CIL2030 COURSE PROJECT
MECHANICS OF SOLIDS

LAYER-WISE BENDING STRESS DISTRIBUTION

Siddharth Chauhan
(B21CI041)

FAILURE THEORIES

A BRIEF INTRODUCTION TO MAXIMUM NORMAL STRESS THEORY AND TRESCA/MAXIMUM SHEAR STRESS THEORY

INTRODUCTION TO FAILURE THEORIES

- Take into account different types of stresses
- Two famous failure theories are Maximum Normal Stress and Maximum Shear Stress
- These theories make an assumption about the behavior of materials under stress
- Both theories are important in designing and understanding the behavior of solid materials
- Understanding the basic failure theories is crucial in engineering design

MAXIMUM NORMAL STRESS THEORY

- Take into account different types of stresses
- Two famous failure theories are Maximum Normal Stress and Maximum Shear Stress
- These theories make an assumption about the behavior of materials under stress
- Both theories are important in designing and understanding the behavior of solid materials
- Understanding the basic failure theories is crucial in engineering design

LIMITATIONS OF MAXIMUM NORMAL STRESS THEORY

- It only considers the extreme values of the normal stress components and ignores the effect of shear stresses. This can result in inaccurate predictions of failure under complex stress states.
- It assumes that the material is isotropic, meaning that its properties are the same in all directions. In reality, most materials are anisotropic, meaning that their properties vary with direction. This can lead to inaccurate predictions of failure in materials that are anisotropic.

- It assumes that yielding occurs when the maximum normal stress reaches the yield strength of the material in tension or compression. However, yielding can also occur when the material reaches the yield strength in shear, which is not considered by the Maximum Normal Stress Theory.
- It does not account for stress concentrations, which can occur at points of discontinuity such as holes, notches, and fillets. These stress concentrations can cause local yielding and failure even when the maximum normal stress is below the yield strength of the material.

TRESCA/MAXIMUM SHEAR STRESS THEORY

- Assumption: The material is isotropic and homogeneous, and it considers only the shear stresses in the material.
- Tresca/Maximum shear stress theory states that a material fails when the maximum shear stress in the material reaches the yield stress in shear.
- The maximum shear stress in a material is the difference between the largest and smallest principal stresses in the material.
- Takes into account the angle shear stress is applied to a material

EXAMPLES OF TRESCA/MAXIMUM SHEAR STRESS THEORY IN USE

- Design of gears: Gears are designed to transmit power and motion between rotating shafts. The Tresca/Maximum Shear Stress Theory is used to ensure that the material used to construct the gear can withstand the maximum shear stress without exceeding the yield strength.

- Design of shafts: Shafts are used to transmit torque and power between different components of a machine. The Tresca/Maximum Shear Stress Theory is used to ensure that the material used to construct the shaft can withstand the maximum shear stress without exceeding the yield strength.
- Design of bolts and fasteners: Bolts and fasteners are used to hold two or more components together. The Tresca/Maximum Shear Stress Theory is used to ensure that the bolt or fastener can withstand the maximum shear stress without breaking or failing.

THE FORMULA FOR MAXIMUM SHEAR STRESS THEORY

$$\tau_{max} \leq \tau_{yield}$$

$$\tau_{max} \leq \frac{\sigma_{yield}}{2}$$

where:

- τ = shear stress (N/m2 or Pa)
- σyield or σy = yield stress (N/m2 or Pa)

$$\tau_{max} = \frac{\sigma_1 - \sigma_2}{2}, \frac{\sigma_2 - \sigma_3}{2}, \frac{\sigma_1 - \sigma_3}{2} = \frac{\sigma_y}{2}$$

