Istanbul Technical University
Graduate School of Science Engineering and Technology

MKC525E
Finite Element Analysis in Engineering

Final Exam

Erdem Çalışkan
503191531
01/07/2020

1 Question 1

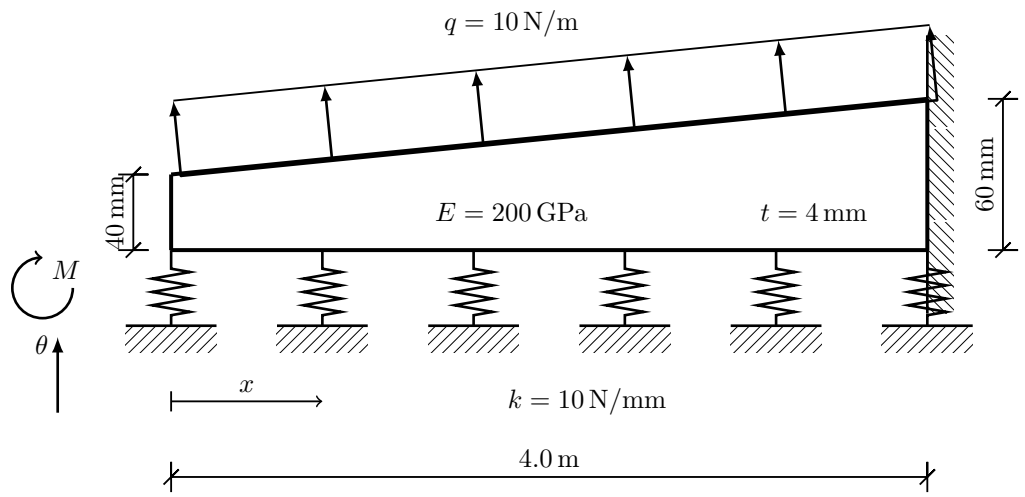


Figure 1: Euler-Bernoulli beam on an elastic foundation under distributed load.

1.1 Strong form of the problem

$l(x)$ distributed vertical load,
Q point force,
M point moment,
$\Omega = [0, 4]$ domain,
E Young modulus,
I moment of inertia.

$$EIu_{,xxxx} = l(x) \text{ on } \Omega \text{ (transverse equilibrium)} \tag{1}$$
$$u(4) = 0 \text{ (zero transverse displacement)} \tag{2}$$
$$u_{,x}(4) = 0 \text{ (zero slope)} \tag{3}$$
$$EIu_{,xx}(0) = M \text{ (prescribed moment)} \tag{4}$$
$$EIu_{,xxx}(0) = Q \text{ (prescribed shear)} \tag{5}$$
$$\tag{6}$$

1.2 Weak form of the problem

Galerkin (weighted residual) method:

$$\text{Residual: } R(x) = EI\frac{\mathrm{d}^4 u}{\mathrm{d}x^4} - l(x) \tag{7}$$
$$\text{Domain: } = \Omega(ie.,\, x \in [0, 4]) \tag{8}$$
$$\int_0^4 \omega(x) R(x)\, \mathrm{d}x = 0 \tag{9}$$
$$\int_0^4 \omega(EIu_{,xxxx} - l)\, \mathrm{d}x = 0 \tag{10}$$
$$\int_0^4 \omega EIu_{,xxxx}\, \mathrm{d}x - \int_0^4 \omega l\, \mathrm{d}x = 0 \tag{11}$$
$$\tag{12}$$

$$(\omega u_{,xxx})_{,x} = \omega_{,x} + \omega u_{,xxxx} \tag{13}$$
$$\omega u_{,xxxx} = (\omega u_{,xxx})_{,x} - \omega_{,x} u_{,xxx} \tag{14}$$

$$(\omega_{,x} u_{,xx})_{,x} = \omega_{,xx} u_{,xx} + \omega_{,x} u_{,xxx} \tag{15}$$
$$\omega_{,x} u_{,xxx} = (\omega_{,x} u_{,xx})_{,x} - \omega_{,xx} u_{,xx} \tag{16}$$

Plug in Equation (4) into (2):

$$\omega u_{,xxxx} = (\omega u_{,xxx})_{,x} - (\omega_{,x} u_{,xx})_{,x} + \omega_{,xx} u_{,xx} \tag{17}$$

$$\int_0^4 (\omega EI u_{,xxx})_{,x} \, \mathrm{d}x + \int_0^4 \omega_{,xx} EI u_{,xx} \, \mathrm{d}x - \int_0^4 (\omega_{,x} EI u_{,xx})_{,x} \, \mathrm{d}x - \int_0^4 \omega l \, \mathrm{d}x = 0 \tag{18}$$

First term of Equation (6):

$$\int_0^4 (\omega EI u_{,xxx})_{,x} \, \mathrm{d}x = (\omega EI u_{,xxx}) \Big|_0^4 \tag{19}$$
$$\omega(4) EI u_{,xxx}(4) - \omega(0) EI u_{,xxx}(0) = -\omega(0) Q(0) \tag{20}$$
$$\omega(4) = 0, \; u_{,xxx}(0) = 0, \; Q(0) = 0 \tag{21}$$

Third term of Equation (6):

$$\int_0^4 (\omega_{,x} EI u_{,xx})_{,x} \, \mathrm{d}x = (\omega_{,x} EI u_{,xx}) \Big|_0^4 \omega_{,x}(4) EI u_{,xx}(4) - \omega_{,x}(0) EI u_{,xx}(0) = -\omega_{,x}(0) M(0) \tag{22}$$
$$\omega_{,x}(4) = 0, \; u_{,xx}(0) = 0, \; M(0) = 0 \tag{23}$$

So equation (6) becomes the weak form of the problem:

$$\int_0^4 \omega_{,xx} EI u_{,xx} \, \mathrm{d}x = \int_0^4 \omega l \, \mathrm{d}x + \omega(0) Q(0) - \omega_{,x}(0) M(0) \tag{24}$$

This equation includes the differantial equation and four boundary condition.
Definition of the opetators:

$$a(\omega, u) = \int_0^4 \omega_{,xx} EI u_{,xx} \, \mathrm{d}x \tag{25}$$
$$a(\omega, u) = \int_0^4 \frac{\mathrm{d}^2 \omega}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} \, \mathrm{d}x \text{ (Energy inner product)} \tag{26}$$
$$(\omega, l) = \int_0^4 \omega l \, \mathrm{d}x \tag{27}$$
$$a(\omega, l) = (\omega, l) + \omega(0) Q - \omega_{,x}(0) M \text{ (Due to tip load and moment)} \tag{28}$$

$$K^e = [K_{ab}] \tag{29}$$
$$K_{ab} = a(N_a, N_b) \tag{30}$$
$$= \int \frac{\mathrm{d}^2 N_a}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 N_b}{\mathrm{d}x^2} \, \mathrm{d}x \tag{31}$$

$$u^h = v^h + g^h \tag{32}$$

ω^h is the shape function

$$a(\omega^h, v^h + g^h) = (\omega^h, l^h) + \omega^h(0) Q - \omega^h_{,x}(0) M \tag{33}$$
$$a(\omega^h, v^h) = (\omega^h, l^h) + \omega^h(0) Q - \omega^h_{,x}(0) M - a(\omega^h, g^h) \tag{34}$$

1.3 Solution

1.3.1 Consistant load vector

The element formulation in the book by Cook, Malkus and Plesha "Concepts and Applications of Finite Element Analysis" p.114:

$$[k] = \int_0^L [B]^T EI[B]\,\mathrm{d}x = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{35}$$

The consistent load vector (Cook, Malkus and Plesha "Concepts and Applications of Finite Element Analysis" p.113:):

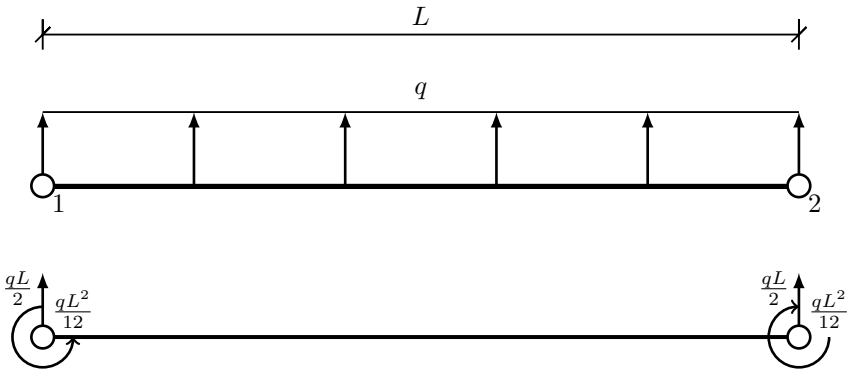


Figure 2: Consistent load vector for 4-DOF element.

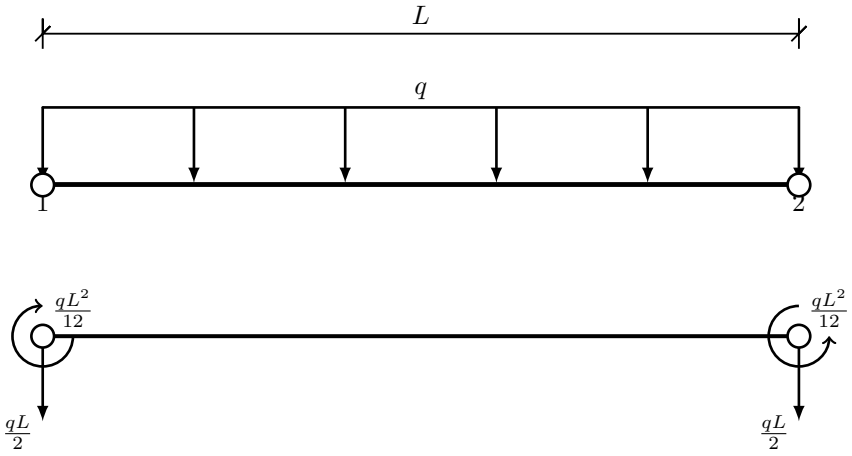


Figure 3: Consistent load vector for 4-DOF element (downward force).

Homogeneous downward force:
$q = 10\,\mathrm{N/m} = 0.01\,\mathrm{N/mm}$
So consitent load vector becomes:

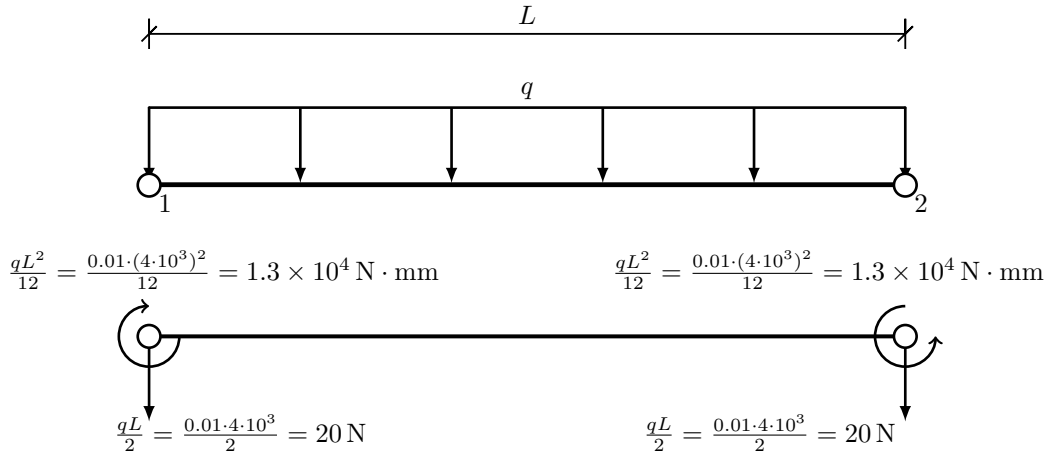


Figure 4: Consistent load vector for one element.

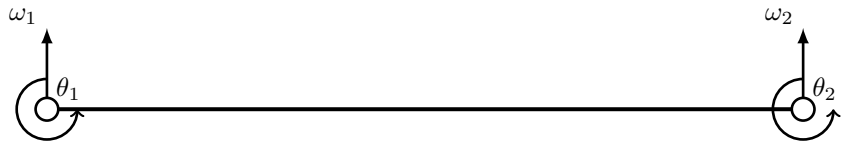


Figure 5: Element degrees of freedom.

