6	0	12	0

```
•Modeling → Create → Lines → Lines → Straight Lines →
•Join the key points according to table
```

Line no	Join
1	1&4
2	4&5
3	5&6
4	2&5
5	3&4

```
•Preprocessor → Meshing → Mesh Attributes → All lines → Select element type Beam 188, Ok
•Meshing → Mesh tool→ set →Global 1Link1→Ok
•Lines→set→3&4 line click→2&5 line click→ok No of divisions 1→ok
• Mesh Tool→Mesh→Mesh only strut→ok Meshing → Mesh tool→ set →Global 2 Beam 188→Ok Lines→set→1&4 line click→4&5 line click→5&6 line click→ok Element egde length→1→ok
•Mesh Tool→Mesh→Mesh only Vertical line→ok Main menu→ plot Cntrls → Style → Size and Shape Click in the box against Display Element Type
```

```
2. Solution:
```

```
•Define Loads → Apply → Structural → Displacement → On key Points → Select keypoints 2 & 3→select UX,UY,UZ,ROTX,ROTY→Ok Select keypoints 2 & 3→select UX,UZ→Ok
•Modeling →Create¿Nodes→Rotate nodes CS¿Byangles¿click 6th keypoint THXY →60→ok •Loads¿Apply→Structural¿Force/Moment→ click On nodes 28/Key point 6→Force/Moment value →10000
•Solution → Solve→ General Post proc→ List results→ Rection solution→ Plot results→ Defromed shape
```

Try

1. Repeat the above analysis for a landing gear subjected to compression load.

14.2 Calculating shear force and bending moment for uniformly distributed load.

A simple retractable landing gear subjected to a load of 10000 N. Find the deformation and stress developed in the landing gear.

```
1. Preferences: Structural Preprocessor:
```

```
•Preferences → Structural → H-Method → OK
•Preprocessor → Element Type → Add → Add → Select Link → 2D spar 1 → Apply
•Preprocessor → Element Type → Add → Add → Select Beam → 2 Node 188 → OK → Close
•RealConstants → Add → Add → Select Type Link 1 → Click OK
```

```
•Enter the cross sectional area =1 → OK → Close
•Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic Enter
the Young's Modulus (EXY) = 3e7
Poisson's Ratio (PRXY) = 0.3
Sections →Beam →Common Sections →Subtype → Select Solid Circle R=0.5 N=20
T=0, Mesh view
•Preprocessor → Modeling → Create → Key points → In Active CS
•Create the key points according to the table
```

KP no	X	Y	Z
1	0	0	0
2	-12	0	0
3	12	0	0
4	0	-12	0
5	0	-12	0
6	0	12	0

```
•Modeling → Create → Lines → Lines → Straight Lines →
•Join the key points according to table
```

Line no	Join
1	1&4
2	4&5
3	5&6
4	2&5
5	3&4

```
•Preprocessor → Meshing → Mesh Attributes → All lines → Select element type
Beam 188, Ok
•Meshing → Mesh tool→ set →Global 1Link1→Ok
•Lines→set→3&4 line click→2&5 line click→ok No of divisions 1→ok
• Mesh Tool→Mesh→Mesh only strut→ok Meshing → Mesh tool→ set →Global 2 Beam
188→Ok Lines→set→1&4 line click→4&5 line click→5&6 line click→ok Element egde
length→1→ok
•Mesh Tool→Mesh→Mesh only Vertical line→ok Main menu→ plot Cntrls → Style →
Size and Shape Click in the box against Display Element Type
```

```
3. Solution:
```

```
• Analysis Type → New analysis → Modal → Ok
• Analysis Option → Block Lancoz → No. of Modes to extract = 5 → No. of Modes
to expand = 5 → Ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for a landing gear subjected to compression load.

14.3. Calculating shear force and bending moment for uniformly varying load

A simple retractable landing gear subjected to a load of 10000 N. Find the deformation and stress developed in the landing gear.

```
1. Preferences: Structural Preprocessor:
```

```
•Preferences → Structural → H-Method → OK
```

```
•Preprocessor → Element Type → Add → Add → Select Link → 2D spar 1 → Apply
•Preprocessor → Element Type → Add → Add → Select Beam → 2 Node 188 → OK →
Close
•RealConstants → Add → Add → Select Type Link 1 → Click OK
•Enter the cross sectional area =1 → OK → Close
•Material Properties → Material Models → Structural → Linear → Elastic →
Isotropic Enter
the Young's Modulus (EXY) = 3e7
Poisson's Ratio (PRXY) = 0.3
Sections →Beam →Common Sections →Subtype → Select Solid Circle R=0.5 N=20
T=0, Mesh view
•Preprocessor → Modeling → Create → Key points → In Active CS
•Create the key points according to the table
```

KP no	X	Y	Z
1	0	0	0
2	-12	0	0
3	12	0	0
4	0	-12	0
5	0	-12	0
6	0	12	0

```
•Modeling → Create → Lines → Lines → Straight Lines →
•Join the key points according to table
```

Line no	Join
1	1&4
2	4&5
3	5&6
4	2&5
5	3&4

```
•Preprocessor → Meshing → Mesh Attributes → All lines → Select element type
Beam 188, Ok
•Meshing → Mesh tool→ set →Global 1Link1→Ok
•Lines→set→3&4 line click→2&5 line click→ok No of divisions 1→ok
• Mesh Tool→Mesh→Mesh only strut→ok Meshing → Mesh tool→ set →Global 2 Beam
188→Ok Lines→set→1&4 line click→4&5 line click→5&6 line click→ok Element egde
length→1→ok
•Mesh Tool→Mesh→Mesh only Vertical line→ok Main menu→ plot Cntrls → Style →
Size and Shape Click in the box against Display Element Type
```

```
2. Solution:
```

```
• Analysis Type → New analysis → Modal → Ok
• Analysis Option → Block Lancoz → No. of Modes to extract = 5 → No. of Modes
to expand = 5 → Ok
• Solve → Current LS → Ok
```

Try

1. Repeat the above analysis for a landing gear subjected to compression load.

V. TEXT BOOKS:

1. Cleve Moler, "Numerical Computing with MATLAB", SIAM, Philadelphia, 2nd edition, 2008.

VI. REFERENCE BOOKS:

1. Dean G. Duffy, "Advanced Engineering Mathematics with MATLAB", CRC Press, Taylor & Francis Group, 6th edition, 2015.

2. Delores M. Etter, David C. Kuncicky, Holly Moore, “Introduction to MATLAB 7”, Pearson Education Inc, 1st edition, 2009.
3. Rao. V. Dukkipati, “MATLAB for ME Engineers”, New Age Science, 1st edition, 2008.

VII. ELECTRONICS RESOURCES:

1. http://www.tutorialspoint.com/matlab/

VIII. MATERIALS ONLINE

1. Course template
2. Lab manual