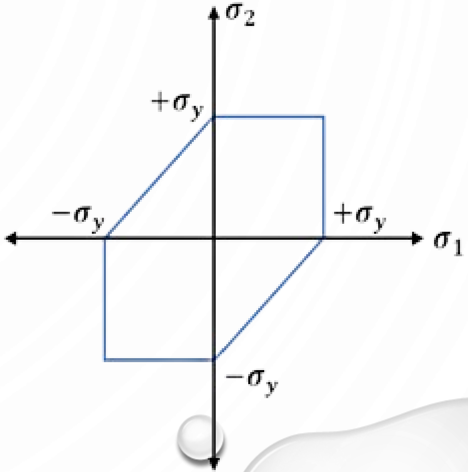
$$\frac{\sigma_1 - \sigma_2}{2} \leq \frac{\sigma_y}{2}$$

$$\frac{\sigma_2 - \sigma_3}{2} \leq \frac{\sigma_y}{2}$$

$$\frac{\sigma_1 - \sigma_3}{2} \leq \frac{\sigma_y}{2}$$

FAILURE ENVELOPE FOR MAXIMUM SHEAR STRESS THEORY

According to the maximum shear stress formula for the biaxial loading condition given in the previous section, a failure envelope can be drawn. The failure envelope for Tresca's theory of failure is hexagonal in shape as shown below.



LIMITATIONS OF TRESCA/MAXIMUM SHEAR STRESS THEORY

- It only considers the extreme values of the shear stress components and ignores the effect of normal stresses. This can result in inaccurate predictions of failure under complex stress states.
- It assumes that the material is isotropic, meaning that its properties are the same in all directions. In reality, most materials are anisotropic, meaning that their properties vary with direction. This can lead to inaccurate predictions of failure in materials that are anisotropic.

- It assumes that yielding occurs when the maximum shear stress reaches the yield strength of the material in shear. However, yielding can also occur when the material reaches the yield strength in tension or compression, which is not considered by the Tresca/Maximum Shear Stress Theory.
- It does not account for stress concentrations, which can occur at points of discontinuity such as holes, notches, and fillets. These stress concentrations can cause local yielding and failure even when the maximum shear stress is below the yield strength of the material.

SUMMARY/ CONCLUSION

Both the theories are useful in determining the failure mode of a material under different loading conditions, and they are commonly used in engineering design and analysis. However, these theories have limitations, such as not taking into account the effects of intermediate principal stresses or the plastic deformation of the material, which can lead to inaccuracies in predicting the failure of materials. Therefore, more sophisticated failure theories have been developed over time to account for these limitations.

भारतीय प्रौद्योगिकी संस्थान जोधपुर
Indian Institute of Technology Jodhpur
॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

PROGRAMMING PROBLEM:

For solving this problem we have to import various modules like pandas, numpy and matplotlib.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

We have written a code which takes input from user regarding which id of material sheet should be tested.

```
inp_str = input().split(" ")
input_ID = [int(inp_str[i]) for i in range(10)]
```

We have taken the 6th material id as our reference material and have selected its modulus of elasticity.(Er)

```
E_r = data_test.iloc[5,1]
E_r = int(E_r)
E_r
```

We have the found out modular ratio i.e, dividing the modulus of elasticity of different materials by Er.

```
modular_ratio = []
for i in data_arr:
  modular_ratio.append(i[1]/E_r)
modular_ratio
```

We have then found out density,area,volume,mass by defining their formulas. Also we have found out force and force per unit length acting of each material sheet

```
volume = 20*0.2*0.04
volume
density = [data_arr[i][3] for i in range(10)]
density
mass = [density[i]*(volume) for i in range(len(density))]
mass
Force = [mass[i]*9.81 for i in range(10)]
Force
force_len = [Force[i]/20 for i in range(10)]
force_len
```

Then we hae calculated the average of all force per unit length acting on material sheet.

```
force_len_mean = sum(force_len)/10
```

```
force_len_mean
```

Then we have multiplied modular ratio of each material by 0.2 i.e, the thickness of each material to find the new width.

```
new_width=[modular_ratio[i]*(0.2) for i in range(len(modular_ratio))]
```

```
new_width
```

Then we have multiplied the new width so found out by 0.04 which is the depth of sheet to find the new area and then we have summed all this area.

```
Area=[new_width[i]*(0.04) for i in range(len(new_width))]
```

```
Area
```

```
Area_sum = sum(Area)
```

```
Area_sum
```

