Delft University of Technology

STABILITY AND ANALYSIS OF STRUCTURES - 1

AE4ASM106

Assignment: 1

Author:

Venkatesh Puchakayala Appaiah Subramanyam
Student ID: 5963540

January 12, 2023

Contents

Question 1: ... 3
Sub question 1: ... 3
Sub question 2: ... 3
Sub question 3: ... 4
Question 2: ... 5
Sub question 1: ... 6
Sub question 2: ... 7
Sub question 3: ... 8
Sub question 4: ... 9
Sub question 5: ... 10
Sub question 6: ... 11
Question 3: ... 12
Sub question a: ... 12
Sub question 1: ... 12
Sub question 2: ... 12
Sub question 3: ... 13
Sub question 4: ... 13
Sub question 5: ... 13
Sub question 6: ... 14
Sub question b: ... 15
Question 4: ... 16
Question 5: ... 19
Sub question 1: ... 19
Sub question 2: ... 21
Sub question 3: ... 23
Sub question 4: ... 25
Question 6: ... 26
Sub question 1: ... 26
Sub question 2: ... 27
Question 7: ... 30
Sub question 1: ... 30
Sub question 2: ... 31
Sub question 3: ... 32
Sub question 4: ... 33
Sub question 5: ... 34
Sub question 6: ... 35
Question 8: ... 40

Sub question 1: 40
Sub question 2: 43
Annexure - A: 45
Question 1: Python Code 45
Question 2: Python code 45
Question 3: Python code 49
Question 6: Python code 52
Question 8: Python Code 53

Question 1:

Stress Tensor: Suppose that at a given point in a structure the stress tensor has been computed (in a global cartesian basis) as (components in MPa)

$$[\sigma]_e = \begin{bmatrix} 150 & 110 & 70 \\ 110 & 160 & 0 \\ 70 & 0 & -140 \end{bmatrix}_e$$

Sub question 1:

i. Compute the principal stresses and order them in descending order.

The principal stresses are the eigenvalues of the stress tensor.

The stress tensor given is, $[\sigma]_e = \begin{bmatrix} 150 & 110 & 70 \\ 110 & 160 & 0 \\ 70 & 0 & -140 \end{bmatrix}_e$

To find the eigenvalues, python has been used and the code is given in Annexure – A.

The eigenvalues i.e., the principal stresses are $\sigma_1 = 270.9657 \text{ MPa}$, $\sigma_2 = 57.1751 \text{ MPa}$ and $\sigma_3 = -158.1409 \text{ MPa}$.

Sub question 2:

ii. Compute the (normalized) principal stress directions.

The principal stress directions are given by the eigenvectors of the stress tensor. Using Python, the eigenvectors have been calculated and they are as follows. The code used is given in Annexure - A.

$$\boldsymbol{n_1} = \begin{bmatrix} -0.70505075 \\ -0.69891447 \\ -0.12009164 \end{bmatrix}$$

$$\boldsymbol{n_2} = \begin{bmatrix} -0.66365531 \\ 0.70996543 \\ -0.23560712 \end{bmatrix}$$

$$\boldsymbol{n_3} = \begin{bmatrix} -0.24993014 \\ 0.08641553 \\ 0.96439996 \end{bmatrix}$$

However, the eigen vectors need to form a right-handed system, i.e., $\boldsymbol{n_1 X\, n_2} = \boldsymbol{n_3}, \boldsymbol{n_2 X\, n_3} = \boldsymbol{n_1}, \boldsymbol{n_3 X\, n_1} = \boldsymbol{n_2}$. The above system doesn't satisfy this condition. Hence, the eigen vectors need to be changed. We can use the fact that multiplication of a factor to an eigen value would still result in an eigen vector for that eigen value.

So changing the sign for vector 1 will give the eigen vectors as:

$$\boldsymbol{n_1} = \begin{bmatrix} 0.70505075 \\ 0.69891447 \\ 0.12009164 \end{bmatrix}$$

$$\boldsymbol{n_2} = \begin{bmatrix} -0.66365531 \\ 0.70996543 \\ -0.23560712 \end{bmatrix}$$

$$\boldsymbol{n_3} = \begin{bmatrix} -0.24993014 \\ 0.08641553 \\ 0.96439996 \end{bmatrix}$$

This would then result into a right hand system.

Sub question 3:

iii. The von Mises criterion indicates that the material remains elastic if the von mises stress is below the yield stress whereas the Tresca criterion indicates that the material remains elastic if the maximum shearing remains below half of the yield stress, i.e.,

$$\sigma_m = \sqrt{\frac{1}{2}((\sigma^{(1)} - \sigma^{(2)})^2 + (\sigma^{(1)} - \sigma^{(3)})^2 + (\sigma^{(2)} - \sigma^{(3)})^2)} \leq \sigma_y$$

$$\tau_{max} = \frac{1}{2}\max\{|\sigma^{(1)} - \sigma^{(2)}|, |\sigma^{(1)} - \sigma^{(3)}|, |\sigma^{(2)} - \sigma^{(3)}|\} \leq \frac{1}{2}\sigma_y$$

Suppose the material has a yield stress of $\sigma_y = 380\ MPa$. Is the material in the elastic range or not?

According to the von Mises criterion,

$$\sigma_m = \sqrt{\frac{1}{2}((\sigma^{(1)} - \sigma^{(2)})^2 + (\sigma^{(1)} - \sigma^{(3)})^2 + (\sigma^{(2)} - \sigma^{(3)})^2)}$$

$$\sigma_m = \sqrt{\frac{1}{2}((270.9657 - 57.1751)^2 + (270.9657 - (-158.1409))^2 + (57.1751 - (-158.1409))^2)}$$

$$\sigma_m = \sqrt{\frac{1}{2}((213.7906)^2 + (429.1066)^2 + (215.316)^2)}$$

$$\sigma_m = \sqrt{\frac{1}{2}((213.7906)^2 + (429.1066)^2 + (215.316)^2)}$$

$$\sigma_m = \sqrt{\frac{1}{2}(276199.8746)}$$

$$\sigma_m = 371.6179\ MPa < 380\ MPa(Yield\ stress)$$

Hence, the material doesn't yield as per von Mises stress criterion.

According to Tresca theory,

$$\tau_{max} = \frac{1}{2}\max\{\left|\sigma^{(1)} - \sigma^{(2)}\right|, \left|\sigma^{(1)} - \sigma^{(3)}\right|, \left|\sigma^{(2)} - \sigma^{(3)}\right|\}$$

$$\tau_{max} = \frac{1}{2}\max\{|270.9657 - 57.1751|, |270.9657 - (-158.1409)|, |57.1751 - (-158.1409)|\}$$

$$\tau_{max} = \frac{1}{2}\max\{|213.7906|, |429.1066|, |215.316|\}$$

$$\tau_{max} = \frac{1}{2}429.1066$$

$$\tau_{max} = 214.5533\ MPa > 190\ MPa\ i.e., \frac{1}{2}(\sigma_y)$$

Hence, the material will yield as per Tresca criterion.

Question 2:

Simple deformations. Compute the infinitesimal strain tensor ϵ_{ij} and the relative change in volume $\epsilon_{11} + \epsilon_{22} + \epsilon_{33}$ of an initially cubic volume element aligned with Cartesian axes x_1, x_2, x_3 subject to the following displacement fields. In each case, sketch the shape of the deformed cubic element.

Before solving the problem, let us define the strain tensor ϵ and its element ϵ_{ij}.

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Substituting in the strain tensor, we get

$$\epsilon = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}$$

This matrix is symmetric in nature.

Sub question 1:

i. Simple extension:

$u_1 = ax_1$

$u_2 = 0$

$u_3 = 0$

where a is a positive constant.

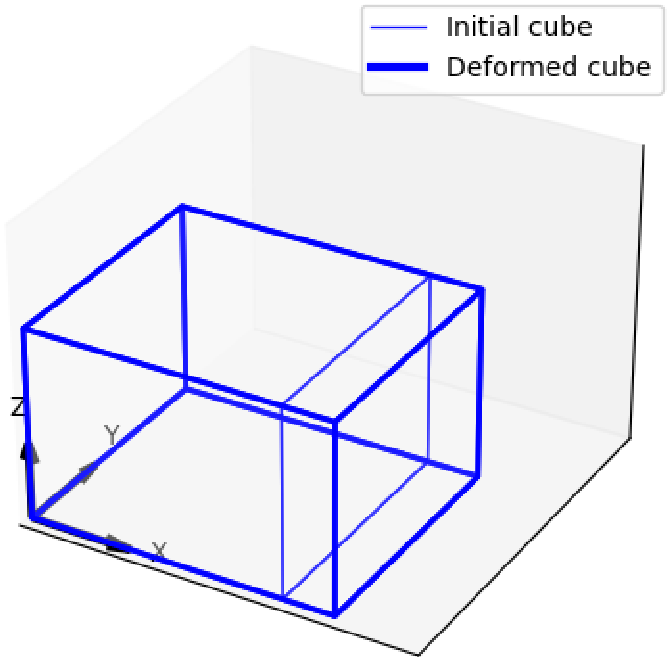
$$\epsilon = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \frac{\partial (ax_1)}{\partial x_1} & \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_2} + \frac{\partial (0)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_3} + \frac{\partial (0)}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_2} + \frac{\partial (0)}{\partial x_1}\right) & \frac{\partial (0)}{\partial x_2} & \frac{1}{2}\left(\frac{\partial (0)}{\partial x_3} + \frac{\partial (0)}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_3} + \frac{\partial (0)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial (0)}{\partial x_3} + \frac{\partial (0)}{\partial x_2}\right) & \frac{\partial (0)}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Relative change in volume is given by, $\epsilon_{11} + \epsilon_{22} + \epsilon_{33} = a$

Deformed cube shape: (Python code used is given in the appendix. The factor a is assumed to be 0.2 of the unit length of the cube.)



Sub question 2:

ii. Extension with lateral contraction:

$u_1 = ax_1$
$u_2 = -bx_2$
$u_3 = -bx_3$
where a>b>0 are constants.

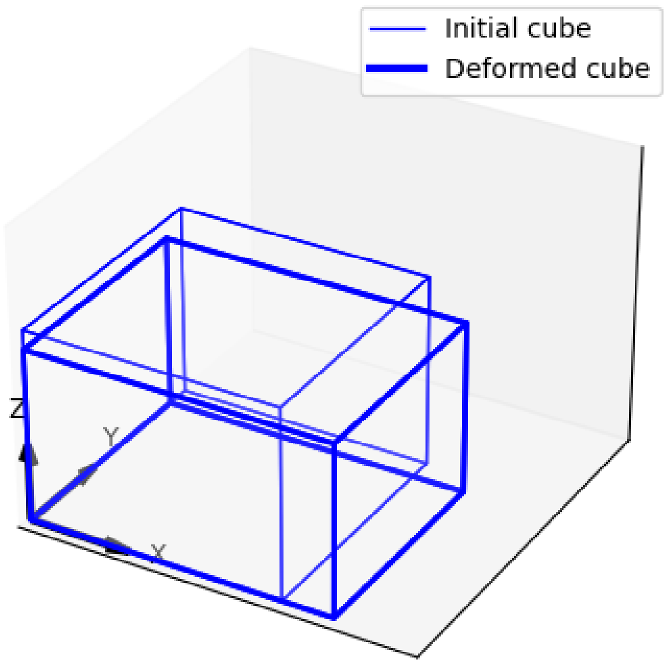
$$\epsilon = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \frac{\partial (ax_1)}{\partial x_1} & \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_2} + \frac{\partial (-bx_2)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_3} + \frac{\partial (-bx_3)}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_2} + \frac{\partial (-bx_2)}{\partial x_1}\right) & \frac{\partial (-bx_2)}{\partial x_2} & \frac{1}{2}\left(\frac{\partial (-bx_2)}{\partial x_3} + \frac{\partial (-bx_3)}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_3} + \frac{\partial (-bx_3)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial (-bx_2)}{\partial x_3} + \frac{\partial (-bx_3)}{\partial x_2}\right) & \frac{\partial (-bx_3)}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} a & 0 & 0 \\ 0 & -b & 0 \\ 0 & 0 & -b \end{bmatrix}$$

Relative change in volume is given by, $\epsilon_{11} + \epsilon_{22} + \epsilon_{33} = a - 2b$

Deformed cube shape: (Python code used is given in the appendix. The factor a is assumed to be 0.2 of the unit length of the cube and b is assumed to be 0.1 of the unit length of the cube.)



Sub question 3:

iii. Uniform volumetric contraction:

$u_1 = -ax_1$
$u_2 = -ax_2$
$u_3 = -ax_3$
where a is a positive constant.

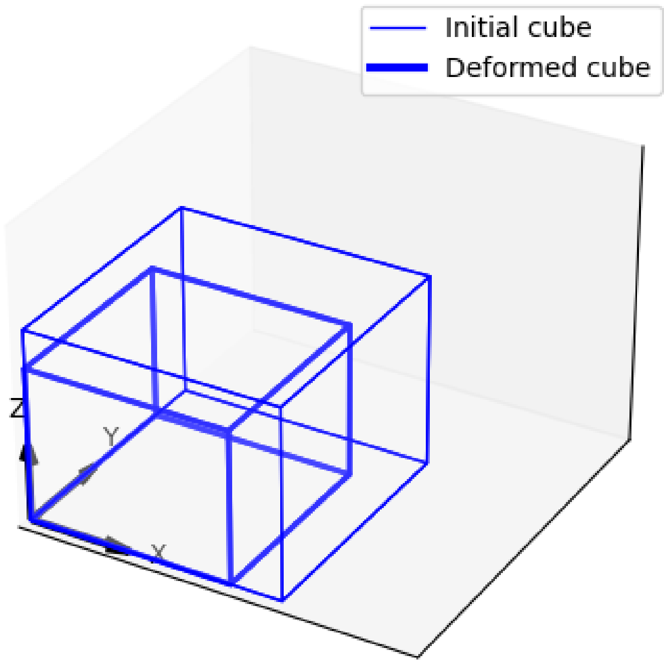
$$\epsilon = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \frac{\partial(-ax_1)}{\partial x_1} & \frac{1}{2}\left(\frac{\partial(-ax_1)}{\partial x_2} + \frac{\partial(-ax_2)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial(-ax_1)}{\partial x_3} + \frac{\partial(-ax_3)}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial(-ax_1)}{\partial x_2} + \frac{\partial(-ax_2)}{\partial x_1}\right) & \frac{\partial(-ax_2)}{\partial x_2} & \frac{1}{2}\left(\frac{\partial(-ax_2)}{\partial x_3} + \frac{\partial(-ax_3)}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial(-ax_1)}{\partial x_3} + \frac{\partial(-ax_3)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial(-ax_2)}{\partial x_3} + \frac{\partial(-ax_3)}{\partial x_2}\right) & \frac{\partial(-ax_3)}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} -a & 0 & 0 \\ 0 & -a & 0 \\ 0 & 0 & -a \end{bmatrix}$$

Relative change in volume is given by, $\epsilon_{11} + \epsilon_{22} + \epsilon_{33} = -3a$

Deformed cube shape: (Python code used is given in the appendix. The factor a is assumed to be 0.2 of the unit length of the cube.)



Sub question 4:

iv. Simple shear in the 1,2-plane:

$u_1 = 2ax_2$
$u_2 = 0$
$u_3 = 0$

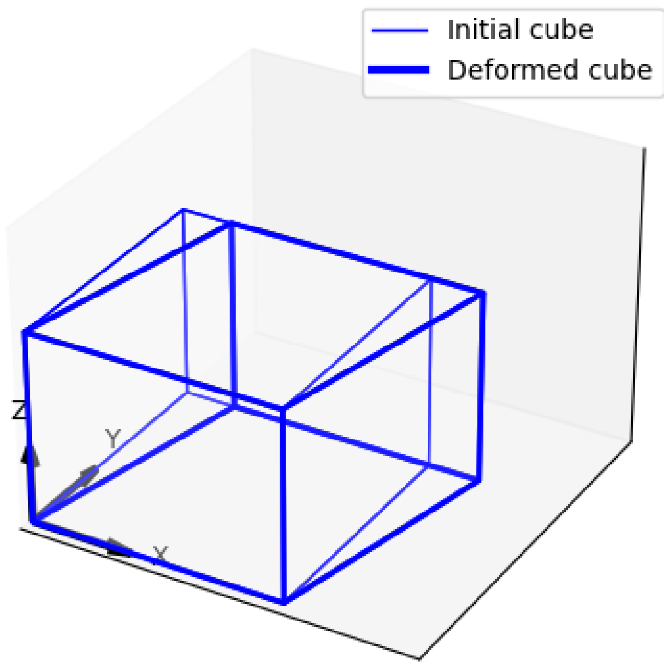
$$\epsilon = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \frac{\partial(2ax_2)}{\partial x_1} & \frac{1}{2}\left(\frac{\partial(2ax_2)}{\partial x_2} + \frac{\partial(0)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial(2ax_2)}{\partial x_3} + \frac{\partial(0)}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial(2ax_2)}{\partial x_2} + \frac{\partial(0)}{\partial x_1}\right) & \frac{\partial(0)}{\partial x_2} & \frac{1}{2}\left(\frac{\partial(0)}{\partial x_3} + \frac{\partial(0)}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial(2ax_2)}{\partial x_3} + \frac{\partial(0)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial(0)}{\partial x_3} + \frac{\partial(0)}{\partial x_2}\right) & \frac{\partial(0)}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} 0 & a & 0 \\ a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Relative change in volume is given by, $\epsilon_{11} + \epsilon_{22} + \epsilon_{33} = 0$

Deformed cube shape: (Python code used is given in the appendix. The factor a is assumed to be 0.2 of the unit length of the cube.)