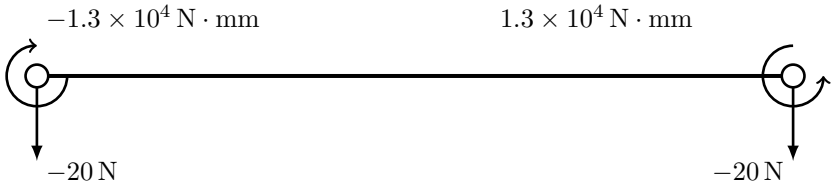


Figure 6: Consistent load vector.

$$[d_e] = \begin{Bmatrix} \omega_1 \\ \theta_1 \\ \omega_2 \\ \theta_2 \end{Bmatrix} \quad [f] = \begin{Bmatrix} -20 \\ -1.3\times10^4 \\ -20 \\ 1.3\times10^4 \end{Bmatrix} \tag{36}$$

1.3.2 Tip deflection for 1 element

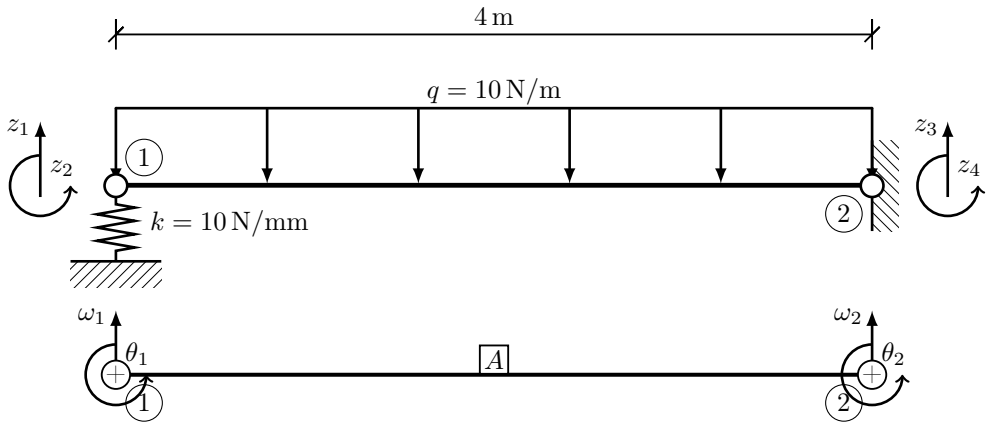


Figure 7: One element.

Schematic for one element is given in Figure 7. In here:

Nodes : (1) (2)
Element : A
Nodal degree of freedoms: $z_1,\ z_2,\ z_3,\ z_4$
Element degree of freedoms: $\omega_1,\ \theta_1,\ \omega_2,\ \theta_2$

$$[\text{Local DOF}] = \begin{bmatrix} w_1 & \theta_1 & w_2 & \theta_2 \end{bmatrix} \tag{37}$$

$$[\text{Top}] = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \tag{38}$$

Boundary condition: (2) is cantilever.
$z_3 = z_4 = 0$

The effect of spring:

Figure 8: Spring force is putted instead of the spring itself.

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{Bmatrix} = \begin{Bmatrix} P - kz_1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{39}$$

Spring constant is added to the corresponding element of $\underline{K}_{global}$:

$$\begin{bmatrix} k_{11}+k & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{Bmatrix} = \begin{Bmatrix} P \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{40}$$

The strain energy is expressed in terms of nodal unknowns and $\underset{\sim}{K}^e$ is founded. $\underset{\sim}{K}^e$ is assembled and $\underset{\sim}{K}_{global}$ is founded (static equation):

$$\underset{\sim}{K}_{global} \cdot \underset{\sim}{d} = \underset{\sim}{f}_{global} \tag{41}$$

The element stiffness matrix

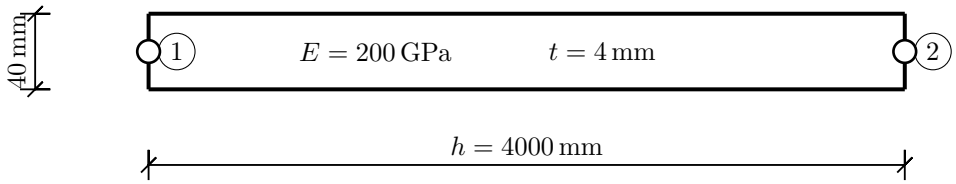


Figure 9: Euler-Bernoulli beam with constant height.

Moment of inertia:

$$I = \frac{bh^3}{12} = \frac{4 * 40^3}{12} = 2.13 \times 10^4\,\mathrm{mm}^4 \tag{42}$$

$$[k] = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{43}$$

$$= \frac{2 \times 10^5 \cdot 2.13 \times 10^4}{(4 \times 10^3)^3}\begin{bmatrix} 12 & 6 \cdot 4 \times 10^3 & -12 & 4 \times 10^3 \\ 6 \cdot 4 \times 10^3 & 4 \cdot (4 \times 10^3)^2 & -6 \cdot 4 \times 10^3 & 2 \cdot (4 \times 10^3)^2 \\ -12 & -6 \cdot 4 \times 10^3 & 12 & -6 \cdot 4 \times 10^3 \\ 6 \cdot 4 \times 10^3 & 2 \cdot (4 \times 10^3)^2 & -6 \cdot 4 \times 10^3 & 4 \cdot (4 \times 10^3)^2 \end{bmatrix} \tag{44}$$

$$= \begin{bmatrix} 0.79875 & 1597.5 & -0.79875 & 1597.5 \\ 1597.5 & 4.26 \times 10^6 & -1597.5 & 2.13 \times 10^6 \\ -0.79875 & -1597.5 & 0.79875 & -1597.5 \\ 1597.5 & 2.13 \times 10^6 & -1597.5 & 4.26 \times 10^6 \end{bmatrix} \tag{45}$$

Spring stiffness is added to the stiffness matrix:

$$\underset{\sim}{K}_{global} = \begin{bmatrix} 10.79875 & 1597.5 & -0.79875 & 1597.5 \\ 1597.5 & 4.26 \times 10^6 & -1597.5 & 2.13 \times 10^6 \\ -0.79875 & -1597.5 & 0.79875 & -1597.5 \\ 1597.5 & 2.13 \times 10^6 & -1597.5 & 4.26 \times 10^6 \end{bmatrix} \tag{46}$$

Global force vector:

$$\underset{\sim}{f}_{global} = f = \begin{Bmatrix} -20 \\ -1.3 \times 10^4 \\ -20 \\ 1.3 \times 10^4 \end{Bmatrix} \tag{47}$$

Displacement vector:

$$d = \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{Bmatrix}, \quad z_3 = 0, \quad z_4 = 0 \tag{48}$$

$$\underset{\sim}{d} = \begin{Bmatrix} z_1 \\ z_2 \\ 0 \\ 0 \end{Bmatrix} \tag{49}$$

So Equation (41) becomes:

$$\begin{bmatrix} 10.79875 & 1597.5 & -0.79875 & 1597.5 \\ 1597.5 & 4.26 \times 10^6 & -1597.5 & 2.13 \times 10^6 \\ -0.79875 & -1597.5 & 0.79875 & -1597.5 \\ 1597.5 & 2.13 \times 10^6 & -1597.5 & 4.26 \times 10^6 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -20 \\ -1.3 \times 10^4 \\ -20 \\ 1.3 \times 10^4 \end{Bmatrix} \tag{50}$$

Apply direct method to get the displacements:

$$\begin{bmatrix} 10.79875 & 1597.5 \\ 1597.5 & 4.26 \times 10^6 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} -20 \\ -1.3 \times 10^4 \end{Bmatrix} \tag{51}$$

Displacements:

$$z_1 = \omega_1 = -1.48\,\text{mm} \;\; \text{(Tip Deflection)} \tag{52}$$

$$z_2 = \theta_1 = -0.002\,\text{rad} \;\; \text{(Tip Rotation)} \tag{53}$$

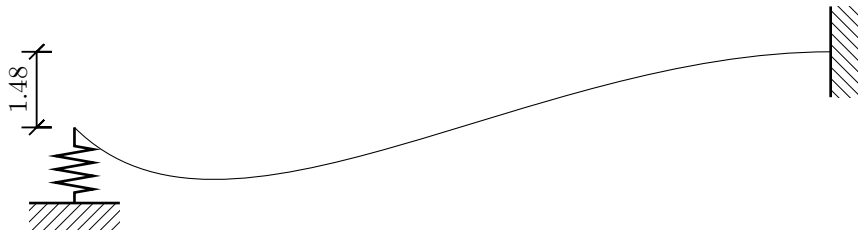


Figure 10: Deformed shape for one element.

1.3.3 Tip deflection for 2 elements

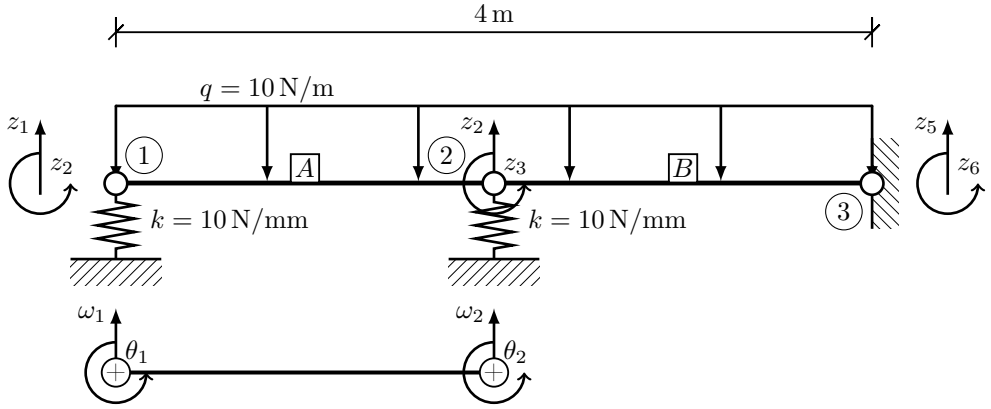


Figure 11: Two elements.

Schematic for two elements is given in Figure 11. In here:

Nodes : (1) (2) (3)
Element : A, B
Nodal degree of freedoms: $z_1,\ z_2,\ z_3,\ z_4,\ z_5,\ z_6$
Element degree of freedoms: $\omega_1,\ \theta_1,\ \omega_2,\ \theta_2$

$$[\text{Local DOF}] = \begin{bmatrix} w_1 & \theta_1 & w_2 & \theta_2 \end{bmatrix} \tag{54}$$

$$[\text{Top}] = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 5 & 6 \end{bmatrix} \tag{55}$$

Boundary condition: (2) is cantilever.
$z_5 = z_6 = 0$

The effect of spring:

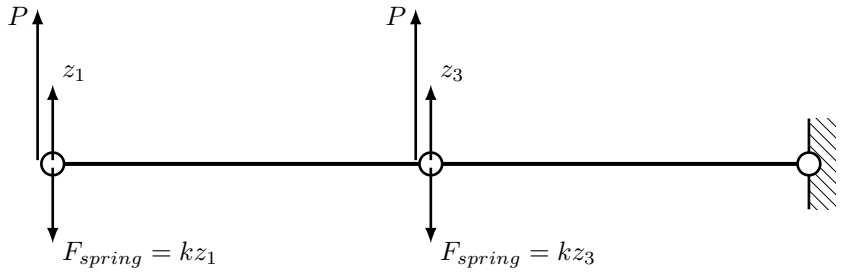


Figure 12: Spring force is putted instead of the spring itself.

Spring constant is added to the corresponding element of $\underset{\sim}{K}_{global}$:

$$\begin{bmatrix} k_{11}+k & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33}+k & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{Bmatrix} = \begin{Bmatrix} P \\ 0 \\ P \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{56}$$

Static equation:

$$\underset{\sim}{K}_{global} \cdot \underset{\sim}{d} = \underset{\sim}{f}_{global} \tag{57}$$

The element stiffness matrix

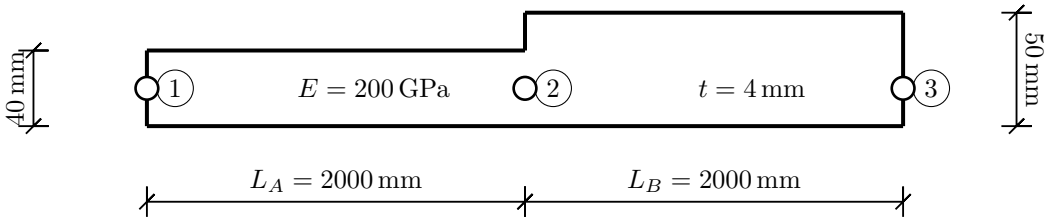


Figure 13: Euler-Bernoulli beam with constant height (Two elements).

