**ME 354**

Beam Bending Lab Write-Up

Konsta Jäske

Section AB

## Numerical Analysis

This lab involves the numerical and experimental analysis of two 6061-T6 aluminum beams subject to 3- and 4-point bending. In it, rectangular beams of length , width and height are supported using pin supports at the two ends of the beam, wherein the displacement at these supports in the longitudinal and transverse directions is zero. A load is applied to their midsection using a turnbuckle, which can vary the applied force by the screwing/unscrewing action of its attached hooks. Four different loads were applied to the beams in increments of up to as measured using a load call at one of the end supports and the resulting deflection was measured using a digital dial indicator placed at the midpoint of the beam. Strain was measured at four distinct points along the beam using a strain gauge rosette for a total of twelve strain gauge measurements on each beam.

### 3-Point Bending Analysis

#### Problem Setup

A single yoke setup was used to apply a displacement to the center of the beam in the 3-point bending setup as illustrated in Figure 1. This illustration also includes all the dimensions of the beam and the positions and orientations of the strain gauges. The corresponding dimensions of the beam as obtained during the experiment are shown in Table 1.

A picture containing diagram

Description automatically generated

**A picture containing clock

Description automatically generated**

**Figure 1**: Beam in a 3-point bending configuration.

**Table 1**: Dimensions of the 3-point bending beam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Length *L* | 0.7 m ± 0.00025 m | | | |
| Width *W* | 0.03835 m ± 0.000025 m | | | |
| Height *H* | 0.02574 m ± 0.000025 m | | | |
| Second moment of inertia *I* | 5.450166E-8 m^4 ± 0.02E-8 m^4 | | | |
| Strain rosette (top) 1, *G*1, **a**b**c | 0.35 m ± 0.0001m | 0° | 45° | 90° |
| Strain rosette (side) 2, *G*2, **a**b**c | 0.375 m ±  0.0001m | 0° | 45° | 90° |
| Strain rosette (side) 2, *Y* | 0 m | | | |
| Strain rosette (bottom) 3, *G*3, **a**b**c | 0.35 m ±  0.0001m | 0° | 45° | 90° |
| Strain rosette (top) 4 , *G*4, **a**b**c | 0.45 m ± 0.0001m | -45° | 0° | 45° |
| Loading position of 3-point bending, *X*load | 0.35 m | | | |

#### Beam Bending Theory

The bending stiffness of a beam is the product of its Young’s modulus and area moment of inertia . For the beams here with a rectangular cross-section, width and height , the area moment of inertia is found to be

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 1 |

For a pinned-pinned beam in a 3-point bending configuration with a point load applied at the midpoint, there is a corresponding reaction force from each of the two supports with a magnitude , as indicated in Figure 1. The corresponding shear forces and moments at each of the strain gauge positions in the beam are

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 2 |

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 3 |

The maximum deflection occurs at the center of the beam with a magnitude

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 4 |

The stresses in the beam are a direct function of the shear forces and the bending moments. In the loading configuration shown in Figures 1, there is a compressive stress on the top side and a tensile stress on the bottom side of the beam. The stresses that arise due to the bending moment and shear force are

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 5 |

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 6 |

The corresponding axial and shear stresses at each of the strain gauge positions are given in Table 2.

**Table 2**: Axial and shear stresses in a 3-point bent beam under 400 N loading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position** | **1** | **2** | **3** | **4** |
|  |  | 0 MPa | 16.530.08 MPa | -11.890.08 MPa |
|  | 0 MPa | -0.304 0.002 MPa | 0 MPa | 0 MPa |

#### Strain Predictions

The strains can be calculated directly from the stresses using the Young’s modulus and Poisson’s ratio of the materials, which are and respectively, using Hooke’s law as

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 7 |

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 8 |

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 9 |

The corresponding axial, transverse and shear strains at each of the gauge positions are given in Table 3.

**Table 3**: Axial, transverse and shear strains in a 3-point bent beam under 400 N loading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position** | **1** | **2** | **3** | **4** |
|  |  | 0 | 2.39E-4 1.16E-6 | -1.72E-4 1.16E-6 |
|  | 7.91E-5 3.83E-7 | 0 | -7.91E-5 3.83E-7 | 5.69E-5 3.83E-7 |
|  | 0 | -1.17E-5 7.71E-8 | 0 | 0 |

For each strain gauge position, there are three strain gauges oriented at different angles. For instance, at position one the strain gauges 1a, 1b and 1c are oriented at 0°, 45° and 90° from the x-axis. In order to compare the strain valued obtained from measurement, the transformation must be done. The equation to rotate the strain is

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 10 |

The 12 strain gauges under the different applied loads are calculated and provided in Table 7.

