ES221- Mechanics of Solids

A
Laboratory Report
on
Estimation of Young's modulus
and yield strength

by

Aakash Venkatesan
Abhishek S
Faayza Vora
Pranav Thakkar
Satchit Velankar
Paras Shivale

INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR

– Table of Contents

– CONTENTS

Estimation of Young's Modulus and yield strength

Aakash Venkatesan (23110002), Abhishek S (23110008), Faayza Vora (23110109), Paras Shirvale (23110232), Pranav Thakkar (23110253), Satchit Velankar (23110291)

***Abstract*—This project focuses on calculating two important mechanical properties: the Young's modulus and yield strength of three materials - aluminium, mild steel, stainless steel using tensile test. Data analysis of the obtained results help us find the Young's Modulus and yield strength of the materials. Consequently, the results of the tensile test are compared with literature.**

I. Problem Statement and Objectives

Properties such as Young's Modulus and yield strength give us information about how a certain material responds to applied loads. This information is important for material selection and design in engineering. A tensile test helps us in understanding the behavior of materials under different loads, which can be used to obtain Young's Modulus and yield strength.

The objective of this project is to evaluate the Young's Modulus and yield strength of different materials by a tensile test using a Universal Testing Machine (UTM), and combining this experimental data with computation to calculate these material properties.

By carrying out this process on different materials, the aim is to experimentally arrive at the values of Young's Modulus and yield strength using computation and programming, and compare them with the available experimental data from literature.

II. Introduction

Tensile testing is fundamental and widely used experiment used to estimate a material's strength and mechanical properties. It shows how the material reacts to uniaxial tensile stress and helps analyse its ability to withstand applied loads, deform elastically, and eventually yield or fracture. We compare the properties of three different materials: stainless steel, mild steel, and aluminum. Comparing different classes of materials through this process offers deeper insights into their mechanical properties and helps us choose correct and reliable material for our application.

These materials were chosen for experiment due to their widespread industrial relevance. All three - aluminum, stainless steel and mild steel differ in their behavior under identical loading conditions. This helps us to study how different materials transition from elastic to plastic deformation and how their failure modes differ under similar loading conditions.

We use the Universal Testing Machine (UTM) for the experiment to visualize and compare their stress-strain curves thus evaluating and comparing their mechanical properties and limits.

Alongside the experimental work, we also developed a Python code to automate the calculation of different mechanical properties from the stress-strain data. This includes the Young's modulus and yield strength.

III. Experimental Procedure

The following steps were carried out to perform the tensile testing on stainless steel, cast iron, and aluminum:

(a) **Sample Preparation:**
Standardized dog bone samples were used for testing for stainless steel and mild steel. Whereas, for aluminum we used a rod with thick ends and thin middle part due to unavailability of aluminum dog bone samples. The tensile test was done on 4 samples of each material to minimize errors. Dimensions such as length and cross-sectional area were measured and recorded before testing.

(b) **Experimental Setup:**
Universal Testing Machine (UTM) was used in this experiment for performing tensile tests on each of the specimen. Each specimen was securely fixed into the grips of the Universal Testing Machine's jaws. Test parameters such as loading rate and maximum load were set according to standard procedure and speed of necking required.

(c) **Conducting the Test:**
A uniaxial tensile load was gradually applied to each sample until fracture. The machine recorded the applied force and elongation data, which we collected in a pendrive.

(d) **Data Collection:**
Load and elongation data was exported from the UTM using pendrive. This data was used to compute various different mechanical properties of the materials.

(e) **Data Analysis Using Code:**
A Python script was made and used to calculate key properties such as Young's modulus and yield stress.

IV. Methodology and code explanation

The aim of this project was to determine the Young's modulus and Yield strength of three different metals - aluminium , mild steel and stainless steel.

In order to do that, we performed tensile test on these metals for 4 samples using universal testing machine. The UTM gives us the values for Load and Elongation. The load was converted to stress using the following formula:

$$\text{Stress } (\sigma) = \frac{\text{load} \times g}{\text{Area of cross section } (A_c)} \tag{1}$$

Where

- g is acceleration due to gravity which is taken to be $9.81 m/s^2$
- A_c was measured to be $6.2 \times 0.3\ mm^2$ for Mild steel and Stainless steel and $50.24\ cm^2$

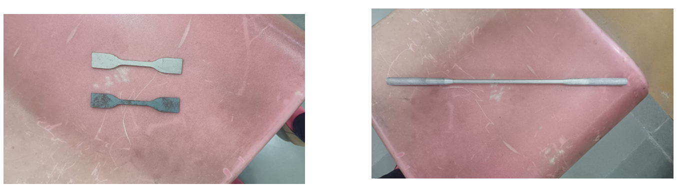
The elongation was converted into strain by using the following formula:

$$\text{Strain} = \frac{\text{Elongation}}{\text{Gage length}} \tag{2}$$

where

- Gage length is 39 mm for mild steel and stainless steel and 19 cm for aluminium.