Moment of inertia for element A:

$$I_A = \frac{bh^3}{12} = \frac{4 * 40^3}{12} = 2.13 \times 10^4\,\text{mm}^4 \tag{58}$$

$$[k_A] = \frac{2 \times 10^5 \cdot 2.13 \times 10^4}{(2 \times 10^3)^3} \begin{bmatrix} 12 & 6 \cdot 2 \times 10^3 & -12 & 2 \times 10^3 \\ 6 \cdot 2 \times 10^3 & 4 \cdot (2 \times 10^3)^2 & -6 \cdot 2 \times 10^3 & 2 \cdot (2 \times 10^3)^2 \\ -12 & -6 \cdot 2 \times 10^3 & 12 & -6 \cdot 2 \times 10^3 \\ 6 \cdot 2 \times 10^3 & 2 \cdot (2 \times 10^3)^2 & -6 \cdot 2 \times 10^3 & 4 \cdot (2 \times 10^3)^2 \end{bmatrix} \tag{59}$$

$$= \begin{bmatrix} 6.39 & 6390 & -6.39 & 6390 \\ 6390 & 8.52 \times 10^6 & -6390 & 4.26 \times 10^6 \\ -6.39 & -6390 & 6.39 & -6390 \\ 6390 & 4.26 \times 10^6 & -6390 & 8.52 \times 10^6 \end{bmatrix} \tag{60}$$

Moment of inertia for element B:

$$I_B = \frac{bh^3}{12} = \frac{4 * 40^3}{12} = 4.16 \times 10^4\,\text{mm}^4 \tag{61}$$

$$[k_B] = \frac{2\times10^5 \cdot 4.16\times10^4}{(2\times10^3)^3}\begin{bmatrix} 12 & 6\cdot2\times10^3 & -12 & 2\times10^3 \\ 6\cdot2\times10^3 & 4\cdot(2\times10^3)^2 & -6\cdot2\times10^3 & 2\cdot(2\times10^3)^2 \\ -12 & -6\cdot2\times10^3 & 12 & -6\cdot2\times10^3 \\ 6\cdot2\times10^3 & 2\cdot(2\times10^3)^2 & -6\cdot2\times10^3 & 4\cdot(2\times10^3)^2 \end{bmatrix} \tag{62}$$

$$= \begin{bmatrix} 12.48 & 12480 & -6.39 & 12480 \\ 12480 & 16.64\times10^6 & -12480 & 8.32\times10^6 \\ -6.39 & -12480 & 6.39 & -12480 \\ 12480 & 8.32\times10^6 & -12480 & 16.64\times10^6 \end{bmatrix} \tag{63}$$

Spring constant is added to the corresponding element of $\underset{\sim}{K}_{global}$:

$$\underset{\sim}{K}_{global} = \begin{bmatrix} k_{11}^A + k & k_{12}^A & k_{13}^A & k_{14}^A & 0 & 0 \\ k_{21}^A & k_{22}^A & k_{23}^A & k_{24}^A & 0 & 0 \\ k_{31}^A & k_{32}^A & k_{33}^A + k_{11}^B + k & k_{34}^A + k_{12}^B & k_{14}^B & k_{14}^B \\ k_{41}^A & k_{42}^A & k_{43}^A + k_{21}^B & k_{44}^A + k_{22}^B & k_{23}^B & k_{24}^B \\ 0 & 0 & k_{31}^B & k_{32}^B & k_{33}^B & k_{34}^B \\ 0 & 0 & k_{41}^B & k_{42}^B & k_{43}^B & k_{44}^B \end{bmatrix} \tag{64}$$

$$= \begin{bmatrix} 16.39 & 6390 & -6.39 & 6390 & 0 & 0 \\ 6390 & 8.52\times10^6 & -6390 & 4.26\times10^6 & 0 & 0 \\ -6.39 & -6390 & 28.87 & -6090 & -6.39 & 12480 \\ 6390 & 4.26\times10^6 & -6090 & 25.16\times10^6 & -12480 & 8.32\times10^6 \\ 0 & 0 & -6.39 & -12480 & 6.39 & -12480 \\ 0 & 0 & 12480 & 8.32\times10^6 & -12480 & 16.64\times10^6 \end{bmatrix} \tag{65}$$

Element consistent load vector:

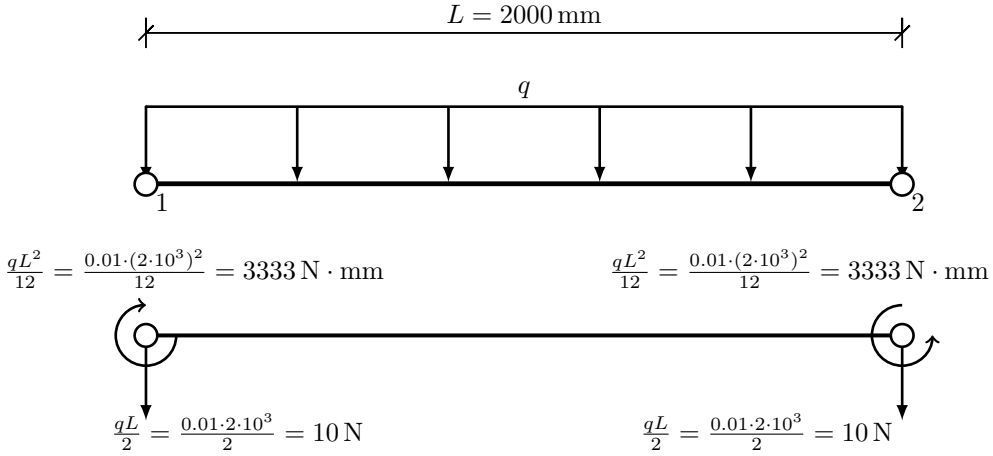


Figure 14: Consistent load vector for one element.

Element force vector:

$$f = \begin{Bmatrix} -10 \\ -3333 \\ -10 \\ 3333 \end{Bmatrix} \tag{66}$$

Global force vector:

$$\underset{\sim}{f}_{global} = \begin{Bmatrix} f_1^A \\ f_2^A \\ f_3^A + f_1^B \\ f_4^A + f_2^B \\ f_3^B \\ f_4^B \end{Bmatrix} = \begin{Bmatrix} -10 \\ -3333 \\ -10{-}10 \\ 3333{-}3333 \\ -10 \\ 3333 \end{Bmatrix} \tag{67}$$

$$= \begin{Bmatrix} -10 \\ -3333 \\ 20 \\ 0 \\ -10 \\ 3333 \end{Bmatrix} \tag{68}$$

Displacement vector:

$$d = \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{Bmatrix}, \quad z_5 = 0, \quad z_6 = 0 \tag{69}$$

$$\underset{\sim}{d} = \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ 0 \\ 0 \end{Bmatrix} \tag{70}$$

So Equation (41) becomes:

$$\begin{bmatrix} 16.39 & 6390 & -6.39 & 6390 & 0 & 0 \\ 6390 & 8.52 \times 10^6 & -6390 & 4.26 \times 10^6 & 0 & 0 \\ -6.39 & -6390 & 28.87 & -6090 & -6.39 & 12480 \\ 6390 & 4.26 \times 10^6 & -6090 & 25.16 \times 10^6 & -12480 & 8.32 \times 10^6 \\ 0 & 0 & -6.39 & -12480 & 6.39 & -12480 \\ 0 & 0 & 12480 & 8.32 \times 10^6 & -12480 & 16.64 \times 10^6 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -10 \\ -3333 \\ 20 \\ 0 \\ -10 \\ 3333 \end{Bmatrix} \tag{71}$$

Apply direct method to get the displacements:

$$\begin{bmatrix} 16.39 & 6390 & -6.39 & 6390 \\ 6390 & 8.52 \times 10^6 & -6390 & 4.26 \times 10^6 \\ -6.39 & -6390 & 28.87 & -6090 \\ 6390 & 4.26 \times 10^6 & -6090 & 25.16 \times 10^6 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{Bmatrix} = \begin{Bmatrix} -10 \\ -3333 \\ 20 \\ 0 \end{Bmatrix} \tag{72}$$

Displacements:

$$z_1 = \omega_1 = -0.6164\,\text{mm} \quad (\textcircled{1}\ \text{Deflection}) \tag{73}$$

$$z_2 = \theta_1 = 0.0004\,\text{rad} \quad (\textcircled{1}\ \text{Rotation}) \tag{74}$$

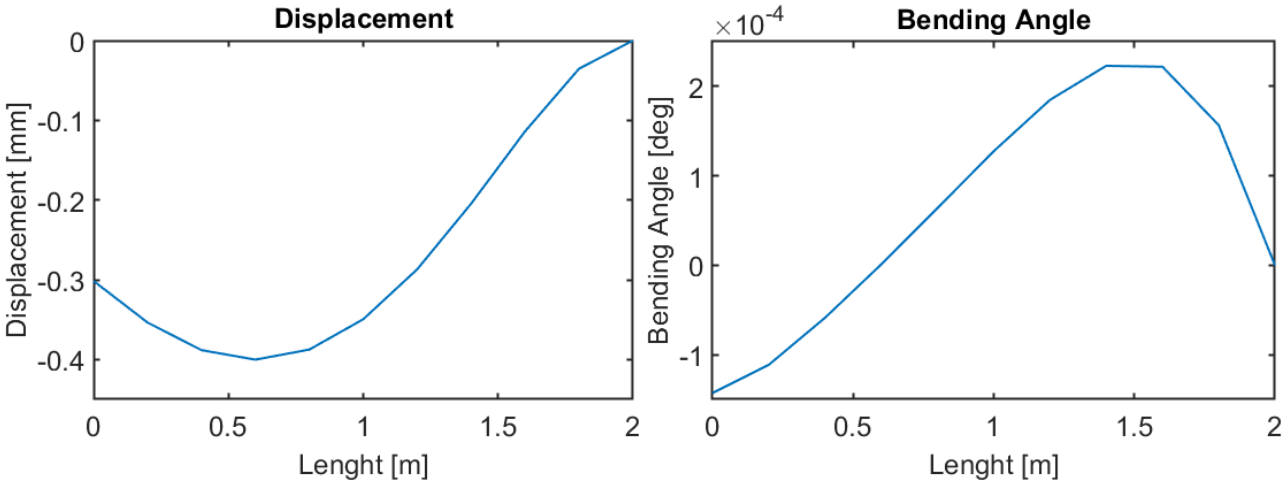
$$z_3 = \omega_2 = 0.715\,73\,\text{m} \quad (\textcircled{2}\ \text{Deflection}) \tag{75}$$

$$z_4 = \theta_2 = 0.0002\,\text{rad} \quad (\textcircled{2}\ \text{Rotation}) \tag{76}$$

1.3.4 Tip deflection for 10 elements

Matlab script for this section is given in Listing 1.

Results:



(a) Deformation at nodes (Matlab).

(b) Bending angle at nodes (Matlab).

Figure 15: Matlab results.

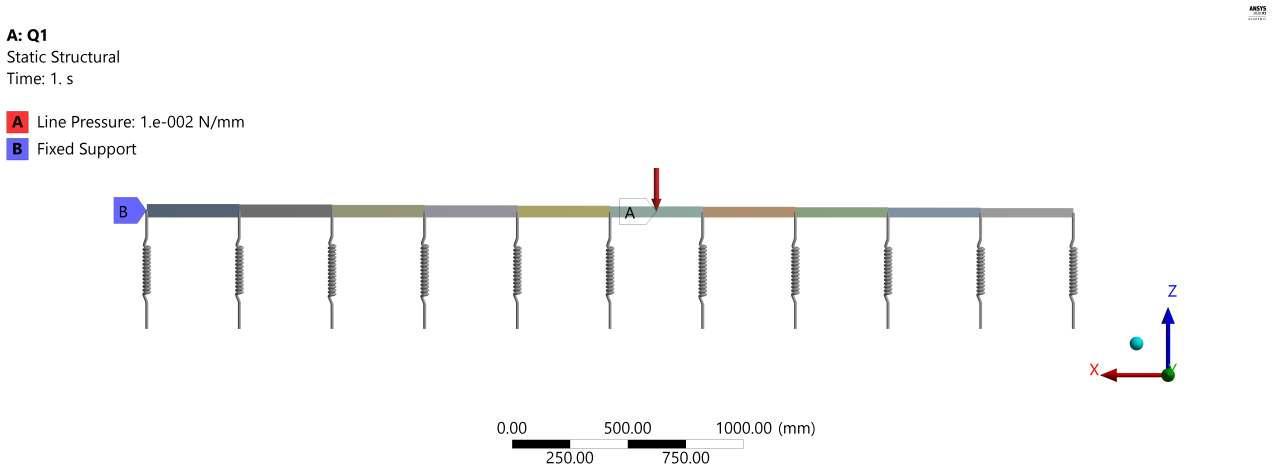


Figure 16: Boundary conditions (Ansys). Linear elements (BEAM188) are used in the model.

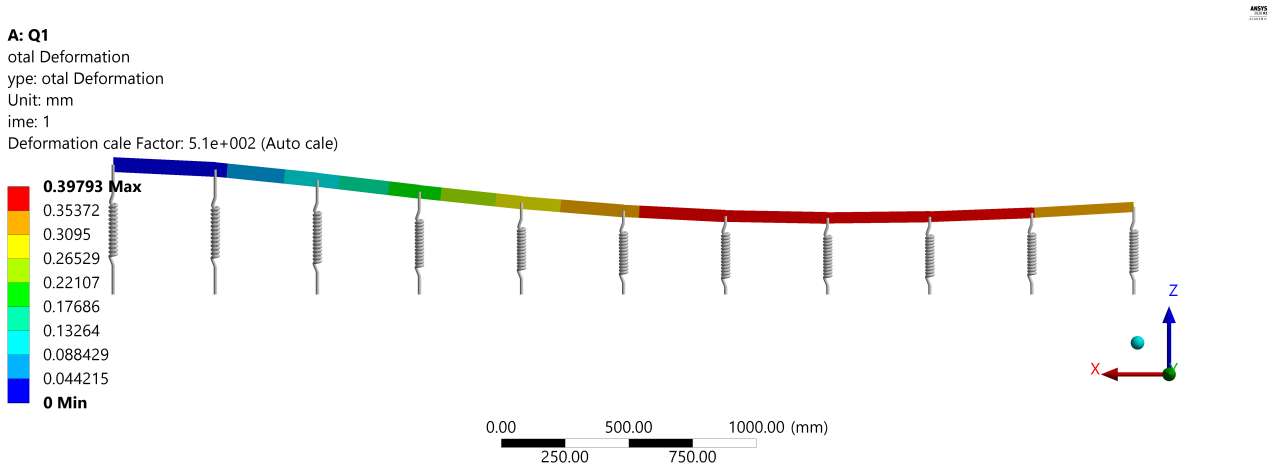


Figure 17: Deformation (Ansys).