Sub question 5:

v. Pure shear in the 1,2-plane:

$u_1 = ax_2$
$u_2 = ax_1$
$u_3 = 0$

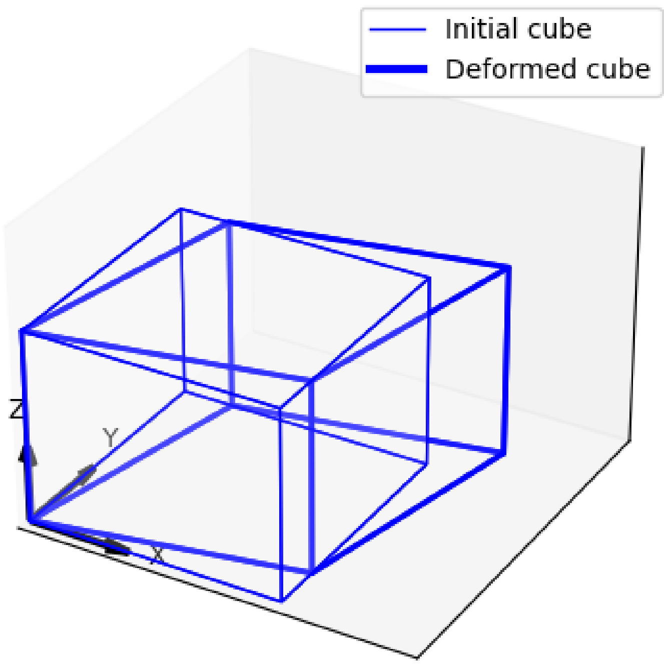
$$\epsilon = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \frac{\partial (ax_2)}{\partial x_1} & \frac{1}{2}\left(\frac{\partial (ax_2)}{\partial x_2} + \frac{\partial (ax_1)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial (ax_2)}{\partial x_3} + \frac{\partial (0)}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial (ax_2)}{\partial x_2} + \frac{\partial (ax_1)}{\partial x_1}\right) & \frac{\partial (ax_1)}{\partial x_2} & \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_3} + \frac{\partial (0)}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial (ax_2)}{\partial x_3} + \frac{\partial (0)}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial (ax_1)}{\partial x_3} + \frac{\partial (0)}{\partial x_2}\right) & \frac{\partial (0)}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} 0 & a & 0 \\ a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Relative change in volume is given by, $\epsilon_{11} + \epsilon_{22} + \epsilon_{33} = 0$

Deformed cube shape: (Python code used is given in the appendix. The factor a is assumed to be 0.2 of the unit length of the cube.)



Sub question 6:

vi. Pure shear:

$u_1 = a(x_2 + x_3)$
$u_2 = a(x_1 + x_3)$
$u_3 = a(x_1 + x_2)$

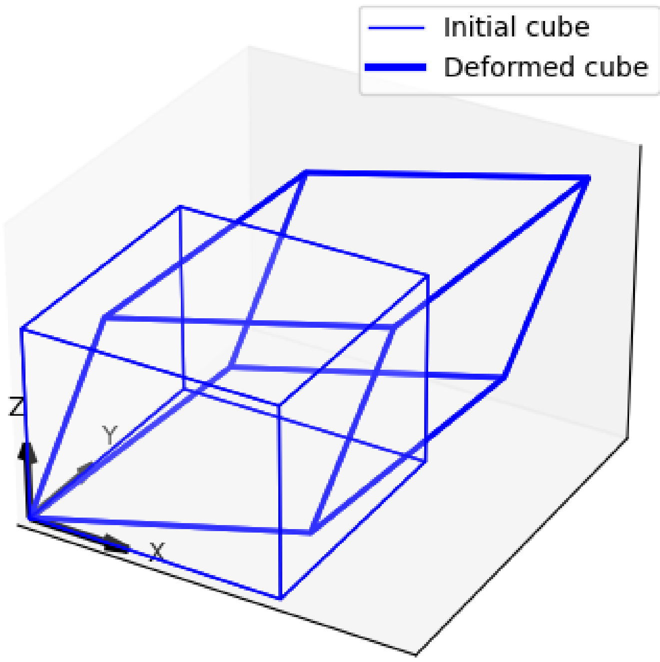
$$\epsilon = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \frac{\partial(a(x_2+x_3))}{\partial x_1} & \frac{1}{2}\left(\frac{\partial(a(x_2+x_3))}{\partial x_2} + \frac{\partial(a(x_1+x_3))}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial(a(x_2+x_3))}{\partial x_3} + \frac{\partial(a(x_1+x_2))}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial(a(x_2+x_3))}{\partial x_2} + \frac{\partial(a(x_1+x_3))}{\partial x_1}\right) & \frac{\partial(a(x_1+x_3))}{\partial x_2} & \frac{1}{2}\left(\frac{\partial(a(x_1+x_3))}{\partial x_3} + \frac{\partial(a(x_1+x_2))}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial(a(x_2+x_3))}{\partial x_3} + \frac{\partial(a(x_1+x_2))}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial(a(x_1+x_3))}{\partial x_3} + \frac{\partial(a(x_1+x_2))}{\partial x_2}\right) & \frac{\partial(a(x_1+x_2))}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} 0 & a & a \\ a & 0 & a \\ a & a & 0 \end{bmatrix}$$

Relative change in volume is given by, $\epsilon_{11} + \epsilon_{22} + \epsilon_{33} = 0$

Deformed cube shape: (Python code used is given in the appendix. The factor a is assumed to be 0.2 of the unit length of the cube.)



Question 3:

Change in length of a fiber. The displacement field of a solid is given by

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{3} u_i(\boldsymbol{x})\boldsymbol{e}_i = ({x_1}^2 + 20)10^{-4}e_1 + 2x_1\, x_2 10^{-3}e_2 + ({x_3}^2 - x_1x_2)10^{-4}e_3.$$

Sub question a:

a) Consider two points, P and Q, that have coordinates (2,5,7) and (3,8,9) in the undeformed body (for accurate results, keep at least 6 significant digits).

The calculations have been coded and attached in the annexure.

Sub question 1:

i. Using the displacement field given above, determine the position vector (coordinates) of points P and Q in the deformed body.

Point P changes as P +u and point Q changes as Q+u+δu.

The displacement gradient function is given by:

$$\delta u = \begin{bmatrix} \delta u_1 \\ \delta u_2 \\ \delta u_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \end{bmatrix}$$

$$\delta x_i = Qi - Pi$$

$$\begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3} \end{bmatrix} \rightarrow all\ of\ these\ use\ point\ P\ alone.$$

The new coordinates of P as P' are found to be:

$$P' = (2.0024e_1 + 5.02e_2 + 7.0039e_3)$$

In coordinates, P' = (2.0024, 5.02, 7.0039)

Using the displacement gradient function, the new coordinates of Q as Q' are found to be:

$$Q' = (3.0028e_1 + 8.042e_2 + 9.0056e_3)$$

In coordinates, Q' = (3.0028, 8.042, 9.0056)

Sub question 2:

ii. Using the coordinates of points P and Q in the undeformed body, compute the distance between P and Q before deformation.

We know the current coordinates of P and Q are (2,5,7) and (3,8,9) respectively. The distance between two points is given by:

$$Distance = \sqrt{(X_{Q1} - X_{P1})^2 + (X_{Q2} - X_{P2})^2 + (X_{Q3} - X_{P3})^2}$$

$$Distance = \sqrt{(3-2)^2 + (8-5)^2 + (9-7)^2}$$
$$Distance = \sqrt{14} = 3.741657 \; units$$

Sub question 3:

iii. Using the coordinates of points P and Q in the deformed body, compute the distance between P and Q after deformation.

We know the current coordinates of P' and Q' are (2.0004, 5.022, 7.0017) and (3.0006, 8.034, 9.0019) respectively. The distance between two points is given by:

$$Distance = \sqrt{(X_{Q'1} - X_{P'1})^2 + (X_{Q'2} - X_{P'2})^2 + (X_{Q'3} - X_{P'3})^2}$$
$$Distance = 3.760330 \; units$$

Sub question 4:

iv. From the results of (ii) and (iii), compute the average relative elongation of a fiber between points P and Q.

We can find the relative elongation of a fiber as

$$\epsilon = \frac{Final\ Length - Initial\ length}{Initial\ length}$$
$$\epsilon = \frac{P'Q' - PQ}{PQ}$$
$$\epsilon = \frac{3.760330 - 3.741657}{3.741657}$$
$$\epsilon = 0.004991$$

Sub question 5:

v. Compute the components of the infinitesimal strain tensor $\epsilon_{ij}(\boldsymbol{x})$. Evaluate the strain tensor at point P.

We know the strain tensor is given by,

$$\epsilon = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \frac{\partial(x_1^2+20)10^{-4}}{\partial x_1} & \frac{1}{2}\left(\frac{\partial(x_1^2+20)10^{-4}}{\partial x_2}+\frac{\partial(2x_1x_210^{-3})}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial(x_1^2+20)10^{-4}}{\partial x_3}+\frac{\partial(x_3^2-x_1x_2)10^{-4}}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial(x_1^2+20)10^{-4}}{\partial x_2}+\frac{\partial(2x_1x_210^{-3})}{\partial x_1}\right) & \frac{\partial(2x_1x_210^{-3})}{\partial x_2} & \frac{1}{2}\left(\frac{\partial(2x_1x_210^{-3})}{\partial x_3}+\frac{\partial(x_3^2-x_1x_2)10^{-4}}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial(x_1^2+20)10^{-4}}{\partial x_3}+\frac{\partial(x_3^2-x_1x_2)10^{-4}}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial(2x_1x_210^{-3})}{\partial x_3}+\frac{\partial(x_3^2-x_1x_2)10^{-4}}{\partial x_2}\right) & \frac{\partial(x_3^2-x_1x_2)10^{-4}}{\partial x_3} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} 2x_1 10^{-4} & x_2 10^{-3} & -0.5\, x_2 10^{-4} \\ x_2 10^{-3} & 2x_1 10^{-3} & -0.5\, x_1 10^{-4} \\ -0.5\, x_2 10^{-4} & -0.5\, x_1 10^{-4} & 2x_3 10^{-4} \end{bmatrix}$$

Substituting for point P with coordinates (2,5,7) we get,

$$\epsilon = \begin{bmatrix} 4 * 10^{-4} & 5 * 10^{-3} & -2.5 * 10^{-4} \\ 5 * 10^{-3} & 4 * \ 10^{-3} & -1 * \ 10^{-4} \\ -2.5 * 10^{-4} & -1 * \ 10^{-4} & 14 * 10^{-4} \end{bmatrix}$$

Sub question 6:

vi. Consider an infinitesimal fiber at point P aligned with the vector PQ. Compute the cartesian components $n_i, i = 1,2,3$ of the unit vector n in the direction of the line PQ. Use the formula developed in class to compute the relative elongation of the infinitesimal fiber at point P. Does your result coincide with the calculation done in (iv)? Why/why not?

To compute unit vector n in directions of line PQ, we need to do:

$$\boldsymbol{n} = \frac{\boldsymbol{PQ}}{||\boldsymbol{PQ}||} = \frac{(3-2)e_1 + (8-5)e_2 + (9-7)e_3}{\sqrt{(3-2)^2 + (8-5)^2 + (9-7)^2}}$$

$$\boldsymbol{n} = \frac{e_1 + 3e_2 + 2e_3}{\sqrt{14}}$$

To compute relative elongation using the in-class method in the direction of n we use the following formula:

$$\epsilon(\boldsymbol{n}) = \sum_{i,j=1}^{3} \epsilon_{ij} n_i n_j;\ n_i n_j = \frac{\delta x_i}{||\delta x||}\frac{\delta x_j}{||\delta x||}$$

$$\epsilon(\boldsymbol{n}) = \frac{\epsilon_{11}\delta x_1\delta x_1+\epsilon_{12}\delta x_1\delta x_2+\epsilon_{13}\delta x_1\delta x_3+\epsilon_{21}\delta x_2\delta x_1+\epsilon_{22}\delta x_2\delta x_2+\epsilon_{23}\delta x_2\delta x_3+\epsilon_{31}\delta x_3\delta x_1+\epsilon_{32}\delta x_3\delta x_2+\epsilon_{33}\delta x_3\delta x_3}{||\delta x||^2}$$

$$\epsilon(\boldsymbol{n}) = \frac{\epsilon_{11}n_1^2 + \epsilon_{12}n_1n_2 + \epsilon_{13}n_1n_3 + \epsilon_{21}n_2n_1 + \epsilon_{22}n_2^2 + \epsilon_{23}n_2n_3 + \epsilon_{31}n_3n_1 + \epsilon_{32}n_3n_2 + \epsilon_{33}n_3^2}{||\boldsymbol{n}||^2}$$

$$\epsilon(\boldsymbol{n}) = 0.004986$$

The strains measured are not exactly equal but are fairly close. This is because in 3(vi) we are using linearized or infinitesimal strain theory. Here, we neglect non-linear terms. As can be seen by the comparison, the strain obtained through this method is a bit smaller than the one measured using 3(iv).

Sub question b:

b) Repeat the calculations from part (a) for the same point P but now consider Q' with the following coordinates in the undeformed body: (2.13363, 5.40089, 7.26726). The line PQ' is parallel to the line PQ but Q' is closer to P. Repeat question (vi) of part (a) with this new data.

The new coordinates of P as P' are found to be:

$$P' = (2.0024e_1 + 5.02e_2 + 7.0039e_3)$$

In coordinates, P' = (2.0024, 5.02, 7.0039)

Using the displacement gradient function, the new coordinates of Q as Q' are found to be:

$$Q' = (2.1360843 + 5.423829e_2 + 7.271387e_3)$$

Q' = (2.1360843, 5.423829, 7.271387)

We know the coordinates of P and Q before deformation are (2,5,7) and (2.1360843, 5.423829, 7.271387) respectively. The distance between two points is given by:

$$Distance = \sqrt{(X_{Q1} - X_{P1})^2 + (X_{Q2} - X_{P2})^2 + (X_{Q3} - X_{P3})^2}$$

$$Distance = \sqrt{(2.13363 - 2)^2 + (5.40089 - 5)^2 + (7.26726 - 7)^2}$$

$$Distance = 0.5\ units$$

We know the current coordinates of P' and Q' are (2.0024,5.02,7.0039) and (2.136142, 5.427091, 7.271619) respectively. The distance between two points is given by:

$$Distance = \sqrt{(X_{Q'1} - X_{P'1})^2 + (X_{Q'2} - X_{P'2})^2 + (X_{Q'3} - X_{P'3})^2}$$

$$Distance = 0.502492\ units$$

We can find the relative elongation of a fiber as

$$\epsilon = \frac{Final\ Length - Initial\ length}{Initial\ length}$$

$$\epsilon = \frac{P'Q' - PQ}{PQ}$$

$$\epsilon = \frac{0.502492 - 0.5}{0.5}$$

$$\epsilon = 0.004991$$

To compute unit vector n in directions of line PQ, we need to do:

$$\boldsymbol{n} = \frac{\boldsymbol{PQ}}{||\boldsymbol{PQ}||} = \frac{(2.13363 - 2)e_1 + (5.40089 - 5)e_2 + (7.26726 - 7)e_3}{\sqrt{(2.13363 - 2)^2 + (5.40089 - 5)^2 + (7.26726 - 7)^2}}$$

$$\boldsymbol{n} = \frac{0.13363e_1 + 0.40089e_2 + 0.26726e_3}{0.5}$$

To compute relative elongation using the in-class method in the direction of n we use the following formula:

$$\epsilon(\boldsymbol{n}) = \sum_{i,j=1}^{3} \epsilon_{ij} n_i n_j;\ n_i n_j = \frac{\delta x_i}{||\delta x||} \frac{\delta x_j}{||\delta x||}$$

$$\epsilon(\boldsymbol{n}) = \frac{\epsilon_{11}\delta x_1 \delta x_1 + \epsilon_{12}\delta x_1 \delta x_2 + \epsilon_{13}\delta x_1 \delta x_3 + \epsilon_{21}\delta x_2 \delta x_1 + \epsilon_{22}\delta x_2 \delta x_2 + \epsilon_{23}\delta x_2 \delta x_3 + \epsilon_{31}\delta x_3 \delta x_1 + \epsilon_{32}\delta x_3 \delta x_2 + \epsilon_{33}\delta x_3 \delta x_3}{||\delta x||^2}$$

$$\epsilon(\boldsymbol{n}) = \frac{\epsilon_{11}{n_1}^2 + \epsilon_{12} n_1 n_2 + \epsilon_{13} n_1 n_3 + \epsilon_{21} n_2 n_1 + \epsilon_{22}{n_2}^2 + \epsilon_{23} n_2 n_3 + \epsilon_{31} n_3 n_1 + \epsilon_{32} n_3 n_2 + \epsilon_{33}{n_3}^2}{||\boldsymbol{n}||^2}$$

$$\epsilon(\boldsymbol{n}) = 10^{-4}\frac{\begin{matrix}4*0.13363^2+50*0.13363*0.40089-2.5*0.13363*0.62726+50*0.13363*0.40089\\+40*0.40089^2-1*0.40089*0.62726-2.5*0.13363*0.62726-1*0.40089*0.62726\\+14*0.62726^2\end{matrix}}{0.756323^2}$$

$$\epsilon(\boldsymbol{n}) = 0.004986$$

Another approach to this question is to use P+u and Q+u for the change in displacement with the corresponding points of P and Q to find u. It has been done and uploaded in the drive. The deformed points were fairly close to each other as done thorough the displacement gradient function, but the strains found out in 3a(iv) and 3b(iv) had slightly more variations using this approach.