### 4-Point Bending Analysis

#### Problem Setup

A double yoke setup was used to apply a displacement to the center of the beam in the 4-point bending setup as illustrated in Figure 2. This illustration also includes all the dimensions of the beam and the positions and orientations of the strain gauges. The corresponding dimensions of the beam as obtained during the experiment are shown in Table 4.

A picture containing diagram

Description automatically generated

**Figure 2**: Beam in a 4-point bending configuration.

**Table 4**: Dimensions of the 4-point bending beam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Length *L* | 0.7 m ± 0.00025 m | | | |
| Width *W* | 0.03835 m ± 0.000025 m | | | |
| Height *H* | 0.02574 m ± 0.000025 m | | | |
| Second moment of inertia *I* | 5.450166E-8 m^4 ± 0.02E-8 m^4 | | | |
| Strain rosette (top) 1, *G*1, **a**b**c | 0.35 m ± 0.0001m | 0° | 45° | 90° |
| Strain rosette (side) 2, *G*2, **a**b**c | 0.375 m ±  0.0001m | 0° | 45° | 90° |
| Strain rosette (side) 2, *Y* | 0 m | | | |
| Strain rosette (bottom) 3, *G*3, **a**b**c | 0.35 m ±  0.0001m | 0° | 45° | 90° |
| Strain rosette (top) 4 , *G*4, **a**b**c | 0.45 m ± 0.0001m | -45° | 0° | 45° |
| Loading position of 4-point bending, *X*load | *X*load1 | | | |

#### Beam Bending Theory

For a pinned-pinned beam in a 4-point bending configuration with a load distributed over two loading points, there is a corresponding reaction force from each of the two supports with a magnitude , as indicated in Figure 2. The corresponding shear forces and moments at each of the strain gauge positions in the beam are

|  |  |  |
| --- | --- | --- |
|  | 0  0  0 | Eq. 11 |

|  |  |
| --- | --- |
|  |  |

The maximum deflection occurs at the center of the beam with a magnitude

|  |  |  |
| --- | --- | --- |
|  |  | Eq. 13 |

Similar to the 4-point beam, there is a compressive stress on the top side and a tensile stress on the bottom side of the beam. The stresses that arise due to the bending moment and shear force are analogous to those for a 3-point beam. The corresponding axial and shear stresses at each of the strain gauge positions for the 4-point beam are given in Table 5.

**Table 5**: Axial and shear stresses in a 4-point bent beam under 400 N loading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position** | **1** | **2** | **3** | **4** |
|  | -9.45 | 0 | 9.45 | -9.45 |
|  | 0 | 0 | 0 | 0 |

#### Strain Predictions

The strains in a 4-point beam can be calculated analogously using Hooke’s law to those of a 3-point beam. The axial, transverse and shear strains at each of the gauge positions are given in Table 3.

**Table 6**: Axial, transverse and shear strains in a 4-point bent beam under 400 N loading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Position** | **1** | **2** | **3** | **4** |
|  |  | 0 |  |  |
|  | 4.52E-5 | 0 | -4.52E-5 | 4.52E-5 |
|  | 0 | 0 | 0 | 0 |

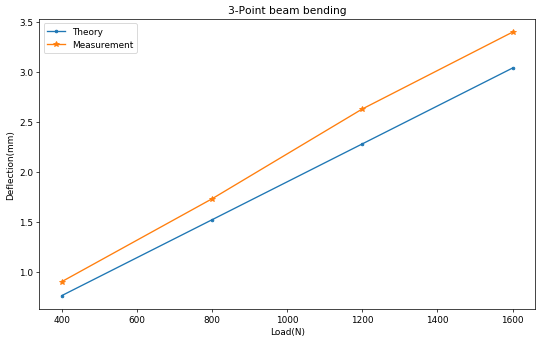
The strain transformation equations are identical to those for a 3-point beam. The 12 strain gauges under the different applied loads are calculated and provided in Table 7.

## Experimental Results

In this section, the measurement data obtained experimentally for each strain gauge will be compared to the theoretically predicted strain value.

### 3-Point Bending Experiment

Data are presented here from the 3-point bending values provided in the datasheet for the lab. The theoretical and experimental deflection at the midpoint are shown plotted in Figure 3, and the full set of theoretical and experimental strain gauge results are given in Table 7.