We have different readings for aluminium as we used a circular cross section sample as the flat ones were not available.
The images of the samples used are attached below:



(a) Mild steel and stainless steel sample

(b) Aluminium sample

Fig. 1: Metal samples used for tensile test

After the Tensile test we analysed the data using Python. Our initial steps were to first convert the load and elongation data to stress and strain data which was done using the following code snippet.

```python
# Keep only necessary columns
df = df[['id', 'LOAD', 'ELONGATION']].copy()
df.rename(columns={'LOAD': 'Load', 'ELONGATION': 'Elongation'}, inplace=True)

df['Stress'] = (df['Load'] * 9.81 ) / area_sqm    # Stress = Load / Area in Pascals
df['Strain'] = df['Elongation'] / 39  # Strain = Change in length / Original length

# Remove the row with index 1
df = df.drop(index=0)

# Reset the index after dropping the row
df = df.reset_index(drop=True)
df['id'] = df['id'] - 1
df.head()
```

Fig. 2: Obtaining Stress Strain Curve

We then neglected some points. This was done because these points were too close to the axes, and some of them were below the y-axis so they may have caused errors in the calculation.
We know that if there is no load applied there will not be any elongation, thus, we shift the origin to the first data point of the modifies database. This is done in the following code snippet.

```python
strain-=strain[0]
stress-=stress[0] # shifting the data points
plt.grid()
plt.axhline()
plt.axvline()
plt.scatter(strain[:], stress[:], facecolors='none', edgecolors='blue', s=10, marker='o')
```

Fig. 3: Shifting the origin

In order to estimate the linearity of the obtained graphs we followed the following steps:

- We applied np.polyfit to perform a linear fit, repeating this process iteratively for every data point. That is this was first done for 2 data points, then 3, then 4, and so on.
- We then computed difference between the y coordinate of the actual point and the point on bestfit line, Calculated root mean square error and minimized that.
- This minimization was done to ensure that we get all the bestfit lines in the early iterations
- We now know the slope of each bestfit line, so we averaged over a range of bestfit lines to get a good estimate of young's modulus

This is done using the following code snippets:

Computing the root mean square errors

```python
(xi,yi) = strain[0],stress[0]
(xf,yf) = strain[-1],stress[-1]

def compute_rmse(arr1,arr2):
    return np.sqrt(np.mean((arr1-arr2)**2))
```

linear fit ine and getting slope and intercept of the same

```python
def linear_fit(x, y):
    m, c = np.polyfit(x, y, 1)

    def g(m_local, c_local, x_val):
        return m_local * x_val + c_local

    return m, c, g
```

Fig. 4: function for computing root mean square errors

Calculating losses between actual point and corrosponding point on best fit line

```python
losses = []
for i in range(2,len(strain)):
    mi1,ci1,gi1 = linear_fit(np.array(strain[:i]), np.array(stress[:i]) )
    loss_1 = compute_rmse(stress[:i], gi1(mi1,ci1,strain[:i]))
    losses.append(((mi1,ci1,loss_1),i))
```

Sorting the losses in ascending order

```python
l2 = losses[:]
l2.sort(key= lambda x: x[0][-1] )
```

Fig. 5: Computing losses and minimizing them

Plotting some graphs for linear fit

```
plt.figure(figsize=(10,100))
for i in range(1,91):
    mm,cc = l2[i-1][0][0], l2[i-1][0][1]
    plt.subplot(30,3,i)
    plt.plot(strain,stress, label="Stress-Strain Curve")
    plt.plot(strain[:100],mm*strain[:100] + cc, label="Stress-Strain Curve fit")
    plt.grid()
    plt.title(f'i=>{i-1}')
    plt.tight_layout()
    plt.legend()


plt.show()
```

Fig. 6: Plotting the obtained graphs

Averaging the slope of the best fits

```
index1 = 20
index2 = 30
ar = l2[index1:index2+1]
mean_slope = np.mean([i[0][0] for i in ar])
```

Young's Modulus

```
Y = float(mean_slope/(10**9))
print(f"{Y=} GPa")
```

```
Y=193.6073726814701 GPa
```

Fig. 7: Computing Young's modulus

Thus, To calculate yield strength we offset the obtained best fit line by 0.2% strain and then find the point of intersection of the orignal curve and line.
This was done by using the following code snippet.

Offsetting the line by 0.002 on strain axis

```
original_strain = df['Strain']
original_stress = df['Stress']

offset_line = (original_strain-0.002)*mean_slope
tangent_line = original_strain*mean_slope
plt.figure(figsize=(10,6))
plt.plot(original_strain,original_stress,label='original')
plt.plot(original_strain,tangent_line,label='tangent')
plt.plot(original_strain,offset_line,label='Offset')
plt.ylim(-1,1.1*max(original_stress))
plt.xlim(-0.001,0.08)
plt.legend()
plt.axhline(c='b')
plt.axvline(c='b')
plt.grid()
```

Outputs are collapsed ...