Question 4:

Anisotropic material. Six separate tests are performed on a material and the stress and corresponding strain tensors are measured, as indicated below. Previous tests showed that the following elastic coefficients are zero: $C_{14} = C_{15} = C_{16} = C_{24} = C_{25} = C_{26} = C_{34} = C_{35} = C_{36} = C_{45} = C_{46} = C_{56} = 0\ (using\ Voigt's notation)$. Based on these tests, determine the elastic coefficients C_{IJ} (provide your results in GPa). What type of symmetry has this materia? (stress values indicated in the table are in MPa and the strains should be multiplied by 10^{-4}).

Tensor	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
σ_{11}	9	3	0.1	0	0	0
σ_{22}	32.1	0.1	1.5	0	0	0
σ_{33}	42.2	0.2	0.1	0	0	0
σ_{23}	0	0	0	0.2	0	0
σ_{13}	0	0	0	0	0.3	0
σ_{12}	0	0	0	0	0	0.4

ϵ_{11} ϵ_{22} ϵ_{33} $2\epsilon_{23}$ $2\epsilon_{13}$ $2\epsilon_{12}$	10 2 2 0 0 0	10 0 0 0 0 0	0 10 0 0 0 0	0 0 0 2 0 0	0 0 0 0 2 0	0 0 0 0 0 2

As per Voigt's notation, the elastic coefficient matrix is given by

$$[C_{IJ}] = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix}$$

Also, Voigt Notation denotes the following $\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 11 \\ 22 \\ 33 \\ 23 \\ 13 \\ 12 \end{bmatrix}$; So, $C_{53} = C_{1333}$

Given in the question that, $C_{14} = C_{15} = C_{16} = C_{24} = C_{25} = C_{26} = C_{34} = C_{35} = C_{36} = C_{45} = C_{46} = C_{56} = 0$. Note that the above matrix is symmetric.

$$[C_{IJ}] = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix}$$

From initial inspection, this system resembles an orthotropic material. However, if additional elastic coefficients after calculation tend to be zero then it could be some other symmetry.

We know the relationship between stress tensor and strain tensor.

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} = C_{IJ}\epsilon_{kl}$$

Lets first find, $C_{44}, C_{55} and\ C_{66}$ as they are simple equations.

From test 4:

$$\sigma_{23} = C_{44}(2\epsilon_{23})$$

$$0.2\ MPa = C_{44} * (2 * 10^{-4})$$

$$C_{44} = \frac{0.2}{4 * 10^{-4}} MPa = 500\ MPa = 1\ GPa$$

From test 5:

$$\sigma_{13} = C_{55}(2\epsilon_{13})$$

$$0.3\ MPa = C_{55} * (2 * 10^{-4})$$

$$C_{55} = \frac{0.3}{4 * 10^{-4}} MPa = 750\ MPa = 1.5\ GPa$$

From test 6:

$$\sigma_{12} = C_{66}(2\epsilon_{12})$$

$$0.4\ MPa = C_{66} * (2 * 10^{-4})$$

$$C_{66} = \frac{0.4}{4 * 10^{-4}} MPa = 1000\ MPa = 2\ GPa$$

Let's now write the equations for test 1, 2 and 3 to find the remaining coefficients:

$$\sigma_{11} = C_{11}(\epsilon_{11}) + C_{12}(\epsilon_{22}) + C_{13}(\epsilon_{33}) \rightarrow 1$$
$$\sigma_{22} = C_{21}(\epsilon_{11}) + C_{22}(\epsilon_{22}) + C_{23}(\epsilon_{33}) \rightarrow 2$$
$$\sigma_{33} = C_{31}(\epsilon_{11}) + C_{32}(\epsilon_{22}) + C_{33}(\epsilon_{33}) \rightarrow 3$$

Substituting the values, we get the following equations:

From test 1:

$$9 = 10^{-4}\,(10C_{11} + 2C_{12} + 2C_{13})$$
$$32.1 = 10^{-4}\,(10C_{21} + 2C_{22} + 2C_{23})$$
$$42.2 = 10^{-4}\,(10C_{31} + 2C_{32} + 2C_{33})$$

From test 2:

$$3 = 10^{-4}\,(10C_{11})$$

$$0.1 = 10^{-4}\,(10C_{21})$$
$$0.2 = 10^{-4}\,(10C_{31})$$

From test 3:

$$0.1 = 10^{-4}\,(10C_{12})$$

$$1.5 = 10^{-4}\,(10C_{22})$$

$$0.1 = 10^{-4}\,(10C_{32})$$

Solving we get,

$$C_{11} = 3\ GPa$$

$$C_{21} = 0.1\ GPa$$

$$C_{31} = 0.2\ GPa$$

$$C_{12} = 0.1\ GP, symmetry\ condition\ satisfied$$

$$C_{22} = 1.5\ GPa$$

$$C_{32} = 0.1\ GPa$$

From test-1 results, we can further calculate:

$$9 = 10^{-4}\,(10C_{11} + 2C_{12} + 2C_{13})$$
$$9 = 10^{-4}\,(10 * 3 * 10^3 + 2 * 0.1 * 10^3 + 2 * C_{13})$$
$$C_{13} = 29.9\ GPa;\ symmetry\ not\ satisfied$$

$$32.1 = 10^{-4}\,(10 * 0.1 * 10^3 + 2 * 1.5 * 10^3 + 2 * C_{23})$$
$$C_{23} = 158.5\ GPa;\ symmetry\ not\ satisfied$$

$$42.2 = 10^{-4}\,(10 * 0.2 * 10^3 + 2 * 0.1 * 10^3 + 2 * C_{33})$$
$$C_{33} = 209.9\ GPa$$

$$[C_{IJ}] = \begin{bmatrix} 3 & 0.1 & 29.9 & 0 & 0 & 0 \\ 0.1 & 1.5 & 158.5 & 0 & 0 & 0 \\ 0.2 & 0.1 & 209.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Generally, the major and minor symmetries are part of the model and not a function of the specific material. For our case, the model is based for linear elastic material. As the symmetry condition doesn't work here, perhaps the material for test 1 has crossed the linear elastic region and has deformed plastically. Hence, the disparity.

Also, on basis of this matrix, we cannot confirm whether the material is orthotropic or monoclinic as it has 11 independent parameters instead of 9 or 13.

Question 5:

Relation between material parameters for isotropic materials. Isotropic materials are characterized by only two material parameters. In practice, several parameters are used (Young's modulus E, Poisson's ratio v, shear modulus μ, Lame modulus λ and bulk modulus κ). To establish the relations between these parameters we can proceed as follows:

Sub question 1:

i. Consider a homogenous, isotropic, linearly elastic material subject to a state of uniaxial tension, i.e., the stress tensor is

$$[\sigma_{ij}] = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where σ is the axial stress. For this stress state, use the strain-stress relation to compute the corresponding strain tensor, i.e., compute $\epsilon_{ij}$from

$$\epsilon_{ij} = \frac{1}{E}\left[(1+v)\sigma_{ij} - v\delta_{ij}\sum_{k=1}^{3}\sigma_{kk}\right]$$

We have the following relation,

$$\epsilon_{ij} = \frac{1}{E}\left[(1+v)\sigma_{ij} - v\delta_{ij}\sum_{k=1}^{3}\sigma_{kk}\right]$$

$$\delta_{ij} = 1 \; if \; i = j, else \; 0$$

$$For \; our \; stress \; tensor, \sum_{k=1}^{3} \sigma_{kk} = \sigma + 0 + 0 = \sigma$$

$$\epsilon_{11} = \frac{1}{E}[(1+v)\sigma_{11} - v\delta_{11} \sum_{k=1}^{3} \sigma_{kk}]$$

$$\epsilon_{11} = \frac{1}{E}[(1+v)\sigma - v * 1 * \sigma]$$

$$\epsilon_{11} = \frac{\sigma}{E}$$

$$\epsilon_{12} = \frac{1}{E}[(1+v)\sigma_{12} - v\delta_{12} \sum_{k=1}^{3} \sigma_{kk}]$$

$$\epsilon_{12} = \frac{1}{E}[(1+v) * 0 - v * 0 * \sigma]$$

$$\epsilon_{12} = 0$$

$$From \; symmetry, \epsilon_{21} = 0$$

$$\epsilon_{13} = \frac{1}{E}[(1+v)\sigma_{13} - v\delta_{13} \sum_{k=1}^{3} \sigma_{kk}]$$

$$\epsilon_{13} = \frac{1}{E}[(1+v) * 0 - v * 0 * \sigma]$$

$$\epsilon_{13} = 0$$

$$From \; symmetry, \epsilon_{31} = 0$$

$$\epsilon_{22} = \frac{1}{E}[(1+v)\sigma_{22} - v\delta_{22} \sum_{k=1}^{3} \sigma_{kk}]$$

$$\epsilon_{22} = \frac{1}{E}[(1+v) * 0 - v * 1 * \sigma]$$

$$\epsilon_{22} = \frac{-v\sigma}{E}$$

$$\epsilon_{23} = \frac{1}{E}[(1+v)\sigma_{23} - v\delta_{23} \sum_{k=1}^{3} \sigma_{kk}]$$

$$\epsilon_{23} = \frac{1}{E}[(1+v) * 0 - v * 0 * \sigma]$$

$$\epsilon_{23} = 0$$

$$From \; symmetry, \epsilon_{32} = 0$$

$$\epsilon_{33} = \frac{1}{E}[(1+v)\sigma_{33} - v\delta_{33} \sum_{k=1}^{3} \sigma_{kk}]$$

$$\epsilon_{33} = \frac{1}{E}[(1+v) * 0 - v * 1 * \sigma]$$

$$\epsilon_{33} = \frac{-v\sigma}{E}$$

The strain tensor is given by,

$$\epsilon = \frac{1}{E}\begin{bmatrix} \sigma & 0 & 0 \\ 0 & -v\sigma & 0 \\ 0 & 0 & -v\sigma \end{bmatrix}$$

Sub question 2:

ii. With the strain tensor computed in (i), use now the stress-strain relation to compute the corresponding stress tensor, i.e.,

$$\sigma_{ij} = \lambda\delta_{ij}\sum_{k=1}^{3}\epsilon_{ij} + 2\mu\epsilon_{ij}$$

The strain tensor we will use is,

$$\epsilon = \frac{1}{E}\begin{bmatrix} \sigma & 0 & 0 \\ 0 & -v\sigma & 0 \\ 0 & 0 & -v\sigma \end{bmatrix}$$

We have the following relation,

$$\sigma_{ij} = \lambda\delta_{ij}\sum_{k=1}^{3}\epsilon_{ij} + 2\mu\epsilon_{ij}$$

$$\delta_{ij} = 1 \; if \; i = j, else \; 0$$

$$For \; our \; strain \; tensor, \sum_{k=1}^{3}\epsilon_{kk} = \frac{\sigma}{E} + \frac{-v\sigma}{E} + \frac{-v\sigma}{E} = \frac{\sigma - 2v}{E}$$

$$\sigma_{11} = \lambda\delta_{11}\sum_{k=1}^{3}\epsilon_{kk} + 2\mu\epsilon_{11}$$

$$\sigma_{11} = \lambda * 1 * \frac{\sigma - 2v\sigma}{E} + 2 * \mu * \frac{\sigma}{E}$$

$$\sigma_{11} = \frac{1}{E}[\lambda * 1 * (\sigma - 2v\sigma) + 2 * \mu * \sigma]$$

$$\sigma_{11} = \frac{1}{E}[\lambda\sigma - 2\lambda v\sigma + 2 * \mu * \sigma]$$

$$\sigma_{11} = \frac{\sigma}{E}[(\lambda + 2\mu) - 2\lambda v]$$

$$\sigma_{11} = \frac{\sigma}{E}[\lambda(1 - 2v) + 2\mu]$$

$$\sigma_{12} = \lambda\delta_{12}\sum_{k=1}^{3}\epsilon_{kk} + 2\mu\epsilon_{12}$$

$$\sigma_{12} = \lambda * 0 * \frac{\sigma - 2v}{E} + 2 * \mu * 0$$

$$\sigma_{12} = 0$$

$$From \; symmetry, \sigma_{21} = 0$$

$$\sigma_{13} = \lambda\delta_{13}\sum_{k=1}^{3}\epsilon_{kk} + 2\mu\epsilon_{13}$$

$$\sigma_{13} = \lambda * 0 * \frac{\sigma - 2v}{E} + 2 * \mu * 0$$

$$\sigma_{13} = 0$$

$$From\ symmetry, \sigma_{31} = 0$$

$$\sigma_{22} = \lambda\delta_{22}\sum_{k=1}^{3}\epsilon_{kk} + 2\mu\epsilon_{22}$$

$$\sigma_{22} = \lambda * 1 * \frac{\sigma - 2v\sigma}{E} + 2 * \mu * \frac{-v\sigma}{E}$$

$$\sigma_{22} = \frac{1}{E}[\lambda * 1 * (\sigma - 2v\sigma) + 2 * \mu * -v\sigma]$$

$$\sigma_{22} = \frac{1}{E}[\lambda\sigma - 2v\lambda\sigma - 2\mu v\sigma]$$

$$\sigma_{22} = \frac{\sigma}{E}[\lambda(1 - 2v) - 2\mu v]$$

$$\sigma_{23} = \lambda\delta_{23}\sum_{k=1}^{3}\epsilon_{kk} + 2\mu\epsilon_{23}$$

$$\sigma_{23} = \lambda * 0 * \frac{\sigma - 2v}{E} + 2 * \mu * 0$$

$$\sigma_{23} = 0$$

$$From\ symmetry, \sigma_{32} = 0$$

$$\sigma_{33} = \lambda\delta_{33}\sum_{k=1}^{3}\epsilon_{kk} + 2\mu\epsilon_{33}$$

$$\sigma_{33} = \lambda * 1 * \frac{\sigma - 2v\sigma}{E} + 2 * \mu * \frac{-v\sigma}{E}$$

$$\sigma_{33} = \frac{1}{E}[\lambda * 1 * (\sigma - 2v\sigma) + 2 * \mu * -v\sigma]$$

$$\sigma_{33} = \frac{1}{E}[\lambda\sigma - 2v\lambda\sigma - 2\mu v\sigma]$$

$$\sigma_{33} = \frac{\sigma}{E}[\lambda(1 - 2v) - 2\mu v]$$

Therefore, our stress tensor will be:

$$\sigma = \begin{bmatrix} \frac{\sigma}{E}[\lambda(1-2v) + 2\mu] & 0 & 0 \\ 0 & \frac{\sigma}{E}[\lambda(1-2v) - 2\mu v] & 0 \\ 0 & 0 & \frac{\sigma}{E}[\lambda(1-2v) - 2\mu v] \end{bmatrix}$$

Sub question 3:

iii. Use the fact that the stress tensor computed in (ii) corresponds to a state of uniaxial tension to express E and v as functions of λ and μ and conversely to express λ and μ as functions of E and v. Compute λ and μ for a steel with E = 200 GPa and $v = 0.3$.

We know that the corresponding stress tensor is for a uniaxial tensile test.

