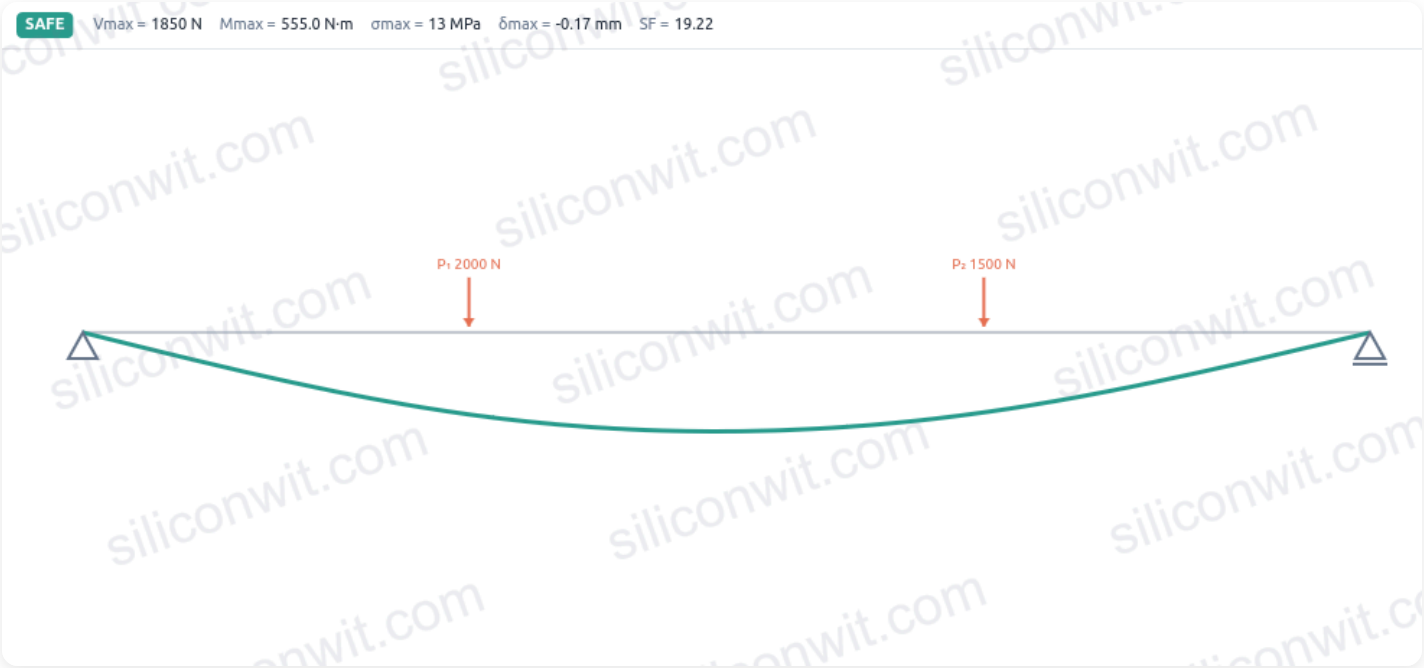
Practical Laboratory Experiments: Structural Analysis with Python and FreeCAD



Modified: Aug. 2, 2026

Published: Nov. 1, 2025



Become Contributor

Expert Support

Hand calculations tell you the stress at a single point under idealized loading. Real structures have complex geometry, multiple load paths, and boundary conditions that are difficult to capture analytically. Finite element analysis handles that complexity, but trusting its output blindly is dangerous: a wrong mesh, a misapplied boundary condition, or a misunderstood result can produce numbers that look plausible but are wrong. The

only reliable check is to solve a simplified version analytically, compare it to FEA, and understand where and why they diverge. These three progressive labs do exactly that for a robotic arm cantilever, a 3D printer gantry rail, and a drone arm under combined loading.

Laboratory Series Overview

This laboratory series provides comprehensive training in structural analysis methods essential for modern engineering practice. You will analyze real engineering components using both analytical methods (Python) and finite element analysis (FreeCAD), developing skills directly applicable to mechatronics, automation, and structural design.

What Makes These Labs Essential

Modern **engine**ering **requ**ires **profic**iency in **mult**iple **anal**ysis **meth**ods: **ha**nd **calcul**ations for **unders**tanding **fundam**entals, **progra**mming for **autom**ation and **param**etric **stud**ies, and **fin**ite **elem**ent **meth**ods for **comp**lex **geome**tries and **load**ing **condi**tions. **Th**is **labor**atory **ser**ies **deve**lops **al**l **thr**ee **compet**encies **thro**ugh **pract**ical **applic**ations.

Core Laboratory Skills:

- **Analytical Methods** - Apply equilibrium principles and stress formulas to real structures
- **Python Programming** - Automate calculations and create professional engineering plots
- **FEM Simulation** - Validate designs using FreeCAD and CalculiX solver
- **Engineering Communication** - Document findings in technical reports and presentations

Laboratory Structure

Lab 1: Shear Force and Bending Moment

Application: Robotic Arm Cantilever Analysis

- Calculate reaction forces and internal loads
- Generate SFD/BMD using Python
- Create 3D visualization in FreeCAD
- Duration: 1 week

Lab 2: Bending Stresses and FEM

Application: 3D Printer Gantry Rail Design

- Apply flexure formula for stress analysis
- Determine critical load positions
- Perform FEM analysis with mesh convergence
- Duration: 1 week

Lab 3: Combined Loading Analysis

Application: Drone Arm Multi-Axis Loading

- Analyze bending and torsion superposition
- Calculate principal stresses
- Apply failure criteria for design verification
- Duration: 1 week

Required Software

Python Environment:

- Python 3.8 or later
- Libraries: numpy, matplotlib, scipy, pandas

CAD/FEM Software:

- FreeCAD 0.20 or later (free and open source)
- Built-in CalculiX solver for FEM analysis

Installation: All software is freely available and runs on Windows, macOS, and Linux.

File Naming and Submission Standards

All submitted files must follow a strict naming convention to ensure proper organization and grading.

Standard File Naming Format

Format: `E022-01-XXXX-YYYY-LabN-Type.extension`

Components:

- `E022-01` = Course code
- `XXXX` = Your 4-digit registration number (e.g., 1234)
- `YYYY` = Registration year (e.g., 2023)
- `LabN` = Lab number (Lab1, Lab2, or Lab3)
- `Type` = File type descriptor
- `.extension` = Appropriate file extension

Example: `E022-01-1234-2023-Lab1-Model.glb`

ⓘ 📋 Required File Types and Naming

For Each Lab, Submit:

1. **3D Model (GLB format)**
   - Name: `E022-01-XXXX-YYYY-LabN-Model.glb`
   - Export from FreeCAD: File → Export → Select GLB format
   - This is the primary 3D deliverable for web visualization
2. **Technical Drawing (PDF format)**
   - Name: `E022-01-XXXX-YYYY-LabN-Drawing.pdf`
   - Create in FreeCAD: TechDraw workbench
   - Must include: dimensions, section views, material specification
   - Professional engineering drawing standards (ISO or ANSI)
3. **Rendered Images (PNG format)**
   - Name: `E022-01-XXXX-YYYY-LabN-Isometric.png`
   - Name: `E022-01-XXXX-YYYY-LabN-SideView.png`
   - Resolution range: 1600×900 – 2560×1440 pixels
   - Show load visualizations and boundary conditions
4. **Python Script (PY format)**
   - Name: `E022-01-XXXX-YYYY-LabN-Analysis.py`
   - Must include comments explaining each section
   - Should run without errors when parameters are set
5. **Analysis Plots (PNG format)**
   - Lab 1: `E022-01-XXXX-YYYY-Lab1-SFD-BMD.png`
   - Lab 2: `E022-01-XXXX-YYYY-Lab2-StressAnalysis.png`
   - Lab 2: `E022-01-XXXX-YYYY-Lab2-Convergence.png`
   - Lab 3: `E022-01-XXXX-YYYY-Lab3-MohrsCircle.png`
   - Lab 3: `E022-01-XXXX-YYYY-Lab3-Optimization.png`
6. **Hand Calculations (PDF format)**

- Name: `E022-01-XXXX-YYYY-LabN-Calculations.pdf`
- Scanned from lab notebook (clear, legible scans)
- Must show FBD, equations, and step-by-step work

7. **Technical Report (PDF format)**
   - Name: `E022-01-XXXX-YYYY-LabN-Report.pdf`
   - Professional formatting with sections as specified
   - Include all plots and tables in report

📦 Submission Package Structure

Create a folder named: `E022-01-XXXX-YYYY-LabN` (the group leader's Reg. No.)