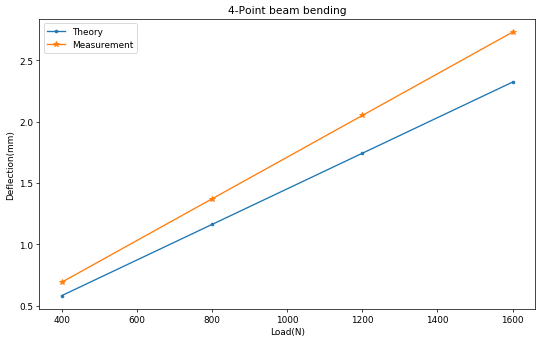
**Figure 3**: Theoretical vs. experimental deflection of a beam in 3-point bending.

**Table 7**: Theoretical vs experimental strain gauge measurements for 3-point bending.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1st Load – P=400 N** | | | **2nd Load – P=800 N** | | | **3rd Load – P=1200 N** | | | **4th Load – P=1600 N** | | |
| Strain Gauge Rosette (με) | Exp | Theory | Diff (%) | Exp | Theory | Diff (%) | Exp | Theory | Diff (%) | Exp | Theory | Diff (%) |
| 1a | -235 | -239 | 1.67 | -460 | -479 | 4.00 | -708 | -718 | 1.40 | -923 | -958 | 3.65 |
| 1b | -78 | -80.3 | 2.9 | -153 | -160 | 4.40 | -228 | -240 | 5.00 | -291 | -321 | 9.35 |
| 1c | 89 | 79.1 | 12 | 176 | 158 | 11.4 | 270 | 237 | 13.9 | 354 | 316 | 12.0 |
| 2a | -2 | 0 | N/A | -2 | 0 | N/A | -10 | 0 | N/A | -5 | 0 | N/A |
| 2b | -1 | -5.86 | 83 | -2 | -11.7 | 83 | 2 | -17.5 | 110 | 8 | -23.4 | 130 |
| 2c | 1 | 0 | N/A | 1 | 0 | N/A | 3 | 0 | N/A | 1 | 0 | N/A |
| 3a | 234 | 239 | 2.09 | 459 | 479 | 4.20 | 707 | 718 | 1.50 | 925 | 958 | 3.45 |
| 3b | 85 | 80.3 | 5.9 | 167 | 160 | 4.38 | 250 | 240 | 4.17 | 321 | 321 | 0 |
| 3c | -85 | -79.1 | 7.5 | -167 | -158 | 5.70 | -258 | -237 | 8.86 | -337 | -316 | 6.65 |
| 4a | -49 | -57.3 | 15 | -96 | -115 | 16 | -156 | -172 | 9.30 | -206 | -229 | 10.0 |
| 4b | -170 | -171 | 0.58 | -333 | -342 | 2.60 | -511 | -513 | 0.390 | -667 | -684 | 2.49 |
| 4c | -61 | -57.3 | 6.5 | -121 | -115 | 5.22 | -178 | -171 | 4.09 | -228 | -229 | 0.437 |

### 4-Point Bending Experiment

Data are presented here from the 4-point bending values provided in the datasheet for the lab. The theoretical and experimental deflection at the midpoint are shown plotted in Figure 4, and the full set of theoretical and experimental strain gauge results are given in Table 8.



**Figure 4**: Theoretical vs. experimental deflection of a beam in 4-point bending.

**Table 8**: Theoretical vs experimental strain gauge measurements for 4-point bending.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1st Load – P=400 N** | | | **2nd Load – P=800 N** | | | **3rd Load – P=1200 N** | | | **4th Load – P=1600 N** | | |
| Strain Gauge Rosette (με) | Exp | Theory | Diff (%) | Exp | Theory | Diff (%) | Exp | Theory | Diff (%) | Exp | Theory | Diff (%) |
| 1a | -134 | -137 | 2.19 | -263 | -274 | 4.01 | -395 | -411 | 3.89 | -531 | -548 | 3.10 |
| 1b | -34 | -45.9 | 26 | -69 | -91.7 | 25 | -106 | -138 | 23.2 | -146 | -183 | 20.2 |
| 1c | 49 | 45.2 | 8.4 | 96 | 90.3 | 6.3 | 144 | 136 | 5.88 | 193 | 181 | 6.63 |
| 2a | 1 | 0 | N/A | 2 | 0 | N/A | 2 | 0 | N/A | 4 | 0 | N/A |
| 2b | 8 | 0 | N/A | 14 | 0 | N/A | 18 | 0 | N/A | 21 | 0 | N/A |
| 2c | 0 | 0 | N/A | 1 | 0 | N/A | 1 | 0 | N/A | 2 | 0 | N/A |
| 3a | 134 | 137 | 2.19 | 265 | 274 | 3.28 | 398 | 411 | 3.16 | 535 | 548 | 2.37 |
| 3b | 35 | 45.9 | 24 | 71 | 91.7 | 23 | 110 | 138 | 20.3 | 151 | 183 | 17.5 |
| 3c | -48 | -45.2 | 6.2 | -94 | -90.3 | 4.1 | -142 | -135 | 5.19 | -190 | -181 | 4.97 |
| 4a | -35 | -45.9 | 24 | -71 | -91.7 | 23 | -109 | -138 | 21.0 | -150 | -183 | 18.0 |
| 4b | -142 | -137 | 4.01 | -278 | -274 | 1.46 | -418 | -411 | 1.70 | -561 | -548 | 2.37 |
| 4c | -58 | -45.9 | 26 | -113 | -91.7 | 23.2 | -167 | -138 | 21.0 | -220 | -183 | 20.2 |