Therefore,

$$\sigma_{11} = \sigma = \frac{\sigma}{E}[\lambda(1-2v) + 2\mu]$$

$$\sigma_{22} = 0 = \frac{\sigma}{E}[\lambda(1-2v) - 2\mu v]$$

$$\sigma_{33} = 0 = \frac{\sigma}{E}[\lambda(1-2v) - 2\mu v]$$

Equation for σ_{22} and σ_{33} are the same. Lets solve one of them to get a relation and then substitute in σ_{11}.

$$\frac{\sigma}{E}[\lambda(1-2v) - 2\mu v] = 0$$

$$\lambda(1-2v) - 2\mu v = 0$$

$$\lambda(1-2v) = 2\mu v$$

$$\lambda = \frac{2\mu v}{1-2v} \rightarrow \mathbf{1}$$

Next, from σ_{11} equation, we have

$$\sigma = \frac{\sigma}{E}[\lambda(1-2v) + 2\mu]$$

$$E = \lambda(1-2v) + 2\mu$$

$$E = \frac{2\mu v}{1-2v}(1-2v) + 2\mu$$

$$E = 2\mu v + 2\mu$$

$$E = 2\mu(1+v) \rightarrow \mathbf{2}$$

$$\mu = \frac{E}{2(1+v)}$$

Now substituting, 2μ from 2 in equation 1

$$\lambda = \frac{\frac{E}{(1+v)}v}{1-2v}$$

$$\lambda = \frac{vE}{(1+v)(1-2v)}$$

From equation 1,

$$\lambda = \frac{2\mu v}{1-2v}$$

$$\lambda(1-2v) = 2\mu v$$

$$\lambda - 2v\lambda = 2\mu v$$
$$\lambda = 2v\lambda + 2\mu v$$
$$\lambda = 2v(\lambda + \mu)$$
$$v = \frac{\lambda}{2(\lambda + \mu)} \rightarrow \mathbf{3}$$

Substituting equation 3 in equation 2, we get

$$E = 2\mu\left(1 + \frac{\lambda}{2(\lambda + \mu)}\right)$$
$$E = 2\mu\left(\frac{2(\lambda + \mu) + \lambda}{2(\lambda + \mu)}\right)$$
$$E = \mu\left(\frac{2\lambda + 2\mu + \lambda}{(\lambda + \mu)}\right)$$
$$E = \mu\left(\frac{3\lambda + 2\mu}{\lambda + \mu}\right)$$

Therefore, the relations sought after are:

$$\mu = \frac{E}{2(1 + v)}$$
$$\lambda = \frac{vE}{(1 + v)(1 - 2v)}$$
$$v = \frac{\lambda}{2(\lambda + \mu)}$$
$$E = \mu\left(\frac{3\lambda + 2\mu}{\lambda + \mu}\right)$$

Given, E = 200 GPa and $v = 0.3$

$$\lambda = \frac{vE}{(1 + v)(1 - 2v)}$$
$$\lambda = \frac{0.3 * 200}{(1 + 0.3)(1 - 2 * 0.3)}$$
$$\lambda = 115.384\ GPa$$
$$\mu = \frac{E}{2(1 + v)}$$
$$\mu = \frac{200}{2(1 + 0.3)}$$
$$\mu = 76.923\ GPa$$

Sub question 4:

iv. For the bulk modulus, we proceed as follows: consider a general deformation with stress tensor σ_{ij} and a corresponding strain tensor ϵ_{ij}. The bulk modulus κ is defined such that

$$(\sigma_{11}+\sigma_{22}+\sigma_{33}) = 3\kappa(\epsilon_{11}+\epsilon_{22}+\epsilon_{33})$$

Use the stress strain relation to express the bulk modulus κ as a function of λ and μ. Repeat the procedure but this time using the strain-stress relation to obtain κ as a function of E and v.

We have the following relation,

$$(\sigma_{11}+\sigma_{22}+\sigma_{33}) = 3\kappa(\epsilon_{11}+\epsilon_{22}+\epsilon_{33})$$

Substituting,

$$\frac{\sigma}{E}[\lambda(1-2v)+2\mu]+\frac{\sigma}{E}[\lambda(1-2v)-2\mu v]+\frac{\sigma}{E}[\lambda(1-2v)-2\mu v] = 3\kappa\left(\frac{\sigma}{E}+\frac{-v\sigma}{E}+\frac{-v\sigma}{E}\right)$$

$$\lambda(1-2v)+2\mu+\lambda(1-2v)-2\mu v+\lambda(1-2v)-2\mu v = 3\kappa(1-2v)$$

$$\kappa = \frac{3\lambda(1-2v)+2\mu-4\mu v}{3(1-2v)}$$

Substituting, $v = \frac{\lambda}{2(\lambda+\mu)}$

$$\kappa = \frac{3\lambda\left(1-2\frac{\lambda}{2(\lambda+\mu)}\right)+2\mu-4\mu\frac{\lambda}{2(\lambda+\mu)}}{3\left(1-2\frac{\lambda}{2(\lambda+\mu)}\right)}$$

$$\kappa = \frac{3\lambda\left(1-\frac{\lambda}{(\lambda+\mu)}\right)+2\mu-2\mu\frac{\lambda}{(\lambda+\mu)}}{3(1-\frac{\lambda}{(\lambda+\mu)})}$$

$$\kappa = \frac{3\lambda\left(\frac{\lambda+\mu-\lambda}{\lambda+\mu}\right)+2\mu(\frac{\lambda+\mu-\lambda}{\lambda+\mu})}{3(\frac{\lambda+\mu-\lambda}{\lambda+\mu})}$$

$$\kappa = \frac{3\lambda\mu+2\mu^2}{3\mu}$$

$$\kappa = \frac{3\lambda+2\mu}{3}$$

Again, we have the following relation,

$$(\sigma_{11}+\sigma_{22}+\sigma_{33}) = 3\kappa(\epsilon_{11}+\epsilon_{22}+\epsilon_{33})$$

Substituting,

$$(\sigma+0+0) = 3\kappa\left(\frac{\sigma}{E}+\frac{-v\sigma}{E}+\frac{-v\sigma}{E}\right)$$

$$\sigma = 3\kappa\left(\frac{\sigma - 2v\sigma}{E}\right)$$

$$\kappa = \frac{E}{3(1-2v)}$$

Question 6:

Axial deformations; isotropic material. Consider a homogeneous, linearly elastic isotropic steel with Young's modulus E and Poisson's ratio v. A rectangular sample of the material is subject to the following tests:

Sub question 1:

i. Simple extension. The displacement field is

$$u_1 = \epsilon x_1$$

$$u_2 = 0$$

$$u_3 = 0$$

where ϵ is the axial strain. Compute the strain tensor. Starting from the strain-stress relation

$$\epsilon_{ij} = \frac{1}{E}\left[(1+v)\sigma_{ij} - v\delta_{ij}\sum_{k=1}^{3}\sigma_{kk}\right]$$

Compute the corresponding stress tensor (alternatively, you may use the stress-strain relation and express λ and μ in terms of E and v using the results from question 1). For E = 200 GPa and v = 0.3, plot the axial stress σ_{11} as a function of the axial strain ϵ_{11}.

Let us first find the strain tensor.

c→ **1**

For finding the stress tensor, the following has been given,

$$\epsilon_{ij} = \frac{1}{E}\left[(1+v)\sigma_{ij} - v\delta_{ij}\sum_{k=1}^{3}\sigma_{kk}\right]$$

We know, $\delta_{ij} = 1 \; if \; i = j \; else \; 0$; Making a strain tensor using this equation we get,

$$\epsilon = \frac{1}{E}\begin{bmatrix}(1+v)\sigma_{11} - v(\sigma_{11}+\sigma_{22}+\sigma_{33}) & (1+v)\sigma_{12} & (1+v)\sigma_{13} \\ (1+v)\sigma_{21} & (1+v)\sigma_{22} - v(\sigma_{11}+\sigma_{22}+\sigma_{33}) & (1+v)\sigma_{23} \\ (1+v)\sigma_{31} & (1+v)\sigma_{32} & (1+v)\sigma_{33} - v(\sigma_{11}+\sigma_{22}+\sigma_{33})\end{bmatrix}$$

$$\epsilon = \frac{1}{E}\begin{bmatrix}\sigma_{11} - v(\sigma_{22}+\sigma_{33}) & (1+v)\sigma_{12} & (1+v)\sigma_{13} \\ (1+v)\sigma_{21} & \sigma_{22} - v(\sigma_{11}+\sigma_{33}) & (1+v)\sigma_{23} \\ (1+v)\sigma_{31} & (1+v)\sigma_{32} & \sigma_{33} - v(\sigma_{11}+\sigma_{22})\end{bmatrix}$$

Equating the obtained tensor with the earlier calculated tensor in 1 gives,

$$\sigma_{12} = \sigma_{21} = 0$$

$$\sigma_{13} = \sigma_{31} = 0$$

$$\sigma_{23} = \sigma_{32} = 0$$

$$\frac{1}{E}[\sigma_{11} - v(\sigma_{22} + \sigma_{33})] = \epsilon$$

$$\frac{1}{E}[\sigma_{22} - v(\sigma_{11} + \sigma_{33})] = 0$$

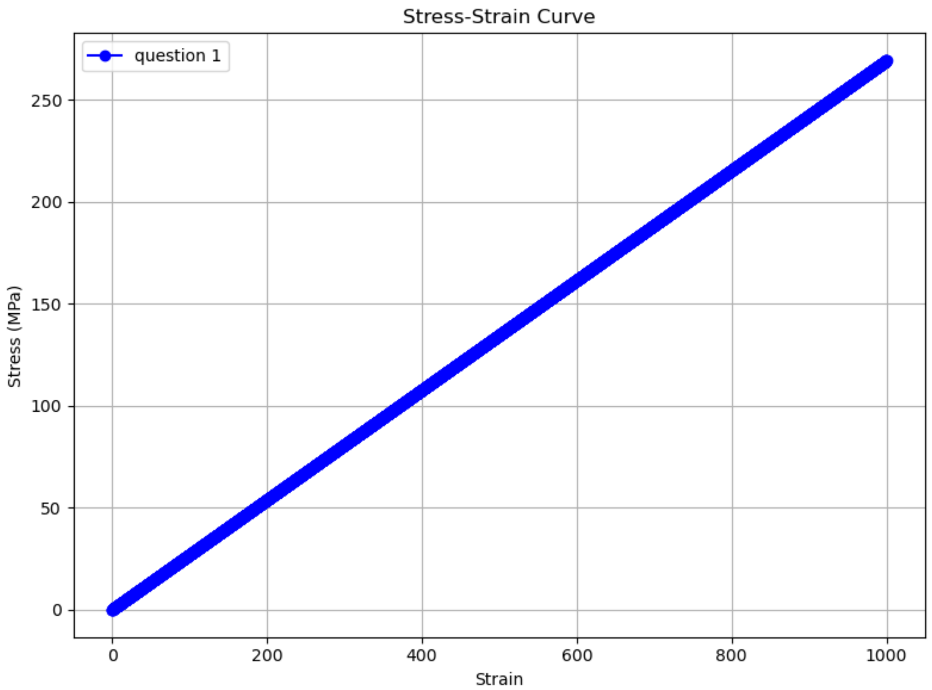
$$\frac{1}{E}[\sigma_{33} - v(\sigma_{11} + \sigma_{22})] = 0$$

Three equations and three unknowns, solving we get:

$$\sigma = \frac{E\epsilon}{(1+v)(1-2v)}\begin{bmatrix} 1-v & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & v \end{bmatrix}$$

Note: The calculation for solving the equations was done by hand. A copy is attached here in the drive.

Plotting σ_{11} as a function of the axial strain $\epsilon_{11}$with E = 200 GPa and $v = 0.3$:



Units of strain is micro strain.

Sub question 2:

ii. Uniaxial tension. The stress tensor is

$$[\sigma_{ij}] = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Where σ is the axial stress. Compute the corresponding strain tensor. Plot the axial stress σ_{11} as a function of the axial strain ϵ_{11}. Comment on the difference between the slopes of this curve and the one obtained in part (i). Provide (in words) a brief physical interpretation of the difference.

For finding the strain tensor, the following has been given in question 6(i),

$$\epsilon_{ij} = \frac{1}{E}\left[(1+v)\sigma_{ij} - v\delta_{ij}\sum_{k=1}^{3}\sigma_{kk}\right]$$

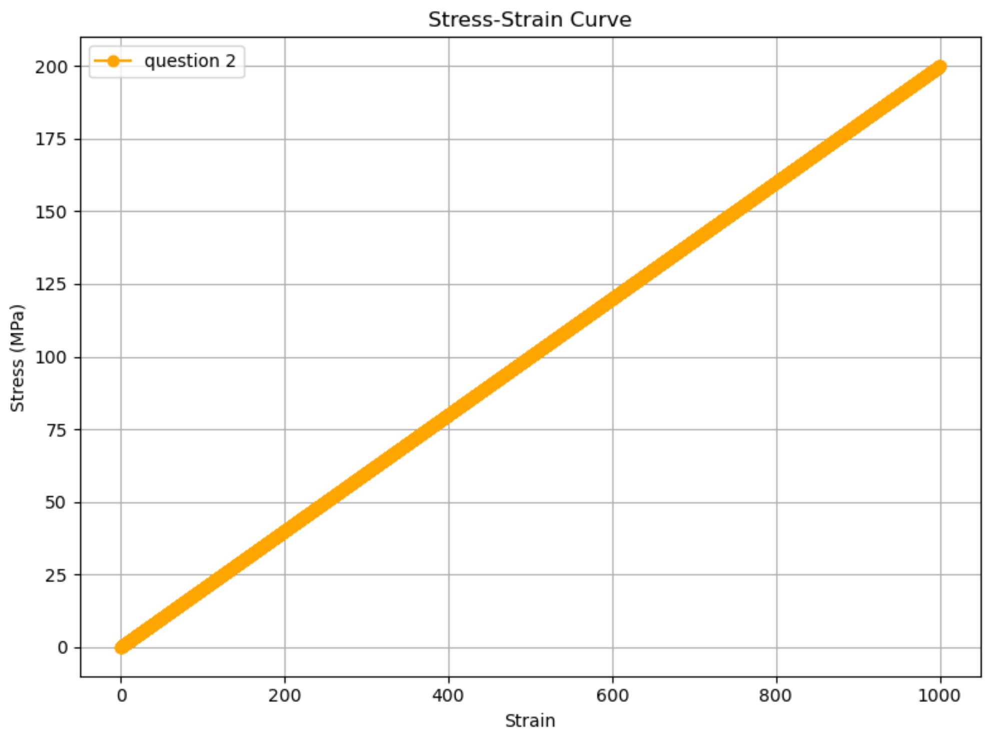
We know, $\delta_{ij} = 1\ if\ i = j\ else\ 0$; Making a strain tensor using this equation we get,

$$\epsilon = \frac{1}{E}\begin{bmatrix}(1+v)\sigma_{11} - v(\sigma_{11}+\sigma_{22}+\sigma_{33}) & (1+v)\sigma_{12} & (1+v)\sigma_{13} \\ (1+v)\sigma_{21} & (1+v)\sigma_{22} - v(\sigma_{11}+\sigma_{22}+\sigma_{33}) & (1+v)\sigma_{23} \\ (1+v)\sigma_{31} & (1+v)\sigma_{32} & (1+v)\sigma_{33} - v(\sigma_{11}+\sigma_{22}+\sigma_{33})\end{bmatrix}$$

$$\epsilon = \frac{1}{E}\begin{bmatrix}\sigma_{11} - v(\sigma_{22}+\sigma_{33}) & (1+v)\sigma_{12} & (1+v)\sigma_{13} \\ (1+v)\sigma_{21} & \sigma_{22} - v(\sigma_{11}+\sigma_{33}) & (1+v)\sigma_{23} \\ (1+v)\sigma_{31} & (1+v)\sigma_{32} & \sigma_{33} - v(\sigma_{11}+\sigma_{22})\end{bmatrix}$$

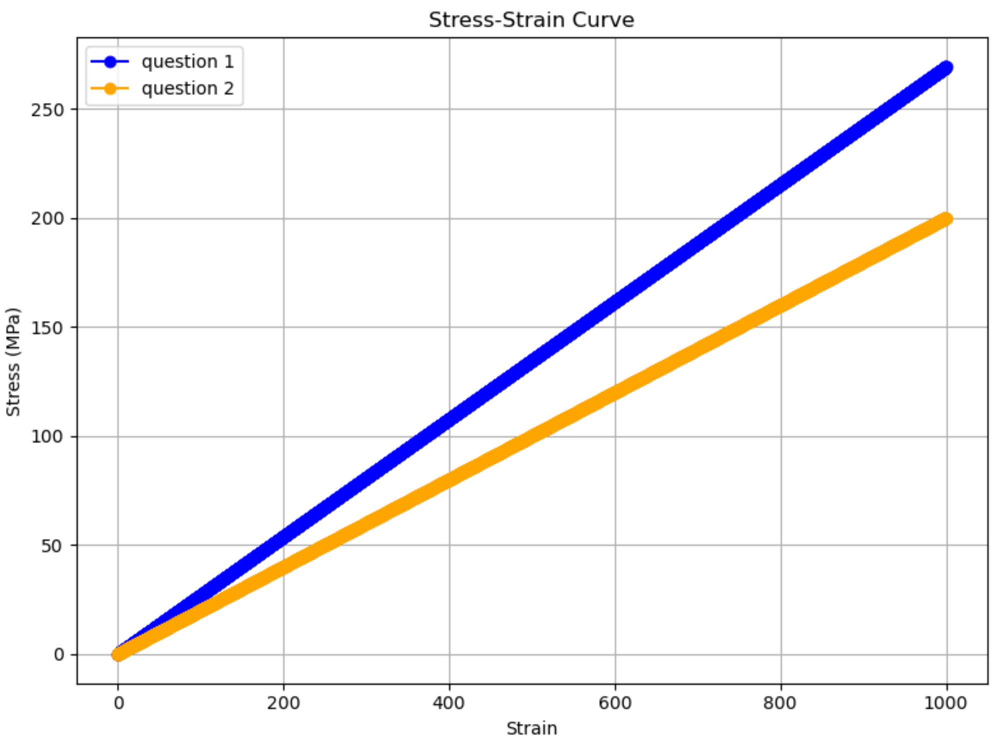
Substituting the corresponding stresses from the given stress tensor, we get

$$\epsilon = \frac{1}{E}\begin{bmatrix}\sigma & 0 & 0 \\ 0 & -v\sigma & 0 \\ 0 & 0 & -v\sigma\end{bmatrix}$$



Units of strain is micro strain.