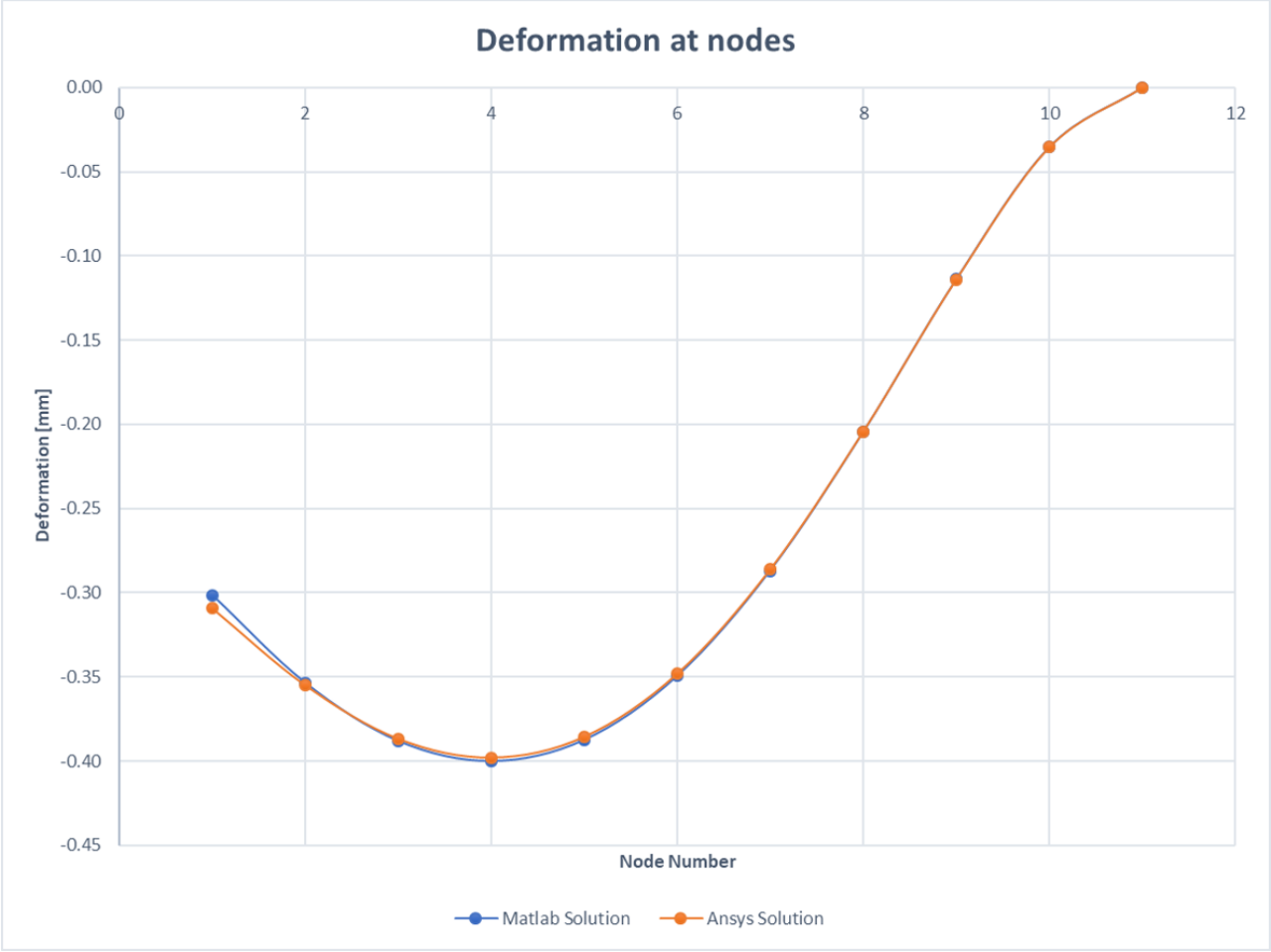


Figure 18: Compression of Ansys and Matlab results (deformation at nodes).

2 Question 2

2.0.1 Element conductivity matrices

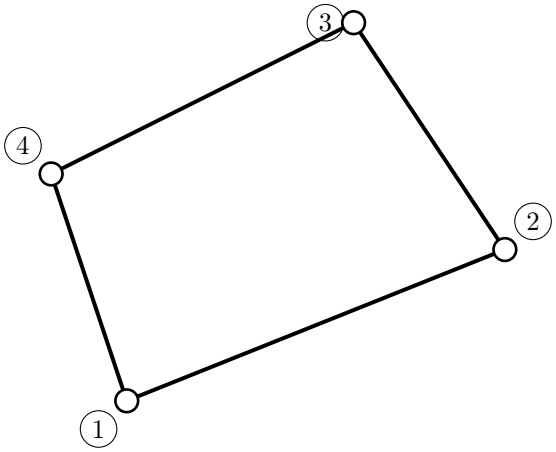


Figure 19: 4-Node element.

In order to evaluate element matrices, Gauss-quadrature rule can be used:

$$\int_{-1}^{+1} f(x)\,\mathrm{d}x = \sum_{i=-1}^{2} \omega_i f(\epsilon_1) \;\; 2 \text{ point}$$

For 2-point Gauss-quadrature rule:
Weights:

$$\omega_1 = \omega_2 = 1 \tag{77}$$

Gauss-points:

$$\epsilon_1 = -\frac{1}{\sqrt{3}} \qquad\qquad \epsilon_2 = \frac{1}{\sqrt{3}} \tag{78}$$

Shape functions:

$$N_1^{4Q}(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta) \tag{79}$$
$$N_2^{4Q}(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta) \tag{80}$$
$$N_3^{4Q}(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta) \tag{81}$$
$$N_4^{4Q}(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta) \tag{82}$$

$$\xi = \epsilon_2 = \frac{1}{\sqrt{3}} \tag{83}$$
$$\eta = \epsilon_1 = -\frac{1}{\sqrt{3}} \tag{84}$$

$$N_1^{4Q} = 0.1667 \tag{85}$$
$$N_2^{4Q} = 0.622 \tag{86}$$
$$N_3^{4Q} = 0.1667 \tag{87}$$
$$N_4^{4Q} = 0.10447 \tag{88}$$

The gradient in the parent domain:

$$\partial N^{4Q} = \frac{1}{4}\begin{bmatrix}\eta-1 & \eta-1 & 1+\eta & -1-\eta\\ \xi-1 & -1-\xi & 1+\xi & 1-\xi\end{bmatrix} \tag{89}$$

Plug in η and ξ values to the gradient:

$$\partial N^{4Q} = \frac{1}{4}\begin{bmatrix}-0.3943 & 0.3943 & 0.1057 & -0.1057\\ -0.1057 & -0.3943 & 0.3943 & 0.1057\end{bmatrix} \tag{90}$$

2.1 For element 1:

The Jacobian matrix:

$$J^{\{1\}} = \left\{\partial N^{4Q}\right\}\left\{x^{\{1\}} \quad y^{\{1\}}\right\} \tag{91}$$

$$= \begin{bmatrix}-0.3943 & 0.3943 & 0.1057 & -0.1057\\ -0.1057 & -0.3943 & 0.3943 & 0.1057\end{bmatrix}\begin{bmatrix}0 & 0.75\\ 1 & 0.85\\ 1 & 1.7\\ 0 & 1.5\end{bmatrix} \tag{92}$$

$$= \begin{bmatrix}0.5 & 0.0606\\ 0 & 0.4144\end{bmatrix} \tag{93}$$

Determinant of the Jacobian matrix:

$$|J^{\{1\}}(\xi,\eta)| = \begin{vmatrix}0.5 & 0.0606\\ 0 & 0.4144\end{vmatrix} = 0.2072 \tag{94}$$

The derivatives of shape functions with respect to the global cartesian coordinates:

$$B^{\{1\}} = \{J^{\{1\}}\}^{-1}(\partial N^{4Q}) = \begin{bmatrix}0.5 & 0.0606\\ 0 & 0.4144\end{bmatrix}^{-1}\begin{bmatrix}-0.3943 & 0.3943 & 0.1057 & -0.1057\\ -0.1057 & -0.3943 & 0.3943 & 0.1057\end{bmatrix} \tag{95}$$

$$= \begin{bmatrix}-0.7577 & 0.9039 & 0.0961 & -0.2423\\ -0.2550 & -0.9514 & 0.9514 & 0.2550\end{bmatrix} \tag{96}$$

The conductance matrix:

$$K = k\sum_{i=1}^{2}\sum_{j=1}^{2}\omega_i\omega_j\left|J^{\{1\}}(\xi_i,\eta_j)\right| {B^{\{1\}}}^{T}(\xi_i,\eta_j)B^{\{1\}}(\xi_i,\eta_j) \tag{97}$$

$$= \begin{bmatrix}13.24 & -9.16 & -6.54 & 2.46\\ -9.16 & 35.69 & -16.96 & -9.57\\ -6.54 & -16.96 & 18.95 & 4.55\\ 2.46 & -9.57 & 4.55 & 2.56\end{bmatrix} \tag{98}$$

2.2 For element 2:

The Jacobian matrix:

$$J^{\{2\}} = \left\{\partial N^{4Q}\right\}\left\{x^{\{2\}} \quad y^{\{2\}}\right\} \tag{99}$$

$$= \begin{bmatrix}-0.3943 & 0.3943 & 0.1057 & -0.1057\\ -0.1057 & -0.3943 & 0.3943 & 0.1057\end{bmatrix}\begin{bmatrix}1 & 0.85\\ 2 & 1.05\\ 2 & 1.9\\ 1 & 1.7\end{bmatrix} \tag{100}$$

$$= \begin{bmatrix}0.5 & 0.1\\ 0 & 0.4250\end{bmatrix} \tag{101}$$

Determinant of the Jacobian matrix:

$$|J^{\{1\}}(\xi,\eta)| = \begin{vmatrix} 0.5 & 0.1 \\ 0 & 0.4250 \end{vmatrix} = 0.2072 \tag{102}$$

$$B^{\{2\}} = \{J^{\{2\}}\}^{-1}(\partial N^{4Q}) = \begin{bmatrix} 0.5 & 0.1 \\ 0 & 0.4250 \end{bmatrix}^{-1} \begin{bmatrix} -0.3943 & 0.3943 & 0.1057 & -0.1057 \\ -0.1057 & -0.3943 & 0.3943 & 0.1057 \end{bmatrix} \tag{103}$$

$$= \begin{bmatrix} -0.738858824 & 0.974152941 & 0.025847059 & -0.261141176 \\ -0.248705882 & -0.927764706 & 0.927764706 & 0.248705882 \end{bmatrix} \tag{104}$$

$$K = k \sum_{i=1}^{2} \sum_{j=1}^{2} \omega_i \omega_j \left| J^{\{2\}}(\xi_i,\eta_j) \right| B^{\{2\}^T}(\xi_i,\eta_j) B^{\{2\}}(\xi_i,\eta_j) \tag{105}$$

$$= \begin{bmatrix} 12.92 & -10.39 & -5.31 & 2.79 \\ -10.39 & 38.46 & -17.76 & -10.31 \\ -5.31 & -17.76 & 18.31 & 4.76 \\ 2.79 & -10.31 & 4.76 & 2.76 \end{bmatrix} \tag{106}$$

2.3 Plate Analysis

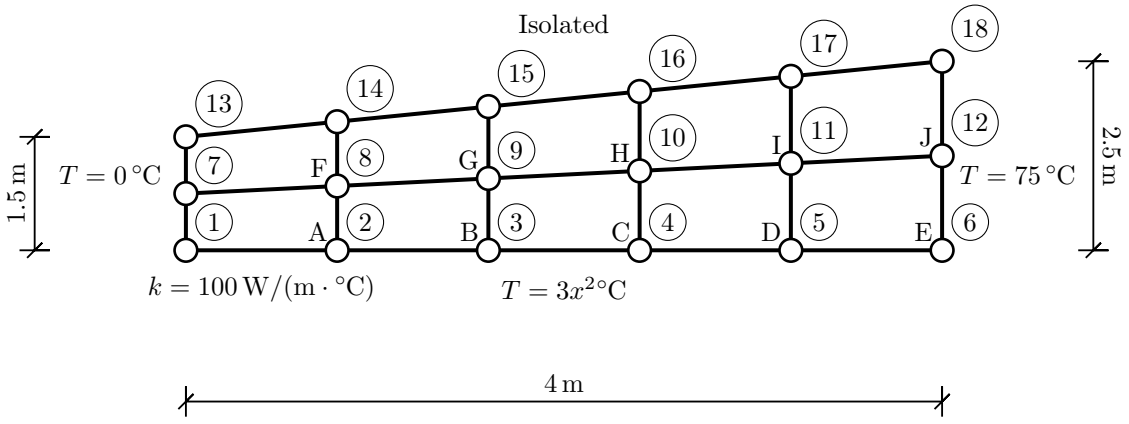


Figure 20: Plate system.

System schematic is given in Figure 20. Here, letters are representing elements and numbers ((1)) are representing nodes.

