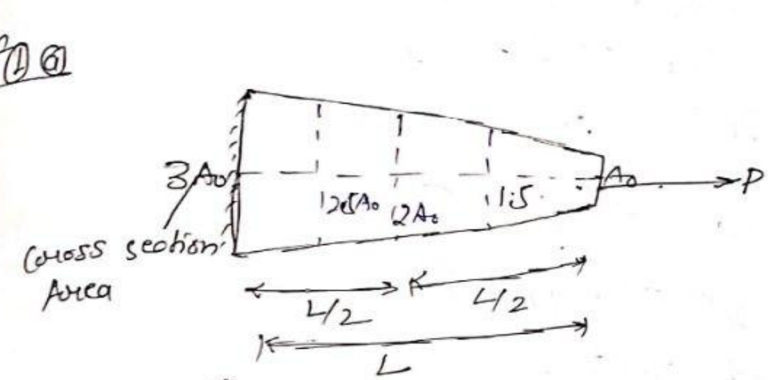
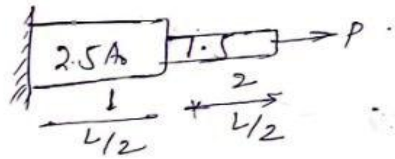
Assignment (3)   22M0090

Sol^n (1) (a)



Uniform thickness

According to question



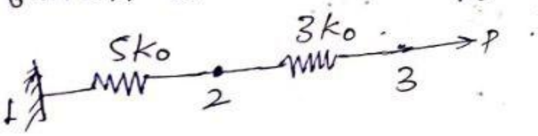
$$k_1 = \frac{2.5 A_0 E}{L/2} = \frac{5 A_0 E}{L}$$

$$k_2 = \frac{1.5 A_0 E}{L/2} = 3\frac{A_0 E}{L}$$

Assume $K_0 = \frac{A_0 E}{L}$

So, $k_1 = 5K_0$ & $k_2 = 3K_0$

Convert stiffness (spring)



Matrix formation

$$\begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix} = \begin{bmatrix} 5K_0 & -5K_0 & 0 \\ -5K_0 & 5K_0+3K_0 & -3K_0 \\ 0 & -3K_0 & 3K_0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}$$

Boundary Condition $U_1 = 0$, $R_1 = R$, $R_2 = 0$, $R_3 = P$

$$\begin{bmatrix} R_1 \\ 0 \\ P \end{bmatrix} = \begin{bmatrix} 5k_0 & -5k_0 & 0 \\ -5k_0 & 8k_0 & -3k_0 \\ 0 & -3k_0 & 3k_0 \end{bmatrix} \begin{bmatrix} 0 \\ U_2 \\ U_3 \end{bmatrix}$$

$$\boxed{R_1 = -5k_0 U_2} \quad \text{— (1)}$$

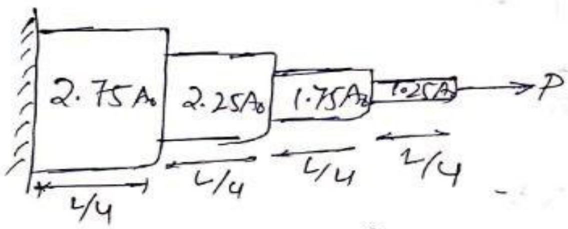
$$\boxed{0 = 8k_0 U_2 - 3k_0 U_3} \quad \text{— (2)}$$

$$\boxed{P = -3k_0 U_2 + 3k_0 U_3} \quad \text{— (3)}$$

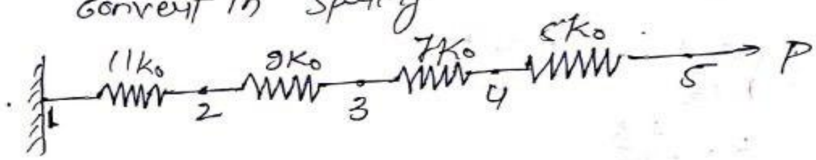
From eqn (2) and eqn (3).