Comparing 1 and 2 we get:



Units of strain is micro strain.

We know that $\sigma_{11} = C_{1111}\epsilon_{11}$

Where $C_{1111} = \frac{E(1-v)}{(1+v)(1-2v)}$, this is the actual elastic constant that needs to be considered while evaluating σ_{11}based on ϵ_{11}. This is what question 6(i) considers while plotting the stress vs strain graph.

However, in general we just write $\sigma_{11} = E\epsilon_{11}$. This is also the exact equation we got in 6(ii) and the equation we use to calculate strain energy of beams. This is done for convenience as for 0.2% strain where the equations are valid (linear region), the differences are reasonably small.

In question (i), the case was that of a plane strain. That is the strains in $\epsilon_{13} = \epsilon_{23} = \epsilon_{33} = 0$, additionally we also have the strains in the y-plane to be zero too. This is reflected in the stress matrix that we obtained, where it shows $\sigma_{22} = \sigma_{33} = +ve\ value$. These stresses are infact trying to pull the body in y and z directions to make $\epsilon_{22} = \epsilon_{33} = 0$.

In question (ii), we have prescribed the stresses, and it was the condition of plane stress or because even the y-plane stresses are zero, it is the condition of uniaxial tension. To achieve these conditions, the body is free to deform in these directions causing compressive strain $\epsilon_{22}\ and \epsilon_{33}$. This is the reason why the stiffness is different in both the cases.

Question 7:

Euler-Bernoulli beam under compressive load. [Note: For this exercise, you may use results from the class notes. You should indicate the meaning of the symbols and quantities that you require for this problem and indicate in words the steps to obtain each formula.]

Consider a homogenous linearly elastic beam subjected to an axial compressive force F applied on both ends along the centerline. Do not consider lateral deflections for this problem.

Sub question 1:

i. Write down the total potential energy of the beam and formulate the axial loading problem as a minimization problem.

Let us consider a Euler Bernoulli beam. The beam is subjected to an axial compressive force of F. Let the beam have a young's modulus E and Area of cross-section A and length L. Let the width of the beam be b and the depth of the beam be h.

$$Area\ of\ the\ beam\ 'A' = b * h$$

Let us assume that the beam is simply supported with the pinned support at x = 0 and the roller support at x = L.

Let us define the displacement function u(x) as

$$\boldsymbol{u}(\boldsymbol{x}) = u_1 e_1 + u_2 e_2 + u_3 e_3$$

Here, no lateral deflections are being considered, hence:

$$u_2, u_3 = 0$$

$$u_1 = u_c(x) \rightarrow Deflection\ along\ the\ centerline$$

$$\boldsymbol{u}(\boldsymbol{x}) = u_c(\boldsymbol{x}) e_1$$

The total potential energy of a beam is given by,

$$P[\boldsymbol{u}] = U[\boldsymbol{u}] + V[\boldsymbol{u}]$$

U: Total strain energy = $\frac{1}{2}\int_\Omega \epsilon . C\epsilon dv = \frac{1}{2}\sum_{i,j,k,l=1}^{3}\int_\Omega \epsilon_{ij} C_{ijkl}\epsilon_{kl} dv$

C_{ijkl}: gives the elastic constant

ϵ_{ij}: The strain normal to the plane i and along the direction j

V: Total potential energy due to external loading = $-\int_\Omega \boldsymbol{b}.\boldsymbol{u} dv - \int_{\partial\Omega_t} \boldsymbol{t}.\boldsymbol{u} da$

b is the body force vector and t is the surface traction vector.

The above equation can be obtained from the strong formulation by applying a small virtual displacement and then integrating by-parts to remove the differentiation on the virtual displacements. The goal is to obtain a formulation of the type:

$$f(\boldsymbol{u}).w = 0\ for\ all\ w$$

$$U[\boldsymbol{u}] = \frac{1}{2}\int \epsilon_{11}\sigma_{11}dv \;\; [as\ other\ components\ are\ zero]$$

$$U[\boldsymbol{u}] = \frac{1}{2}\int \epsilon_{11}\sigma_{11}dv = \frac{E}{2}\int {\epsilon_{11}}^{2}dv$$

$$U[\boldsymbol{u}] = \frac{E}{2}\int_{0}^{L}\int_{-\frac{b}{2}}^{\frac{b}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}}\left(\frac{du_c}{dx}\right)^{2} dzdydx$$

As u_c is not a function of y and z we can integrate it directly.

$$U[\boldsymbol{u}] = \frac{EA}{2}\int_{0}^{L}\left(\frac{du_c}{dx}\right)^{2} dx$$

Our beam is assumed to be a simply supported beam and only an axial compressive force is acting on it and no body forces.

$$V[\boldsymbol{u}] = \int_{\partial\Omega_t} \boldsymbol{t}.\boldsymbol{u}da = Fu_c(0) - Fu_c(L) = -\int_{0}^{L}(F\frac{du_c(x)}{dx})\, dx$$

Total Potential energy is given by,

$$P[\boldsymbol{u}] = U[\boldsymbol{u}] + V[\boldsymbol{u}] = \frac{EA}{2}\int_{0}^{L}\left(\frac{du_c}{dx}\right)^{2} dx + \int_{0}^{L}(F\frac{du_c(x)}{dx})\, dx$$

$$P[\boldsymbol{u}] = \int_{0}^{L}(\frac{EA}{2}\left(\frac{du_c}{dx}\right)^{2} + F\left(\frac{du_c}{dx}\right))\, dx$$

Sub question 2:

ii. Consider Ritz method for the axial deformation and choose a single function to approximate the axial displacement. Does this function need to be zero at the end points? Why/why not? Use Ritz's method to solve the problem (clearly indicate all the steps in your solution method).

We need to give an approximation to our displacement function u_c.

$$u_c(x) = c^{(0)}\varphi^{(0)}(x) + \sum_{i=1}^{N} c^{(n)}\varphi^{(n)}(x)$$

Our boundary conditions for displacement are as follows:

At x = o, $u_c = 0$, therefore, $\varphi^{(0)}(x) = 0$

At x = L, u_c is not zero because it is a roller support, and it can move axially at this end. Hence, let us choose a single polynomial function.

$$c^{(n)} = c^{(1)} = c$$

$$\varphi^{(n)} = \varphi^{(1)} = x^3$$

We are choosing $\varphi^{(1)}$such that it is at least differentiable once.

$$u_c(x) = cx^3$$

Substituting in the equation P[**u**] we get

$$P[\boldsymbol{u}] = \int_0^L (\frac{EA}{2}\left(\frac{du_c}{dx}\right)^2 + F\left(\frac{du_c}{dx}\right))\, dx$$

$$P[\boldsymbol{u}] = \int_0^L (\frac{EA}{2}(3cx^2)^2 + F(3cx^2))\, dx$$

$$P[\boldsymbol{u}] = \frac{EA}{2}\frac{9c^2x^5}{5} + Fcx^3]_0^L$$

$$P[\boldsymbol{u}] = \frac{9EAc^2L^5}{10} + FcL^3$$

Next, we need to find the stationary point 'c' for which, $\frac{dP}{dc} = 0$

$$\frac{dP}{dc} = \frac{9EAcL^5}{5} + FL^3 = 0$$

$$\frac{9EAcL^5}{5} = -FL^3$$

$$c = \frac{-5F}{9EAL^2}$$

Therefore,

$$u(x) = u_c(x)e_1 = \frac{-5F}{9EAL^2}\, x^3 e_1$$

Sub question 3:

iii. From the potential energy, formulate the principle of virtual work for the axial loading problem.

We have,

$$P[\boldsymbol{u}] = \int_0^L (\frac{EA}{2}\left(\frac{du_c}{dx}\right)^2 + F\left(\frac{du_c}{dx}\right))\, dx$$

$$P[u] = \int_0^L H(u'_c)dx$$

The first variation can be given as,

$$\delta P[u_c] = \int_0^L \frac{\partial H}{\partial u'_c}\delta u'_c\, dx$$

$$\frac{\partial H}{\partial u'_c} = EAu'_c + F$$

$$\delta P[u_c] = \int_0^L (\,EAu'_c + F)\delta u'_c dx$$

As per the principle of virtual work,

$$\delta P[u_c] = 0$$

Sub question 4:

iv. Using the same function as for Ritz method, use Galerkin's method to solve the problem (clearly indicate all the steps in your solution method). Compare the solutions from Galerkin and Ritz's methods.

Using same function as Ritz method,

$$\boldsymbol{u}(\boldsymbol{x}) = u_c(\boldsymbol{x})e_1$$

$$u_c(\boldsymbol{x}) = cx^3$$

$$u'_c = 3cx^2$$

Again for the virtual displacement δu_c, the same boundary conditions apply. At x=0, $\delta u_c = 0$ due to pin support and at x =L, it is a roller, hence the displacement need not be zero.

$$\delta u_c(\boldsymbol{x}) = dx^3$$

$$\delta u'_c(\boldsymbol{x}) = 3dx^2$$

Substituting in

$$\delta P[u_c] = 0$$

$$\delta P[u_c] = \int_0^L (EAu'_c + F)\delta u'_c dx = 0$$

$$\int_0^L (EA(3cx^2) + F)(3dx^2)dx = 0$$

$$\int_0^L (9EAcdx^4 + 3Fdx^2)dx = 0$$

$$\frac{9EAcdx^5}{5} + Fdx^3]_0^L = 0$$

$$\left(\frac{9EAcL^5}{5} + FL^3\right)d = 0$$

If this has to be true for any d, then the term in parentheses must be zero.

$$\frac{9EAcL^5}{5} + FL^3 = 0$$

Solving for c we get,

$$c = \frac{-5F}{9EAL^2}$$

Therefore,

$$u(x) = u_c(x)e_1 = \frac{-5F}{9EAL^2}\, x^3 e_1$$

The solution obtained from Galerkin and Ritz's methods are the same.

Sub question 5:

v. From the principle of virtual work, obtain the strong formulation for the axial loading problem. Indicate the possible boundary conditions at both ends. What type of boundary conditions were used in the formulation above?

From the principle of virtual work,

$$\delta P[u_c] = \int_0^L (\, EAu'_c + F)\delta u'_c dx$$

To convert it into the strong formulation we need to bring it into the form,

$$\delta P[u_c] = \int_0^L (f_u \delta u_c) dx = 0 \; for \; all \; \delta u_c$$

To convert it, we can integrate once by parts

$$[(\, EAu'_c + F)\delta u_c]' = (\, EAu'_c + F)\delta u'_c + (\, EAu'_c + F)'\delta u_c$$

$$(\, EAu'_c + F)\delta u'_c = [(\, EAu'_c + F)\delta u_c]' - (\, EAu'_c + F)'\delta u_c$$

$$(\, EAu'_c + F)\delta u'_c = [(\, EAu'_c + F)\delta u_c]' - (\, EAu''_c)\delta u_c$$

$$\delta P[u_c] = \int_0^L (\, EAu'_c + F)\delta u'_c dx$$

$$= \int_0^L ([(\, EAu'_c + F)\delta u_c]' - (\, EAu''_c)\delta u_c)\, dx$$

$$\delta P[u_c] = [(\, EAu'_c + F)\delta u_c]_0^L - \int_0^L (\, EAu''_c)\delta u_c)\, dx = 0$$

Therefore, the boundary conditions would be:

$$[(\, EAu'_c + F)\delta u_c]_{x=0} = 0$$

$$[(\, EAu'_c + F)\delta u_c]_{x=L} = 0$$

That is,

$$EAu'_c(0) = -F \; or \; \delta u_c(0) = 0$$

and ,

$$EAu'_c(L) = -F \; or \; \delta u_c(L) = 0$$

And the equation of equilibrium being,

$$EAu''_c = 0 \; for \; all \; x \; \epsilon \; [0, L]$$

For our beam, we will be considering it to be simply supported with a pin support at end x = 0 and roller support at x = L. Therefore, the boundary conditions at the two ends must be:

$$\delta u_c(0) = 0$$

And,

$$EAu'_c(L) = -F$$

Sub question 6:

vi. Solve the problem again using direct integration. Is this solution the same or different than the solutions above (minimum potential energy, principle of virtual work)?

Our ODE obtained is,

$$EAu''_c = 0$$

With boundary conditions,

$$u_c = 0$$
$$EAu'_c(L) = -F$$

Solving the ODE,

$$EAu'_c = C_1$$

$$EAu_c = C_1x + C_2$$

Substituting u_c=0,

$$EA(0) = C_1(0) + C_2$$
$$C_2 = 0$$

Substituting,

$$EAu'_c(L) = -F = C_1$$

Therefore,

$$EAu_c = -Fx$$
$$u_c = -\frac{Fx}{EA}$$
$$u(x) = u_c(x)e_1 = -\frac{Fx}{EA}e_1$$

The solution obtained here is different compared to the ones obtained in Ritz and Galerkin method. We had assumed a cubic polynomial as our initial displacement as we had no idea about how it varies with length. If instead we use $u_c(x) = cx$, then we would end up with the same solution as the analytical solution in Galerkin and Ritz method too.

Post processing and comparison between the methods:

Before that, we need to define a few parameters to be able to visualize the comparison.

Axial Force, F = 1000 N

The young's modulus, E = 200*10^3 N/mm^2

Area, A = 100 mm^2

Length of the beam, L = 1000 mm
Poisson's ratio v = 0.3

1. Displacements:

   We know, the following are the functions for the displacements.

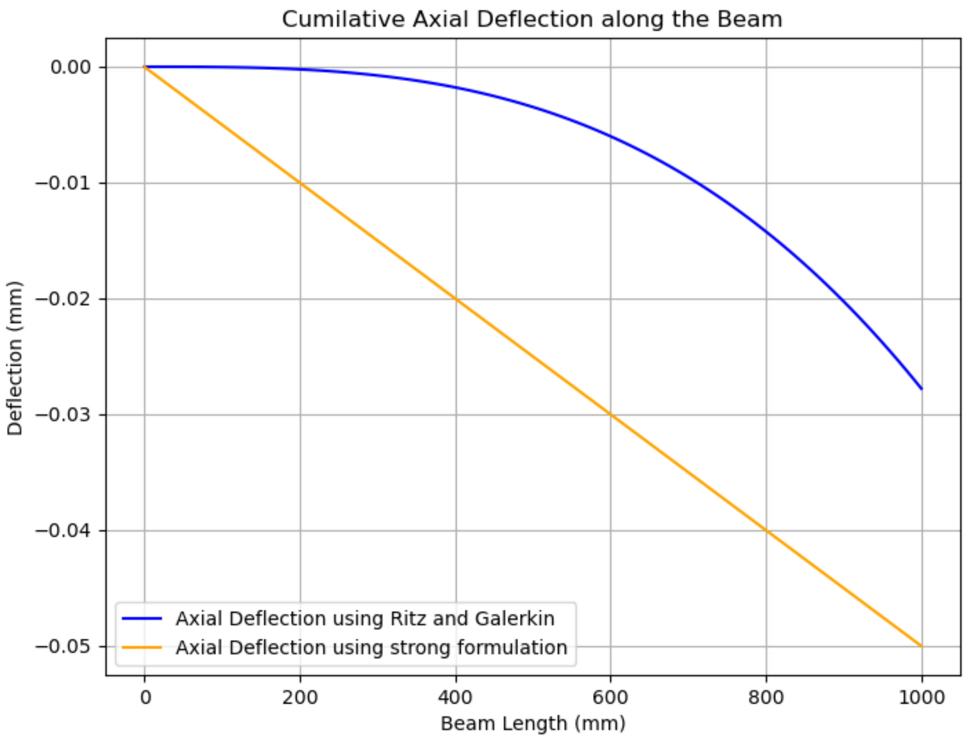
   For Ritz method and Galerkin method:

$$u(x) = u_c(x)e_1 = \frac{-5F}{9EAL^2}\, x^3 e_1$$

   For strong formulation:

$$u(x) = u_c(x)e_1 = -\frac{Fx}{EA} e_1$$

   Plotting the cumulative axial displacements along the length of the beam, we get the following:



   As we can see from the graph, choosing a function couple of orders away from the analytical solution obtained from the strong formulation introduced a significant error into the calculations. Better approximations to the initial function would help in reducing these errors.