We have then found out the centroid of each sheet and then the second moment of inertia of sheet.

```
Centroid=[0.02]
for i in range(1,10):
  x = round(Centroid[i-1]+0.04,2)
  Centroid.append(x)
Centroid
Area_Centroid_sum = 0
for i in range(len(Centroid)):
  Area_Centroid_sum += Area[i]*Centroid[i]
Area_Centroid_sum
Second_MOI = [I_1_lst[i]+I_2_lst[i] for i in range(len(I_1_lst))]
sum(Second_MOI)
```

We then found out the total moment of the beam and also the radius of curvature

```
moment_load = force_len_mean*(20**2)/8
moment_beam = 20000*20/2
Total_Moment = (moment_load) + (moment_beam)
Total_Moment
R_cur = Total_Moment/(E_r*sum(Second_MOI))
R_cur
```

Also we found out the deflection of each sheet.

```
defl = [(Y_1[i]-Y_0) for i in range(10)]
defl
```

With help of deflection we found out the strain in each sheet.

```
Strain = [(-1*defl[i])*R_cur for i in range(10)]
Strain
```

We have then found out the centroid of each sheet and then the second moment of inertia of sheet.

```
Centroid=[0.02]
for i in range(1,10):
  x = round(Centroid[i-1]+0.04,2)
  Centroid.append(x)
Centroid
Area_Centroid_sum = 0
for i in range(len(Centroid)):
  Area_Centroid_sum += Area[i]*Centroid[i]
Area_Centroid_sum
Second_MOI = [I_1_lst[i]+I_2_lst[i] for i in range(len(I_1_lst))]
sum(Second_MOI)
```

We then found out the total moment of the beam and also the radius of curvature

```
moment_load = force_len_mean*(20**2)/8
moment_beam = 20000*20/2
Total_Moment = (moment_load) + (moment_beam)
Total_Moment
R_cur = Total_Moment/(E_r*sum(Second_MOI))
R_cur
```

Also we found out the deflection of each sheet.

```
defl = [(Y_1[i]-Y_0) for i in range(10)]
defl
```

With help of deflection we found out the strain in each sheet.

```
Strain = [(-1*defl[i])*R_cur for i in range(10)]
Strain
```

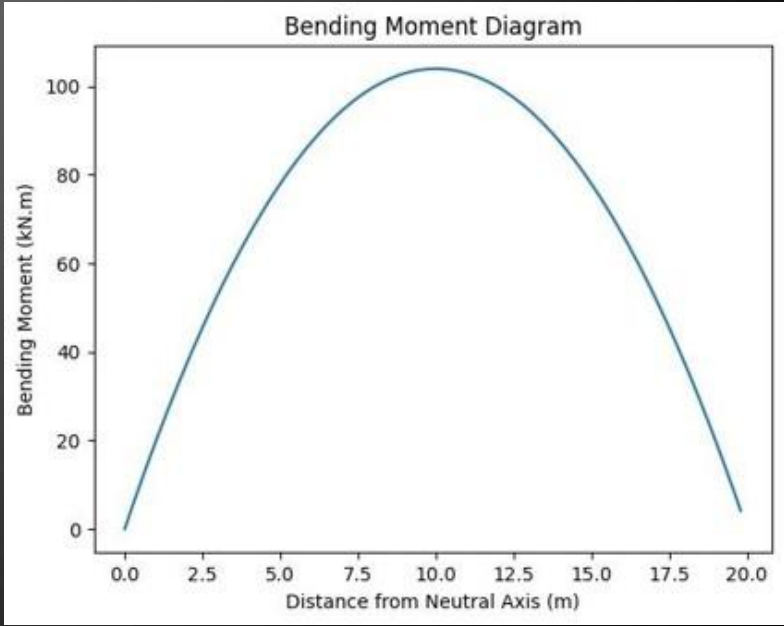
Now we found out bending moment and then plotted its graphs.

```
Bending_Moment = [((force_len[i]*20/1000)+20)*5 for i in range(10)]
Bending_Moment
import matplotlib.pyplot as plt
i = 0
step = 0.2
x = []
for j in range(100):
    x.append(i)
    i += step
for bd in Bending_Moment:
    y = []
    for n in x:
        y.append((-bd/100)*n*n + (bd/5)*n)
    plt.figure()
    plt.plot(x, y)
    plt.title("Bending Moment Diagram")
    plt.xlabel("Distance from Neutral Axis (m)")
    plt.ylabel("Bending Moment (kN.m)")
    plt.show()
```

