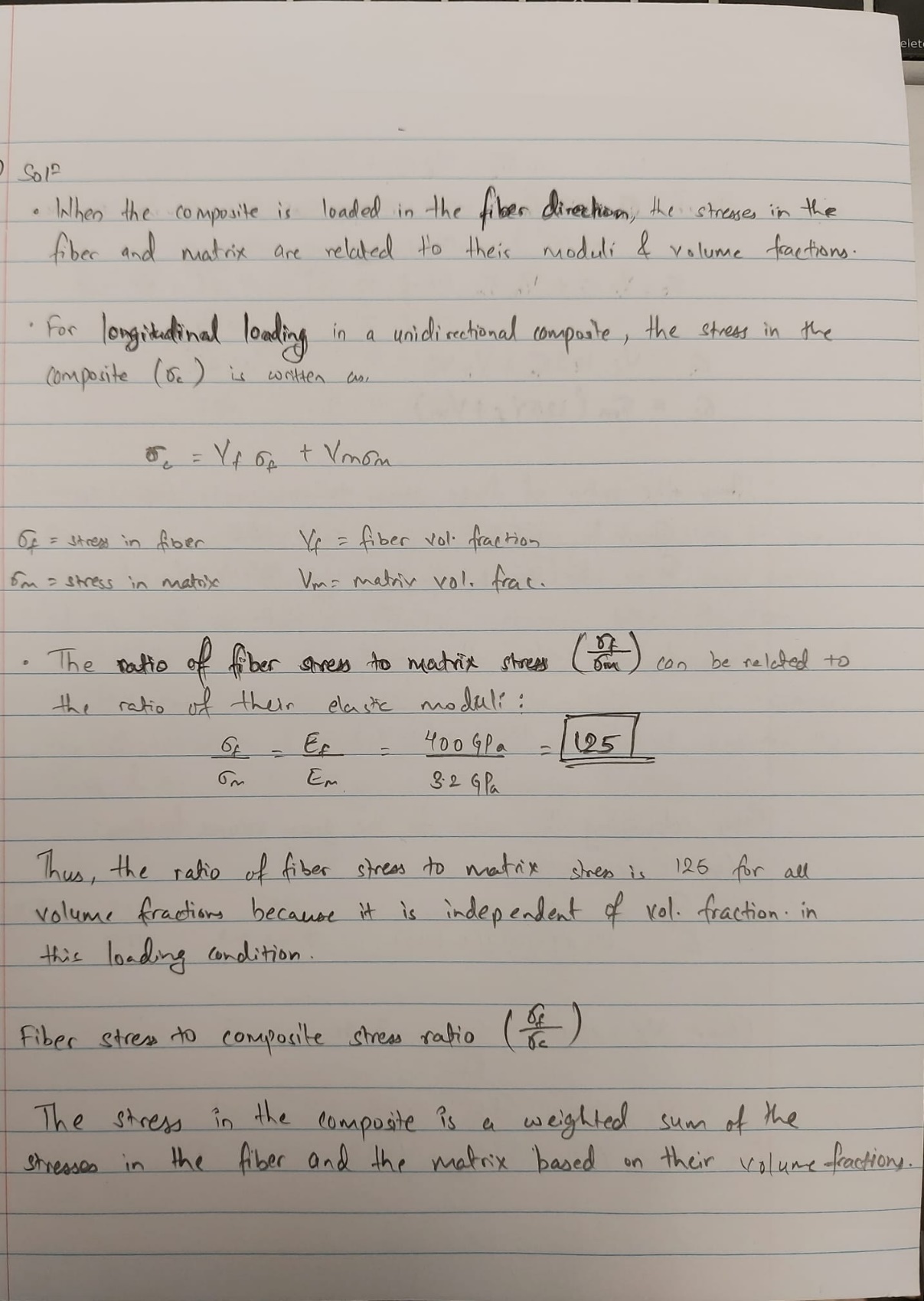
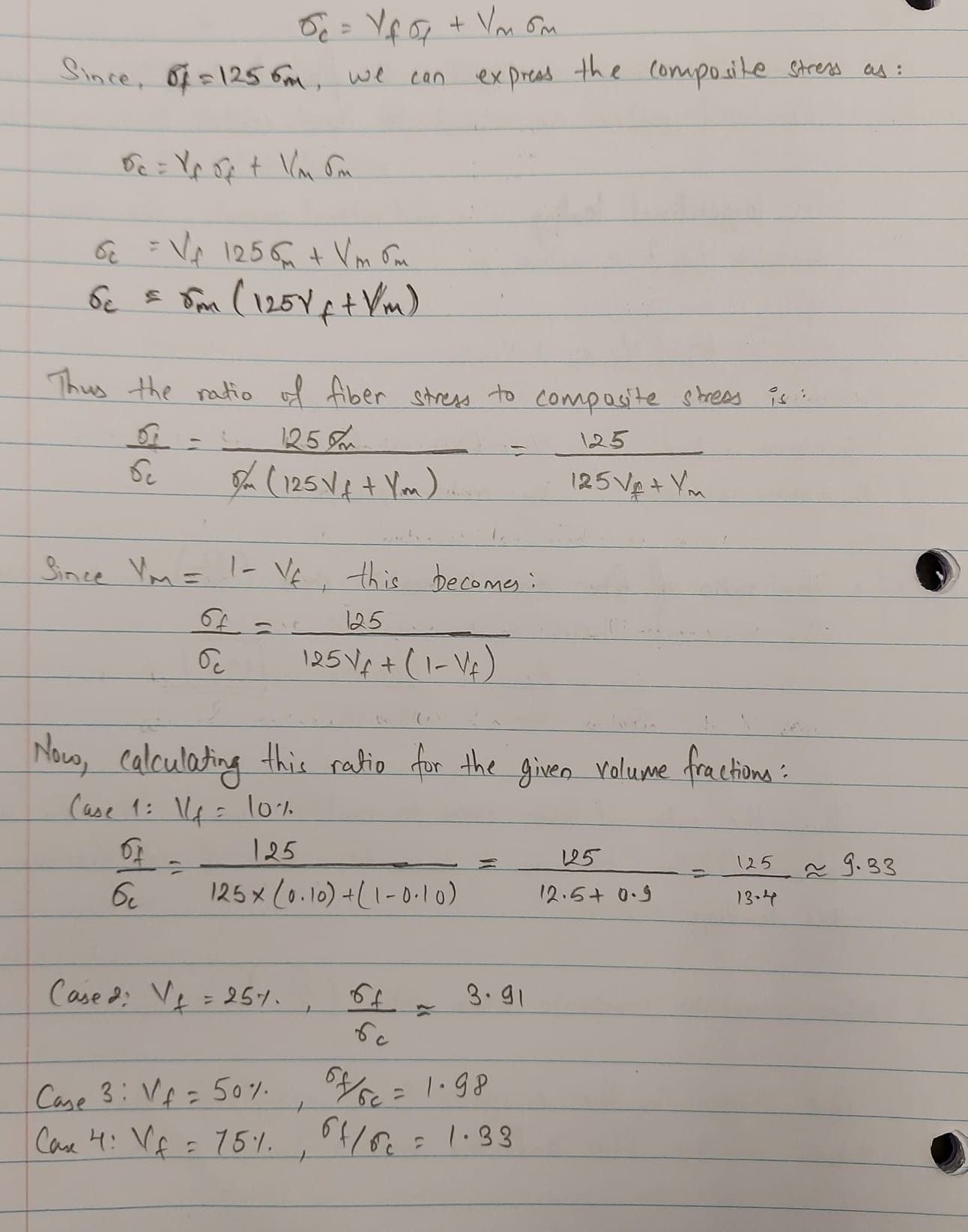
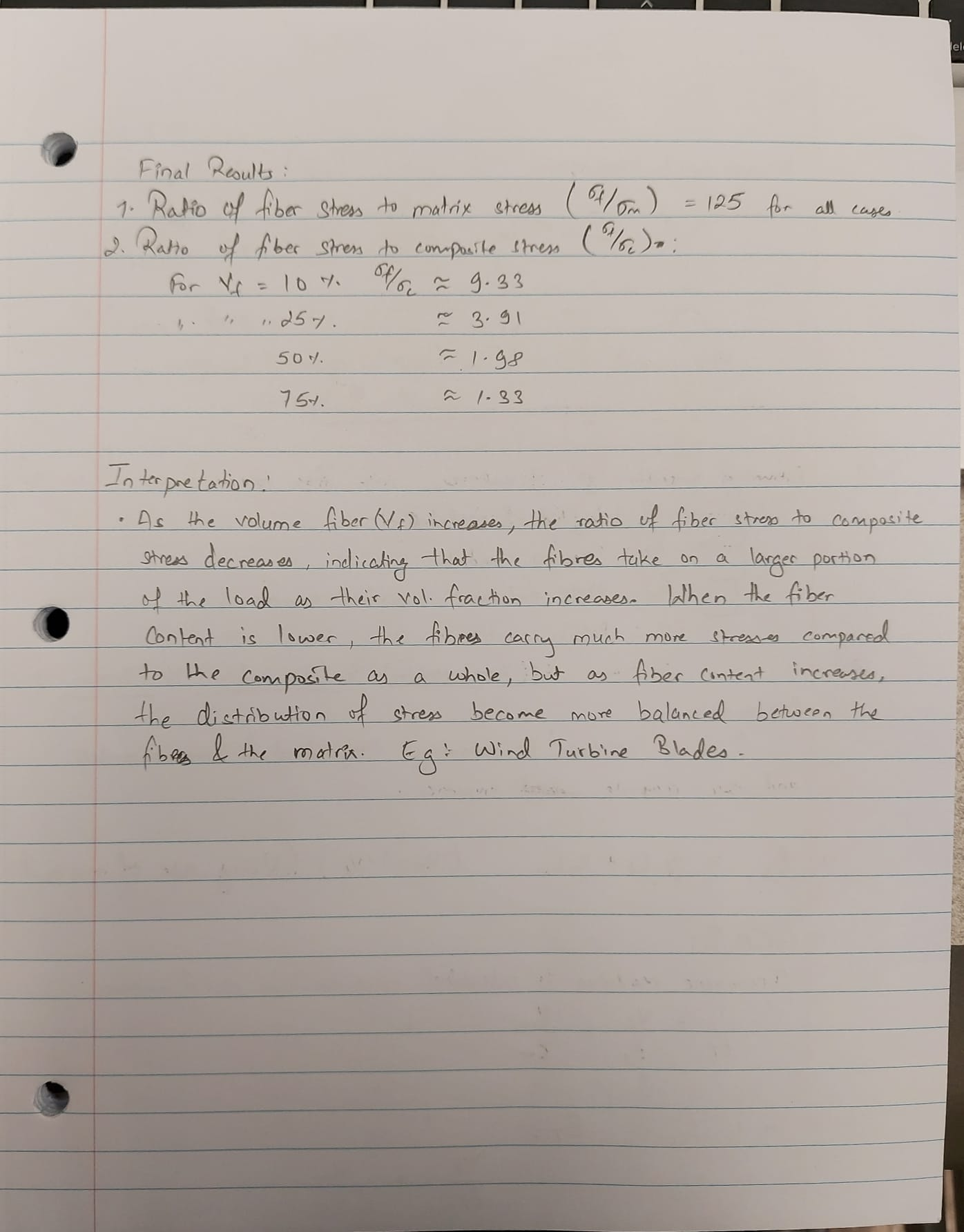
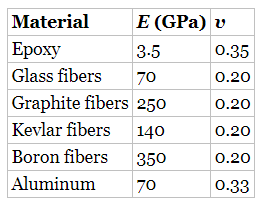
**3.6.** If unidirectional composites are loaded in the fiber direction, calculate the ratios of fiber stress to matrix stress and fiber stress to composite stress when *Vf* = 10%, 25%, 50%, and 75%. Assume  *Ef* = 400 GPa and *Em* = 3.2 GPa.