2. Strains:
   Let us find the strain tensors for both cases:

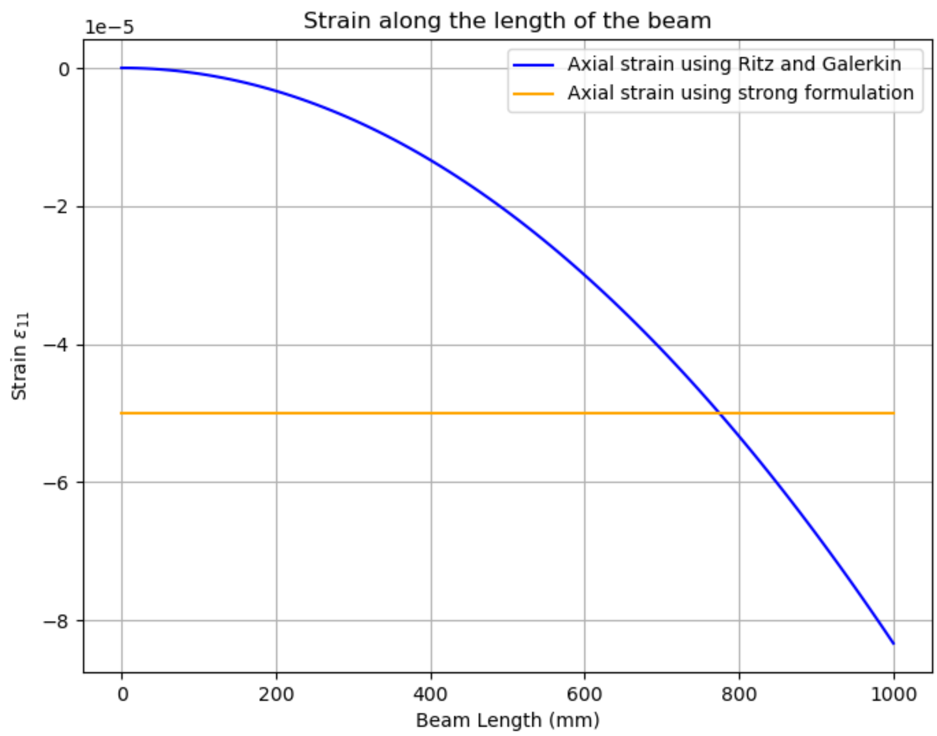
   Galerkin/Ritz method:

$$\epsilon = \begin{bmatrix} -\dfrac{5Fx^2}{3EAL^2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Strong formulation:

$$\epsilon = \begin{bmatrix} -\dfrac{F}{EA} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

We only have ϵ_{11}. Let us plot it along the length to find the differences.



We know that for a beam/bar with constant axial loading, the strain must be constant. This is verified using the strong formulation. However, when we used a function other than a linear function, we have caused the strains to vary at various points. Even though we didn't know what type of function to use earlier, at this point it should be clear that our displacements must be linear to get a constant strain along the length of the beam.

3. Stresses:
   In question 6(i), for a similar displacement field we derived the stress tensor.

$$\sigma = \frac{E\epsilon}{(1+v)(1-2v)} \begin{bmatrix} 1-v & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & v \end{bmatrix}$$

[We could have also used the approximation $\sigma_{11} = E\epsilon_{11}$ but for now let us use this expression]

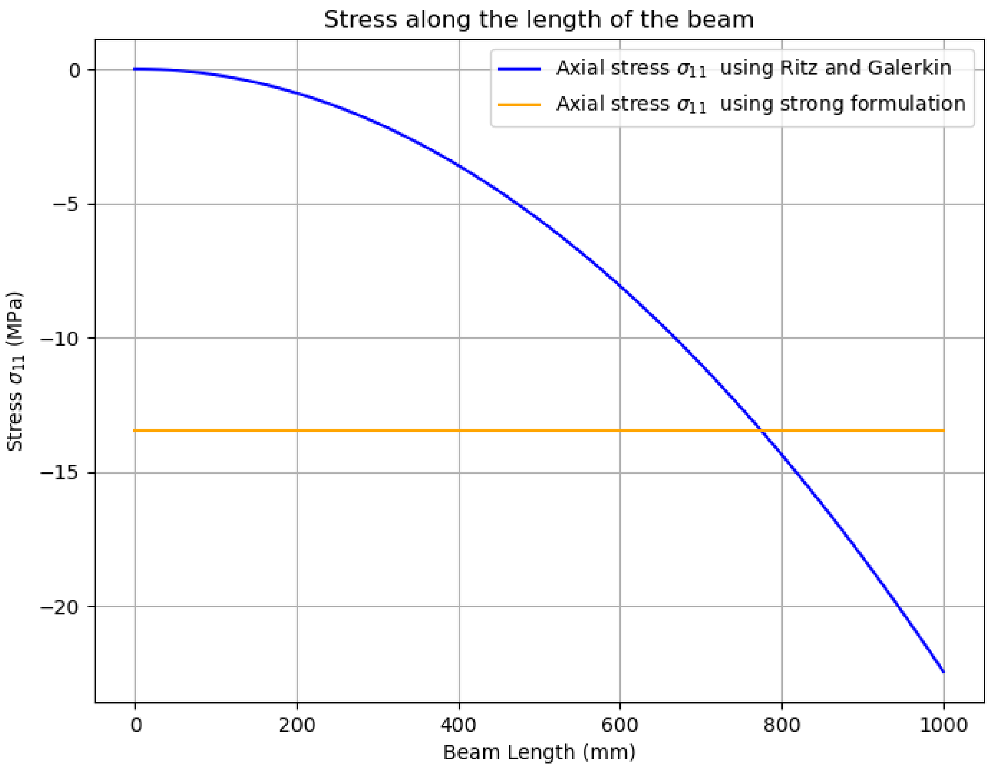
Stress tensor for Galerkin/Ritz method:

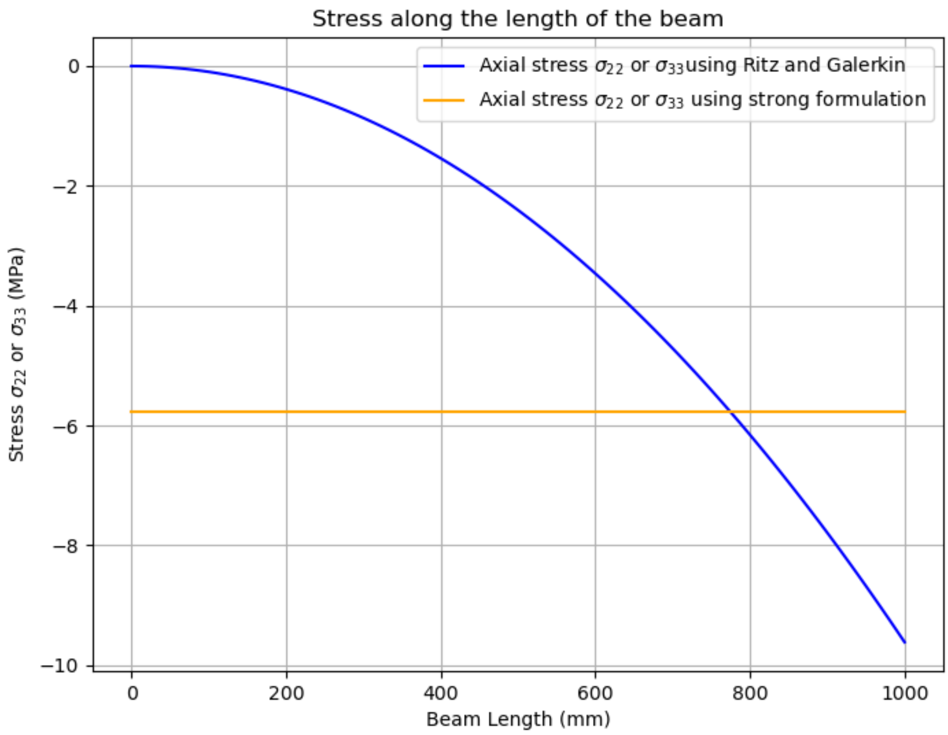
$$\sigma = \frac{\left(-\frac{5Fx^2}{3AL^2}\right)}{(1+v)(1-2v)}\begin{bmatrix}1-v & 0 & 0\\ 0 & v & 0\\ 0 & 0 & v\end{bmatrix}$$

Stress tensor for strong formulation:

$$\sigma = \frac{\left(-\frac{F}{A}\right)}{(1+v)(1-2v)}\begin{bmatrix}1-v & 0 & 0\\ 0 & v & 0\\ 0 & 0 & v\end{bmatrix}$$

Plotting σ_{11} along the length of the beam:

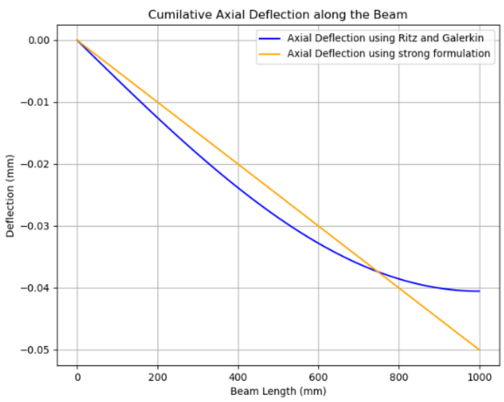
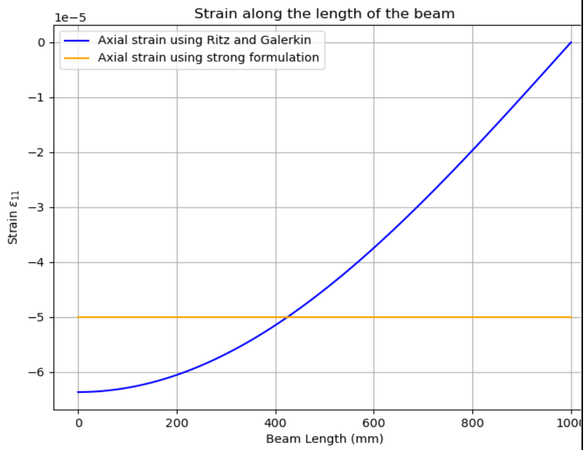


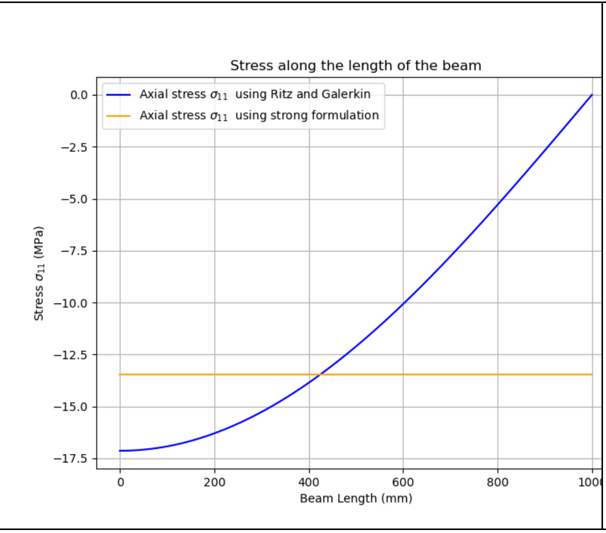


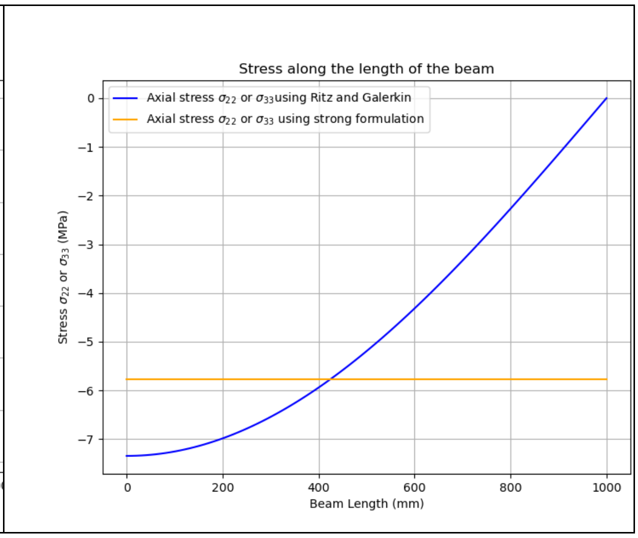
The trends for all three stresses are similar to strain. The differences are only due to the change in order of the initial assumption.

Also, a sinusoidal function was used separately to calculate and check the values with strong formulation compared with the one obtained using Ritz/Galerkin. The code used is shown in the drive to avoid cluttering here. The results are shown below briefly:

$$u_c(x) = c \sin\left(\frac{\pi x}{2L}\right)$$

Axial deflections	**Strain in 11 direction**
	
Stress in 11 direction	**Strain in 22 and 33 direction**





We have got a much better approximation of the deflection using the sinusoidal function. However, we still have the same issue with strain and stress.

Question 8:

Simply supported Kirchhoff-Love plate with uniform lateral load. [Note: For this exercise you may use results from the class notes. You should indicate the meaning of the symbols and quantities that you require for this problem and indicate in words the steps to obtain each formula.]

Consider a homogenous, isotropic linearly elastic plate of dimensions $a\ x\ b\ x\ h$ subjected to a uniform lateral load q_0 (i.e., pressure-like distributed force).

Sub question 1:

i. Complete the missing steps in the class notes regarding the solution to this problem using Galerkin's method (use the same single term approximation for the lateral displacement as used for Ritz' method). Is the solution from Galerkin's method the same as from Ritz' method or not?

The principle of virtual work for Kirchhoff-love plate with uniform lateral load is given as

$$\delta P = \delta U_b + \delta V = 0\ for\ all\ \delta w$$

δU_b: $Variation\ of\ the\ bending\ strain\ energy$

δV: $Variation\ of\ the\ potential\ of\ external\ lateral\ loading\ q(x, y)$

w: $w(x, y)\ is\ the\ lateral\ displacement$

δw: $the\ variation\ of\ the\ lateral\ displacement$ or the virtual displacement

$$\delta U_b[w] = \frac{D}{2}\int_0^a \int_0^b [2\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}\right)\left(\frac{\partial^2(\delta w)}{\partial x^2}+\frac{\partial^2(\delta w)}{\partial y^2}\right)$$
$$-2(1-v)(\frac{\partial^2 w}{\partial y^2}\frac{\partial^2(\delta w)}{\partial x^2}+\frac{\partial^2 w}{\partial x^2}\frac{\partial^2(\delta w)}{\partial y^2}-2\frac{\partial^2 w}{\partial x\partial y}\frac{\partial^2(\delta w)}{\partial x\partial y})]dydx$$

$$\delta V[w] = -\int_0^a \int_0^b q(x,y)\delta w dydx$$

Here, D is the bending stiffness and $D = \frac{Eh^3}{12(1-v^2)}$

$h: the\ height/depth\ of\ the\ plate$

$v: poisson's ratio$

$E: The\ young's\ modulus$

Using the same approximation used in Ritz's method,

$$w(x,y) = c\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$$

The approximation for the virtual displacement is given by,

$$\delta w(x,y) = d\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$$

$a: Length\ of\ the\ beam\ in\ x-direction$

$b: Length\ of\ the\ beam\ in\ y-direction$

$c: a\ constant/parameter\ that\ needs\ to\ be\ determined$

$d: a\ constant$

Computing the derivatives,

$$\frac{\partial^2 w}{\partial x^2} = -c\frac{\pi^2}{a^2}\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$$
$$\frac{\partial^2 w}{\partial y^2} = -c\frac{\pi^2}{b^2}\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$$
$$\frac{\partial^2 w}{\partial x\partial y} = c\frac{\pi^2}{ab}\cos\left(\frac{\pi x}{a}\right)\cos\left(\frac{\pi y}{b}\right)$$
$$\frac{\partial^2(\delta w)}{\partial x^2} = -d\frac{\pi^2}{a^2}\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$$
$$\frac{\partial^2(\delta w)}{\partial y^2} = -d\frac{\pi^2}{b^2}\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$$
$$\frac{\partial^2(\delta w)}{\partial x\partial y} = d\frac{\pi^2}{ab}\cos\left(\frac{\pi x}{a}\right)\cos\left(\frac{\pi y}{b}\right)$$
$$\delta P = \delta U_b + \delta V$$

$$\delta P=\frac{D}{2}\int_0^a\int_0^b[2\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}\right)\left(\frac{\partial^2(\delta w)}{\partial x^2}+\frac{\partial^2(\delta w)}{\partial y^2}\right)-2(1-v)\left[\left(\frac{\partial^2 w}{\partial y^2}\frac{\partial^2(\delta w)}{\partial x^2}+\frac{\partial^2 w}{\partial x^2}\frac{\partial^2(\delta w)}{\partial y^2}-2\frac{\partial^2 w}{\partial x\partial y}\frac{\partial^2(\delta w)}{\partial x\partial y}\right)\right]dydx-\int_0^a\int_0^b q(x,y)\delta w dydx$$

$$\delta P=\frac{D}{2}\int_0^a\int_0^b[2\left(\frac{c\pi^2}{a^2}+\frac{c\pi^2}{b^2}\right)\left(\frac{d\pi^2}{a^2}+\frac{d\pi^2}{b^2}\right)\sin^2\left(\frac{\pi x}{a}\right)\sin^2\left(\frac{\pi y}{b}\right)dydx-2(1-v)\left[\left(\frac{-c\pi^2}{b^2}\right)\left(\frac{-d\pi^2}{a^2}\right)\sin^2\left(\frac{\pi x}{a}\right)\sin^2\left(\frac{\pi y}{b}\right)+\left(\frac{-c\pi^2}{a^2}\right)\left(\frac{-d\pi^2}{b^2}\right)\sin^2\left(\frac{\pi x}{a}\right)\sin^2\left(\frac{\pi y}{b}\right)-2cd\frac{\pi^4}{a^2b^2}\cos^2\left(\frac{\pi x}{a}\right)\cos^2\left(\frac{\pi y}{b}\right)\right]dydx-q_0 d\int_0^a\int_0^b\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)dydx$$

$$\delta P=\frac{D}{2}\int_0^a\int_0^b[2\left(\frac{c\pi^2}{a^2}+\frac{c\pi^2}{b^2}\right)\left(\frac{d\pi^2}{a^2}+\frac{d\pi^2}{b^2}\right)\sin^2\left(\frac{\pi x}{a}\right)\sin^2\left(\frac{\pi y}{b}\right)dydx-2(1-v)\left[\left(\frac{2cd\pi^4}{a^2b^2}\right)\sin^2\left(\frac{\pi x}{a}\right)\sin^2\left(\frac{\pi y}{b}\right)-2cd\frac{\pi^4}{a^2b^2}\cos^2\left(\frac{\pi x}{a}\right)\cos^2\left(\frac{\pi y}{b}\right)\right]dydx-q_0 d\int_0^a\int_0^b\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)dydx$$