We then calculated bending stress and then checked if material can withstand or not

```
Bending_Stress =
[(Bending_Moment[i]*Y_1[i]/sum(Second_MOI))/100
0 for i in range(10)]
Bending_Stress

for i in range(len(data_arr)):
  if abs(Bending_Stress[i])<=float(data_arr[i][2]):
    print("Material ID = {} ;
Pass\n".format(data_arr[i][0]))
  else:
    print("Material ID = {} ;
Fail\n".format(data_arr[i][0]))
```

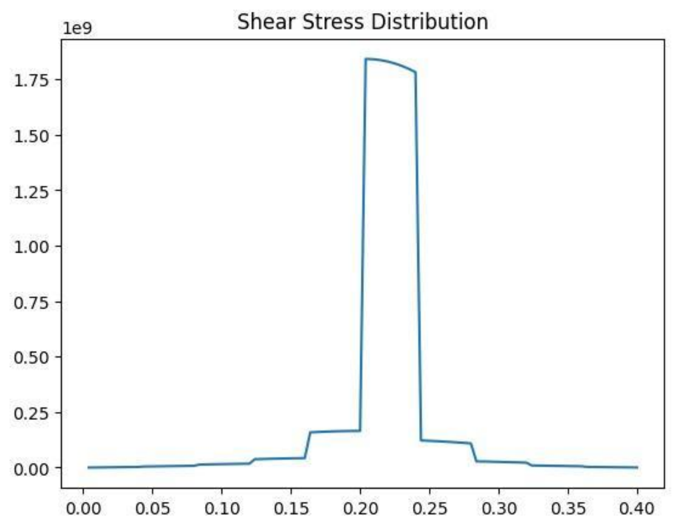


Verifying Toolkit for homogenous orientation

```
h=0.4
w=0.2
Volume  =20*0.2*0.04*10
S_Moi = (w*(h**3))/12
B_Moment_val=[]
for i in range(10):
  B_Moment_val.append(((force_len[i]*20/1000)+20)*20/4)
print(B_Moment_val)
Max_stress = []
for i in range(10):
  Max_stress.append(B_Moment_val[i]*Y_1[i]/(S_Moi*1000))
print(Max_stress)
for i in range(len(data_arr)):
  if abs(Max_stress[i])<=float(data_arr[i][2]):
    print("Material ID = {} ; Pass\n".format(data_arr[i][0]))
  else:
    print("Material ID = {} ; Fail\n".format(data_arr[i][0]))
```

Finding Shear Stress Distribution Curve

```
def V(q):
  x = (q*20/2) + 20000
  return x
def Q(y):
  a = (y*0.2*(0.2-(y/2)))
  return a
X_axis = np.linspace(0,0.4,101)
X_axis
Y_axis = []
for i in range(len(X_axis)-1):
  Y_axis.append((V(int(i/10))*Q(X_axis[i])/(Second_MOI[int(i/10)]*0.04)))
X_axis = X_axis[1:len(X_axis)]
plt.plot(X_axis,Y_axis)
plt.title("Shear Stress Distribution")
```



Finding Cost Estimation

```
Cost =[]
for i in range(10):
  print("Material ID = {} ; Price = 
{}".format(float(data_arr[i][0]),float(data_arr[i][4])*new_width[i]*20))
  Cost.append(float(data_arr[i][4])*new_width[i]*20)
print("\nTotal Price for the composite Beam = {}".format(sum(Cost)))
```

THANK YOU…