## Discussion

The experimental results overall had a mixed quantitative agreement with the numerical predictions. There exist several possible reasons that may explain the discrepancies observed in both the 3-point and 4-point bending test results and numerical predictions. In the application of the Euler-Bernoulli theory the applied load is assumed to be a point load, whereas in reality the applied load is distributed over the contact area that the beam shares with the bracket attached to the turnbuckle. This inconsistency is not accounted for in my calculations and may skew the numerical predictions from the test results. When applying the Euler-Bernoulli theory to calculate the strains in the beam the strain directions and orientations are perfectly consistent with the established beam axes, while in the experiment the gauges were perhaps not perfectly aligned with the beam axes. This would cause discrepancy between the experimental and numerical data for every measurement. Categorically, the experimental applied loads did not match the theoretically applied loads. While the theoretical loads were exactly 400 N, 800 N, 1200 N, and 1600 N, the load cell readings for the experimental results indicate that in reality the applied load was either several Newtons above or below the theoretically prescribed loading magnitude. This could also cause significant differences between the experimental and numerical results. In attaining the theoretical strain values, the beam dimensions were assumed to be exact. In practice the beam dimensions may vary from the assumed dimensions and are also subject to measurement uncertainty due to the inaccuracy inherent to the measuring equipment.

### 3-Point Bending

For the 3-point bending case the greatest discrepancies between experimental and numerical results were found in the 1c, 2b, and 4a gauges. The experimental and numerical results for each differed by an average of 12.3%, 101%, and 12.6% respectively wherein each numerically obtained result was greater than its corresponding experimental result. Under application of the Euler-Bernoulli beam theory it is assumed that the beam is stiffer than it really is, as shear deformations inside of the beam are neglected. This assumption must be what causes the large divergence in the experimental and numerical results for the 2b gauge. With Euler-Bernoulli theory we assume that the cross-sections of the beam remain perpendicular to the neutral axis, so the overall angle change and shear strain for each point on the beam is greater in theory than in reality. It is a mystery to me why the 4c gauge exhibited greater strain than the 4a gauge throughout all of the 3-point bending loads, as in theory the two readings should be similar. Perhaps this indicates that strain rosette 4 was not aligned properly to the beam axes.

### 4-Point Bending

For the 4-point bending case the greatest discrepancies between the experimental and numerical results were in the 1b, 3b, 4a, and 4c gauges. The experimental and numerical results for each differed by an average of 23.6%, 21.2%, 21.5%, and 22.6% respectively. The theoretical results of gauges 1b, 3b, and 4a all overestimated the strain in comparison to the experimental results, while gauge 4c underestimated the strains. The gauges 1b, 3b, and 4c are all oriented in the same direction while 4a is symmetrically opposite to them about the beam x-axis; this makes it unsurprising that the percent difference for each gauge is roughly similar to the other. To obtain the theoretical figures for this testing configuration I used Euler-Bernoulli theory which tends to overestimate the stiffness of the beam. A possible reason this theory overestimated the 1b, 3b, and 4a strains is that the theory does not account for shear effects inside the beam, so theoretically inside the beam there are less crystal slip planes which results in a greater shear strain than if these slip planes were accounted for through shear effects.