Boundary conditions:

$$
\begin{aligned}
T_1 &= 0 \\
T_2 &= 3 \\
T_3 &= 12 \\
T_4 &= 27 \\
T_5 &= 48 \\
T_6 &= 75 \\
T_7 &= 0 \\
T_{12} &= 75 \\
T_{13} &= 0 \\
T_{18} &= 75
\end{aligned}
$$

2.3.1 Results

Temperatures (°C) and heat fluxes (W) at nodes:

$$
\begin{aligned}
T_1 &= 0 & q_1 &= -1347.46 \\
T_2 &= 3 & q_2 &= -7527.78 \\
T_3 &= 12 & q_3 &= -18273.34 \\
T_4 &= 27 & q_4 &= -26860.57 \\
T_5 &= 48 & q_5 &= -27898.38 \\
T_6 &= 75 & q_6 &= 34529.31 \\
T_7 &= 0 & q_7 &= -24284.37 \\
T_8 &= 13.58 & q_8 &= 0 \\
T_9 &= 26.59 & q_9 &= 0 \\
T_{10} &= 41.17 & q_{10} &= 0 \\
T_{11} &= 57.75 & q_{11} &= 0 \\
T_{12} &= 75 & q_{12} &= 66452.83 \\
T_{13} &= 0 & q_{13} &= -6492.64 \\
T_{14} &= 14.34 & q_{14} &= 0 \\
T_{15} &= 27.98 & q_{15} &= 0 \\
T_{16} &= 42.87 & q_{16} &= 0 \\
T_{17} &= 59.74 & q_{17} &= 0 \\
T_{18} &= 75 & q_{18} &= 11702.39
\end{aligned}
$$

Matlab code for Question 2 can be found in Listing 2.

3 Question 3

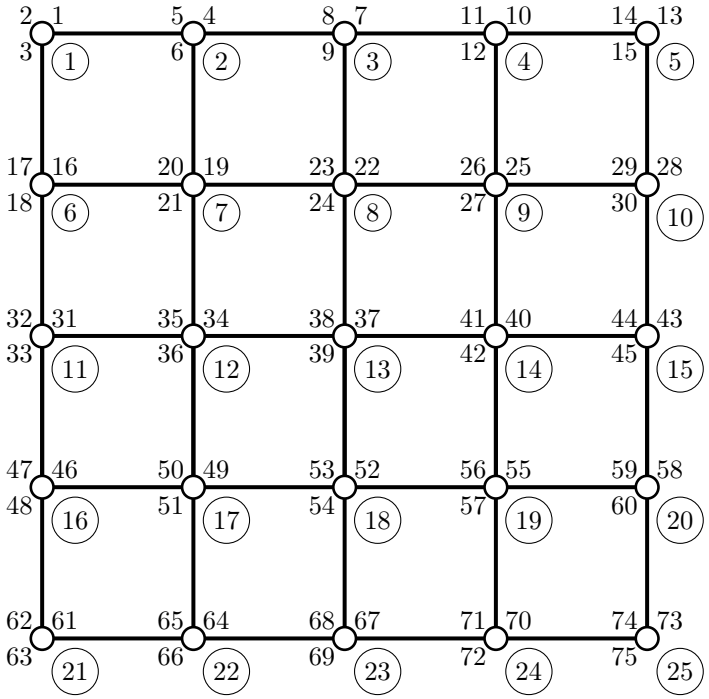


Figure 21: 4x4 meshed plate. (+) direction is counter clockwise.

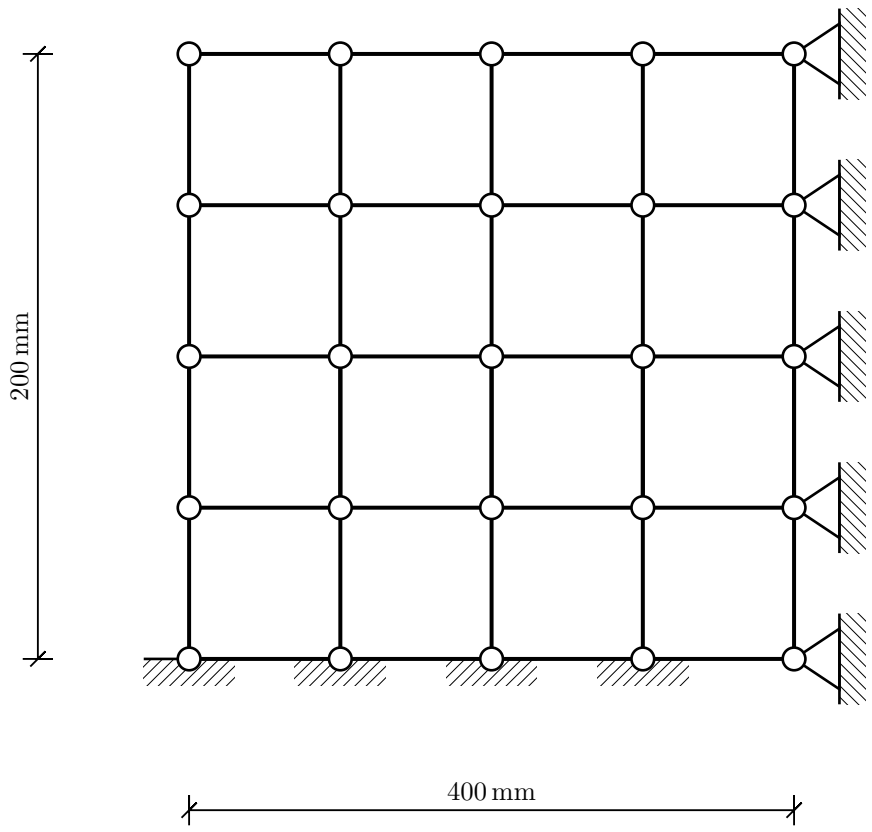


Figure 22: Dimensions and boundary conditions for the plate with thickness of 4 mm. Bottom edge is cantilever and right edge is simply supported. Initialy at rest.

3.1 Topology matrix

$$\begin{array}{cccccccccccc}
16 & 17 & 18 & 19 & 20 & 21 & 4 & 5 & 6 & 1 & 2 & 3 \\
19 & 20 & 21 & 22 & 23 & 24 & 7 & 8 & 9 & 4 & 5 & 6 \\
22 & 23 & 24 & 25 & 26 & 27 & 10 & 11 & 12 & 7 & 8 & 9 \\
25 & 26 & 27 & 28 & 29 & 30 & 13 & 14 & 15 & 10 & 11 & 12 \\
31 & 32 & 33 & 34 & 35 & 36 & 19 & 20 & 21 & 16 & 17 & 18 \\
34 & 35 & 36 & 37 & 38 & 39 & 22 & 23 & 24 & 19 & 20 & 21 \\
37 & 38 & 39 & 40 & 41 & 42 & 25 & 26 & 27 & 22 & 23 & 24 \\
40 & 41 & 42 & 43 & 44 & 45 & 28 & 29 & 30 & 25 & 26 & 27 \\
46 & 47 & 48 & 49 & 50 & 51 & 34 & 35 & 36 & 31 & 32 & 33 \\
49 & 50 & 51 & 52 & 53 & 54 & 37 & 38 & 39 & 34 & 35 & 36 \\
52 & 53 & 54 & 55 & 56 & 57 & 40 & 41 & 42 & 37 & 38 & 39 \\
55 & 56 & 57 & 58 & 59 & 60 & 43 & 44 & 45 & 40 & 41 & 42 \\
61 & 62 & 63 & 64 & 65 & 66 & 49 & 50 & 51 & 46 & 47 & 48 \\
64 & 65 & 66 & 67 & 68 & 69 & 52 & 53 & 54 & 49 & 50 & 51 \\
67 & 68 & 69 & 70 & 71 & 72 & 55 & 56 & 57 & 52 & 53 & 54 \\
70 & 71 & 72 & 73 & 74 & 75 & 58 & 59 & 60 & 55 & 56 & 57
\end{array} \tag{107}$$

3.2 Consistent load vector for the first element

3.3 Deflections for static vertical distributed loading of $0.3\,\mathrm{N/mm^2}$

Nodal deflection matrix is given as 5x5. Nodes and their position in the matrix corresponds to the grid in Figure 21.

$$[\mathrm{ff}] = \begin{Bmatrix} 1500 \\ 25000 \\ -50000 \\ 1500 \\ 25000 \\ 50000 \\ 1500 \\ -25000 \\ 50000 \\ 1500 \\ -25000 \\ -50000 \end{Bmatrix} \qquad [\text{Nodal Deflections}] = \begin{bmatrix} 52.65 & 47.00 & 43.77 & 36.66 & 0 \\ 49.51 & 43.62 & 40.58 & 33.37 & 0 \\ 40.03 & 34.17 & 31.77 & 25.80 & 0 \\ 23.15 & 18.96 & 17.71 & 14.33 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{108}$$

$$\text{Maximum deflection} = 52.65\,\mathrm{mm} \tag{109}$$

3.4 First 10 modes of the plate

First 10 normal modes and constrained modes of the plate are compared. As we can see in Figure 23, normal modes are less than constrained modes in every mode. Also, first normal mode is zero which corresponds to the rigid motion of the plate.

Mode	Natural frequency (Normal) [Hz]	Natural frequency (Constrained) [Hz]
1	0	80.72
2	37.35	100.31
3	74.85	129.74
4	112.56	163.41
5	148.82	197.88
6	157.95	232.79
7	169.48	246.80
8	184.23	258.09
9	200.01	278.25
10	220.56	296.96

Table 1: First 10 modes of the plate

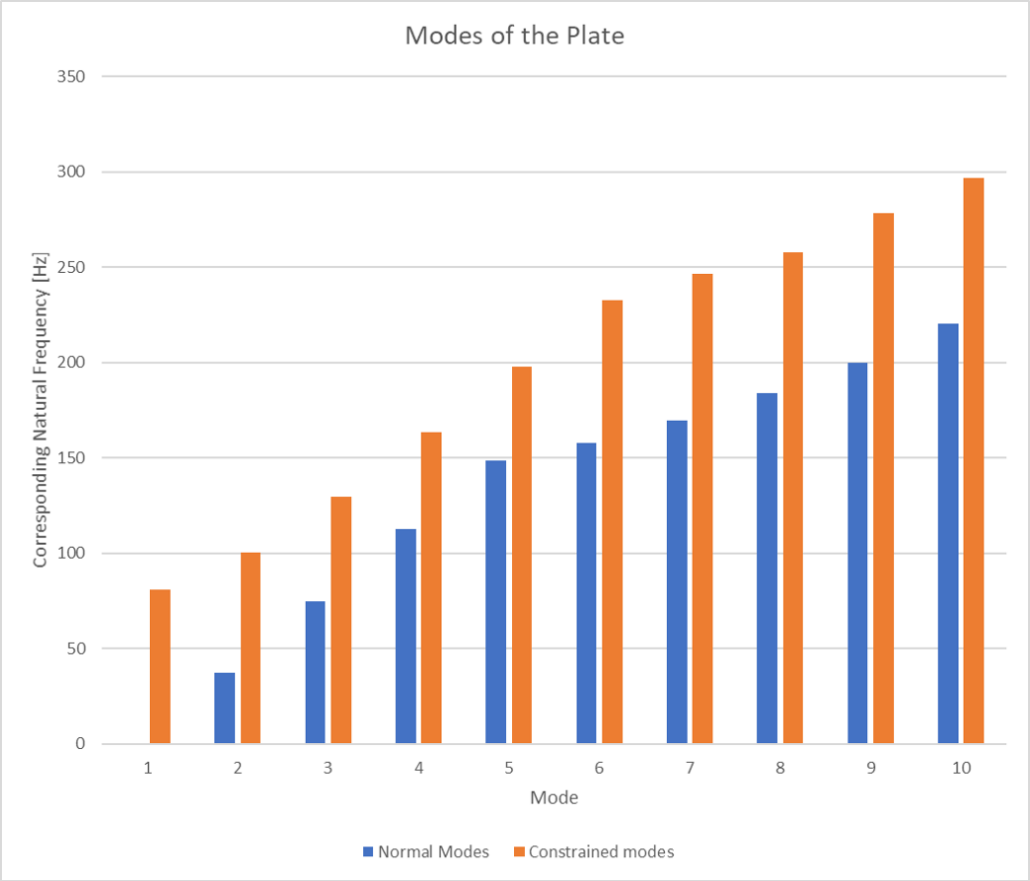
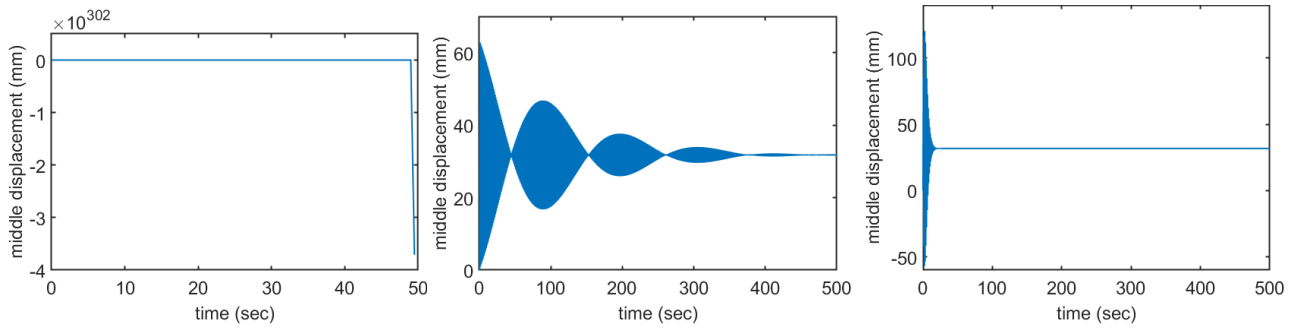