Inside the folder, organize files:

```
E022-01-1234-2023-Lab1/
├── E022-01-1234-2023-Lab1-Model.glb
├── E022-01-1234-2023-Lab1-Drawing.pdf
├── E022-01-1234-2023-Lab1-Isometric.png
├── E022-01-1234-2023-Lab1-SideView.png
├── E022-01-1234-2023-Lab1-Analysis.py
├── E022-01-1234-2023-Lab1-SFD-BMD.png
├── E022-01-1234-2023-Lab1-Calculations.pdf
└── E022-01-1234-2023-Lab1-Report.pdf
```

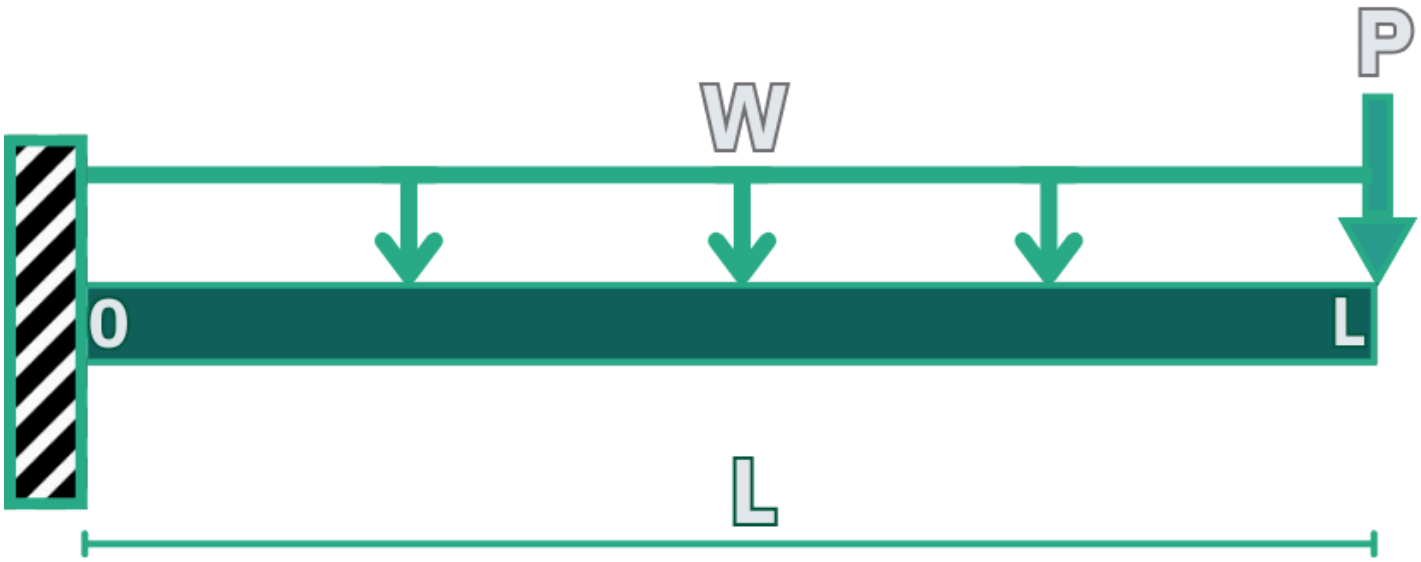
Compress to ZIP: `E022-01-XXXX-YYYY-LabN.zip`

Submit to the instructor before deadline

> ⚠ **Important Submission Rules**
>
> - **Late submissions** lose 10% per day
> - **Incorrect file naming** results in 5% penalty
> - **Missing files** cannot be added after deadline
> - **Group members** ensure your reg. no. appears in at least one file in the ZIP, otherwise, you'll be marked absent
> - **File size limit:** Maximum 50 MB per ZIP file
> - **Verify before submitting:** Check all files open correctly

Lab 1: Shear Force and Bending Moment Analysis

Application: Robotic Arm Design



Lab Duration: 1 week

Learning Objectives

By the end of this laboratory, you will be able to:

1. **Calculate** reaction forces using equilibrium equations for cantilever beam configurations
2. **Write** Python code to generate shear force and bending moment diagrams automatically
3. **Identify** critical sections in beam structures where maximum internal forces occur
4. **Create** professional engineering plots with proper labels, units, and formatting
5. **Relate** analysis results to practical robot design decisions and safety requirements

Real-World Engineering Context

You are designing a 6-DOF industrial robotic manipulator for material handling operations. The forearm segment connects the wrist joint to the gripper and must support the payload while maintaining sufficient rigidity to ensure positioning accuracy. The arm is modeled as a cantilever beam fixed at the elbow joint.

Design Challenge: The beam must safely support operational loads while minimizing weight to reduce motor torque requirements and improve dynamic response. Excessive deflection would compromise positioning accuracy, while insufficient strength could lead to structural failure.

Problem Statement

📋 Robotic Arm Specifications

Geometry:

- Beam Length (L): **500 mm**
- Cross-section: **Rectangular 60 mm × 20 mm** (width × height)
- Moment of Inertia (I): **4.0×10^4 mm⁴** (calculated as $bh^3/12 = 60 \times 20^3/12$)
- Distance to outer fiber (c): **10 mm** (half of height)

Loading:

- Point Load at Free End (P): **500 N** (gripper + payload weight)
- Applied at: x = 500 mm (free end)

Material:

- Material: **Aluminum alloy 6061-T6**
- Yield Strength (σ_{yield}): **275 MPa**
- Young's Modulus (E): **69 GPa**

Support Configuration:

- **Cantilever** (fixed at x = 0, free at x = 500 mm)
- Fixed support provides both vertical reaction force and reaction moment

> **Group-Specific Parameters**
>
> Your instructor will assign specific parameters to your group from the Group Parameter Table at the end of this document. Each group receives slightly different beam length and load values to ensure independent work while maintaining consistent difficulty.

Critical Question: What are the maximum shear force and bending moment for this robotic arm, and where do they occur?

Part 1: Analytical Solution

Before writing any code, develop fundamental understanding through hand calculations.

1. **Draw Free Body Diagram (Equivalent System Model)**

   Sketch the beam showing:

   - Fixed support at left end (x = 0) with reaction force R_y and reaction moment M_A
   - Downward force P = 500 N at right end (x = 500 mm)
   - Coordinate system with x-axis along beam length

2. **Calculate Reaction Forces**

   For cantilever beam with end load, apply equilibrium:

   Vertical force equilibrium (↑ positive): $\sum F_y = 0 : R_y - P = 0$

   $R_y = P = 500 \text{ N (upward)}$

   Moment equilibrium about fixed support (counterclockwise positive):
   $\sum M_A = 0 : M_A - P \times L = 0$

   $M_A = 500 \text{ N} \times 0.5 \text{ m} = 250 \text{ N·m (counterclockwise)}$

3. **Determine Shear Force Distribution**

   For cantilever with end load, shear force is constant along length:

$$V(x) = -500 \text{ N for } 0 \leq x < 500 \text{ mm}$$

(Negative because load creates downward shear on right side of any cut)

At free end after load point: V = 0 N

4. **Determine Bending Moment Distribution**

Bending moment varies linearly from support to free end:

$$M(x) = -P \times (L - x) = -500 \times (0.5 - x) \text{ N·m}$$

At key points:

- At x = 0 (fixed end): M(0) = **-250 N·m**
- At x = 250 mm (midpoint): M(0.25) = **-125 N·m**
- At x = 500 mm (free end): M(0.5) = **0 N·m**

5. **Calculate Maximum Bending Stress**

Use the flexure formula:

$$\sigma_{max} = \frac{M_{max} \times c}{I}$$

$$\sigma_{max} = \frac{250 \times 10^3 \text{ N·mm} \times 10 \text{ mm}}{4.0 \times 10^4 \text{ mm}^4}$$

$$\sigma_{max} = 62.5 \text{ MPa}$$

6. **Calculate Safety Factor**

$$SF = \frac{\sigma_{yield}}{\sigma_{max}} = \frac{275 \text{ MPa}}{62.5 \text{ MPa}} = 4.4$$

Engineering Insight: This safety factor of 4.4 is appropriate for robotic applications. In practice, SF = 3-5 is typical for robotics where dynamic loads, impact, and reliability are important considerations. This design balances safety with reasonable material usage.

> **Checkpoint**
>
> Complete these hand calculations in your lab notebook. These values will validate your Python code in the next section.

Part 2: Python Analysis

Now automate this analysis using Python for rapid design iteration and parametric studies.

Step 2.1: Setup Python Environment

Install required libraries:

```
pip install numpy matplotlib scipy pandas
```

Step 2.2: Complete Python Analysis Code

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

# ===== PROBLEM PARAMETERS =====
# Geometry
L = 500  # Beam length in mm
b = 60   # Width in mm
h = 20   # Height in mm

# Loading
P = 500  # Point load in N

# Material
sigma_yield = 275  # Yield strength in MPa

# Section properties
I = (b * h**3) / 12  # Moment of inertia in mm^4
c = h / 2            # Distance to outer fiber in mm

print(f'Moment of Inertia: {I:.2e} mm^4')
print(f'Distance to outer fiber: {c:.1f} mm')

# ===== ANALYSIS =====
# Create position array (0 to L)
x = np.linspace(0, L, 100)

# Shear force (constant for cantilever with end load)
V = -np.ones_like(x) * P  # Negative shear throughout

# Bending moment (linear variation)
M = -P * (L - x)  # Negative indicates tension on top fiber

# Find maximum values
V_max = np.max(np.abs(V))
M_max = np.max(np.abs(M))
M_max_location = x[np.argmax(np.abs(M))]

# Calculate maximum bending stress
```

```
sigma_max = (M_max * c) / I  # in N/mm^2 = MPa

# Safety factor
SF = sigma_yield / sigma_max

# ===== PRINT RESULTS =====
print('\n===== ANALYSIS RESULTS =====')
print(f'Maximum Shear Force: {V_max:.1f} N')
print(f'Maximum Bending Moment: {M_max:.1f} N·mm = {M_max/1000:.2f} N·m')
print(f'Location of M_max: {M_max_location:.1f} mm from fixed end')
print(f'Maximum Bending Stress: {sigma_max:.2f} MPa')
print(f'Safety Factor: {SF:.2f}')

# ===== CREATE PLOTS =====
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))

# Shear Force Diagram
ax1.plot(x, V, 'b-', linewidth=2, label='Shear Force')
ax1.axhline(y=0, color='k', linestyle='--', linewidth=0.5)
ax1.fill_between(x, 0, V, alpha=0.3, color='blue')
ax1.set_xlabel('Position along beam (mm)', fontsize=12)
ax1.set_ylabel('Shear Force (N)', fontsize=12)
ax1.set_title('Shear Force Diagram - Robotic Arm Cantilever', fontsize=14,
fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend()

# Bending Moment Diagram
ax2.plot(x, M/1000, 'r-', linewidth=2, label='Bending Moment')
ax2.axhline(y=0, color='k', linestyle='--', linewidth=0.5)
ax2.fill_between(x, 0, M/1000, alpha=0.3, color='red')
ax2.set_xlabel('Position along beam (mm)', fontsize=12)
ax2.set_ylabel('Bending Moment (N·m)', fontsize=12)
ax2.set_title('Bending Moment Diagram - Robotic Arm Cantilever',
fontsize=14, fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend()

plt.tight_layout()
plt.savefig('SFD_BMD_RoboticArm.png', dpi=300, bbox_inches='tight')
```

```
plt.show()

print('\nPlots saved as SFD_BMD_RoboticArm.png')
```

Expected Console Output:

```
Moment of Inertia: 4.00e+04 mm^4
Distance to outer fiber: 10.0 mm

===== ANALYSIS RESULTS =====
Maximum Shear Force: 500.0 N
Maximum Bending Moment: 250000.0 N·mm = 250.00 N·m
Location of M_max: 0.0 mm from fixed end
Maximum Bending Stress: 62.50 MPa
Safety Factor: 4.40
```

> **Validation**
>
> Compare these values with your hand calculations. They should match within rounding errors. If not, check your equations carefully.

Your Plots Should Show:

Shear Force Diagram:

- Horizontal line at V = -500 N (constant negative shear)
- Blue filled area below zero line
- Clear axis labels and grid

Bending Moment Diagram:

- Linear decrease from -250 N·m at fixed end to 0 at free end
- Red filled area below zero line (negative moment indicates tension on top fiber)
- Maximum moment clearly visible at x = 0

Part 3: FreeCAD Visualization

Create a 3D model of the robotic arm in FreeCAD to visualize the component and understand its geometry.

Beam Geometry and Orientation:

```
Coordinate System (FreeCAD Standard):
┌──────────────────────────────────────────────────────┐
│  X-axis: Beam LENGTH (500 mm) → extends left-right
│  Y-axis: Beam WIDTH (60 mm) → extends front-back
│  Z-axis: Beam HEIGHT (20 mm) → extends up-down
│
│  Cross-section: 60 mm (Y) × 20 mm (Z) rectangle
│  Extrusion: Along X-axis for 500 mm
│
│  Load: P = 500 N pointing DOWN (-Z direction)
│        at x = 500 mm (free end, right side)
│
│  Support: Fixed at x = 0 mm (left side)
└──────────────────────────────────────────────────────┘
```

> **FreeCAD Sketch Reference**
>
> - **Horizontal** in sketch view = **Y-axis** (width = 60 mm)
> - **Vertical** in sketch view = **Z-axis** (height = 20 mm)
> - **Extrude/Pad** direction = **X-axis** (length = 500 mm)

1. **Create Basic Beam Model**
   - Open FreeCAD and create new document (File → New)
   - Switch to Part Design workbench
   - Create new body (right-click in tree → Create Body)
   - Create sketch on **XY plane** (this will be the cross-section face)
2. **Draw Rectangular Cross-Section (60 mm wide × 20 mm tall)**
   - Click **Rectangle tool**

- Draw rectangle centered on origin (or from origin)
- Constrain **horizontal dimension** (width): **60 mm** (Y-direction)
- Constrain **vertical dimension** (height): **20 mm** (Z-direction)
- Close sketch

3. **Extrude Beam Along Length (500 mm)**
   - Click **'Pad'** button (extrusion tool)
   - Enter **length: 500 mm**
     - This extrudes the cross-section along the **X-axis** (beam length direction)
   - Click OK
   - **Result**: You now have a beam 500 mm long (X) × 60 mm wide (Y) × 20 mm tall (Z)
4. **Add Material Properties**
   - Right-click part → Appearance → Set Material
   - Choose 'Aluminum' from material library
   - Set aluminum silver color
5. **Add Load Visualization (Arrow Pointing Downward)**
   - Switch to **Part workbench**
   - Create arrow shaft: **Part → Primitives → Cylinder**
     - Radius: 8 mm, Height: 80 mm
     - Position: Above beam at free end (x = 500 mm, z = top of beam + 80 mm)
     - **Rotate 180°** around Y-axis to point **downward**
   - Create arrow head: **Part → Primitives → Cone**
     - Base radius: 15 mm, Height: 25 mm
     - Position: At bottom of shaft, pointing at beam
     - **Rotate 180°** around Y-axis to point **downward**
   - Fuse shaft and head: Select both → **Part → Boolean → Union**
   - Color it **orange/red** (right-click → Appearance → Shape color)
   - Add text label using **Draft workbench**: Text = "P = 500 N"

> **Load Direction**
>
> The arrow must point **downward** (negative Z-direction) to represent the gravitational load from the gripper and payload.

6. **Add Fixed Support Visualization**

   - At the left end (x = 0), add a wall/block to represent the fixed support
   - Use **Part → Box** with dimensions 30 mm × 70 mm × 50 mm
   - Position it at the left end of beam
   - Color it **gray** to distinguish from beam
   - Add text label: “Fixed Support”

7. **Create Technical Drawing (TechDraw Workbench)**

   - Switch to **TechDraw workbench**
   - Insert → Default Page (A4 landscape)
   - Insert View → Insert View → Select your beam body

   Add Multiple Views:

   - **Front view**: Shows length (500 mm, X-axis) and height (20 mm, Z-axis)
   - **Top view**: Shows length (500 mm, X-axis) and width (60 mm, Y-axis)
   - **Right side view**: Shows width (60 mm, Y-axis) and height (20 mm, Z-axis)
   - **Section A-A**: Cut through beam to show rectangular cross-section clearly

   Add Dimensions:

   - Insert → Dimensions → Length: 500 mm (beam length)
   - Insert → Dimensions → Width: 60 mm (cross-section width)
   - Insert → Dimensions → Height: 20 mm (cross-section height)

   Add Annotations:

   - Insert → Annotation → Material: “Aluminum Alloy 6061-T6”
   - Insert → Annotation → Load: “P = 500 N (downward) at x = 500 mm”

- Insert → Annotation → Support: “Fixed support at x = 0 mm”

Export Drawing:

- File → Export as PDF
- Save as: `E022-01-XXXX-YYYY-Lab1-Drawing.pdf`

8. **Export 3D Model and Images**

Set Isometric View:

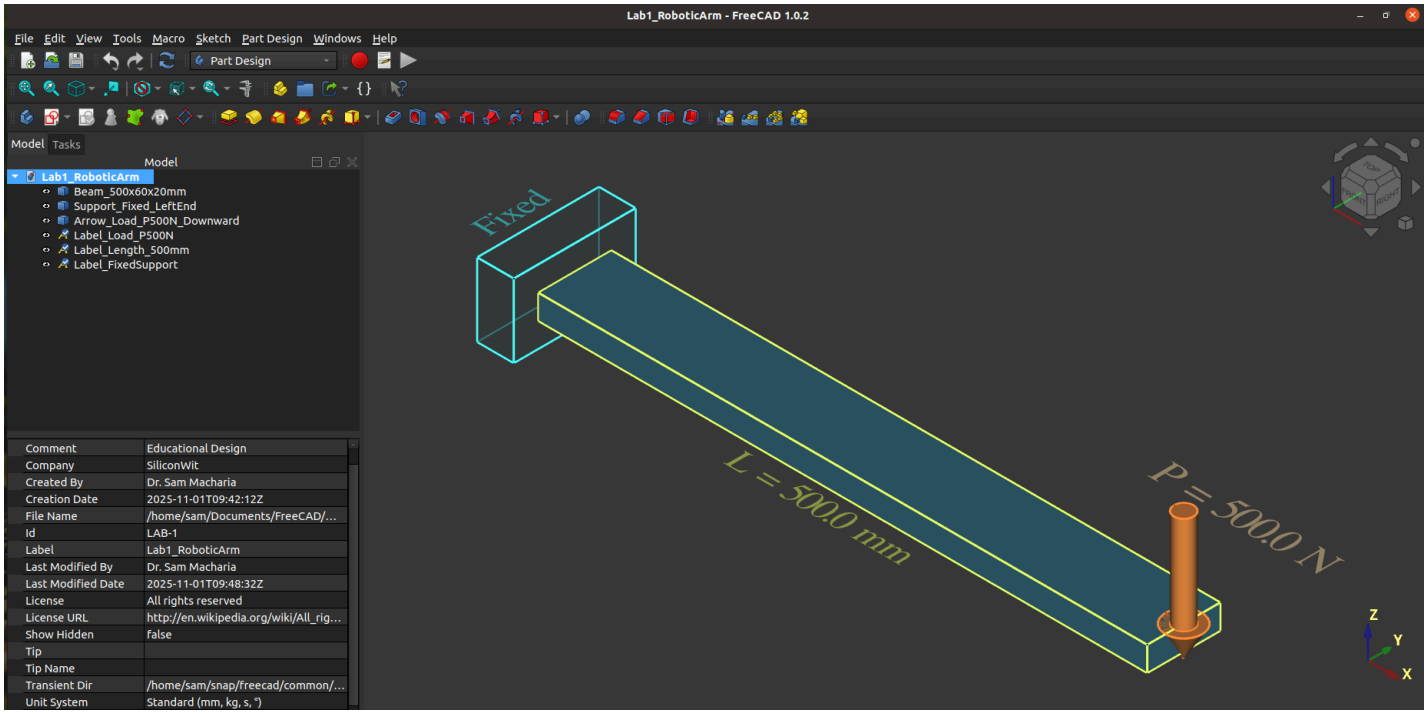
- Click **View → Standard Views → Isometric**
- Or press **“0”** on numpad for isometric view
- Adjust camera to show beam, load arrow, and support clearly

Take Screenshots:

- **Isometric view**: View → Screenshot → Save as `E022-01-XXXX-YYYY-Lab1-Isometric.png`
  - Resolution: according to the File Naming and Submission Standards above
- **Side view**: Rotate to side → Screenshot → Save as `E022-01-XXXX-YYYY-Lab1-SideView.png`
  - Should clearly show beam length and load arrow

Export 3D Model (GLB for Web):

- Select all model objects in tree view (Ctrl+A or select individually)
- File → Export
- In the file type dropdown, select **“glTF 2.0 (*.glb .gltf)”
- **Important**: Change the extension from `.gltf` to `.glb` before saving
  - Default filename: `E022-01-XXXX-YYYY-Lab1-Model.gltf`
  - Change to: `E022-01-XXXX-YYYY-Lab1-Model.glb`
- Click **Export** or **Save**
- **Verify your GLB file**: Visit https://siliconwit.com/product-development/glb-viewer/
  - Drag and drop your `.glb` file to confirm it displays correctly



Example FreeCAD model showing beam, fixed support, and load arrow

> **Important**
>
> This lab focuses on visualization and familiarity with FreeCAD. FEM analysis will be introduced in Lab 2.

> **Instrument it**
>