The integrals calculated are,

$$\int_0^a\int_0^b\sin^2\left(\frac{\pi x}{a}\right)\sin^2\left(\frac{\pi y}{b}\right)dydx=\frac{ab}{4}$$

$$\int_0^a\int_0^b\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)dydx=\frac{4ab}{\pi^2}$$

$$\int_0^a\int_0^b\cos^2\left(\frac{\pi x}{a}\right)\cos^2\left(\frac{\pi y}{b}\right)dydx=\frac{ab}{4}$$

$$\delta P=Dcd\pi^4\left(\frac{1}{a^2}+\frac{1}{b^2}\right)^2\frac{ab}{4}-2(1-v)\left[\left(\frac{2cd\pi^4}{a^2b^2}\right)\left(\frac{ab}{4}\right)-\left(\frac{2cd\pi^4}{a^2b^2}\right)\left(\frac{ab}{4}\right)\right]-q_0 d\frac{4ab}{\pi^2}=0$$

$$\delta P = Dcd\pi^4 \frac{\left(a^2+b^2\right)^2}{4a^3b^3} - q_0 d\frac{4ab}{\pi^2} = 0$$

$$\delta P = \left[Dc\pi^4 \frac{\left(a^2+b^2\right)^2}{4a^3b^3} - q_0 \frac{4ab}{\pi^2}\right] d = 0$$

For δP to be zero for all d values, the box needs to be zero.

$$Dc\pi^4 \frac{\left(a^2+b^2\right)^2}{4a^3b^3} - q_0 \frac{4ab}{\pi^2} = 0$$

$$c = \frac{16a^4b^4}{\pi^6(a^2+b^2)^2}\left(\frac{q_0}{D}\right)$$

$$w(x,y) = \frac{16a^4b^4}{\pi^6(a^2+b^2)^2}\left(\frac{q_0}{D}\right)\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$$

Sub question 2:

ii. Consider the solution given in the lecture notes in terms of Fourier series. Choose your own values for the plate dimensions, material properties and loading (clearly indicate all the values used, including units). Compute the maximum lateral deflection of the plate using the single term Galerkin solution for (i) and the the fourier series solutions using 1,2 and up to 3 terms (you can consider more if you want). What can you say about the error in the computed displacement in terms of the number of terms used? What about Galerkin's solution?

Let us assume the following values:

$$a = 1000\ mm$$

$$b = 1000\ mm$$

$$h = 100\ mm$$

$$v = 0.3$$

$$q_0 = 500\frac{N}{mm}$$

$$E = 70\ \ GPa$$

Calculating the bending stiffness D,

$$D = \frac{Eh^3}{12(1-v^2)} = \frac{70 * 10^3 * (100)^3}{12(1-0.3^2)} = 6.4102 * 10^9\ N - mm$$

From Galerkin solution computed in (i) we have,

$$w(x,y) = \frac{16a^4b^4}{\pi^6(a^2+b^2)^2}\left(\frac{q_0}{D}\right)\sin\left(\frac{\pi x}{a}\right)\sin\left(\frac{\pi y}{b}\right)$$

We know a sine function is maximum at 90 degrees. Therefore, x = a/2 and y =b/2

Therefore, the maximum lateral deflection would be

$$w = \frac{16 * 1000^4 * 1000^4}{\pi^6(1000^2 + 1000^2)^2}\left(\frac{500}{8.333 * 10^9}\right)$$

$$w = 324.5303\ mm$$

Now, using the strong formulation with Fourier series for determining the maximum lateral deflection for a constant load q_0.

$$w(x,y) = \sum_{m=1}^{\propto}\sum_{n=1}^{\propto} w^{(m,n)} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right)$$

$$w^{(m,n)} = \frac{1}{\pi^4 D}\frac{q^{(m,n)}}{\left[\left(\frac{m}{a}\right)^2 + \left(\frac{n}{b}\right)^2\right]^2}$$

$$q^{(m,n)} = \frac{16q_0}{mn\pi^2}\sin^2\left(\frac{m\pi}{2}\right)\sin^2\left(\frac{n\pi}{2}\right)$$

Considering 1 term:

The deflection is maximum at x = a/2 and y = b/2

$$w(x,y) = w^{(1,1)} = 324.5303\ mm$$

This matches exactly with the solution obtained from the Galerkin solution.

Considering 2 terms:

The deflection, we know, is maximum at x = a/2 and y = b/2

$$w(x,y) = w^{(1,1)} + w^{(2,2)} = 324.5303\ mm$$

This matches exactly with the solution obtained from the Galerkin solution as $w^{(2,2)}$ turned out to be zero.

Considering 3 terms:

The deflection, we know, is maximum at x = a/2 and y = b/2

$$w(x,y) = w^{(1,1)} + w^{(2,2)} + w^{(3,3)} = 324.9755\ mm$$

This deviates a little from the solution we obtained from the Galerkin solution using a single term. However, we can see that as we are adding more terms the solution is converging in strong form.

The strong formulation converges with 13 terms with 1e-4 accuracy. The code is attached in the annexure.

Similarly, adding terms for galerkin gives the same exact solution when compared with strong formulation for a particular number of terms used. It converges at 13 terms for 1e-4 accuracy. The code used for galerkin is also given in the annexure.

An observation can be made that whenever the no. of terms used is even, the formulation does not add any value (can also be seen in the above formulation for 2 terms) as it is the same as the previous iteration.

Another approach could be by using terms in the Fourier and Galerkin including m!=n terms. Though the strong formulation with Fourier series is relatively faster for running using Python, the Galerkin method was computationally running for 1.5 hours with m=n architecture. Computing that for m!=n, is expected to be at least a few hours of computational time. For this reason, this approach has not been pursued. Also, the question mentioned use 1,2 and 3 terms but if m!=n, then the next in the series after 1 term would be 4 terms.

Annexure - A:

Question 1: Python Code

```
import numpy as np
import math

stress = np.array([[150,110,70],[110,160,0],[70,0,-140]])
eigvals, eigvecs = np.linalg.eig(stress)
print("The eigen values are",eigvals)
print("The eigen vectors are", eigvecs)
```

Output:

```
The eigen values are [ 270.96576977   57.17515774 -158.1409275 ]
The eigen vectors are [[-0.70505075 -0.66365531 -0.24993014]
 [-0.69891447  0.70996543  0.08641553]
 [-0.12009164 -0.23560712  0.96439996]]
```

Post this, rearranging has been done to obtain the vectors in a right handed system, i.e., $\boldsymbol{n_1 X\, n_2 = n_3, n_2 X\, n_3 = n_1, n_3 X\, n_1 = n_2.}$

Question 2: Python code

Only the vertices of 2(i) question have been left uncommented here. Other vertices of the questions 2(ii) to 2(vi) are commented out and can beuncommented based on the need.

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.art3d import Poly3DCollection
from matplotlib.lines import Line2D

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Defining the vertices of the first cube
p1 = [0,0,0]
p2 = [1,0,0]
p3 = [1,1,0]
p4 = [0,1,0]
p5 = [0,0,1]
p6 = [1,0,1]
p7 = [1,1,1]
p8 = [0,1,1]
```

```python
vertices1 = [
    p1,p2,p3,p4,p5,p6,p7,p8
]

# Defining the faces of the first cube using the vertices
faces1 = [
    [vertices1[0], vertices1[1], vertices1[2], vertices1[3]],
    [vertices1[4], vertices1[5], vertices1[6], vertices1[7]],
    [vertices1[0], vertices1[1], vertices1[5], vertices1[4]],
    [vertices1[2], vertices1[3], vertices1[7], vertices1[6]],
    [vertices1[1], vertices1[2], vertices1[6], vertices1[5]],
    [vertices1[0], vertices1[3], vertices1[7], vertices1[4]]
]

#Question 2(i)
a = 0.2
p1_ = [x + y for x, y in zip(p1, [a*p1[0],0,0])]
p2_ = [x + y for x, y in zip(p2, [a*p2[0],0,0])]
p3_ = [x + y for x, y in zip(p3, [a*p3[0],0,0])]
p4_ = [x + y for x, y in zip(p4, [a*p4[0],0,0])]
p5_ = [x + y for x, y in zip(p5, [a*p5[0],0,0])]
p6_ = [x + y for x, y in zip(p6, [a*p6[0],0,0])]
p7_ = [x + y for x, y in zip(p7, [a*p7[0],0,0])]
p8_ = [x + y for x, y in zip(p8, [a*p8[0],0,0])]

# #Question 2(ii)
# a = 0.2
# b = 0.1
# p1_ = [x + y for x, y in zip(p1, [a*p1[0],-b*p1[1],-b*p1[2]])]
# p2_ = [x + y for x, y in zip(p2, [a*p2[0],-b*p2[1],-b*p2[2]])]
# p3_ = [x + y for x, y in zip(p3, [a*p3[0],-b*p3[1],-b*p3[2]])]
# p4_ = [x + y for x, y in zip(p4, [a*p4[0],-b*p4[1],-b*p4[2]])]
# p5_ = [x + y for x, y in zip(p5, [a*p5[0],-b*p5[1],-b*p5[2]])]
# p6_ = [x + y for x, y in zip(p6, [a*p6[0],-b*p6[1],-b*p6[2]])]
# p7_ = [x + y for x, y in zip(p7, [a*p7[0],-b*p7[1],-b*p7[2]])]
# p8_ = [x + y for x, y in zip(p8, [a*p8[0],-b*p8[1],-b*p8[2]])]

# #Question 2(iii)
# a = 0.2
# p1_ = [x + y for x, y in zip(p1, [-a*p1[0],-a*p1[1],-a*p1[2]])]
# p2_ = [x + y for x, y in zip(p2, [-a*p2[0],-a*p2[1],-a*p2[2]])]
# p3_ = [x + y for x, y in zip(p3, [-a*p3[0],-a*p3[1],-a*p3[2]])]
# p4_ = [x + y for x, y in zip(p4, [-a*p4[0],-a*p4[1],-a*p4[2]])]
# p5_ = [x + y for x, y in zip(p5, [-a*p5[0],-a*p5[1],-a*p5[2]])]
# p6_ = [x + y for x, y in zip(p6, [-a*p6[0],-a*p6[1],-a*p6[2]])]
# p7_ = [x + y for x, y in zip(p7, [-a*p7[0],-a*p7[1],-a*p7[2]])]
# p8_ = [x + y for x, y in zip(p8, [-a*p8[0],-a*p8[1],-a*p8[2]])]
```

```
# #Question 2(iv)
# a = 0.2
# p1_ = [x + y for x, y in zip(p1, [a*p1[1],0,0])]
# p2_ = [x + y for x, y in zip(p2, [a*p2[1],0,0])]
# p3_ = [x + y for x, y in zip(p3, [a*p3[1],0,0])]
# p4_ = [x + y for x, y in zip(p4, [a*p4[1],0,0])]
# p5_ = [x + y for x, y in zip(p5, [a*p5[1],0,0])]
# p6_ = [x + y for x, y in zip(p6, [a*p6[1],0,0])]
# p7_ = [x + y for x, y in zip(p7, [a*p7[1],0,0])]
# p8_ = [x + y for x, y in zip(p8, [a*p8[1],0,0])]

# #Question 2(v)
# a = 0.2
# p1_ = [x + y for x, y in zip(p1, [a*p1[1],a*p1[0],0])]
# p2_ = [x + y for x, y in zip(p2, [a*p2[1],a*p2[0],0])]
# p3_ = [x + y for x, y in zip(p3, [a*p3[1],a*p3[0],0])]
# p4_ = [x + y for x, y in zip(p4, [a*p4[1],a*p4[0],0])]
# p5_ = [x + y for x, y in zip(p5, [a*p5[1],a*p5[0],0])]
# p6_ = [x + y for x, y in zip(p6, [a*p6[1],a*p6[0],0])]
# p7_ = [x + y for x, y in zip(p7, [a*p7[1],a*p7[0],0])]
# p8_ = [x + y for x, y in zip(p8, [a*p8[1],a*p8[0],0])]

# #Question 2(vi)
# a = 0.2
# p1_ = [x + y for x, y in zip(p1,
[a*(p1[1]+p1[2]),a*(p1[0]+p1[2]),a*(p1[0]+p1[1])])]
# p2_ = [x + y for x, y in zip(p2,
[a*(p2[1]+p2[2]),a*(p2[0]+p2[2]),a*(p2[0]+p2[1])])]
# p3_ = [x + y for x, y in zip(p3,
[a*(p3[1]+p3[2]),a*(p3[0]+p3[2]),a*(p3[0]+p3[1])])]
# p4_ = [x + y for x, y in zip(p4,
[a*(p4[1]+p4[2]),a*(p4[0]+p4[2]),a*(p4[0]+p4[1])])]
# p5_ = [x + y for x, y in zip(p5,
[a*(p5[1]+p5[2]),a*(p5[0]+p5[2]),a*(p5[0]+p5[1])])]
# p6_ = [x + y for x, y in zip(p6,
[a*(p6[1]+p6[2]),a*(p6[0]+p6[2]),a*(p6[0]+p6[1])])]
# p7_ = [x + y for x, y in zip(p7,
[a*(p7[1]+p7[2]),a*(p7[0]+p7[2]),a*(p7[0]+p7[1])])]
# p8_ = [x + y for x, y in zip(p8,
[a*(p8[1]+p8[2]),a*(p8[0]+p8[2]),a*(p8[0]+p8[1])])]


# Plotting the first cube
cube1 = Poly3DCollection(faces1, linewidths=1, edgecolors='blue',
facecolors='white', alpha=0.1)
ax.add_collection3d(cube1)
```

```
vertices2 = [
    p1_,p2_,p3_,p4_,p5_,p6_,p7_,p8_
]


# Defining the faces of the second cube using the vertices
faces2 = [
    [vertices2[0], vertices2[1], vertices2[2], vertices2[3]],
    [vertices2[4], vertices2[5], vertices2[6], vertices2[7]],
    [vertices2[0], vertices2[1], vertices2[5], vertices2[4]],
    [vertices2[2], vertices2[3], vertices2[7], vertices2[6]],
    [vertices2[1], vertices2[2], vertices2[6], vertices2[5]],
    [vertices2[0], vertices2[3], vertices2[7], vertices2[4]]
]

# Plotting the second cube
cube2 = Poly3DCollection(faces2, linewidths=2, edgecolors='blue',
facecolors='white', alpha=0.1)
ax.add_collection3d(cube2)

#Arrows, legends and etc modifications
arrow_length = 0.4
ax.quiver(0, 0, 0, 0.4, 0, 0, color='black', arrow_length_ratio=0.25,
linewidth=3)
ax.quiver(0, 0, 0, 0, 0.4, 0, color='black', arrow_length_ratio=0.25,
linewidth=3)
ax.quiver(0, 0, 0, 0, 0, 0.4, color='black', arrow_length_ratio=0.25,
linewidth=3)
legend_elements = [
    Line2D([0], [0], color='blue', lw=1, label='Initial cube'),
    Line2D([0], [0], color='blue', lw=3, label='Deformed cube')
]
ax.legend(handles=legend_elements)
ax.set_xlim(0, 1.5)
ax.set_ylim(0, 1.5)
ax.set_zlim(0, 1.5)
ax.set_xticks([])
ax.set_yticks([])
ax.set_zticks([])
arrow_label_offset = 0.15
ax.text(arrow_length + arrow_label_offset, 0, 0, "X", color='black',
fontsize=10, ha='right')
ax.text(0, arrow_length + arrow_label_offset, 0, "Y", color='black',
fontsize=10, ha='right')
ax.text(0, 0, arrow_length + arrow_label_offset, "Z", color='black',
fontsize=10, ha='right')
plt.show()
```

Output: Shown in the question 2 itself for each case.

Question 3: Python code