Figure 23: First 10 modes of the plate

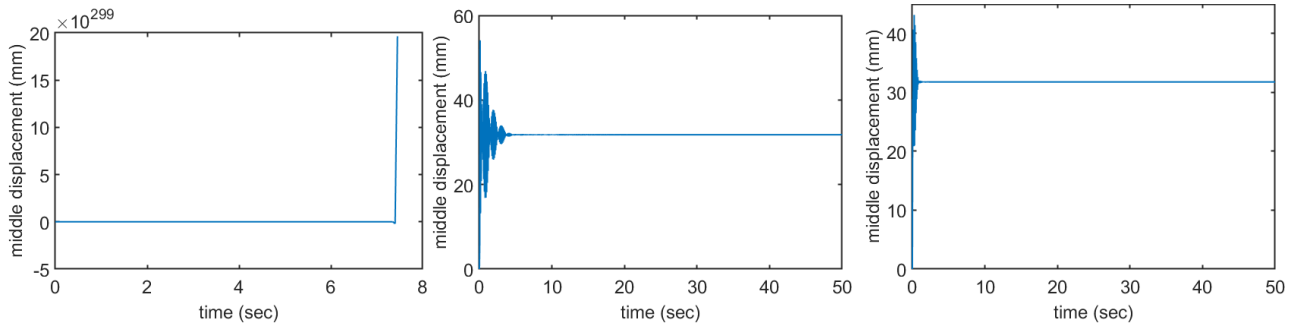
3.5 Deflection at the middle of the plate when a sudden vertical distributed loading of $0.3\,\mathrm{N/mm^2}$ is applied at $t = 0$

In this section three different time steps are compared using Central Difference, Trapezoidal and Damped Newark methods. 1000 steps is used for all analyses. Deflection at the middle is compared using 0.5, 0.05 and 1×10^{-4} second time steps.



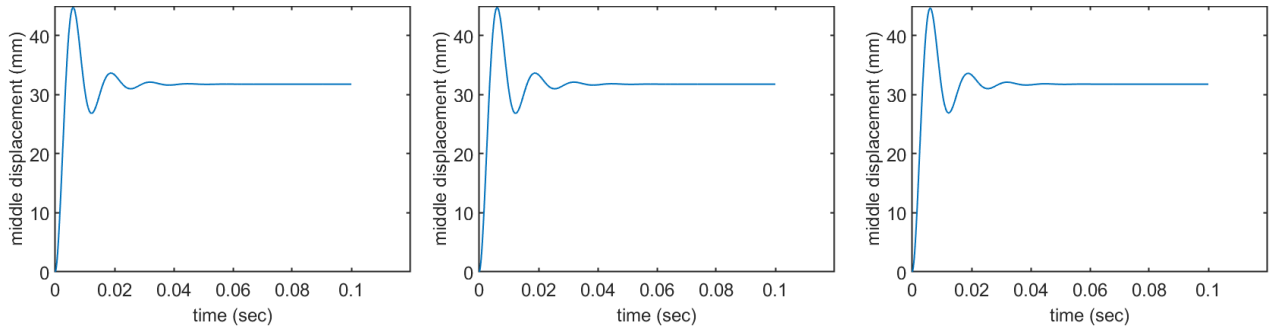
(a) Central Difference (b) Trapezoidal (c) Damped Newark

Figure 24: Deflection at the middle of the plate ($\Delta t = 0.5$).



(a) Central Difference (b) Trapezoidal (c) Damped Newark

Figure 25: Deflection at the middle of the plate ($\Delta t = 0.05$).



(a) Central Difference (b) Trapezoidal (c) Damped Newark

Figure 26: Deflection at the middle of the plate ($\Delta t = 1 \times 10^{-4}$).

It is clear that at $\Delta t = 1 \times 10^{-4}$, all methods gives the same result. If time step is smaller than this value, central difference method diverges, trapezoidal and damped Newark methods oscillates. One of the oscillating section of the trapezoidal in Figure 24 (b) is given in Figure 27. Time steps larger than $\Delta t = 1 \times 10^{-4}$ will give the corresponding section of time in Figure 26. For example, if $\Delta t = 1 \times 10^{-5}$, displacements until $0.01s$ will appear ($10^{-5} \cdot 10^{-2} = 0.01$). So, converging value will not be captured.

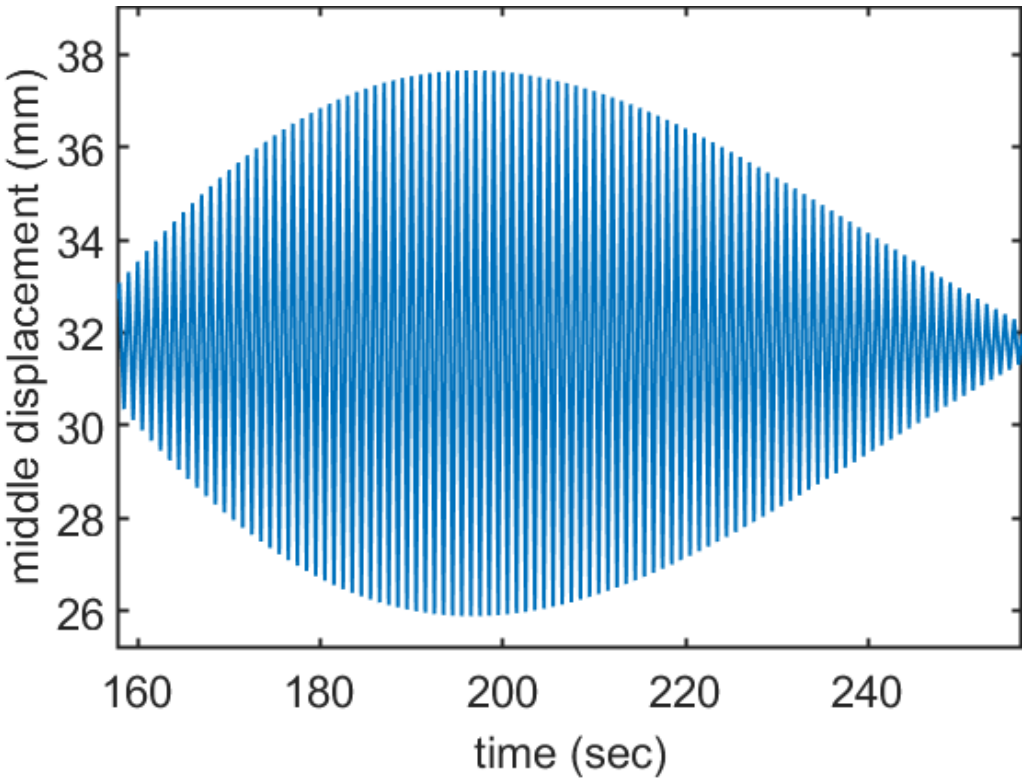


Figure 27: Closer look at one of the blue bars appeared in trapezoidal method in Figure 24 (b) ($\Delta t = 1 \times 10^{-4}$).

3.6 Reaction forces, moments and element stresses

$$\underset{\sim}{F}_{reaction} = \underset{\sim}{K}_{global}\underset{\sim}{d} - \underset{\sim}{F}_{global} \tag{110}$$

Force reaction output from Matlab is given in Table 2. If rounding errors are eliminated (3) it can be seen that reaction forces are only lie on constrained DOF's.

Element 1	F_1	-4.32E-12	1.23E-11	-7.73E-12	-1.09E-11	-3420
	M_1	3.64E-11	-5.09E-11	9.46E-11	1.09E-10	3.64E-12
	M_2	1.60E-10	3.08E-11	1.98E-11	-6.67E-11	-2.18E-11
Element 6	F_1	4.55E-12	3.37E-11	-3.64E-12	-2.73E-12	-4718
	M_1	7.21E-11	2.38E-11	6.56E-11	7.69E-11	1.19E-11
	M_2	2.18E-10	-1.05E-11	-4.50E-11	-4.11E-11	-1.02E-10
Element 11	F_1	4.09E-12	-2.55E-11	-1.64E-11	-1.00E-11	-3879
	M_1	9.76E-11	-2.88E-11	2.08E-10	5.29E-11	9.88E-12
	M_2	-4.37E-11	1.71E-11	-9.20E-11	-6.71E-11	8.73E-11
Element 16	F_1	-6.82E-12	2.46E-11	2.36E-11	5.46E-12	-3207
	M_1	1.15E-10	8.07E-11	3.36E-11	1.88E-11	9.44E-15
	M_2	0	-5.13E-12	1.99E-11	2.67E-11	-2.91E-11
Element 21	F_1	-14885	-24379	-22978	-18483	-50.97197177
	M_1	-176885	-288663	-267272	-195531	-3.64E-12
	M_2	398794	-15561	-22617	-167552	-4.37E-11

Table 2: Force reactions for nodal DOF's. Elements in the matrix given in same order with schematic 20.

3.7 Natural frequencies using lumped mass matrices.

Element 1	F_1	0	0	0	0	-3420
	M_1	0	0	0	0	0
	M_2	0	0	0	0	0
Element 6	F_1	0	0	0	0	-4718
	M_1	0	0	0	0	0
	M_2	0	0	0	0	0
Element 11	F_1	0	0	0	0	-3879
	M_1	0	0	0	0	0
	M_2	0	0	0	0	0
Element 16	F_1	0	0	0	0	-3207
	M_1	0	0	0	0	0
	M_2	0	0	0	0	0
Element 21	F_1	-14885	-24379	-22978	-18483	-51
	M_1	-176885	-288663	-267272	-195531	0
	M_2	398794	-15561	-22617	-167552	0

Table 3: Force reactions for nodal DOF's. Elements in the matrix given in same order with schematic 20.

Element 1	Element 2	Element 3	Element 4	Element 5
-61.1	-40.7	-44.4	-150.5	-1112.8
-6.9	-1.9	-7.2	25.1	-656.3
6.9	-2.5	11.0	-45.8	-124.8
Element 6	Element 7	Element 8	Element 9	Element 10
14.0	59.1	46.1	-86.8	-889.5
256.9	287.2	263.5	204.3	-311.8
7.0	-6.2	2.4	-5.0	-49.0
Element 11	Element 12	Element 13	Element 14	Element 15
106.0	156.9	131.8	-3.8	-780.8
551.4	552.9	505.0	385.2	-230.2
-18.2	-12.5	-3.6	-12.2	0.8
Element 16	Element 17	Element 18	Element 19	Element 20
205.8	217.6	182.1	74.8	-732.9
739.1	680.1	617.7	481.1	-174.7
-7.4	-4.8	-8.7	-23.2	91.5
Element 21	Element 22	Element 23	Element 24	Element 25
0.0	0.0	0.0	0.0	-735.9
0.0	0.0	0.0	0.0	23.7
4.4	4.7	10.4	100.1	214.7

Table 4: Element stresses [MPa].

1	0 + 1048.29 i
2	0 + 121.54 i
3	0 + 120.72 i
4	0 + 114.32 i
5	0 + 105.79 i
6	0 + 97.92 i
7	0 + 64.74 i
8	0 + 28.04 i
9	137.09 + 0 i
10	151.17 + 0 i

Table 5: Natural frequencies, found using lumped mass matrices, corresponding to constrained modes [Hz]. It distinct from natural frequencies obtained by the consistent mass matrix by complex eigenvalues. Because there is no damping the plate might be unstable in this modes. Real modes are where the vibration decays over time.