> This lab calculates and simulates the bending stress; on the real beam you would close the loop and measure it. A foil strain gauge bonded at the section of maximum moment, read through an HX711 or an instrumentation amplifier, gives the strain directly, and multiplying by the modulus gives a stress to compare against your analytical and FEM numbers. An ESP32 or STM32 reads the gauge and logs the readings, turning the lab from a calculation into a measurement you can trust.

Lab 1 Deliverables

Submit according to the File Naming and Submission Standards above.

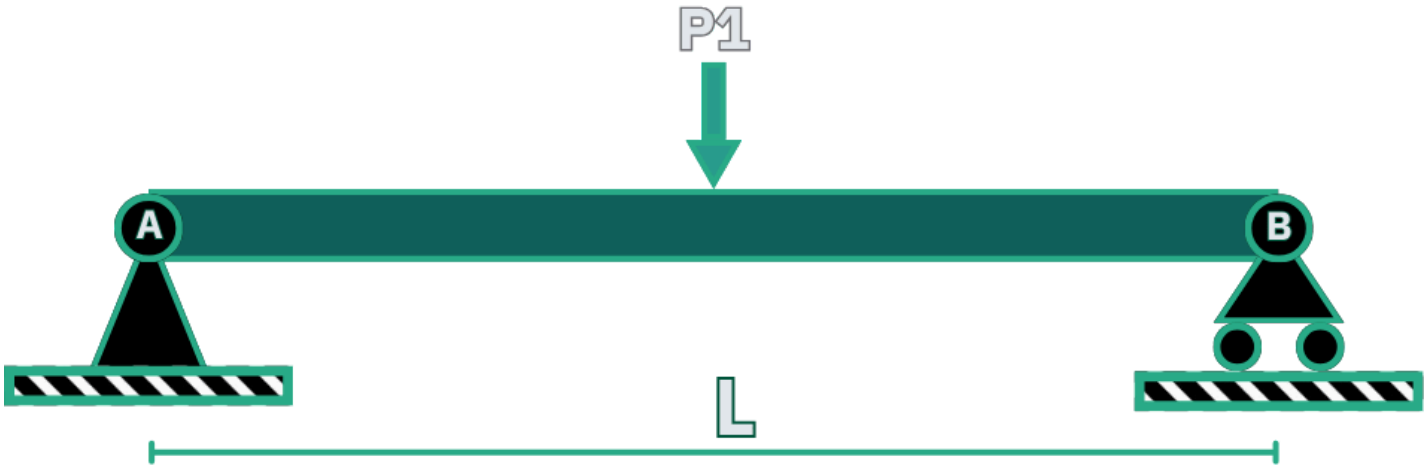
Required Files:

1. **3D Model** - `E022-01-XXXX-YYYY-Lab1-Model.glb`
    - Exported from FreeCAD showing beam geometry with load visualization
2. **Technical Drawing** - `E022-01-XXXX-YYYY-Lab1-Drawing.pdf`
    - Created using FreeCAD TechDraw workbench
    - Must include: orthographic views, dimensions (500mm length, 60×20mm section), material specification (Al 6061-T6), load annotation (500 N)
3. **Rendered Images**
    - `E022-01-XXXX-YYYY-Lab1-Isometric.png` (isometric view showing load and support)
    - `E022-01-XXXX-YYYY-Lab1-SideView.png` (side elevation showing beam dimensions)
4. **Python Script** - `E022-01-XXXX-YYYY-Lab1-Analysis.py`
    - Complete code with comments for SFD and BMD analysis
5. **Analysis Plot** - `E022-01-XXXX-YYYY-Lab1-SFD-BMD.png`
    - Combined shear force and bending moment diagrams
6. **Hand Calculations** - `E022-01-XXXX-YYYY-Lab1-Calculations.pdf`
    - Scanned lab notebook pages with FBD, equilibrium equations, and results
7. **Technical Report** - `E022-01-XXXX-YYYY-Lab1-Report.pdf` (2-3 pages)
    - Introduction (robotic arm application and analysis importance)
    - Problem statement with specifications
    - Methodology (key equations used)
    - Results (comparison table, plots, safety factor)
    - Discussion (SF appropriateness, weight reduction strategies, load sensitivity)
    - Conclusion (key findings and design recommendations)

Submission: Compress all files into `E022-01-XXXX-YYYY-Lab1.zip` and submit to the instructor.

Lab 2: Bending Stresses and FEM Introduction

Application: 3D Printer Gantry Rail Design



Lab Duration: 1 week

Learning Objectives

By the end of this laboratory, you will be able to:

1. **Apply** the flexure formula σ = Mc/I to calculate bending stresses in beam structures

2. **Determine** critical load positions that produce maximum stress in simply supported beams
3. **Create and mesh** 3D models in FreeCAD for finite element analysis
4. **Set up and run** FEM analyses using CalculiX solver with appropriate boundary conditions
5. **Validate** FEM results against analytical calculations and understand sources of discrepancy
6. **Understand** mesh convergence concepts and perform convergence studies

Real-World Engineering Context

Desktop 3D printers use a horizontal rail (gantry) that supports the print head as it moves back and forth during the additive manufacturing process. The rail must be stiff enough to prevent deflection during printing, which would cause layer misalignment and poor print quality. The print head weight creates a moving concentrated load that varies in position during operation.

Design Challenge: The gantry must maintain positioning accuracy within ±0.05 mm while minimizing weight to enable faster print speeds. Understanding where maximum stresses occur is critical for optimizing the rail cross-section and selecting appropriate materials.

Problem Statement

📋 3D Printer Gantry Specifications

Geometry:

- Rail Length (L): **800 mm**
- Cross-section: **Aluminum tube OD 30 mm × ID 24 mm × 3 mm wall thickness** (hollow circular)
- Moment of Inertia (I): **2.35×10^4 mm⁴** (hollow circular: $I = \pi(OD^4-ID^4)/64$)
- Distance to outer fiber (c): **15 mm** (outer radius)

Loading:

- Print Head Weight (P): **200 N** (includes extruder, hotend, cooling fans, wiring)
- Load position: **Variable** ($0 \leq x \leq 800$ mm during operation)

Material:

- Material: **Aluminum alloy 6061-T6**
- Young's Modulus (E): **70 GPa**
- Poisson's Ratio (ν): **0.33**
- Yield Strength (σ_{yield}): **275 MPa**

Support Configuration:

- **Simply supported** (pinned at $x = 0$, roller at $x = 800$ mm)
- Supports provide vertical reactions only, allowing thermal expansion

> **Group-Specific Parameters**
>
> Your instructor will assign specific parameters to your group from the Group Parameter Table at the end of this document.

Critical Questions:

1. Where should the print head be positioned to cause maximum bending stress?
2. What is the maximum stress value at this critical position?
3. Does FEM analysis confirm the analytical solution?

Part 1: Analytical Solution

First, determine the critical load position and maximum stress analytically.

Theory: Moving Load on Simply Supported Beam

For a simply supported beam with point load P at distance 'a' from left support:

Left reaction: $R_A = \frac{P(L-a)}{L}$

Right reaction: $R_B = \frac{Pa}{L}$

Bending moment under load: $M(a) = \frac{Pa(L-a)}{L}$

To find position ‘a’ that gives absolute maximum moment, differentiate and set to zero:

$$\frac{dM}{da} = \frac{P(L-2a)}{L} = 0$$

Therefore: Maximum bending moment occurs when load is at **midspan (a = L/2 = 400 mm)**

1. **Calculate Maximum Moment**

   With load at midspan (a = 400 mm):

   $$M_{max} = \frac{PL}{4} = \frac{200\ \text{N} \times 800\ \text{mm}}{4} = 40{,}000\ \text{N·mm} = 40\ \text{N·m}$$

2. **Calculate Maximum Bending Stress**

   Apply flexure formula:

   $$\sigma_{max} = \frac{M_{max} \times c}{I}$$

   $$\sigma_{max} = \frac{40{,}000\ \text{N·mm} \times 15\ \text{mm}}{6.0 \times 10^4\ \text{mm}^4}$$

   $$\sigma_{max} = 10.0\ \text{MPa}$$

3. **Calculate Safety Factor**

   $$SF = \frac{\sigma_{yield}}{\sigma_{max}} = \frac{275\ \text{MPa}}{10.0\ \text{MPa}} = 27.5$$

4. **Engineering Assessment**

   This extremely high static safety factor indicates the rail is over-designed for static loading. However, 3D printers involve:

   - **Dynamic loading** (acceleration and deceleration of print head)
   - **Vibration** during rapid movements
   - **Fatigue** from millions of load cycles

> These dynamic effects can create stress amplification factors of 2-5×. A high static SF provides margin for dynamic effects and ensures long service life.

> **Checkpoint**
>
> Record these analytical results in your lab notebook. You will compare them with Python calculations and FEM results.

Part 2: Python Parametric Analysis

Create Python code to analyze stress at multiple load positions and generate visualization plots.