```
import numpy as np
import sympy as sp
import math

P = np.array([2,5,7])
Q = np.array([3,8,9])
x1 = sp.symbols('x1',real = True)
x2 = sp.symbols('x2',real = True)
x3 = sp.symbols('x3',real = True)
u1 = (x1**2+20)*10**-4
u2 = (2*x1*x2)*10**-3
u3 = (x3**2-x1*x2)*10**-4

du1 = u1 + sp.diff(u1,x1)*(Q[0]-P[0])+sp.diff(u1,x2)*(Q[1]-
P[1])+sp.diff(u1,x3)*(Q[2]-P[2])
dup1 = u1.subs({x1: P[0],x2: P[1],x3: P[2]})
duq1 = du1.subs({x1: P[0],x2: P[1],x3: P[2]})
#duq1 = du1.subs({x1: Q[0],x2: Q[1],x3: Q[2]})
du2 = u2 + sp.diff(u2,x1)*(Q[0]-P[0])+sp.diff(u2,x2)*(Q[1]-
P[1])+sp.diff(u2,x3)*(Q[2]-P[2])
dup2 = u2.subs({x1: P[0],x2: P[1],x3: P[2]})
duq2 = du2.subs({x1: P[0],x2: P[1],x3: P[2]})
#duq2 = du2.subs({x1: Q[0],x2: Q[1],x3: Q[2]})
du3 = u3 + sp.diff(u3,x1)*(Q[0]-P[0])+sp.diff(u3,x2)*(Q[1]-
P[1])+sp.diff(u3,x3)*(Q[2]-P[2])
dup3 = u3.subs({x1: P[0],x2: P[1],x3: P[2]})
duq3 = du3.subs({x1: P[0],x2: P[1],x3: P[2]})
#duq3 = du3.subs({x1: Q[0],x2: Q[1],x3: Q[2]})
P_deformed = [x + y for x, y in zip(P, [dup1,dup2,dup3])]
Q_deformed = [x + y for x, y in zip(Q, [duq1,duq2,duq3])]
print("3 part a")
print("The deformed coordinates of P are " +str(P_deformed))
print("The deformed coordinates of Q are " +str(Q_deformed))

Distance_PQ = np.sqrt((Q[0]-P[0])**2+(Q[1]-P[1])**2+(Q[2]-P[2])**2)
print("The distance of P and Q in the undeformed body is " +str(Distance_PQ))

Distance_PQ_deformed = math.sqrt((Q_deformed[0]-
P_deformed[0])**2+(Q_deformed[1]-P_deformed[1])**2+(Q_deformed[2]-
P_deformed[2])**2)
print("The distance of P and Q in the deformed body is "
+str(Distance_PQ_deformed))

Relative_elongation = (Distance_PQ_deformed-Distance_PQ)/Distance_PQ
print("The relative elongation or the strain is given as "
+str(Relative_elongation))
```

```
strain_tensor = np.array([[sp.diff(u1,x1).subs({x1: P[0],x2: P[1],x3:
P[2]}),0.5*(sp.diff(u1,x2).subs({x1: P[0],x2: P[1],x3:
P[2]})+sp.diff(u2,x1).subs({x1: P[0],x2: P[1],x3: P[2]})),
0.5*(sp.diff(u1,x3).subs({x1: P[0],x2: P[1],x3:
P[2]})+sp.diff(u3,x1).subs({x1: P[0],x2: P[1],x3: P[2]}))],
                          [0.5*(sp.diff(u1,x2).subs({x1: P[0],x2: P[1],x3:
P[2]})+sp.diff(u2,x1).subs({x1: P[0],x2: P[1],x3:
P[2]})),sp.diff(u2,x2).subs({x1: P[0],x2: P[1],x3: P[2]}),
0.5*(sp.diff(u2,x3).subs({x1: P[0],x2: P[1],x3:
P[2]})+sp.diff(u3,x2).subs({x1: P[0],x2: P[1],x3: P[2]}))],
                          [0.5*(sp.diff(u1,x3).subs({x1: P[0],x2: P[1],x3:
P[2]})+sp.diff(u3,x1).subs({x1: P[0],x2: P[1],x3:
P[2]})),0.5*(sp.diff(u2,x3).subs({x1: P[0],x2: P[1],x3:
P[2]})+sp.diff(u3,x2).subs({x1: P[0],x2: P[1],x3: P[2]})),
sp.diff(u3,x3).subs({x1: P[0],x2: P[1],x3: P[2]})]])

print("The strain tensor at point P with coordinates (2,5,7) is \n" +
str(strain_tensor))

n_vector = [x - y for x, y in zip(Q, P)]
n_unit_vector = n_vector/np.linalg.norm(n_vector)
print("The unit vector n in direction of line PQ is given by \n"
+str(n_unit_vector))

Relative_elongation_method2 = 0
for i in range(0,3):
    for j in range(0,3):
        Relative_elongation_method2 += strain_tensor[i][j] * n_unit_vector[i]
* n_unit_vector[j]

print("The relative elongation or the strain using in class relation is given
as " +str(Relative_elongation_method2))

#part B:
print("3 part b")
P_b = np.array([2,5,7])
Q_b = np.array([2.13363,5.40089,7.26726])
du1_b = u1 + sp.diff(u1,x1)*(Q_b[0]-P_b[0])+sp.diff(u1,x2)*(Q_b[1]-
P_b[1])+sp.diff(u1,x3)*(Q_b[2]-P_b[2])
du2_b = u2 + sp.diff(u2,x1)*(Q_b[0]-P_b[0])+sp.diff(u2,x2)*(Q_b[1]-
P_b[1])+sp.diff(u2,x3)*(Q_b[2]-P_b[2])
du3_b = u3 + sp.diff(u3,x1)*(Q_b[0]-P_b[0])+sp.diff(u3,x2)*(Q_b[1]-
P_b[1])+sp.diff(u3,x3)*(Q_b[2]-P_b[2])
dup1_b = u1.subs({x1: P_b[0],x2: P_b[1],x3: P_b[2]})
duq1_b = du1_b.subs({x1: P_b[0],x2: P_b[1],x3: P_b[2]})
#duq1_b = du1_b.subs({x1: Q_b[0],x2: Q_b[1],x3: Q_b[2]})
dup2_b = u2.subs({x1: P_b[0],x2: P_b[1],x3: P_b[2]})
duq2_b = du2_b.subs({x1: P_b[0],x2: P_b[1],x3: P_b[2]})
```

```
#duq2_b = du2_b.subs({x1: Q_b[0],x2: Q_b[1],x3: Q_b[2]})
dup3_b = u3.subs({x1: P_b[0],x2: P_b[1],x3: P_b[2]})
duq3_b = du3_b.subs({x1: P_b[0],x2: P_b[1],x3: P_b[2]})
#duq3_b = du3_b.subs({x1: Q_b[0],x2: Q_b[1],x3: Q_b[2]})
P_deformed_b = [x + y for x, y in zip(P_b, [dup1_b,dup2_b,dup3_b])]
Q_deformed_b = [x + y for x, y in zip(Q_b, [duq1_b,duq2_b,duq3_b])]
print("The deformed coordinates of P are " +str(P_deformed_b))
print("The deformed coordinates of Q are " +str(Q_deformed_b))

Distance_PQ_b = np.sqrt((Q_b[0]-P_b[0])**2+(Q_b[1]-P_b[1])**2+(Q_b[2]-
P_b[2])**2)
print("The distance of P and Q in the undeformed body is "
+str(Distance_PQ_b))

Distance_PQ_deformed_b = math.sqrt((Q_deformed_b[0]-
P_deformed_b[0])**2+(Q_deformed_b[1]-P_deformed_b[1])**2+(Q_deformed_b[2]-
P_deformed_b[2])**2)
print("The distance of P and Q in the deformed body is "
+str(Distance_PQ_deformed_b))

Relative_elongation_b = (Distance_PQ_deformed_b-Distance_PQ_b)/Distance_PQ_b
print("The relative elongation or the strain is given as "
+str(Relative_elongation_b))

n_vector_b = [x - y for x, y in zip(Q_b, P_b)]
n_unit_vector_b = n_vector_b/np.linalg.norm(n_vector_b)
print("The unit vector n in direction of line PQ is given by \n"
+str(n_unit_vector_b))

Relative_elongation_method2_b = 0
for i in range(0,3):
    for j in range(0,3):
        Relative_elongation_method2_b += strain_tensor[i][j] *
n_unit_vector_b[i] * n_unit_vector_b[j]

print("The relative elongation or the strain using in class relation is given
as " +str(Relative_elongation_method2_b))
```

Output:

```
3 part a
The deformed coordinates of P are [2.00240000000000, 5.02000000000000, 7.00390000000000]
The deformed coordinates of Q are [3.00280000000000, 8.04200000000000, 9.00560000000000]
The distance of P and Q in the undeformed body is 3.7416573867739413
The distance of P and Q in the deformed body is 3.7603307101902623
The relative elongation or the strain is given as 0.004990655606878305
The strain tensor at point P with coordinates (2,5,7) is
[[0.000400000000000000 0.00500000000000000 -0.000250000000000000]
 [0.00500000000000000 0.00400000000000000 -0.000100000000000000]
 [-0.000250000000000000 -0.000100000000000000 0.00140000000000000]]
The unit vector n in direction of line PQ is given by
[0.26726124 0.80178373 0.53452248]
The relative elongation or the strain using in class relation is given as 0.0049857142857l429
3 part b
The deformed coordinates of P are [2.00240000000000, 5.02000000000000, 7.00390000000000]
The deformed coordinates of Q are [2.13608345200000, 5.42382986000000, 7.27138717100000]
The distance of P and Q in the undeformed body is 0.4999976765946023
The distance of P and Q in the deformed body is 0.502492992802726
The relative elongation or the strain is given as 0.004990655606879 6595
The unit vector n in direction of line PQ is given by
[0.26726124 0.80178373 0.53452248]
```

Question 6: Python code

```python
import matplotlib.pyplot as plt

E = 200
v = 0.3
# Stress and strain data
strain_11 = []  # Example strain values
stress_11_q1 = []  # Example stress values
stress_11_q2 = []
i = 0
j = 0
while j<1000:
    strain_11.append(j)
    j=j+0.1

for x in strain_11:
    stress_11_q1.append(E*x*(1-v)/((1+v)*(1-2*v))/1000)
    stress_11_q2.append(E*x/1000)
    i = i+1
# Plotting stress-strain curve
plt.figure(figsize=(8, 6))
plt.plot(strain_11, stress_11_q1, marker='o', linestyle='-', color = 'blue',
label = 'question 1')
plt.plot(strain_11, stress_11_q2, marker='o', linestyle='-', color = 'orange',
label = 'question 2')

# Title and labels
plt.title('Stress-Strain Curve')
plt.xlabel('Strain')
plt.ylabel('Stress (MPa)')
```

```
plt.legend()
# Grid and display
plt.grid(True)
plt.tight_layout()
plt.show()
```

Output: Plots shown in the respective questions

Question 8: Python Code

Strong formulation and single term galerkin

```
import numpy as np
a = 1000
b = 1000
h = 100
v = 0.3
q0 = 500
E = 70e3

D = E*h**3/(12*(1-v**2))
w_galerkin = 16 * (a**4) * (b**4) * q0 /(np.pi**6 * (a**2+b**2)**2 * D)
print("The solution using single term in Galerkin methods is " +
str(w_galerkin))
w_strong = 0

# for i in range(1,6):
#     for j in range(1,6):
#         q = 16 * q0 * (np.sin(i*np.pi/2))**2 * (np.sin(j*np.pi/2))**2 /(i *
j * np.pi**2)
#         w_strong += q/(np.pi ** 4 * D * ((i/a)**2+(j/b)**2)**2)

# print(w_strong)

for i in range(1,100):
    q = 16 * q0 * (np.sin(i*np.pi/2))**2 * (np.sin(i*np.pi/2))**2 /(i * i *
np.pi**2)
    w_old = w_strong
    w_strong += q/(np.pi ** 4 * D * ((i/a)**2+(i/b)**2)**2)
    print("The strong form displacement for " + str(i) + " terms is " +
str(w_strong))
    if i % 2 != 0 and w_strong - w_old < 1e-4:
        print("The solution converged at " + str(i) + " terms")
        break
    else:
        continue
```

Output:

```
The solution using single term in Galerkin methods is 324.53037966830595
The strong form displacement for 1 terms is 324.53037966830595
The strong form displacement for 2 terms is 324.53037966830595
The strong form displacement for 3 terms is 324.97555165687703
The strong form displacement for 4 terms is 324.97555165687703
The strong form displacement for 5 terms is 324.9963216011758
The strong form displacement for 6 terms is 324.9963216011758
The strong form displacement for 7 terms is 324.99908006388836
The strong form displacement for 8 terms is 324.99908006388836
The strong form displacement for 9 terms is 324.9996907250524
The strong form displacement for 10 terms is 324.9996907250524
The strong form displacement for 11 terms is 324.9998739139912
The strong form displacement for 12 terms is 324.9998739139912
The strong form displacement for 13 terms is 324.99994114896566
The solution converged at 13 terms
```

Galerkin method for multiple terms:

```
import sympy as sp
import numpy as np


a = 1000
b = 1000
h = 100
v = 0.3
q = 500
E = 70e3
D = E * h**3/(12*(1-v**2))
x = sp.symbols('x',real = True)
y = sp.symbols('y',real = True)

number_terms = 5
w_old = np.array([0])
for i in range(1,100):
    number_terms = i
    for i in range(number_terms+1):
        variables_c = [sp.symbols(f'c{i}') for i in range(1, number_terms +
1)]
        variables_d = [sp.symbols(f'd{i}') for i in range(1, number_terms +
1)]

    w = 0
    dw = 0

    for i in range(number_terms+1):
        w += variables_c[i-1] * sp.sin(i*np.pi*x/a) * sp.sin(i*np.pi*y/b)
        dw += variables_d[i-1] * sp.sin(i*np.pi*x/a) * sp.sin(i*np.pi*y/b)

    w11 = sp.diff(sp.diff(w,x),x)
    w22 = sp.diff(sp.diff(w,y),y)
    w12 = sp.diff(sp.diff(w,x),y)
    dw11 = sp.diff(sp.diff(dw,x),x)
    dw22 = sp.diff(sp.diff(dw,y),y)
    dw12 = sp.diff(sp.diff(dw,x),y)
```

```
    variation_U = D*((w11+w22)*(dw11+dw22)-(1-v)*(w22*dw11+w11*dw22-
2*w12*dw12))
    variation_U_integrate = sp.integrate(sp.integrate(variation_U,
(y,0,b)),(x,0,a))
    variation_V = - q*dw
    variation_V_integrate = sp.integrate(sp.integrate(variation_V,
(y,0,b)),(x,0,a))
    #variation_V_integrate = variation_V_integrate.args[0][0].as_expr()
    variation_PE_integrate = variation_U_integrate + variation_V_integrate
    equation = sp.Eq(variation_PE_integrate,0)
    equation = sp.simplify(equation)

    equations = []
    expr = equation.lhs
    for d in variables_d:
        equations.append(sp.collect(expr,d).coeff(d))
    c_galerkin = sp.linsolve(equations,variables_c)
    list_from_finite_set = list(c_galerkin)
    c_galerkin = list_from_finite_set[0]
    w_galerkin = sum(element for element in c_galerkin)
    print("The Galerkin displacement for " + str(i) + " terms is " +
str(w_galerkin))
    if i % 2 != 0 and w_galerkin - w_old < 1e-4:
        print("The solution converged at " + str(i) + " terms")
        break
    else:
        w_old = w_galerkin
        continue
```

Output:

```
The Galerkin displacement for 1 terms is 324.530379668306
The Galerkin displacement for 2 terms is 324.530379668306
The Galerkin displacement for 3 terms is 324.975551656877
The Galerkin displacement for 4 terms is 324.975551656877
The Galerkin displacement for 5 terms is 324.996321601176
The Galerkin displacement for 6 terms is 324.996321601176
The Galerkin displacement for 7 terms is 324.999080063888
The Galerkin displacement for 8 terms is 324.999080063888
The Galerkin displacement for 9 terms is 324.999690725052
The Galerkin displacement for 10 terms is 324.999690725052
The Galerkin displacement for 11 terms is 324.999873913991
The Galerkin displacement for 12 terms is 324.999873913991
The Galerkin displacement for 13 terms is 324.999941148965
The solution converged at 13 terms
```