Matlab Scripts

Listing 1: Matlab code for question 1

```
%Final Q1

clear all; close all; clc; format short e;

% 1) INPUT
% 1.1) Nodes
coordinates_of_nodes=[  1 0 0;        % Location (mm)
                        2 0 400;
                        3 0 800;
                        4 0 1200;
                        5 0 1600;
                        6 0 2000;
                        7 0 2400;
                        8 0 2800;
                        9 0 3200;
                        10 0 3600;
                        11 0 4000];
connectivity=[  1 2;
                2 3;
                3 4;
                4 5;
                5 6;
                6 7;
                7 8;
                8 9;
                9 10;
                10 11];              % Connectivity of node

restrained_nodes=[11];               % Restrained degree of freedom

% a2) Uniform Load

uniform_load=0.01;                   % (N/mm)
nodes_load={[1 0 0] [2 0 0]};        % Nodal load (kN)

% 1.2) Elements

b=4;                                 % width of beam (mm)
h=[40 42 44 46 48 50 52 54 56 58];   % height of beam (mm)

angle= [0 0 0 0 0 0 0 0 0 0]';       % angle with respect to X axis(Degree)
m_e=2e5;                             % (N/mm^2)
k=10;                                % spring stiffness(N/mm)

% 2) TOPOLOGY MATRIX

nel=size(connectivity,1);
for i=1:nel
    top1(i,1)=connectivity(i,1)*2-1;
    top1(i,2)=connectivity(i,1)*2;
    top1(i,3)=connectivity(i,2)*2-1;
    top1(i,4)=connectivity(i,2)*2;
end

% 3) NUMBERS OF DOF RESTRAINED
```

```
56
57 for i=1:length(restrained_nodes)
58     restrained_DoF_number(2*i-1,1)=restrained_nodes(i)*2-1;
59     restrained_DoF_number(2*i,1)=restrained_nodes(i)*2;
60 end
61
62 % 4) BEAM ELEMENTS
63
64 for i=1:nel
65
66     % a) Lenght (m)
67     l_e(i,1)...
68         =sqrt((coordinates_of_nodes(connectivity(i,2),2)...
69         -coordinates_of_nodes(connectivity(i,1),2))^2....
70         +(coordinates_of_nodes(connectivity(i,2),3)...
71         -coordinates_of_nodes(connectivity(i,1),3))^2);
72
73     % b) Local Stiffness matrix
74
75     moment_of_inertia(i)=(b*h(i)^3)/12;
76     stiffness_of_elements_local{i,1}...
77         =(m_e*moment_of_inertia(i))/(l_e(i)^3)*...
78         [12 6*l_e(i) -12 6*l_e(i)
79          6*l_e(i) 4*(l_e(i)^2) -6*l_e(i) 2*(l_e(i)^2)
80          -12 -6*l_e(i) 12 -6*l_e(i)
81          6*l_e(i) 2*(l_e(i)^2) -6*l_e(i) 4*(l_e(i)^2)];
82
83     % c) Transformation matrix ( from global coordinate to element local
84     %coordinate)
85
86     transformation_matrix{i,1}=...
87         [ cos(angle(i)*((2*pi)/360)) sin(angle(i)*((2*pi)/360)) 0 0 ;
88         -sin(angle(i)*((2*pi)/360)) cos(angle(i)*((2*pi)/360)) 0 0 ;
89         0 0 cos(angle(i)*((2*pi)/360)) sin(angle(i)*((2*pi)/360));
90         0 0 -sin(angle(i)*((2*pi)/360)) cos(angle(i)*((2*pi)/360)) ];
91
92     % d) Global Stiffness Matrix
93
94     stiffness_of_elements_global{i,1}=...
95         transformation_matrix{i}'*stiffness_of_elements_local{i}...
96         *transformation_matrix{i};
97
98     % e) Uniform Load
99     uniform_loading_of_elements_local{i,1}...
100        =[(uniform_load*l_e(i))/2;...
101        (uniform_load*l_e(i)^2)/12;(uniform_load*l_e(i))/2;...
102        -(uniform_load*l_e(i)^2)/12];
103
104    fixed_f{i,1}=uniform_loading_of_elements_local{i,1};
105    fix_end_forces_and_moments_of_elements_global{i,1}...
106        =transformation_matrix{i}'*fixed_f{i,1};
107
108 end
109 % 5) SYSTEM
110
111 nodes_loads_of_system{1}=zeros(2*coordinates_of_nodes(end,1),1);
112 % Loading for nodes(kN)
113
```

```
for i=1:size(nodes_load,1)
    nodes_loads_of_system{1}([nodes_load{i}(1)*2-1 nodes_load{i}(1)*2])...
        =nodes_loads_of_system{1}...
        ([nodes_load{i}(1)*2-1 nodes_load{i}(1)*2])...
        +(nodes_load{i}(2:end))';
end

stiffness_of_system{1}...
    =zeros(2*coordinates_of_nodes(end,1)); % Stiffness matrix of system

for i=1:nel
    stiffness_of_system{1}(top1(i,:),top1(i,:))...
    =stiffness_of_system{1}(top1(i,:),top1...
    (i,:))+stiffness_of_elements_global{i};
end

for ii=[1:2:19]
    stiffness_of_system{1}(ii,ii)...
    =stiffness_of_system{1}(ii,ii)+k; % spring constant
end

fix_end_forces_and_moments_of_system{1}...
    =zeros(2*coordinates_of_nodes(end,1),1); %Uniform distributed load

for i=1:nel
    fix_end_forces_and_moments_of_system{1}(top1(i,:))...
        =fix_end_forces_and_moments_of_system{1}(top1(i,:))...
        +fix_end_forces_and_moments_of_elements_global{i};
end

% 6) LINEAR STATIC ANALYSIS FOR SYSTEM
% 6.1) Matching vector for obtaining global matrix

all_DoF_number=(1:2*coordinates_of_nodes(end,1))';

% All DoF
restrained_DoF_number_for_matching=restrained_DoF_number;

% Restrained DoF
unrestrained_DoF_number_for_matching=all_DoF_number;
unrestrained_DoF_number_for_matching...
    (restrained_DoF_number_for_matching)=[];

% Unrestrained DoF
matching_vector_first=[unrestrained_DoF_number_for_matching;
restrained_DoF_number_for_matching]; % Matching vector for reorganize

for i=1:length(matching_vector_first)
    matching_vector_end(i,1)=find(matching_vector_first==i);
    % Matching vector for orijinal position
end

% 6.2) Reorganize loading and stiffness matrix
loading_for_nodes_reorganize=nodes_loads_of_system{1}...
    (matching_vector_first);

stiffness_of_system_global_reorganize=stiffness_of_system{1}...
    (matching_vector_first,matching_vector_first);
```

```
172
173 fix_end_forces_and_moments_of_system_reorganize...
174     =fix_end_forces_and_moments_of_system{1}(matching_vector_first);
175
176 % 6.3) Linear solution with finite element method for system
177
178 % a) Input
179 % a1) Forces
180
181 F_A=loading_for_nodes_reorganize...
182     (1:length(unrestrained_DoF_number_for_matching));
183
184 % a2) Stiffness matrix partition
185
186 K_AA=stiffness_of_system_global_reorganize...
187     ([1:length(unrestrained_DoF_number_for_matching)],...
188     [1:length(unrestrained_DoF_number_for_matching)]);
189
190 K_AB=stiffness_of_system_global_reorganize...
191     ([1:length(unrestrained_DoF_number_for_matching)],...
192     [length(unrestrained_DoF_number_for_matching)+1:end]);
193
194 K_BA=stiffness_of_system_global_reorganize...
195     ([length(unrestrained_DoF_number_for_matching)+1:end],...
196     [1:length(unrestrained_DoF_number_for_matching)]);
197
198 K_BB=stiffness_of_system_global_reorganize...
199     ([length(unrestrained_DoF_number_for_matching)+1:end],...
200     [length(unrestrained_DoF_number_for_matching)+1:end]);
201
202 % a3) Displacements ( Support Settlements )
203
204 D_B=[0 0 ]';
205
206 % a4) Fix end forces and moments
207
208 P_A=fix_end_forces_and_moments_of_system_reorganize...
209     (1:length(unrestrained_DoF_number_for_matching));
210
211 P_B=fix_end_forces_and_moments_of_system_reorganize...
212     (length(unrestrained_DoF_number_for_matching)+1:end);
213
214 % b) Solution
215 D_A=K_AA\(F_A-K_AB*D_B-P_A); %Specify nodal displacement (m)
216
217 F_B_second_part_joint_reactions=K_BA*D_A+K_BB*D_B+P_B;
218
219 %Specify support reaction force (kN)
220
221 % 6.4) Orijinal position displacement and force matrix
222
223 displacements_of_system_reorganized=[D_A;D_B];
224
225 displacements_of_system_orijinal_position...
226     =displacements_of_system_reorganized(matching_vector_end)
227
228 nodes_loads_both_point_loads_and_joint_reactions{1}...
229     =nodes_loads_of_system{1};
```

```
230
231 nodes_loads_both_point_loads_and_joint_reactions{1}...
232     (restrained_DoF_number_for_matching)...
233     =nodes_loads_both_point_loads_and_joint_reactions{1}...
234     (restrained_DoF_number_for_matching)+F_B_second_part_joint_reactions;
235
236 u = displacements_of_system_orijinal_position(1:2:22,1); % vertical
    displacements
237 phi = displacements_of_system_orijinal_position(2:2:22,1);     % bending angle
238 lenght = (0:10)/10*2;
239
240 figure('Name','Displacement');
241 plot(lenght,u)
242 title('Displacement')
243 xlabel('Lenght [m]')
244 ylabel('Displacement [mm]')
245
246 figure('Name','Bending Angle');
247 plot(lenght,phi)
248 title('Bending Angle')
249 xlabel('Lenght [m]')
250 ylabel('Bending Angle [deg]')
```

Listing 2: Matlab code for question 2

```
%Final Q2

clear all; close all; clc; format short;

width=5; left=1.5; right=2.5; %dimensions (m)

seed_x=6;
seed_y=3;

num_node=18;                    %number of total nodes
num_elem=10;                    %number of total elements

element_type=4;                 %number of nodes on each element
ndof=1;                         %degrees-of-freedom per node

num_eq=num_node*ndof;           %number of equations

z=[1 7 13 19];
k=100;

%%---------------Coordinates---------------%%

x_dummy=linspace(0,width,seed_x);
y_right=linspace(0,right,seed_y);
y_left=linspace(0,left,seed_y);

x_coord=[];
y_dummy=[];
y_coord=[];

for i=1:seed_y
    x_coord=[x_coord x_dummy];
    difference=(y_right(i)-y_left(i));
    y0=linspace(0,difference,seed_x);
    y_dummy=[y_dummy y0];
    y_increased=y_left(i)+y_dummy(z(i):1:z(i+1)-1);
    y_coord=[y_coord y_increased];
end

Coords=[x_coord', y_coord'];

%%-------topology matrix-------%%

rowcount=0;

for elementcount=1:num_elem             %topology matrix

    Topology(1,elementcount)=elementcount+rowcount;
    Topology(2,elementcount)=elementcount+1+rowcount;
    Topology(3,elementcount)=elementcount+seed_x+rowcount+1;
    Topology(4,elementcount)=elementcount+seed_x+rowcount;

    if mod(elementcount,seed_x-1)==0    %end of element in a row
        rowcount=rowcount+1;
    end

end
```

```
58
59 topology=Topology';
60
61 %%------Gauss Quadrature and Shape function ------%%
62
63 num_gauss_point=2;
64
65 if num_gauss_point==1
66     gp=0;
67     w=2;
68     elseif num_gauss_point==2
69         %if four gauss points are used (two in each direction)
70     gp=[-0.57735027, 0.57735027]; %[eta psi]
71     w= [1,1];
72 end
73
74 eta=gp(1);
75 psi=gp(2);
76
77 N=(1/4)*[(1-psi)*(1-eta) (1+psi)*(1-eta) (1+psi)*(1+eta) (1-psi)*(1+eta)];
78 %shape functions
79
80 dN=(1/4)*[eta-1 1-eta 1+eta -eta-1; % Gradient
81           psi-1 -psi-1 1+psi 1-psi];
82
83 %%------Obtaining Element Stiffness Matrix------%%
84
85 ke=zeros(element_type,element_type);
86 kglobal=zeros(num_eq);
87
88 for e=1:1:num_elem
89
90     elem_top=[topology(e,:)];
91     J=dN*Coords(elem_top',:);                    % compute Jacobian matrix
92     detJ=det(J);                                 % Jacobian
93     B=J\dN;                                      % compute the B matrix
94     D=k*eye(2);
95     for i=1:num_gauss_point
96         for j=1:num_gauss_point
97             ke=ke+w(i)*w(j)*B'*D*B*detJ;         % element conductance matrix
98             kglobal(elem_top,elem_top)=kglobal(elem_top,elem_top)+ke;
99         end
100    end
101 end
102
103 % At node 1,2,3,4,5,6,7,12,13 and 18, the temperature is to be fixed
104 fix_temperature_nodes=[1 2 3 4 5 6 7 12 13 18];
105
106 % The temperature is to be fixed at a particular value ( C )
107 fix_temperature=[0 3 12 27 48 75 0 75 0 75];
108
109 % Nodal force(heat source) vector
110 nodal_force{1}=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]';
111
112 % 4) LINEAR 2-D HEAT CONDUCTION ANALYSIS
113 % 4.1) Matching vector for obtaining global matrix
114
115 % All DoF
```

```
all_DoF_number=(1:num_node)';

% Restrained DoF
restrained_DoF_number_for_matching=fix_temperature_nodes';

% Unrestrained DoF
unrestrained_DoF_number_for_matching=all_DoF_number;
unrestrained_DoF_number_for_matching(restrained_DoF_number_for_matching)...
    =[];

matching_vector_first=[unrestrained_DoF_number_for_matching;
restrained_DoF_number_for_matching];        % Matching vector for reorganize

for i=1:length(matching_vector_first)
    matching_vector_end(i,1)=find(matching_vector_first==i);
    % Matching vector for orijinal position
end

% 4.2) Reorganize nodal force and stiffness matrix
nodal_force_reorganize=nodal_force{1}(matching_vector_first);

stiffness_of_system_reorganize=...
    kglobal(matching_vector_first,matching_vector_first);

% 4.3) Linear solution with finite element method
% a) Input

% a1) Forces
F_A=nodal_force_reorganize(1:length(unrestrained_DoF_number_for_matching));

% a2) Stiffness matrix partition
K_AA=stiffness_of_system_reorganize...
    ([1:length(unrestrained_DoF_number_for_matching)],...
    [1:length(unrestrained_DoF_number_for_matching)]);

K_AB=stiffness_of_system_reorganize...
    ([1:length(unrestrained_DoF_number_for_matching)],...
    [length(unrestrained_DoF_number_for_matching)+1:end]);

K_BA=stiffness_of_system_reorganize...
    ([length(unrestrained_DoF_number_for_matching)+1:end],...
    [1:length(unrestrained_DoF_number_for_matching)]);

K_BB=stiffness_of_system_reorganize...
    ([length(unrestrained_DoF_number_for_matching)+1:end],...
    [length(unrestrained_DoF_number_for_matching)+1:end]);

% a3) Fix tempreture ( F )
D_B=fix_temperature';

% b) Solution
% Specify nodal tempreture ( C )
D_A=K_AA\(F_A-K_AB*D_B);
% Specify nodal in-bound heat fluxes(Watts)
F_B_heat_fluxes=K_BA*D_A+K_BB*D_B;

% 4.4) Orijinal position tempreture and heat fluxes