```python
import numpy as np
import matplotlib.pyplot as plt

# ===== PROBLEM PARAMETERS =====
# Geometry
L = 800  # Rail length in mm
OD = 30  # Outer diameter in mm
t = 3  # Wall thickness in mm
ID = OD - 2*t  # Inner diameter in mm

# Loading
P = 200  # Print head weight in N

# Material
sigma_yield = 275  # Yield strength in MPa
E = 70000  # Young's modulus in MPa

# Section properties (hollow circular)
I = (np.pi / 64) * (OD**4 - ID**4)  # Moment of inertia in mm^4
c = OD / 2  # Distance to outer fiber in mm

print(f'Moment of Inertia: {I:.2e} mm^4')
print(f'Distance to outer fiber: {c:.1f} mm')

# ===== PARAMETRIC ANALYSIS =====
# Analyze load at multiple positions
a_positions = np.linspace(0, L, 200)  # Load positions

# Calculate moment at each position
M_at_load = P * a_positions * (L - a_positions) / L

# Calculate stress at each position
sigma = (M_at_load * c) / I

# Find maximum values
a_critical = a_positions[np.argmax(sigma)]
M_max = np.max(M_at_load)
sigma_max = np.max(sigma)
SF = sigma_yield / sigma_max
```

```
# ===== PRINT RESULTS =====
print('\n===== ANALYSIS RESULTS =====')
print(f'Critical load position: {a_critical:.1f} mm from left support')
print(f'Maximum bending moment: {M_max:.1f} N·mm = {M_max/1000:.2f} N·m')
print(f'Maximum bending stress: {sigma_max:.3f} MPa')
print(f'Safety factor: {SF:.1f}')

# ===== CREATE PLOTS =====
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))

# Bending Moment vs Load Position
ax1.plot(a_positions, M_at_load/1000, 'b-', linewidth=2)
ax1.axvline(x=a_critical, color='r', linestyle='--', linewidth=1,
label=f'Critical position: {a_critical:.0f} mm')
ax1.axhline(y=M_max/1000, color='gray', linestyle=':', linewidth=1)
ax1.set_xlabel('Load Position (mm)', fontsize=12)
ax1.set_ylabel('Bending Moment (N·m)', fontsize=12)
ax1.set_title('Bending Moment vs Print Head Position', fontsize=14,
fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend()

# Bending Stress vs Load Position
ax2.plot(a_positions, sigma, 'r-', linewidth=2)
ax2.axvline(x=a_critical, color='b', linestyle='--', linewidth=1,
label=f'Max stress: {sigma_max:.3f} MPa')
ax2.axhline(y=sigma_max, color='gray', linestyle=':', linewidth=1)
ax2.axhline(y=sigma_yield, color='orange', linestyle='--', linewidth=2,
label=f'Yield strength: {sigma_yield} MPa')
ax2.set_xlabel('Load Position (mm)', fontsize=12)
ax2.set_ylabel('Maximum Bending Stress (MPa)', fontsize=12)
ax2.set_title('Bending Stress vs Print Head Position', fontsize=14,
fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend()

plt.tight_layout()
plt.savefig('GantryRail_Stress_Analysis.png', dpi=300, bbox_inches='tight')
plt.show()
```

```
print('\nPlots saved as GantryRail_Stress_Analysis.png')
```

Expected Results:

- Critical position: 400.0 mm (midspan)
- Maximum moment: 40.0 N·m
- Maximum stress: 10.0 MPa
- Safety factor: 27.5

Part 3: FEM Analysis in FreeCAD

Now validate the analytical solution using finite element analysis.

1. **Create 3D Model of Hollow Circular Tube**
   - Open FreeCAD, create new document
   - Part Design workbench → Create Body
   - Create sketch on XY plane
   - Draw **circle with radius 15 mm** (OD = 30 mm diameter) centered on origin
   - Close sketch, Pad **800 mm** along length (beam length)
   - Create second sketch on same face
   - Draw **circle with radius 12 mm** (ID = 24 mm diameter) centered on origin (inner cavity)
   - Close sketch, Pocket through all (creates hollow circular tube with 3 mm wall thickness)
2. **Switch to FEM Workbench**
   - Workbench dropdown → FEM
   - Create Analysis container: Model → Analysis container
3. **Add Material**
   - Model → Material → System → Standard → Metal
   - Select material: Aluminum-Generic
   - Verify: E = 70 GPa, v = 0.33

- Select solid body, click OK

4. **Apply Boundary Conditions - Left Support (Pin)**
   - Analysis → Constraint → Constraint fixed
   - Select left end face
   - Constrain: Fix X, Fix Y, Fix Z (pinned support approximation)
   - Click OK
5. **Apply Boundary Conditions - Right Support (Roller)**
   - Analysis → Constraint → Constraint displacement
   - Select right end face
   - Set: Y displacement = 0 (vertical support only)
   - Leave X and Z free (allows expansion)
   - Click OK
6. **Apply Load at Midspan**
   - Analysis → Constraint → Constraint force
   - Select small face at top center (x = 600 mm position)
   - Force magnitude: 250 N
   - Direction: negative Y (downward)
   - Click OK
7. **Create Mesh**
   - Analysis → Mesh → FEM mesh from shape by Gmsh
   - Max element size: 20 mm (coarse mesh for first run)
   - Click OK, then Compute
8. **Run Solver**
   - Analysis → Solver → Solver CalculiX Standard
   - Click Write .inp file
   - Click Run CalculiX
   - Wait for solution (should take 10-30 seconds)
9. **View Results**

- Double-click “Results” in tree
- Change field to “von Mises stress”
- Note maximum stress value in color bar
- Verify location is at midspan, top/bottom surfaces

10. **Compare with Analytical Solution**
    - Analytical: σ_max = 0.682 MPa
    - FEM: (read from color bar, should be 0.6-0.7 MPa)
    - Calculate percent difference

Part 4: Mesh Convergence Study

Verify that your FEM results are mesh-independent by refining the mesh.

🔬 Mesh Convergence Procedure

Why Convergence Studies Matter:

FEM approximates reality by dividing continuous structure into discrete elements. Coarse meshes give approximate results; finer meshes converge toward exact solution. Engineers must verify their mesh is sufficiently refined.

Procedure:

1. Run analysis with max element size = 20 mm (already done)
2. Record maximum von Mises stress
3. Refine mesh: max element size = 10 mm
4. Run analysis again, record stress
5. Refine mesh: max element size = 5 mm
6. Run analysis again, record stress
7. Plot stress vs number of elements

Convergence Criterion:

Solution has converged when refining mesh changes result by less than 2-5%.

Create Python script to analyze convergence:

```python
import matplotlib.pyplot as plt

# Mesh convergence data (fill in from your FEM runs)
element_sizes = [20, 10, 5]  # mm
num_elements = []  # Record from FreeCAD
max_stress_fem = []  # Record from FEM results (MPa)

# Analytical solution for comparison
sigma_analytical = 0.682  # MPa

# Calculate percent error
percent_error = [(abs(s - sigma_analytical) / sigma_analytical * 100) for s
in max_stress_fem]

# Plot convergence
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

# Stress vs number of elements
ax1.plot(num_elements, max_stress_fem, 'bo-', linewidth=2, markersize=8,
label='FEM Results')
ax1.axhline(y=sigma_analytical, color='r', linestyle='--', linewidth=2,
label='Analytical Solution')
ax1.set_xlabel('Number of Elements', fontsize=12)
ax1.set_ylabel('Maximum Stress (MPa)', fontsize=12)
ax1.set_title('Mesh Convergence Study', fontsize=14, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend()

# Percent error vs number of elements
ax2.plot(num_elements, percent_error, 'ro-', linewidth=2, markersize=8)
ax2.axhline(y=5, color='gray', linestyle='--', linewidth=1, label='5% error
threshold')
ax2.set_xlabel('Number of Elements', fontsize=12)
ax2.set_ylabel('Percent Error (%)', fontsize=12)
ax2.set_title('FEM Error vs Mesh Refinement', fontsize=14,
fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend()