Finding the point of intersection i.e yield strength

```
yeild_strain_idx = np.argmin(np.abs(offset_line-original_stress))
yeild_strength = original_stress[yeild_strain_idx]
float(yeild_strength)/10**6
```

Fig. 8: Calculating the yield strength

V. Experimental Results

A. Young's Modulus

Young's modulus is same as the magnitude of the linear region of stress strain curve. The linear region and the slope is estimated using the method mentioned in section (IV) and the results obtained are tabulated :

Metal	Sample Number	Young's Modulus in (GPa)	Average Value
Aluminum	1	69.77	69.55
	2	69.76	
	3	69.33	
	4	69.33	
Mild Steel	1	209.28	190.28
	2	194.27	
	3	177.02	
	4	180.57	
Stainless Steel	1	187.50	184.89
	2	192.52	
	3	193.60	
	4	165.93	

TABLE I: Young's Modulus

B. Yield Strength

Yield Strength is the point at which the stress strain curve and the offset line intersect and is evaluated using the code snippet given in section (IV) and the results obtained are tabulated below:

Metal	Sample Number	Yeild Strength (MPa)	Average Value
Aluminium	1	5344.65	4967.33
	2	3697.96	
	3	5438.28	
	4	5388.44	
Mild Steel	1	2870.12	2702.58
	2	2777.44	
	3	2907.03	
	4	2255.75	
Stainless Steel	1	2103.36	3140.14
	2	1888.40	
	3	2380.09	
	4	6188.71	

TABLE II: Yield Strength

VI. Observations and Discussion

The values available in the literature are :

1) Aluminium
   - Young's modulus: 69 Gpa reference
   - Yield strength: 395 Mpa reference
2) Mild Steel

- Young's modulus: $180 - 250$ Gpa reference
- Yield strength: $270 - 370$ Mpa reference

3) Stainless Steel

- Young's modulus: 180 Gpa reference
- Yield strength: 502 Mpa reference

The values of Young's Modulus that we get are similar to the above mentioned values. The deviations for young's modulus obtained in the above readings are 0.55 for aluminium, 10 for mild steel (considering lower limit and 4.89 for stainless steel. These deviation are obtained due to the approximations used while getting the best fit line and some noise in the obtained data.

However, large deviations are observed in the results for yield strength, which can be due to different composition of the metal. We do not know about the homogeneity of the manufactured sample and also its grain composition. Moreover the tangent line is approximated to be the best fit line which included many approximations that may have caused error in the results.

Also, the experiment involves machine errors as from the data we can observe that there are certain datapoints which are missing which contribute to the errors.

We observed a gradual necking in aluminium but not in the case of stainless steel and mild steel thus proving that both the types of steel are more brittle than aluminium. The position of necking particularly for aluminium was different thus showing that the composition of the material was not homogeneous.

The Python code and raw data obtained from the UTM, along with the animations of the plotted graphs can be found in the link here.

VII. Conclusion

Through this study, we successfully estimated some mechanical properties—namely, the Young's modulus, and yield strength of aluminum, stainless steel, and mild steel samples using a simple tensile test. The experimental setup using the Universal Testing Machine (UTM) provided load-elongation data, which we transformed into stress-strain curves. Using Python-based numerical tools, we could derive meaningful values for each property and observe how the test data varied with different materials.

While some deviations were observed between theoretical and experimental values, owing to factors like grain structure and data noise, our results emphasize the importance of accurate measurement and computational refinement. This approach of combining experimental results with programming-based data analysis not only improved the precision of our findings but also demonstrated the power of simulation and programming application in engineering practice.

Overall, the project strengthened our understanding of material behavior under tensile loading and underscored the role of computational tools in interpreting experimental data. These helped us understand how materials must be selected for specific tasks and how the behavior of any material affects the design of the machine/device to be made.

VIII. Learning Outcomes

We learned how to operate the universal testing machine (UTM) and then extracted the data from it. While using the universal testing machine (UTM), we observed how the data obtained from the tests varied with changes in material, size, and dimensions. Through these experiments, we gained insights into the mechanical properties of different materials such as tensile strength.

We also enhanced our Python proficiency by applying it to analyze the data collected from the UTM tests. This helped us interpret the results more efficiently and visualize the trend of this data.

Furthermore, we realized that real-life applications and experimental results do not always align accurately with the theoretical data, which deepened our understanding of practical limitations and experimental uncertainties. Also, we understood how programming languages like Python can be effectively used in analyzing data by performing calculations and visualization using their powerful libraries. This helped us understand the importance of these programming languages in mechanical engineering.

IX. Acknowledgements

We would like to thank to Prof.Harmeet Singh for teaching us the principles to enable us to successfully execute this project and also for guiding us through every step of the project. We also thank the MAchine lab staff particularly Mr. Tarun Sharma and Mr. Dev Meena for helping us conducting the tensile test and collecting the required data and supervising us during the same. Their presence and support were invaluable to us. We also thank IIT Gandhinagar for providing us with the platform and support necessary to carry out this experiment. Lastly, we would like to thank our peers, whose continued support was instrumental in completing this project successfully.

X. References

- S. H. Çrandall , N. C. Dahl, T.J.Lardner and M.s.Sivakumar , An introduction to solid mechanics of solids, 3rd., 2012, SI.Units.edition.
- https://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MA6061T91
- https://www.xometry.com/resources/materials/mild-steel/
- https://www.engineeringtoolbox.com/young-modulus-d_417.html