```

```
temperature_of_system_reorganized=[D_A;D_B];

temperature_of_system_orijinal_position...
    =temperature_of_system_reorganized(matching_vector_end)
heat_fluxes_of_system_orijinal_position_{1}...
    =nodal_force{1};

heat_fluxes_of_system_orijinal_position_{1}...
(restrained_DoF_number_for_matching)...
    =heat_fluxes_of_system_orijinal_position_{1}...
    (restrained_DoF_number_for_matching)+F_B_heat_fluxes;

heat_fluxes_of_system_orijinal_position...
    =heat_fluxes_of_system_orijinal_position_{1}
```

Listing 3: Matlab code for question 3

```
%Final Q3

clc
clear all
close all
format compact
format short e
tic

nel=16; %Total Number of Elements
nnode=25; %Total Number of Nodes
ndof=3; %Nodal Degree of Freedom

h=4; %Thickness
E=2.1e5; %Young Modulus
v=0.33; %Poisson Ratio

a=100; %Length of 1 element
b=50; %Width of 1 element
ro=7.85e-9; %Density

pz=0.3; %Distributed load

%Gauss Points (4 points)

gpoints = (1/35)*[ sqrt(525-70*sqrt(30))
                  -sqrt(525-70*sqrt(30))
                   sqrt(525+70*sqrt(30))
                  -sqrt(525+70*sqrt(30)) ];

gweights = (1/36)*[18+sqrt(30)
                   18+sqrt(30)
                   18-sqrt(30)
                   18-sqrt(30)];

%Element matrices

E1 = E/(1-v^2);

kk = zeros(12);
mm = zeros(12);
ff = zeros(12,1);

Kelstiff = zeros(12,12,nel);

N = zeros(1,12);
n = sym('n',[4 3]);
ns = sym('ns',[4 3]);
nt = sym('nt',[4 3]);
ns2 = sym('ns2',[4 3]);
nt2 = sym('nt2',[4 3]);
nsnt = sym('nsnt',[4 3]);

syms s t
for i=1:length (gweights)
    for j=1:length(gweights)
        for node=1:4
```

```
            if node==1
            sc=-1;
            tc=-1;
            elseif node==2
            sc=1;
            tc=-1;
            elseif node==3
            sc=1;
            tc=1;
            else
            sc=-1;
            tc=1;
            end
            n(node,:)=[(1/8)*(1+sc*s)*(1+tc*t)*(2+sc*s+tc*t-s^2-t^2),...
            (b/8)*(1+sc*s)*(tc+t)*(t^2-1), ...
            -(a/8)*(sc+s)*(s^2-1)*(1+tc*t)];
        end

        ns=diff(n,s);
        nt=diff(n,t);
        ns2=diff(n,s,2);
        nt2=diff(n,t,2);
        nsnt=diff(ns,t);

        N=[n(1,:) n(2,:) n(3,:) n(4,:)];
        Ns2=[ns(1,:) ns(2,:) ns(3,:) ns(4,:)];
        Nt2=[nt(1,:) nt(2,:) nt(3,:) nt(4,:)];
        Nsnt=[nsnt(1,:) nsnt(2,:) nsnt(3,:) nsnt(4,:)];

        B=[Ns2/a^2; Nt2/b^2; 2/(a*b)*Nsnt];

        D=[ E1 E1*v 0;...
            E1*v E1 0;...
            0 0 E/(2*(1+v))];

        %Element stiffness matrix
        kk=kk+a*b*h^3/12*B'*D*B*gweights(i)*gweights(j);
        kk=subs(kk,s,gpoints(i));
        kk=subs(kk,t,gpoints(j));

        %Element mass matrix
        mm=mm+ro*h*a*b*(N'*N)*gweights(i)*gweights(j);
        mm=subs(mm,s,gpoints(i));
        mm=subs(mm,t,gpoints(j));

        %Element consistent load vector
        ff=ff+a*b*N'*pz*gweights(i)*gweights(j);
        ff=subs(ff,s,gpoints(i));
        ff=subs(ff,t,gpoints(j));

    end
end

kk=double(kk);
mm=double(mm);
ff=double(ff);

%Topology Matrix
```

```
top=[16 17 18 19 20 21 4 5 6 1 2 3;
    19 20 21 22 23 24 7 8 9 4 5 6;
    22 23 24 25 26 27 10 11 12 7 8 9;
    25 26 27 28 29 30 13 14 15 10 11 12;
    31 32 33 34 35 36 19 20 21 16 17 18;
    34 35 36 37 38 39 22 23 24 19 20 21;
    37 38 39 40 41 42 25 26 27 22 23 24;
    40 41 42 43 44 45 28 29 30 25 26 27;
    46 47 48 49 50 51 34 35 36 31 32 33;
    49 50 51 52 53 54 37 38 39 34 35 36;
    52 53 54 55 56 57 40 41 42 37 38 39;
    55 56 57 58 59 60 43 44 45 40 41 42;
    61 62 63 64 65 66 49 50 51 46 47 48;
    64 65 66 67 68 69 52 53 54 49 50 51;
    67 68 69 70 71 72 55 56 57 52 53 54;
    70 71 72 73 74 75 58 59 60 55 56 57];

%Global stiffness matrix

Kglob=zeros(nnode*ndof);
for nel=1:16
    for i=1:12
        for j=1:12
            Kglob(top(nel,i),top(nel,j))= ...
            Kglob(top(nel,i),top(nel,j))+kk(i,j);
        end
    end
end

%Global mass matrix
Mglob=zeros(nnode*ndof);
for nel=1:16
    for i=1:12
        for j=1:12
            Mglob(top(nel,i),top(nel,j))= ...
            Mglob(top(nel,i),top(nel,j))+mm(i,j);
        end
    end
end

%Lumped mass matrix using row sum method
MglobLum=zeros(nnode*ndof);
for i=1:nnode*ndof
    MglobLum(i,i)=sum(Mglob(i,:));
end

%Global force vector
Fglob=zeros(nnode*ndof,1);
for nel=1:16
    for i=1:12
            Fglob(top(nel,i),1)= ...
            Fglob(top(nel,i),1)+ff(i,1);
    end
end

%Boundary conditions (Vertical displacements of all edges are zero)
% BCs=[1;4;7;10;13;16;28;31;43;46;58;61;64;67;70;73];

```

```
%BCs=[1;4;7;10;13;61;64;67;70;73];
%Boundary conditions (Bottom edge cantilever, right edge simply supported)
BCs=[13;28;43;58;61;62;63;64;65;66;67;68;69;70;71;72;73];
activeDof=setdiff(1:nnode*ndof',BCs);
Kglobactive=Kglob(activeDof,activeDof);
Mglobactive=Mglob(activeDof,activeDof);
MglobLumactive=MglobLum(activeDof,activeDof);
Fglobactive=Fglob(activeDof,1);

%Solve
d=Kglobactive\Fglobactive;
do(activeDof,1)=d;

%Obtaining only vertical displacement DOFs

count=1;
for i=1:25
    dw(i)=do(count); %Nodal displacements
    count=count+3;
end

maxdw=max(dw) %Maximum deflection

%Reaction forces and moments at nodes
Freaction=Kglob*do-Fglob;

%Strain at nodes (eps x, eps y, gama xy)
B1=subs(B,s,-1);
B1=subs(B1,t,1);
B1=double(B1);
strain=zeros(3,1,25);
e=0;

for i=[1:4 6:9 11:14 16:19] %Nodes [1:4 6:9 11:14 16:19]
    e=e+1;
    strain(:,:,i)=B1*do(top(e,:));
end

B2=subs(B,s,1);
B2=subs(B2,t,1);
B2=double(B2);

e=0;
for i=[5 10 15 20] %Nodes [5 10 15 20]
    e=e+4;
    strain(:,:,i)=B2*do(top(e,:));
end

B3=subs(B,s,-1);
B3=subs(B3,t,-1);
B3=double(B3);
e=12;

for i=[21:24] %Nodes [21 24]
    e=e+1;
    strain(:,:,i)=B3*do(top(e,:));
end

```

```
B4=subs(B,s,1);
B4=subs(B4,t,-1);
B4=double(B4);
strain(:,:,25)=B4*do(top(16,:)); %Node 25

%Stress at nodes (sigma x, sigma y,tau xy)
stress=zeros(3,1,25);
for i=1:25
    stress(:,:,i)=D*strain(:,:,i)
end

%Normal modes and corresponding natural frequencies
[vecfreq, freq]=eig(Kglob,Mglob);
freq=diag(freq);
[freq,I1]=sort(freq, 'ascend');      %Sorted eigenvalues
vecfreq=vecfreq(I1,:);               %Sorted eigenvalues
freq=sqrt(freq);                     %UNITS : rad per sec
freqHz=freq/(2*pi);                  %UNITS : Hertz

%Constrained modes and corresponding natural frequencies
[vecfreqc,freqc]=eig(Kglobactive,Mglobactive);
freqc=diag(freqc);
[freqc,I2]=sort(freqc, 'ascend');    %Sorted eigenvalues
vecfreqc=vecfreqc(I2,:);             %Sorted eigenvalues
freqc=sqrt(freqc);                   %UNITS : rad per sec
freqcHz=freqc/(2*pi);                %UNITS : Hertz

%Constrained modes and corresponding natural frequencies
%with lumped mass matrix
[vecfreqcLum,freqcLum]=eig(Kglobactive,MglobLumactive);
freqcLum=diag(freqcLum);
[freqcLum,I3]=sort(freqcLum, 'ascend'); %Sorted eigenvalues
vecfreqcLum=vecfreqcLum(I3,:);          %Sorted eigenvalues
freqcLum=sqrt(freqcLum);                %UNITS : rad per sec
freqcLumHz=freqcLum/(2*pi);             % UNITS : Hertz

%Rayleigh damping model
cglob1=0.001*Kglobactive+0.02*Mglobactive;
em=Mglobactive;
ka=Kglobactive;
fe=Fglobactive;
ce=cglob1;
[mg1,junk1]=size(Mglobactive);

%INTEGRATION USING NEWMARK METHOD
%Central difference formula
gama=0.5;
beta=0;
ksi=(0.001/max(freqc)+0.02/max(freqc))/2;
om=(ksi*(gama-0.5)+sqrt(gama/2-beta+ksi^2*(gama-0.5)^2))/(gama/2-beta);
deltatcrit=om/max(freqc);
deltat=1e-4;
delta=deltat;

%trapezoidal rule
gama=0.5;
beta=0.25;

```

```
290 %% Damped Newmark Method
291 % gama=0.6;
292 % beta=0.3025;
293
294 dold=zeros(mg1,1);
295 vold=zeros(mg1,1);
296 say=0;
297 say=say+1;
298
299 rubbish=inv(em+gama*delta*ce+beta*delta*delta*ka);
300 aold=inv(em)*(fe-ce*vold-ka*dold);
301 dtildanew=dold+delta*vold+0.5*delta*delta*(1-2*beta)*aold;
302 vtildanew=vold+(1-gama)*delta*aold;
303 anew=rubbish*(fe-ce*vtildanew-ka*dtildanew);
304 dnew=dtildanew+beta*delta*delta*anew;
305 vnew=vtildanew+gama*delta*anew;
306 aold=anew;
307 vold=vnew;
308 dold=dnew;
309 dtildaold=dtildanew;
310 vtildaold=vtildanew;
311 time1(say)=0;
312 middledisplacement(say)=0;
313
314 for i=1:1000
315     say=say+1;
316     time1(say)=time1(say-1)+delta;
317     dtildanew=dold+delta*vold+0.5*delta*delta*(1-2*beta)*aold;
318     vtildanew=vold+(1-gama)*delta*aold;
319     anew=rubbish*(fe-ce*vtildanew-ka*dtildanew);
320     dnew=dtildanew+beta*delta*delta*anew;
321     vnew=vtildanew+gama*delta*anew;
322     aold=anew;
323     vold=vnew;
324     dold=dnew;
325     dtildaold=dtildanew;
326     vtildaold=vtildanew;
327     df(activeDof)=dold;
328     middledisplacement(say)=df(37);
329 end
330
331 toc
332
333 figure, plot(time1, middledisplacement)
334 xlabel('time (sec)')
335 ylabel('middle displacement (mm)')
```