plt.tight_layout()
```

```
plt.savefig('Mesh_Convergence_Study.png', dpi=300, bbox_inches='tight')
plt.show()
```

Lab 2 Deliverables

Submit according to the File Naming and Submission Standards above.

Required Files:

1. **3D FEM Model** - `E022-01-XXXX-YYYY-Lab2-Model.glb`
    - Exported from FreeCAD showing gantry rail with FEM mesh and boundary conditions
2. **Technical Drawing** - `E022-01-XXXX-YYYY-Lab2-Drawing.pdf`
    - FreeCAD TechDraw with orthographic views, section view showing hollow tube (OD 30mm, ID 24mm)
    - Dimensions, material (Al 6061-T6), support annotations (pin and roller)
3. **Rendered Images**
    - `E022-01-XXXX-YYYY-Lab2-Isometric.png` (FEM stress contours visible)
    - `E022-01-XXXX-YYYY-Lab2-SideView.png` (showing support conditions and load position)
4. **Python Scripts**
    - `E022-01-XXXX-YYYY-Lab2-Analysis.py` (parametric stress analysis)
    - `E022-01-XXXX-YYYY-Lab2-Convergence.py` (mesh convergence study)
5. **Analysis Plots**
    - `E022-01-XXXX-YYYY-Lab2-StressAnalysis.png` (stress vs load position, moment vs load position)
    - `E022-01-XXXX-YYYY-Lab2-Convergence.png` (convergence plot showing stress vs element count)
6. **Hand Calculations** - `E022-01-XXXX-YYYY-Lab2-Calculations.pdf`
    - Critical load position derivation, maximum stress calculation, safety factor

7. **Technical Report** - `E022-01-XXXX-YYYY-Lab2-Report.pdf` (3-4 pages)
    - Introduction (3D printer application and moving load analysis)
    - Analytical solution (critical position derivation, stress calculations)
    - FEM methodology (model setup, mesh details, boundary conditions)
    - Results (analytical vs FEM comparison, mesh convergence data)
    - Discussion (why midspan is critical, sources of error, convergence assessment)
    - Conclusion (design recommendations for weight reduction)

Submission: Compress all files into `E022-01-XXXX-YYYY-Lab2.zip` and submit to the instructor.

Lab 3: Combined Loading and Failure Analysis



Application: Drone Arm Under Multi-Axis Loading



Lab Duration: 1 week

Learning Objectives

By the end of this laboratory, you will be able to:

1. **Analyze** structures under combined bending and torsional loading using superposition
2. **Calculate** principal stresses from combined stress states
3. **Construct** 3D Mohr's circles both by hand and using Python
4. **Apply** Von Mises and Tresca failure criteria to verify design safety
5. **Perform** FEM analysis with combined loading conditions in FreeCAD
6. **Optimize** designs for weight reduction while maintaining structural integrity

Real-World Engineering Context

Quadcopter drones use four motor arms extending from the central body to support the motors and propellers. Each arm experiences complex loading: vertical thrust forces from motors, aerodynamic drag creating bending, and gyroscopic moments from spinning propellers creating torsion. The arms must withstand these combined loads while minimizing weight for maximum flight time.

Design Challenge: Optimize the arm cross-section to achieve a safety factor of 3.0 under combined loading while minimizing mass. Every gram saved increases flight time, but insufficient strength leads to catastrophic failure during flight.

Problem Statement

📋 Drone Arm Specifications

Geometry:

- Arm Length (L): **200 mm** (center to motor mount)
- Cross-section: **Circular tube OD 16 mm × ID 12 mm** (carbon fiber composite)
- Polar Moment of Inertia (J): **4.64×10^3 mm⁴**
- Moment of Inertia (I): **2.32×10^3 mm⁴**

Loading:

- Vertical Thrust Force (P): **30 N** at arm tip (motor thrust)
- Aerodynamic Drag Force (F_drag): **5 N** horizontal at tip
- Gyroscopic Torque (T): **1.0 N·m** about arm axis (propeller gyroscopic effect)

Material:

- Material: **Carbon Fiber Composite** (unidirectional)
- Tensile Strength (σ_t): **600 MPa**
- Compressive Strength (σ_c): **500 MPa**
- Shear Strength (τ_ultimate): **70 MPa**

Support Configuration:

- **Cantilever** fixed to drone central body at origin

Critical Questions:

1. What are the maximum principal stresses in the arm?
2. Which failure criterion is most appropriate for this composite material?
3. What is the safety factor, and is the design adequate?

Part 1: Combined Stress Analysis

Analyze each loading component separately, then combine using superposition.

1. **Analyze Bending from Vertical Thrust**

   Vertical load P = 50 N creates bending moment at fixed end:

   $$M_y = P \times L = 50\ \text{N} \times 250\ \text{mm} = 12{,}500\ \text{N·mm}$$

   Maximum bending stress (at top and bottom of section):

   $$\sigma_x = \frac{M_y \cdot c}{I} = \frac{12{,}500 \times 10}{2{,}898} = 43.1\ \text{MPa}$$

   (Tension on bottom, compression on top)

2. **Analyze Bending from Horizontal Drag**

   Horizontal load F = 8 N creates bending in orthogonal plane:

   $$M_z = F_{drag} \times L = 8\ \text{N} \times 250\ \text{mm} = 2{,}000\ \text{N·mm}$$

   Maximum bending stress (at sides of section):

   $$\sigma_x = \frac{M_z \cdot c}{I} = \frac{2{,}000 \times 10}{2{,}898} = 6.9\ \text{MPa}$$

3. **Analyze Torsion from Gyroscopic Moment**

   Applied torque T = 2 N·m = 2000 N·mm creates shear stress:

$$\tau_{xy} = \frac{T \cdot r}{J} = \frac{2{,}000 \times 10}{5{,}796} = 3.45 \text{ MPa}$$

4 Identify Critical Point

The critical point is at the fixed end, at the location where bending stresses from both planes combine constructively. For circular tube, consider point at 45° where both bending components contribute:

$$\sigma_x = \sqrt{(43.1)^2 + (6.9)^2} = 43.6 \text{ MPa (tensile)}$$

$$\tau_{xy} = 3.45 \text{ MPa (shear from torsion)}$$

5 Calculate Principal Stresses

For plane stress state (σ_x, τ_xy):

$$\sigma_1 = \frac{\sigma_x}{2} + \sqrt{\left(\frac{\sigma_x}{2}\right)^2 + \tau_{xy}^2}$$

$$\sigma_1 = \frac{43.6}{2} + \sqrt{\left(\frac{43.6}{2}\right)^2 + (3.45)^2} = 44.4 \text{ MPa}$$

$$\sigma_2 = \frac{\sigma_x}{2} - \sqrt{\left(\frac{\sigma_x}{2}\right)^2 + \tau_{xy}^2}$$

$$\sigma_2 = \frac{43.6}{2} - \sqrt{\left(\frac{43.6}{2}\right)^2 + (3.45)^2} = -0.8 \text{ MPa}$$

$$\sigma_3 = 0 \text{ (plane stress)}$$

6 Apply Von Mises Failure Criterion

For ductile materials (conservative for composites):

$$\sigma_{VM} = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}{2}}$$

$$\sigma_{VM} = \sqrt{\frac{(44.4 - (-0.8))^2 + (-0.8 - 0)^2 + (0 - 44.4)^2}{2}} = 44.2 \text{ MPa}$$

Safety factor:

$$SF = \frac{\sigma_{yield}}{\sigma_{VM}} = \frac{500}{44.2} = 11.3$$

7 Apply Maximum Stress Criterion (More Appropriate for Composites)

> For brittle composites, compare maximum stress to strength:
>
> $SF_{tension} = \frac{600}{44.4} = 13.5$
>
> $SF_{compression} = \frac{500}{0.8} = 625$ (compression negligible)
>
> **Governing safety factor: 13.5**

Engineering Assessment: The safety factor exceeds 10, indicating significant over-design. This provides margin for:

- Dynamic loads during maneuvers (up to 3× static)
- Manufacturing defects in composite layup
- Fatigue from vibration
- Impact loads during crashes

However, weight reduction could increase flight time. Target SF = 3-4 would be more appropriate.

Part 2: Python Analysis with Mohr's Circle

Automate combined stress analysis and visualize stress state using Mohr's circle.

```python
import numpy as np
import matplotlib.pyplot as plt

# ===== PROBLEM PARAMETERS =====
# Geometry
L = 250  # Arm length in mm
OD = 20  # Outer diameter in mm
ID = 16  # Inner diameter in mm
r_outer = OD / 2
r_inner = ID / 2

# Loading
P_vertical = 50  # N
F_horizontal = 8  # N
T_torque = 2000  # N·mm

# Material (carbon fiber)
sigma_tensile = 600  # MPa
sigma_compressive = 500  # MPa
tau_ultimate = 70  # MPa

# Section properties
J = (np.pi / 32) * (OD**4 - ID**4)  # Polar moment
I = (np.pi / 64) * (OD**4 - ID**4)  # Moment of inertia

print(f'Polar Moment of Inertia: {J:.1f} mm^4')
print(f'Moment of Inertia: {I:.1f} mm^4')

# ===== STRESS ANALYSIS =====
# Bending moments
M_vertical = P_vertical * L
M_horizontal = F_horizontal * L

# Bending stresses at critical point
sigma_vertical = (M_vertical * r_outer) / I
sigma_horizontal = (M_horizontal * r_outer) / I

# Combined bending stress (vector sum for worst case)
sigma_x = np.sqrt(sigma_vertical**2 + sigma_horizontal**2)
```

```
# Torsional shear stress
tau_xy = (T_torque * r_outer) / J

print('\n===== STRESS COMPONENTS =====')
print(f'Bending stress (vertical load): {sigma_vertical:.2f} MPa')
print(f'Bending stress (horizontal load): {sigma_horizontal:.2f} MPa')
print(f'Combined bending stress: {sigma_x:.2f} MPa')
print(f'Torsional shear stress: {tau_xy:.2f} MPa')

# ===== PRINCIPAL STRESSES =====
# For plane stress: sigma_x, tau_xy
sigma_avg = sigma_x / 2
R = np.sqrt((sigma_x/2)**2 + tau_xy**2)

sigma_1 = sigma_avg + R
sigma_2 = sigma_avg - R
sigma_3 = 0  # plane stress

print('\n===== PRINCIPAL STRESSES =====')
print(f'σ₁ (maximum principal): {sigma_1:.2f} MPa')
print(f'σ₂ (minimum principal): {sigma_2:.2f} MPa')
print(f'σ₃ (out-of-plane): {sigma_3:.2f} MPa')

# ===== FAILURE CRITERIA =====
# Von Mises stress
sigma_vm = np.sqrt(((sigma_1-sigma_2)**2 + (sigma_2-sigma_3)**2 + (sigma_3-
sigma_1)**2) / 2)

# Tresca (maximum shear stress)
tau_max = (sigma_1 - sigma_2) / 2

# Safety factors
SF_von_mises = sigma_compressive / sigma_vm
SF_max_stress_tensile = sigma_tensile / sigma_1
SF_max_stress_shear = tau_ultimate / tau_max

print('\n===== FAILURE ANALYSIS =====')
print(f'Von Mises equivalent stress: {sigma_vm:.2f} MPa')
print(f'Maximum shear stress (Tresca): {tau_max:.2f} MPa')
print(f'\nSafety Factors:')
```

```
print(f'  Von Mises criterion: {SF_von_mises:.2f}')
print(f'  Maximum stress (tension): {SF_max_stress_tensile:.2f}')
print(f'  Maximum shear: {SF_max_stress_shear:.2f}')
print(f'\nGoverning safety factor: {min(SF_von_mises,
SF_max_stress_tensile, SF_max_stress_shear):.2f}')

# ===== MOHR'S CIRCLE VISUALIZATION =====
fig, ax = plt.subplots(figsize=(10, 8))

# Circle parameters
center = sigma_avg
radius = R

# Draw circle
theta = np.linspace(0, 2*np.pi, 100)
circle_x = center + radius * np.cos(theta)
circle_y = radius * np.sin(theta)

ax.plot(circle_x, circle_y, 'b-', linewidth=2, label="Mohr's Circle")

# Draw axes
ax.axhline(y=0, color='k', linestyle='-', linewidth=0.5)
ax.axvline(x=0, color='k', linestyle='-', linewidth=0.5)

# Plot principal stresses
ax.plot([sigma_1, sigma_2], [0, 0], 'ro', markersize=10, label='Principal
Stresses')
ax.text(sigma_1, 2, f'σ₁ = {sigma_1:.1f} MPa', fontsize=11, ha='center')
ax.text(sigma_2, -2, f'σ₂ = {sigma_2:.1f} MPa', fontsize=11, ha='center')

# Plot original stress state
ax.plot([sigma_x, 0], [tau_xy, -tau_xy], 'gs', markersize=8,
label='Original State')

# Draw line connecting original state points
ax.plot([sigma_x, 0], [tau_xy, -tau_xy], 'g--', linewidth=1)

# Labels and formatting
ax.set_xlabel('Normal Stress σ (MPa)', fontsize=13)
ax.set_ylabel('Shear Stress τ (MPa)', fontsize=13)
```