$$\boxed{U_3 = 0.534 P/k_0} \quad \text{Ans-} \quad \boxed{U_2 = 0.2 P/k_0}$$

Second part



convert in spring



$$k_1 = \frac{2.75 A_0 E}{L/4} = \frac{11 A_0 E}{L} = 11 k_0$$

$$k_2 = \frac{2.25 A_0 E}{L/4} = \frac{9 A_0 E}{L} = 9 k_0$$

$$k_3 = \frac{1.75 A_0 E}{L/4} = \frac{7 A_0 E}{L} = 7 k_0$$

$$k_4 = \frac{1.25 A_0 E}{L/4} = \frac{5 A_0 E}{L} = 5 k_0$$

Boundary condition:

$R_1 = R$, $R_2 = 0$, $R_3 = 0$, $R_4 = 0$, $R_5 = P$

$U_1 = 0$, $U_2 = ?$, $U_3 = ?$,

$$\begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{bmatrix} = \begin{bmatrix} 11K_0 & -11K_0 & 0 & 0 & 0 \\ -11K_0 & 11K_0+9K_0 & -9K_0 & 0 & 0 \\ 0 & -9K_0 & 9K_0+7K_0 & -7K_0 & 0 \\ 0 & 0 & -7K_0 & 7K_0+5K_0 & -5K_0 \\ 0 & 0 & 0 & -5K_0 & 5K_0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix}$$

On solving these eq^n (Matrix).

$$U_4 = 0.629U_5 .$$

$$-5K_0U_4 + 5K_0U_5 = P .$$

$$U_5 = \frac{P/K_0}{1.85} = .540 \, P/K_0 \text{ Ans.}$$

$$\boxed{U_5 = 0.540\frac{P}{K_0}}$$

(b) Actual According to question

$$U_5 = \frac{PL}{2A_0E} \ln 3$$

$$= \frac{P}{K_0}(0.549)$$

— for 2 Element

$$\text{percentage Error} = \frac{\left(0.549\frac{P}{K_0}\right) - \left(0.534\frac{P_0}{K_0}\right)}{0.549\, P/K_0} .$$

$$= 2.9\% . \text{ Ans.}$$

— for 4 Element

$$\text{percentage Error} = \frac{\left(0.549\frac{P}{K_0}\right) - \left(0.534\frac{P_0}{K_0}\right)}{0.549\, P/K_0}$$

$$= 1.62\% \text{ - Ans}$$

More Element system is more accurate.

Input for C, D & E:-

```
Young's Modulus of bar material: 5000000
Length of bar: 500
Cross section Area at the free end: 10
Value of force applied at end: 1000
Input array of no. of Quadratic elements to discretize: [2 4 6 8 10]
```

C.

```
% Linear elements - Displacement of bar due to load at free end, fixed at
another

clc,clearvars,format compact,close all
E = input("Young's Modulus of bar material: ");
l = input("Length of bar: ");
A_l = input("cross section Area at the free end: ");
P = input("Value of force applied at end: ");
n = input("Input array of no. of linear elements to discretize: ");
disp_end_lin = [];
for i = 1:length(n)
 e_n = zeros(n(i),2); %element node connectivity relation
 for j = 1:n(i)
 e_n(j,:) = [j j+1]; %i th row gives i th element's node connectivity
 end
 l_e = l/n(i); %Length of each element
 no_nodes = max(e_n,[],'all'); %no. of nodes
 force = zeros(no_nodes,1);
 stiffness = zeros(no_nodes);
 displacement = zeros(no_nodes,1);


 g = 1;
 h = setdiff(1:no_nodes,g);
 r = no_nodes;
 force(r,1) = P;
 reduced_force = force(h, :);
 %Creating Stiffness matrix
```

```
 for j = 1:n(i)


 t = 3 - (((2*j)-1)/n(i));
 T = (E*A_l*t)/l_e;
 r = e_n(j,1); % left node no. of element j
 s = e_n(j,2); % right node no. of element j
 stiffness(r,r) = stiffness(r,r) + T;
 stiffness(s,s) = stiffness(s,s) + T;
 stiffness(r,s) = stiffness(r,s) - T;
 stiffness(s,r) = stiffness(s,r) - T;
 end
 zero_disp_nodes = g;
 active_nodes = setdiff(1:no_nodes,zero_disp_nodes); % nodes with unknown
displacement
 reduced_stiffness = stiffness;
 reduced_stiffness = reduced_stiffness(:,active_nodes); %A([3,9],:) = [];
Deletes rows
 reduced_stiffness = reduced_stiffness(active_nodes,:); %keeps rows and
columns of active nodes in stifness matrix in order to solve for unknown
displacements.
 unknown_disp = inv(reduced_stiffness)*reduced_force;
 for j = 1:length(active_nodes)
 displacement(active_nodes(j)) = unknown_disp(j);
 end
 fprintf("\n%d linear elements", n(i))
 fprintf("\nDisplacement at end = %d", displacement(no_nodes))
 disp_end_lin(end+1) = displacement(no_nodes);
end
%Plotting
X = n;
Y = disp_end_lin;
plot(X,Y);
hold on
plot(X,Y,'bv','MarkerFaceColor','b');
title('Displacement at End vs. No. of Elements')
ylabel('Displacement')
xlabel('No. of Elements')
```

```
grid on
analy_soln = (P*l*log(3))/(2*E*A_l);
yline(analy_soln,'m','LineWidth',1.5)
legend('','FEM Solution(Linear)','Exact
Solution','Location','Northwest','Orientation','vertical');
```