Appendix

## Geometry dimensions, Two setups

#### 3-point beam bending



L3 = 0.7 #3-point length

b3 = 0.03835 #3-point width

h3 = 0.02574 #3-point height

#### 4-point beam bending



L4 = 0.7

b4 = 0.03835

h4 = 0.02574

a = 0.2

b = 0.5

c = 0.15

#### Material Properties



E = 69000e6 # Elastic Modulus

v = 0.33 # Poisson's Ratio

Y = 275e6 # Yield Strength

## Calculate the second moment of inertia I



#This is a function

def areaMomentOfInertia(b,h):

I = b\*h\*\*3/12

return I

### Max Deflection



# 3-point max deflection

def threePointDeflection(P,E,I,L):

w = P\*L\*\*3 /(48\*E\*I)

return w

​

# 4-point max deflection

def fourPointDeflection(P,E,I,L,a):

w = P\*a\*(3\*L\*\*2-4\*a\*\*2)/(48\*E\*I)

return w

​

#Calculate deflection for 3-point beam

load = [400 , 800, 1200, 1600] # input a list of the loads in Newtons i.e. [L0, L1, L2,...],

#Notice that the values in excel file is the reaction force, make sure the load is 2 times of the reaction force

I3 = areaMomentOfInertia(b3,h3) #calculate the 2nd moment of area for 3-point beam

deflection3 = [threePointDeflection(P,E,I3,L3)\*1000 for P in load]

print(deflection3)

​

#Calculate deflection for 4-point beam

I4 = areaMomentOfInertia(b4,h4)

deflection4 = [fourPointDeflection(P,E,I4,L4,a)\*1000 for P in load]

print(deflection4)



import matplotlib.pyplot as plt

​

# Deflection data from measurement

D3 = [0.90,1.73,2.63,3.4] #input a list of deflection measurements Data from excel file

D4 = [0.69,1.37,2.05,2.73]

​

#Plot all the things

fig = plt.figure(1,figsize=(10,6),dpi = 64) #initiate the figure

ax = fig.gca() #get the axes

ax.plot(load,deflection3,'.-',label = 'Theory') #plot the theoretical deflection

ax.plot(load,D3,'\*-',label = 'Measurement') #plot the experimental deflection

#note: if you want to use error bars then run the following

#ax.errorbar(load,deflection3,yerr=[listOfYErrors],label = 'Theory')

ax.legend()

ax.set\_xlabel('Load(N)')

ax.set\_ylabel('Deflection(mm)')

plt.title('3-Point beam bending')

​

fig = plt.figure(2,figsize=(10,6),dpi = 64)

ax = fig.gca()

ax.plot(load,deflection4,'.-', label = 'Theory')

ax.plot(load,D4,'\*-', label = 'Measurement')

ax.legend()

ax.set\_xlabel('Load(N)')

ax.set\_ylabel('Deflection(mm)')

plt.title('4-Point beam bending')

### Here are the Functions Shear and Moment in Bending

#### For each strain gauge rosette, we have two dimension values x and z



#Shear force for 3-point bending

def shearForce3(P,L,x):

if x<= L/2:

Q = P/2

else:

Q = -P/2

return Q

​

# Moment M for 3-points

def bendingMoment3(P,L,x):

if x<= L/2:

M = P\*x/2

else:

M = P\*(L-x)/2

return M

​

##### Write your own function for 4-point bending shear force and moment here #####

L4 = 0.7

b4 = 0.03835

h4 = 0.02574

a = 0.2

b = 0.5

c = 0.15

​

def shearForce4(P,L,x):

if x<= a:

Q = P/2

elif x<= L-a and x>a:

Q = 0

elif x<=L and x>L-a:

Q = -P/2

return Q

​

def bendingMoment4(P,L,x):

if x<=a:

M = (P\*x)/2

elif x>a and x<=L-a:

M = (P\*a)/2

elif x>L-a and x<=L:

M = (P\*(x-L)\*0.5)

return M

​

# Axial stress by moment

def axialStress(M,I,z):

sigma = M\*z/I

return sigma

# Shear stress by Shear force, since we only care about the top and bottom surface and neutral axix of the beam

def shearStress(Q,b,h,z):

if z == 0:

tau = 3\*Q/(2\*b\*h)

else:

tau = 0

return tau

### Rotation Function for Strain



import numpy as np

def strain1Rotate(epsilon,th):

'''Rotate a strain epsilon =[e\_x,e\_y,gamma\_xy] by an angle th given in radians.'''

T = [cos(th)\*\*2, sin(th)\*\*2, sin(th)\*cos(th)]

return np.dot(T,epsilon)

## 3 - Point Bending Strain Calculations

### Rosette 1



rosettePosition = [0.35,-h3/2] #[x\_coordinate, z\_coordinate] this is rosette 1, note that z value equals to zero at neutral axis

Loads =[400,800,1200,1600]

for P in Loads:

Q = shearForce3(P,L3,rosettePosition[0])

M = bendingMoment3(P,L3,rosettePosition[0])

sig = axialStress(M,I3,rosettePosition[1])

tau = shearStress(Q,b3,h3,rosettePosition[1])