```
ax.set_title("Mohr's Circle - Drone Arm Combined Loading", fontsize=15,
fontweight='bold')
ax.grid(True, alpha=0.3)
ax.legend(fontsize=11)
ax.axis('equal')

plt.tight_layout()
plt.savefig('Mohrs_Circle_DroneArm.png', dpi=300, bbox_inches='tight')
plt.show()

print("\nMohr's circle saved as Mohrs_Circle_DroneArm.png")
```

Part 3: FEM Analysis with Combined Loading

Validate analytical results using FreeCAD FEM with multiple load types.

1. **Create Circular Tube Model**
   - Part Design workbench → Create Body
   - Create sketch on XY plane
   - Draw circle centered at origin, diameter 20 mm
   - Close sketch, Pad 250 mm
   - Create sketch on same face, draw circle diameter 16 mm
   - Pocket through all (creates hollow tube)
2. **Setup FEM Analysis**
   - FEM workbench → Create Analysis
   - Add material: Carbon-Fiber-Reinforced-Polymer
   - Manual input: E = 70 GPa, v = 0.1 (orthotropic, use equivalent)
3. **Apply Fixed Constraint**
   - Constraint fixed on left end face (drone body connection)
   - Fix all DOF (X, Y, Z, RotX, RotY, RotZ)
4. **Apply Vertical Force**
   - Constraint force on right end face center point

- Force: 50 N in negative Z direction (downward thrust)

5. **Apply Horizontal Force**
   - Constraint force on same point
   - Force: 8 N in positive Y direction (drag)
6. **Apply Torque**
   - This requires workaround in FreeCAD
   - Create couple forces: two equal opposite forces offset from center
   - Calculate: F × d = 2000 N·mm, use d = 8 mm → F = 250 N
   - Apply +250 N and -250 N at top/bottom of end face
7. **Mesh and Solve**
   - Create mesh: max element 5 mm (refined from start)
   - Compute mesh
   - Run CalculiX solver
8. **Analyze Results**
   - View von Mises stress distribution
   - Verify maximum occurs at fixed end
   - Record maximum stress value
   - Compare with analytical $\sigma_{VM} = 44.2$ MPa
9. **View Principal Stresses**
   - Change display to “Major principal stress”
   - Verify $\sigma_1 \approx 44.4$ MPa at critical location
   - Export stress contour images

Part 4: Design Optimization Study

Use Python to perform parametric study for weight optimization.

```python
import numpy as np
import matplotlib.pyplot as plt

# Design constraints
SF_target = 3.0  # Target safety factor
L = 250  # Length fixed
sigma_allowable = sigma_compressive / SF_target  # 500/3 = 166.7 MPa

# Vary tube dimensions
OD_range = np.linspace(12, 24, 30)  # Outer diameter range
thickness = 2  # Fix wall thickness at 2mm
ID_range = OD_range - 2*thickness

# Material density
rho = 1.6e-6  # kg/mm^3 (carbon fiber)

masses = []
safety_factors = []

for OD, ID in zip(OD_range, ID_range):
    r_outer = OD / 2
    r_inner = ID / 2

    # Section properties
    J = (np.pi / 32) * (OD**4 - ID**4)
    I = (np.pi / 64) * (OD**4 - ID**4)

    # Stress analysis (same loading)
    M_v = P_vertical * L
    M_h = F_horizontal * L
    sigma_v = (M_v * r_outer) / I
    sigma_h = (M_h * r_outer) / I
    sigma_x = np.sqrt(sigma_v**2 + sigma_h**2)
    tau_xy = (T_torque * r_outer) / J

    # Principal stress
    R = np.sqrt((sigma_x/2)**2 + tau_xy**2)
    sigma_1 = sigma_x/2 + R

    # Von Mises
```

```
    sigma_vm = np.sqrt(sigma_1**2 + 3*tau_xy**2)  # Simplified for
comparison

    # Safety factor
    SF = sigma_compressive / sigma_vm
    safety_factors.append(SF)

    # Mass calculation
    volume = np.pi * (r_outer**2 - r_inner**2) * L
    mass = volume * rho * 1000  # Convert to grams
    masses.append(mass)

# Find optimal design
optimal_idx = np.argmin([abs(sf - SF_target) for sf in safety_factors])
optimal_OD = OD_range[optimal_idx]
optimal_mass = masses[optimal_idx]
optimal_SF = safety_factors[optimal_idx]

# Plotting
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))

# Mass vs Outer Diameter
ax1.plot(OD_range, masses, 'b-', linewidth=2)
ax1.plot(optimal_OD, optimal_mass, 'ro', markersize=12, label=f'Optimal:
OD={optimal_OD:.1f}mm, m={optimal_mass:.1f}g')
ax1.set_xlabel('Outer Diameter (mm)', fontsize=12)
ax1.set_ylabel('Mass (grams)', fontsize=12)
ax1.set_title('Arm Mass vs Tube Diameter', fontsize=14, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend()

# Safety Factor vs Outer Diameter
ax2.plot(OD_range, safety_factors, 'r-', linewidth=2)
ax2.axhline(y=SF_target, color='g', linestyle='--', linewidth=2,
label=f'Target SF = {SF_target}')
ax2.plot(optimal_OD, optimal_SF, 'ro', markersize=12, label=f'Optimal: SF=
{optimal_SF:.2f}')
ax2.set_xlabel('Outer Diameter (mm)', fontsize=12)
ax2.set_ylabel('Safety Factor', fontsize=12)
```

```
ax2.set_title('Safety Factor vs Tube Diameter', fontsize=14,
fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend()

plt.tight_layout()
plt.savefig('Design_Optimization_Study.png', dpi=300, bbox_inches='tight')
plt.show()

print('\n===== OPTIMIZATION RESULTS =====')
print(f'Optimal outer diameter: {optimal_OD:.1f} mm')
print(f'Optimal mass per arm: {optimal_mass:.1f} grams')
print(f'Safety factor achieved: {optimal_SF:.2f}')
print(f'\nOriginal design: OD=20mm, mass=
{masses[list(OD_range).index(20)]:.1f}g')
print(f'Mass savings: {masses[list(OD_range).index(20)] -
optimal_mass:.1f}g ({(masses[list(OD_range).index(20)] -
optimal_mass)/masses[list(OD_range).index(20)]*100:.1f}%)')
print(f'Total savings for 4 arms: {4*(masses[list(OD_range).index(20)] -
optimal_mass):.1f}g')
```

Lab 3 Deliverables

Submit according to the File Naming and Submission Standards above.

Required Files:

1. **3D FEM Model** - `E022-01-XXXX-YYYY-Lab3-Model.glb`
    - Exported from FreeCAD showing drone arm with combined loading (bending + torsion)
2. **Technical Drawing** - `E022-01-XXXX-YYYY-Lab3-Drawing.pdf`
    - FreeCAD TechDraw showing circular tube (OD 16mm, ID 12mm), length 200mm
    - Section view of tube, material (Carbon Fiber), all three loading components annotated
3. **Rendered Images**

- `E022-01-XXXX-YYYY-Lab3-Isometric.png` (FEM von Mises stress contours)
- `E022-01-XXXX-YYYY-Lab3-SideView.png` (principal stress visualization)

4. **Python Scripts**
   - `E022-01-XXXX-YYYY-Lab3-Analysis.py` (combined stress and Mohr's circle analysis)
   - `E022-01-XXXX-YYYY-Lab3-Optimization.py` (design optimization study)
5. **Analysis Plots**
   - `E022-01-XXXX-YYYY-Lab3-MohrsCircle.png` (Mohr's circle with principal stresses labeled)
   - `E022-01-XXXX-YYYY-Lab3-Optimization.png` (mass vs safety factor, optimal design point)
6. **Hand Calculations** - `E022-01-XXXX-YYYY-Lab3-Calculations.pdf`
   - Combined stress superposition, principal stress calculations, Mohr's circle construction by hand
   - Von Mises and Tresca criteria application, safety factor calculations
7. **Comprehensive Technical Report** - `E022-01-XXXX-YYYY-Lab3-Report.pdf` (5-6 pages)
   - Introduction (drone arm application, combined loading importance)
   - Combined loading methodology (superposition principle, stress components)
   - Principal stress analysis (calculations, Mohr's circle interpretation)
   - Failure criteria comparison (Von Mises, Maximum Stress, Tresca - which is most appropriate for composites?)
   - FEM validation (analytical vs FEM comparison, percent error analysis)
   - Design optimization (parametric study results, optimal tube dimensions)
   - Discussion (dynamic load factors for flight applications, fatigue considerations)
   - Conclusion (final design specification with justified safety factor)

Submission: Compress all files into `E022-01-XXXX-YYYY-Lab3.zip` and submit to the instructor.

Appendices

Appendix A: Group Parameter Assignments

To ensure independent work while maintaining collaborative learning, each group receives slightly different parameters. Your instructor will assign one of these parameter sets to your group.

Lab 1: Robotic Arm - Group Parameters

Group	Length (mm)	Load (N)	Width (mm)	Height (mm)
1	501	542	60	20
2	464	521	60	20

Group	Length (mm)	Load (N)	Width (mm)	Height (mm)
3	510	470	60	20
4	532	536	60	20
5	524	524	60	20
6	537	549	60	20
7	473	452	60	20
8	471	502	60	20
9	451	537	60	20
10	479	487	60	20
11	451	513	60	20
12	509	470	60	20
13	482	525	60	20
14	507	471	60	20
15	538	498	60	20
16	540	508	60	20
17	491	541	60	20
18	509	529	60	20
19	464	511	60	20
20	511	496	60	20

Note: Cross-section dimensions (60mm × 20mm) remain constant for all groups. Only beam length and load vary.

Lab 2: 3D Printer Gantry - Group Parameters

Group	Length (mm)	Load (N)	Outer Dia (mm)	Wall Thickness (mm)
1	818	180	30	3
2	799	201	30	3
3	808	196	30	3
4	801	197	30	3
5	809	207	30	3
6	816	210	30	3
7	837	180	30	3
8	785	191	30	3
9	839	181	30	3
10	784	191	30	3
11	816	215	30	3
12	754	216	30	3
13	826	204	30	3
14	797	212	30	3
15	824	189	30	3

Group	Length (mm)	Load (N)	Outer Dia (mm)	Wall Thickness (mm)
16	784	184	30	3
17	778	189	30	3
18	757	200	30	3
19	797	220	30	3
20	803	212	30	3

Note: Tube dimensions (OD=30mm, wall=3mm) remain constant. Only beam length and load vary.

Lab 3: Drone Arm - Group Parameters

Group	Length (mm)	P_vert (N)	F_horiz (N)	Torque (N·mm)	OD (mm)	ID (mm)
1	200	30	5	1073	16	12
2	183	27	4	1031	16	12
3	203	33	5	1095	16	12
4	187	30	6	1009	16	12
5	216	32	5	901	16	12
6	194	27	4	993	16	12
7	201	30	4	974	16	12
8	216	28	6	972	16	12

Group	Length (mm)	P_vert (N)	F_horiz (N)	Torque (N·mm)	OD (mm)	ID (mm)
9	209	32	4	974	16	12
10	202	29	6	1044	16	12
11	188	30	6	918	16	12
12	208	32	5	963	16	12
13	192	29	5	966	16	12
14	189	27	5	908	16	12
15	216	29	6	1023	16	12
16	186	32	4	921	16	12
17	193	30	5	1023	16	12
18	195	31	4	1089	16	12
19	197	27	6	996	16	12
20	215	33	5	925	16	12

Note: Tube dimensions (OD=16mm, ID=12mm) remain constant. Loading parameters vary by ±10-15%.

Appendix B: Python Environment Setup

Required Libraries:

```
pip install numpy matplotlib scipy pandas
```

Version Verification:

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

print(f"NumPy: {np.__version__}")
print(f"Matplotlib: {matplotlib.__version__}")
print(f"SciPy: {scipy.__version__}")
print(f"Pandas: {pd.__version__}")
```

Recommended: Use Anaconda distribution or virtual environment to avoid conflicts.

Appendix B: FreeCAD FEM Workflow Checklist

Complete Procedure for Finite Element Analysis:

1. **Model Creation** (Part Design workbench)
   - Create 3D solid geometry
   - Ensure no small features that complicate meshing
   - Verify dimensions match analytical model
2. **FEM Setup** (FEM workbench)
   - Create Analysis container
   - Add material from library (or input custom properties)
   - Verify E, ν, and other properties
3. **Boundary Conditions**
   - Apply fixed constraints (select appropriate faces/edges)
   - Apply displacement constraints (roller supports)
   - Check constraint symbols in viewport
4. **Loading**
   - Apply forces (select application points/faces)
   - Apply pressures (select faces)

- Apply torques (use force couples if direct torque unavailable)

5. **Meshing**
   - Choose Gmsh or Netgen
   - Set maximum element size
   - Compute mesh, verify element count
   - Check mesh quality visually
6. **Solving**
   - Write .inp file
   - Run CalculiX solver
   - Monitor console for errors
   - Wait for completion message
7. **Post-Processing**
   - View von Mises stress
   - View displacement magnitude
   - View principal stresses
   - Export images and data
8. **Validation**
   - Compare maximum stress with analytical
   - Check deformation pattern makes physical sense
   - Perform mesh convergence if needed

Appendix C: Common Material Properties

📊 Material Properties Reference Table

Material	E (GPa)	ν	σ_yield (MPa)	σ_ultimate (MPa)	Density (g/cm³)
Aluminum 6061-T6	69	0.33	275	310	2.70
Steel AISI 1020	200	0.29	350	420	7.85
Carbon Fiber (UD)	70-140	0.10	500-600	600-1000	1.55-1.65
Titanium Ti-6Al-4V	114	0.34	880	950	4.43
ABS Plastic	2.3	0.35	40	45	1.05

Notes:

- UD = Unidirectional (properties along fiber direction)
- For composites, use equivalent isotropic properties for preliminary FEM
- Always verify material properties from manufacturer datasheets for final designs

Appendix D: Creating Technical Drawings in FreeCAD

Professional engineering documentation requires proper technical drawings following standard conventions.

📐 TechDraw Workbench Quick Guide

Basic Workflow:

1. **Setup Drawing Page**
    - Switch to TechDraw workbench

- Insert → Default Page (select template: A4 Portrait or Landscape)
- Edit title block with your information

2. **Insert Views**
   - **Front View:** Insert → Insert View → Select part body → Position
   - **Top View:** Insert → Insert View (link to front view for alignment)
   - **Side View:** Insert → Insert View (link to front view)
   - **Section View:** Select edge on view → Insert → Insert Section View → Specify cutting plane
3. **Add Dimensions**
   - **Length Dimension:** Insert → Dimension → Length Dimension → Click two points
   - **Radius Dimension:** Insert → Dimension → Radius → Click arc/circle
   - **Angle Dimension:** Insert → Dimension → Angle → Click two lines
   - **Hole Dimension:** For hollow sections, dimension both OD and ID
4. **Add Annotations**
   - **Material Note:** Insert → Annotation → Rich Text Annotation
     - Example: “Material: Aluminum Alloy 6061-T6”
   - **Load Notes:** Add text boxes indicating applied forces
     - Example: “P = 500 N applied at free end”
   - **Support Notes:** Annotate boundary conditions
     - Example: “Fixed support (all DOF constrained)”
5. **Dimensioning Best Practices**
   - Dimension to centerlines where appropriate
   - Show all critical dimensions (length, cross-section, wall thickness)
   - Use section views to show internal features (hollow tubes)
   - Follow standard dimensioning conventions (ISO or ANSI)
   - Don’t over-dimension (each feature once only)
6. **Export Drawing**

- File → Export as PDF
- Resolution: 300 DPI minimum
- Ensure all text is legible

☑ Drawing Standards Checklist

Your technical drawing must include:

- ✅ **Title block** with name, registration number, date, lab number
- ✅ **Orthographic views** (minimum 2: front + top, or front + side)
- ✅ **Section view** for hollow/internal features
- ✅ **All critical dimensions** clearly labeled with units (mm)
- ✅ **Material specification** annotation
- ✅ **Load annotations** showing magnitude and location
- ✅ **Support/boundary condition** annotations
- ✅ **Scale** indicated (1:1, 1:2, etc.)
- ✅ **Clean line weights** (thick for outlines, thin for dimensions)
- ✅ **Professional appearance** suitable for manufacturing

Appendix E: Troubleshooting Guide

Common Python Errors:

Import Errors | Runtime Errors | Plotting Issues

Error: `ModuleNotFoundError: No module named 'numpy'`

Solution: Install missing library

```
pip install numpy
```

If using Anaconda:

```
conda install numpy
```

Common FreeCAD FEM Errors:

Meshing Failures | Solver Errors | Unrealistic Results

Error: "Meshing failed" or very poor quality mesh

Solutions:

- Simplify geometry (remove fillets, chamfers, small holes)
- Increase max element size
- Use Gmsh instead of Netgen (or vice versa)
- Check for overlapping faces or duplicate geometry

Appendix F: Key Formulas Summary

📐 Essential Structural Analysis Formulas

Beam Bending: $M = \frac{PL}{4}$ (simply supported, center load)

$\sigma = \frac{Mc}{I}$ (flexure formula)

Torsion: $\tau = \frac{Tr}{J}$ (shear stress from torsion)

$\theta = \frac{TL}{GJ} \quad \text{(angle of twist)}$

Principal Stresses: $\sigma_{1,2} = \frac{\sigma_x + \sigma_y}{2} \pm \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2}$

Von Mises Stress: $\sigma_{VM} = \sqrt{\sigma_1^2 - \sigma_1\sigma_2 + \sigma_2^2}$

Tresca (Maximum Shear Stress): $\tau_{max} = \frac{\sigma_1 - \sigma_3}{2}$

Section Properties:

Solid circular: $I = \frac{\pi d^4}{64}, \quad J = \frac{\pi d^4}{32}$

Hollow circular: $I = \frac{\pi(d_o^4 - d_i^4)}{64}, \quad J = \frac{\pi(d_o^4 - d_i^4)}{32}$

Rectangular: $I = \frac{bh^3}{12}, \quad J = \beta bh^3$ (β depends on b/h ratio)

Summary and Course Integration

This laboratory series provides comprehensive hands-on experience with structural analysis methods essential for mechatronics and mechanical engineering practice. You

have developed proficiency in:

Core Competencies Developed:

1. Analytical Methods - Apply equilibrium principles and stress formulas to real engineering problems
2. Programming Skills - Automate calculations and create parametric design studies using Python
3. FEM Simulation - Validate analytical solutions with finite element analysis
4. Design Optimization - Make informed design decisions balancing performance, weight, and cost
5. Technical Communication - Document engineering analysis in professional reports

Integration with Course Topics:

These labs directly support and extend concepts from:

- Lesson 2.1: Shear Force and Bending Moment (Lab 1)
- Lesson 2.2: Bending Stresses in Simple Beams (Lab 2)
- Lesson 2.4: Combined Bending and Torsion (Lab 3)
- Lesson 2.6: Principal Stresses and Failure Analysis (Lab 3)

Next Steps in Your Engineering Education:

- Apply these analysis techniques to your design projects
- Explore advanced FEM capabilities (nonlinear analysis, dynamic simulation, thermal-structural coupling)
- Investigate optimization algorithms for automated design
- Study fatigue analysis and life prediction for cyclic loading
- Learn topology optimization for advanced lightweight structures

Career Relevance

The skills developed in this laboratory series are directly applicable to careers in:

- **Robotics Engineering** (manipulator design, structural optimization)
- **Aerospace Engineering** (aircraft/drone structural analysis)
- **Mechanical Design** (product development with CAD/CAE integration)
- **Automotive Engineering** (chassis and component analysis)
- **Additive Manufacturing** (support structure design and validation)

Comments

Loading comments...

© 2021-2026 SiliconWit®. All rights reserved.

Previous
Chapter 2 Assignments

Next
Planar Mechanics Overview

Privacy · Cookie settings