Output:-

```
2 linear elements
Displacement at end = 5.333333e-03

4 linear elements
Displacement at end = 5.448773e-03

6 linear elements
Displacement at end = 5.472906e-03

8 linear elements
Displacement at end = 5.481624e-03

10 linear elements
Displacement at end = 5.485710e-03
```

D.

```
% Quadratic elements
clc,clearvars,format compact,close all
E = input("Young's Modulus of bar material: ");
l = input("Length of bar: ");
A_l = input("cross section Area at the free end: ");
P = input("Value of force applied at end: ");
n = input("Input array of no. of Quadratic elements to
discretize: ");
disp_end_quad = [];
for i = 1:length(n)
 e_n = zeros(n(i),3); %element node connectivity relation


 v = 1;
```

```
for j = 1:n(i)
 e_n(j,:) = [v v+1 v+2]; %i th row gives i th element's node
connectivity
 v = v+2;
 end
 l_e = l/n(i); %Length of each element
 no_nodes = max(e_n,[],'all'); %no. of nodes
 force = zeros(no_nodes,1);
 stiffness = zeros(no_nodes);
 displacement = zeros(no_nodes,1);


 g = 1;
 h = setdiff(1:no_nodes,g);
 r = no_nodes;
 force(r,1) = P;
 reduced_force = force(h, :);
 %Creating Stiffness matrix
 for j = 1:n(i)


 t = 3 - (((2*j)-1)/n(i));
 T = (E*A_l*t)/(3*l_e);
 q = e_n(j,1); % left node no. of element j
 r = e_n(j,2); % middle node no. of element j
 s = e_n(j,3); % right node no. of element j
 %Diagonal elements
 stiffness(q,q) = stiffness(q,q) + (7*T);
 stiffness(r,r) = stiffness(r,r) + (16*T);
 stiffness(s,s) = stiffness(s,s) + (7*T);


 %Upper diagonal elements
 stiffness(q,r) = stiffness(r,s) - (8*T);
 stiffness(q,s) = stiffness(s,r) + T;
 stiffness(r,s) = stiffness(r,s) - (8*T);


 %Lower diagonal elements
```