​

from numpy import sin,cos,pi

​

# Already have stresses values at each strain rosette, now we can compute the strains using Hooke's law

C = np.array([[1/E, -v/E,0],[-v/E,1/E,0],[0,0,2\*(1+v)/E]]) #stiffness tensor

​

# Let's take strain rosette 1 as example. We need both axial strains and the shear strain

epsilon = np.dot(C,np.array([sig,0,tau])) # No stress on y direction

​

# Now transform from coordinate strains to the direction of strain gauge

ros1 = [0,45,90] #all three gauge directions

e1a = strain1Rotate(epsilon,ros1[0]\*pi/180)

e1b = strain1Rotate(epsilon,ros1[1]\*pi/180)

e1c = strain1Rotate(epsilon,ros1[2]\*pi/180)

epsilon\_rosette1 = np.array([e1a,e1b,e1c])

​

#This can also be done using list comprehensions as

epsilon\_rosette1Alt = np.array([strain1Rotate(epsilon,th\*pi/180) for th in ros1])

​

# Since the unit of measurement is micron strain, the results here need to be multiplied by 1e6

epsilon\_rosette1 \*= 1e6 #this multiplies the original value by 1e6

epsilon\_rosette1Alt \*= 1e6

print(epsilon\_rosette1)

### Rosette 2



rosettePosition = [0.375,0] #[x\_coordinate, z\_coordinate] this is rosette 1, note that z value equals to zero at neutral axis

Loads =[400,800,1200,1600]

for P in Loads:

Q = shearForce3(P,L3,rosettePosition[0])

M = bendingMoment3(P,L3,rosettePosition[0])

sig = axialStress(M,I3,rosettePosition[1])

tau = shearStress(Q,b3,h3,rosettePosition[1])

​

from numpy import sin,cos,pi

​

# Already have stresses values at each strain rosette, now we can compute the strains using Hooke's law

C = np.array([[1/E, -v/E,0],[-v/E,1/E,0],[0,0,2\*(1+v)/E]]) #stiffness tensor

​

# Let's take strain rosette 1 as example. We need both axial strains and the shear strain

epsilon = np.dot(C,np.array([sig,0,tau])) # No stress on y direction

​

# Now transform from coordinate strains to the direction of strain gauge

ros1 = [0,45,90] #all three gauge directions

e1a = strain1Rotate(epsilon,ros1[0]\*pi/180)

e1b = strain1Rotate(epsilon,ros1[1]\*pi/180)

e1c = strain1Rotate(epsilon,ros1[2]\*pi/180)

epsilon\_rosette1 = np.array([e1a,e1b,e1c])

​

#This can also be done using list comprehensions as

epsilon\_rosette1Alt = np.array([strain1Rotate(epsilon,th\*pi/180) for th in ros1])

​

# Since the unit of measurement is micron strain, the results here need to be multiplied by 1e6

epsilon\_rosette1 \*= 1e6 #this multiplies the original value by 1e6

epsilon\_rosette1Alt \*= 1e6

print(epsilon\_rosette1)

### Rosette 3



rosettePosition = [0.35,h3/2] #[x\_coordinate, z\_coordinate] this is rosette 1, note that z value equals to zero at neutral axis

Loads =[400,800,1200,1600]

for P in Loads:

Q = shearForce3(P,L3,rosettePosition[0])

M = bendingMoment3(P,L3,rosettePosition[0])

sig = axialStress(M,I3,rosettePosition[1])

tau = shearStress(Q,b3,h3,rosettePosition[1])

​

from numpy import sin,cos,pi

​

# Already have stresses values at each strain rosette, now we can compute the strains using Hooke's law

C = np.array([[1/E, -v/E,0],[-v/E,1/E,0],[0,0,2\*(1+v)/E]]) #stiffness tensor

​

# Let's take strain rosette 1 as example. We need both axial strains and the shear strain

epsilon = np.dot(C,np.array([sig,0,tau])) # No stress on y direction

​

# Now transform from coordinate strains to the direction of strain gauge

ros1 = [0,45,90] #all three gauge directions

e1a = strain1Rotate(epsilon,ros1[0]\*pi/180)

e1b = strain1Rotate(epsilon,ros1[1]\*pi/180)

e1c = strain1Rotate(epsilon,ros1[2]\*pi/180)

epsilon\_rosette1 = np.array([e1a,e1b,e1c])

​

#This can also be done using list comprehensions as

epsilon\_rosette1Alt = np.array([strain1Rotate(epsilon,th\*pi/180) for th in ros1])