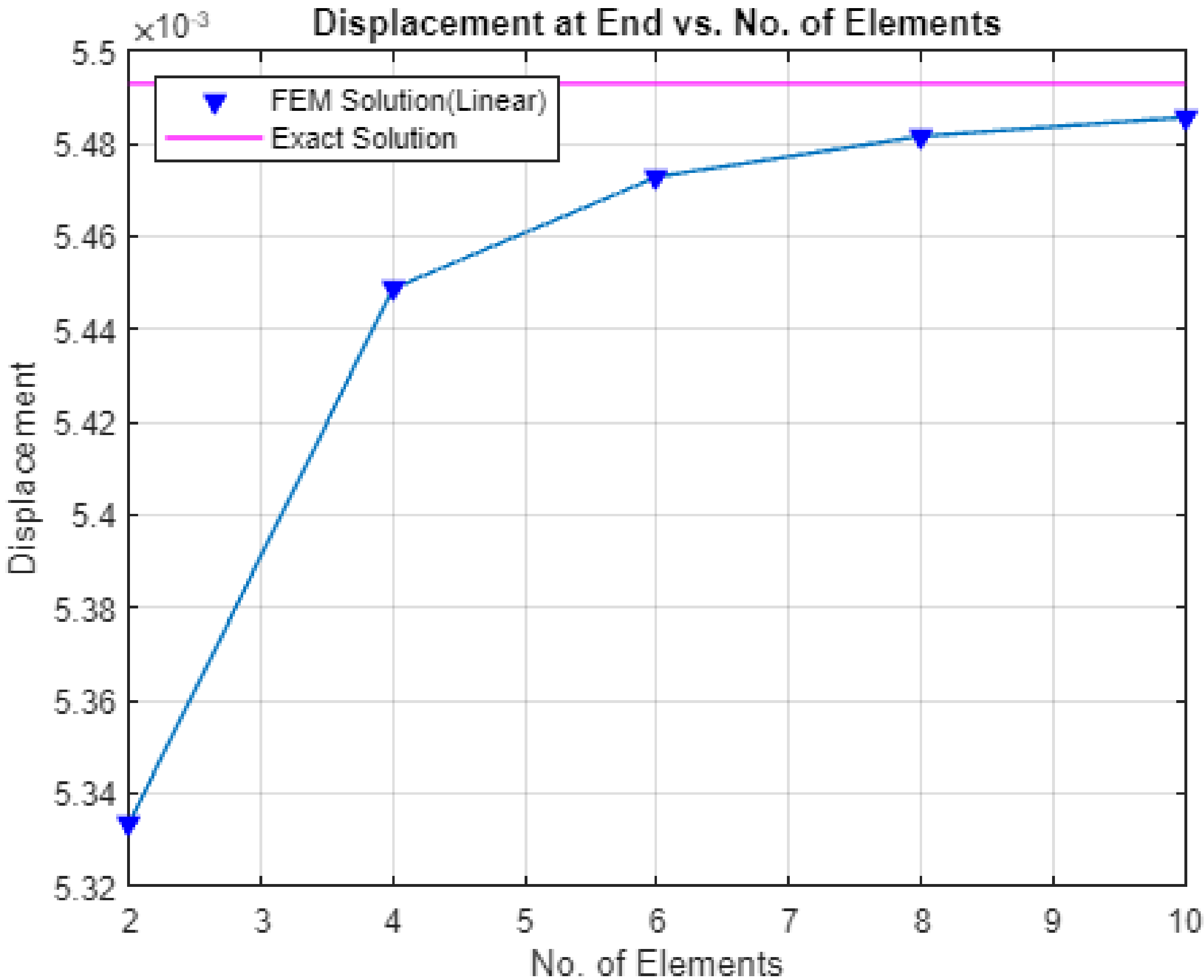
```
 stiffness(r,q) = stiffness(q,r);
 stiffness(s,q) = stiffness(q,s);
 stiffness(s,r) = stiffness(r,s);


 end
 zero_disp_nodes = g;
 active_nodes = setdiff(1:no_nodes,zero_disp_nodes); % nodes with
unknown displacement
 reduced_stiffness = stiffness;
 reduced_stiffness = reduced_stiffness(:,active_nodes);
%A([3,9],:) =[]; Deletes rows
 reduced_stiffness = reduced_stiffness(active_nodes,:); %keeps
rows and columns of active nodes in stifness matrix in order to
solve for unknown displacements.
 unknown_disp = inv(reduced_stiffness)*reduced_force;
 for j = 1:length(active_nodes)
 displacement(active_nodes(j)) = unknown_disp(j);
 end
 fprintf("\n%d Quadratic elements", n(i))
 fprintf("\nDisplacement at end = %d", displacement(no_nodes))
 disp_end_quad(end+1) = displacement(no_nodes);
end

%Plotting
X = n;
Y = disp_end_quad;
plot(X,Y);
hold on
plot(X,Y,'bv','MarkerFaceColor','b');
title('Displacement at End vs. No. of Elements')
ylabel('Displacement')
xlabel('No. of Elements')
grid on
analy_soln = (P*l*log(3))/(2*E*A_l);
yline(analy_soln,'m','LineWidth',1.5)
legend('','FEM Solution','Exact
Solution','Location','Northwest','Orientation','vertical');
```