​

# Since the unit of measurement is micron strain, the results here need to be multiplied by 1e6

epsilon\_rosette1 \*= 1e6 #this multiplies the original value by 1e6

epsilon\_rosette1Alt \*= 1e6

print(epsilon\_rosette1)

​

### Rosette 4



rosettePosition = [0.45,-h3/2] #[x\_coordinate, z\_coordinate] this is rosette 1, note that z value equals to zero at neutral axis

Loads =[400,800,1200,1600]

for P in Loads:

Q = shearForce3(P,L3,rosettePosition[0])

M = bendingMoment3(P,L3,rosettePosition[0])

sig = axialStress(M,I3,rosettePosition[1])

tau = shearStress(Q,b3,h3,rosettePosition[1])

​

from numpy import sin,cos,pi

​

# Already have stresses values at each strain rosette, now we can compute the strains using Hooke's law

C = np.array([[1/E, -v/E,0],[-v/E,1/E,0],[0,0,2\*(1+v)/E]]) #stiffness tensor

​

# Let's take strain rosette 1 as example. We need both axial strains and the shear strain

epsilon = np.dot(C,np.array([sig,0,tau])) # No stress on y direction

​

# Now transform from coordinate strains to the direction of strain gauge

ros1 = [-45,0,45] #all three gauge directions

e1a = strain1Rotate(epsilon,ros1[0]\*pi/180)

e1b = strain1Rotate(epsilon,ros1[1]\*pi/180)

e1c = strain1Rotate(epsilon,ros1[2]\*pi/180)

epsilon\_rosette1 = np.array([e1a,e1b,e1c])

​

#This can also be done using list comprehensions as

epsilon\_rosette1Alt = np.array([strain1Rotate(epsilon,th\*pi/180) for th in ros1])

​

# Since the unit of measurement is micron strain, the results here need to be multiplied by 1e6

epsilon\_rosette1 \*= 1e6 #this multiplies the original value by 1e6

epsilon\_rosette1Alt \*= 1e6

print(epsilon\_rosette1)

## 4 - Point Bending Strain Calculations

### Rosette 1



rosettePosition = [0.35,-h3/2] #[x\_coordinate, z\_coordinate] this is rosette 1, note that z value equals to zero at neutral axis

Loads =[400,800,1200,1600]

for P in Loads:

Q = shearForce4(P,L4,rosettePosition[0])

M = bendingMoment4(P,L4,rosettePosition[0])

sig = axialStress(M,I3,rosettePosition[1])

tau = shearStress(Q,b3,h3,rosettePosition[1])

​

from numpy import sin,cos,pi

​

# Already have stresses values at each strain rosette, now we can compute the strains using Hooke's law

C = np.array([[1/E, -v/E,0],[-v/E,1/E,0],[0,0,2\*(1+v)/E]]) #stiffness tensor

​

# Let's take strain rosette 1 as example. We need both axial strains and the shear strain

epsilon = np.dot(C,np.array([sig,0,tau])) # No stress on y direction

​

# Now transform from coordinate strains to the direction of strain gauge

ros1 = [0,45,90] #all three gauge directions

e1a = strain1Rotate(epsilon,ros1[0]\*pi/180)

e1b = strain1Rotate(epsilon,ros1[1]\*pi/180)

e1c = strain1Rotate(epsilon,ros1[2]\*pi/180)

epsilon\_rosette1 = np.array([e1a,e1b,e1c])

​

#This can also be done using list comprehensions as

epsilon\_rosette1Alt = np.array([strain1Rotate(epsilon,th\*pi/180) for th in ros1])

​

# Since the unit of measurement is micron strain, the results here need to be multiplied by 1e6

epsilon\_rosette1 \*= 1e6 #this multiplies the original value by 1e6

epsilon\_rosette1Alt \*= 1e6

print(epsilon\_rosette1)

### Rosette 2



rosettePosition = [0.375,0] #[x\_coordinate, z\_coordinate] this is rosette 1, note that z value equals to zero at neutral axis

Loads =[400,800,1200,1600]

for P in Loads:

Q = shearForce4(P,L4,rosettePosition[0])

M = bendingMoment4(P,L4,rosettePosition[0])

sig = axialStress(M,I3,rosettePosition[1])

tau = shearStress(Q,b3,h3,rosettePosition[1])