Output:-

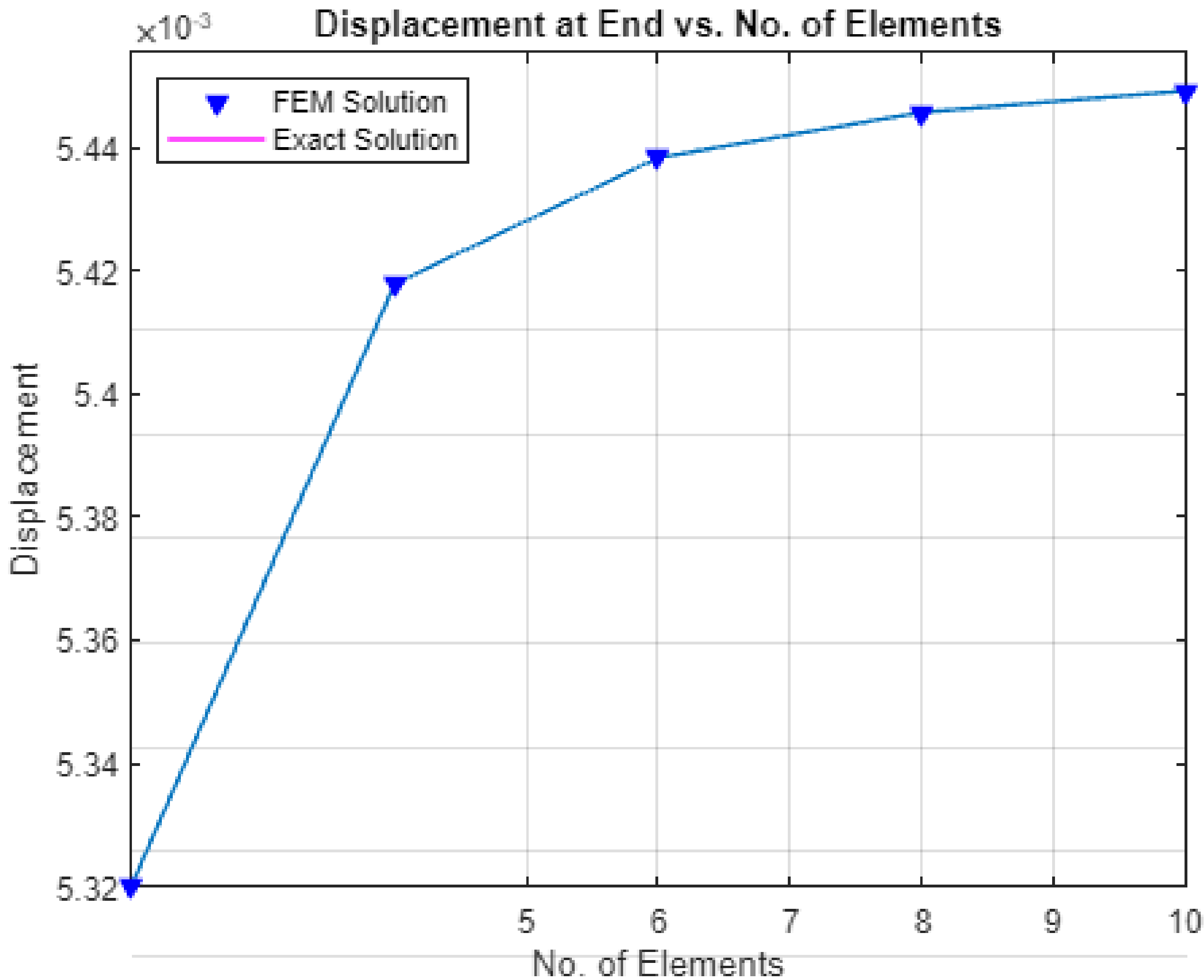
```
2 Quadratic elements
Displacement at end = 5.333333e-03
4 Quadratic elements
Displacement at end = 5.448773e-03
6 Quadratic elements
Displacement at end = 5.472906e-03
8 Quadratic elements
Displacement at end = 5.481624e-03
10 Quadratic elements
Displacement at end = 5.485710e-03
```

E.

Linear elements



Quadratic elements



F. Conclusions:-

- ✓ As number of elements increase, the rate of convergence to exact solution decreases, for both the linear and quadratic elements.
- ✓ So, for infinite number of elements, FEM model gives approximate exact solution.
- ✓ Even with finite number of elements, the error in FEM solution is very small, so can be used up to good approximation.
- ✓ According to the application of problem and computation power available, one can decide the extent of discretization of model, which comes with the penalty of errors.