​

from numpy import sin,cos,pi

​

# Already have stresses values at each strain rosette, now we can compute the strains using Hooke's law

C = np.array([[1/E, -v/E,0],[-v/E,1/E,0],[0,0,2\*(1+v)/E]]) #stiffness tensor

​

# Let's take strain rosette 1 as example. We need both axial strains and the shear strain

epsilon = np.dot(C,np.array([sig,0,tau])) # No stress on y direction

​

# Now transform from coordinate strains to the direction of strain gauge

ros1 = [0,45,90] #all three gauge directions

e1a = strain1Rotate(epsilon,ros1[0]\*pi/180)

e1b = strain1Rotate(epsilon,ros1[1]\*pi/180)

e1c = strain1Rotate(epsilon,ros1[2]\*pi/180)

epsilon\_rosette1 = np.array([e1a,e1b,e1c])

​

#This can also be done using list comprehensions as

epsilon\_rosette1Alt = np.array([strain1Rotate(epsilon,th\*pi/180) for th in ros1])

​

# Since the unit of measurement is micron strain, the results here need to be multiplied by 1e6

epsilon\_rosette1 \*= 1e6 #this multiplies the original value by 1e6

epsilon\_rosette1Alt \*= 1e6

print(epsilon\_rosette1)

### Rosette 3



rosettePosition = [0.35,h3/2] #[x\_coordinate, z\_coordinate] this is rosette 1, note that z value equals to zero at neutral axis

Loads =[400,800,1200,1600]

for P in Loads:

Q = shearForce4(P,L4,rosettePosition[0])

M = bendingMoment4(P,L4,rosettePosition[0])

sig = axialStress(M,I3,rosettePosition[1])

tau = shearStress(Q,b3,h3,rosettePosition[1])

​

from numpy import sin,cos,pi

​

# Already have stresses values at each strain rosette, now we can compute the strains using Hooke's law

C = np.array([[1/E, -v/E,0],[-v/E,1/E,0],[0,0,2\*(1+v)/E]]) #stiffness tensor

​

# Let's take strain rosette 1 as example. We need both axial strains and the shear strain

epsilon = np.dot(C,np.array([sig,0,tau])) # No stress on y direction

​

# Now transform from coordinate strains to the direction of strain gauge

ros1 = [0,45,90] #all three gauge directions

e1a = strain1Rotate(epsilon,ros1[0]\*pi/180)

e1b = strain1Rotate(epsilon,ros1[1]\*pi/180)

e1c = strain1Rotate(epsilon,ros1[2]\*pi/180)

epsilon\_rosette1 = np.array([e1a,e1b,e1c])

​

#This can also be done using list comprehensions as

epsilon\_rosette1Alt = np.array([strain1Rotate(epsilon,th\*pi/180) for th in ros1])

​

# Since the unit of measurement is micron strain, the results here need to be multiplied by 1e6

epsilon\_rosette1 \*= 1e6 #this multiplies the original value by 1e6

epsilon\_rosette1Alt \*= 1e6

print(epsilon\_rosette1)

### Rosette 4[¶](http://localhost:8888/notebooks/Desktop/Autumn%202020/ME%20354/Lab%201%20-%20Beam%20Bending/Beam%20Bending%20Lab-Copy1.ipynb#Rosette-4)



rosettePosition = [0.45,-h3/2] #[x\_coordinate, z\_coordinate] this is rosette 1, note that z value equals to zero at neutral axis

Loads =[400,800,1200,1600]

for P in Loads:

Q = shearForce4(P,L4,rosettePosition[0])

M = bendingMoment4(P,L4,rosettePosition[0])

sig = axialStress(M,I3,rosettePosition[1])

tau = shearStress(Q,b3,h3,rosettePosition[1])

​

from numpy import sin,cos,pi

​

# Already have stresses values at each strain rosette, now we can compute the strains using Hooke's law

C = np.array([[1/E, -v/E,0],[-v/E,1/E,0],[0,0,2\*(1+v)/E]]) #stiffness tensor

​

# Let's take strain rosette 1 as example. We need both axial strains and the shear strain

epsilon = np.dot(C,np.array([sig,0,tau])) # No stress on y direction

​

# Now transform from coordinate strains to the direction of strain gauge

ros1 = [-45,0,45] #all three gauge directions

e1a = strain1Rotate(epsilon,ros1[0]\*pi/180)

e1b = strain1Rotate(epsilon,ros1[1]\*pi/180)

e1c = strain1Rotate(epsilon,ros1[2]\*pi/180)

epsilon\_rosette1 = np.array([e1a,e1b,e1c])

​

#This can also be done using list comprehensions as

epsilon\_rosette1Alt = np.array([strain1Rotate(epsilon,th\*pi/180) for th in ros1])

​

# Since the unit of measurement is micron strain, the results here need to be multiplied by 1e6

epsilon\_rosette1 \*= 1e6 #this multiplies the original value by 1e6

epsilon\_rosette1Alt \*= 1e6

print(epsilon\_rosette1)