**3.7.** Estimate *EL*, *ET*, *GLT*, and *νLT*, of glass–epoxy, graphite–epoxy, Kevlar–epoxy, and boron–aluminum composites with *Vf* = 25%, 50%, and 75%. Constituent properties are



For the purpose of calculations, assume all fibers to be isotropic.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

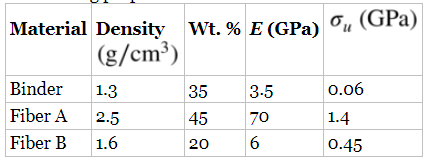
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**3.8.** A rod consists of a binder and two types of fibers with the following properties:



1. What maximum load can this rod carry without rupturing any of the constituents? (Assume that the cross-sectional area of the rod = 10 cm2.)
2. What is the maximum load the rod can carry?
3. What constituent will rupture last?
4. Plot the load–elongation curves for the rod to failure in load- maintained and elongation-maintained tests.

import matplotlib.pyplot as plt

# Material properties from the table (Density, Weight %, Young's Modulus, Tensile Strength)

materials = {

    'Binder': {

        'density': 1.3,  # g/cm^3

        'weight\_percentage': 35,  # %

        'E': 3.5,  # Young's modulus in GPa

        'sigma\_max': 0.06  # Maximum tensile strength in GPa

    },

    'Fiber A': {

        'density': 2.5,  # g/cm^3

        'weight\_percentage': 45,  # %

        'E': 70,  # Young's modulus in GPa

        'sigma\_max': 1.4  # Maximum tensile strength in GPa

    },

    'Fiber B': {

        'density': 1.6,  # g/cm^3

        'weight\_percentage': 20,  # %

        'E': 6,  # Young's modulus in GPa

        'sigma\_max': 0.45  # Maximum tensile strength in GPa

    }

}

# Total cross-sectional area of the rod

total\_area = 10  # cm^2

# Calculate volume fractions based on weight percentages and densities

volume\_fractions = {}

total\_inverse\_density\_weight\_ratio = 0

# First, compute the sum of weight% / density for normalization

for material, props in materials.items():

    total\_inverse\_density\_weight\_ratio += props['weight\_percentage'] / props['density']

# Now calculate the volume fractions and area for each material

for material, props in materials.items():

    volume\_fraction = (props['weight\_percentage'] / props['density']) / total\_inverse\_density\_weight\_ratio

    volume\_fractions[material] = volume\_fraction

    materials[material]['area'] = volume\_fraction \* total\_area  # Cross-sectional area

# Function to calculate maximum load before failure

*def* calculate\_max\_load(*materials*):

    max\_loads = {}

    for material, properties in *materials*.items():

        max\_stress = properties['sigma\_max']  # Maximum stress the material can handle

        area = properties['area']  # Cross-sectional area of the material

        # Calculate the maximum load the material can carry (force = stress \* area)

        max\_load = max\_stress \* area  # Load in GPa \* cm² = GPa \* cm²

        max\_loads[material] = max\_load  # Store the max load for each material

    return max\_loads

# Get the maximum loads for each material

max\_loads = calculate\_max\_load(materials)

# Find the limiting load (smallest maximum load)

limiting\_material = min(max\_loads, *key*=max\_loads.get)

limiting\_load = max\_loads[limiting\_material]

print("Maximum loads each material can carry before failure:")

for material, load in max\_loads.items():

    print(*f*"{material}: {load*:.2f*} GPa\*cm²")

    print(*f*"{material} Area: {materials[material]['area']*:.2f*} cm²")

print(*f*"\nThe limiting material is '{limiting\_material}' which fails first.")

print(*f*"The maximum load the rod can carry is {limiting\_load*:.2f*} GPa\*cm².")

# Plotting load vs. elongation for each material (just for illustration)

*def* plot\_load\_elongation(*materials*, *max\_load*):

    loads = []

    elongations = {material: [] for material in *materials*}

    increments = 50  # Number of increments for the plot

    for load in range(1, increments+1):

        current\_load = (load / increments) \* *max\_load*

        loads.append(current\_load)

        for material, props in *materials*.items():

            elongation = current\_load / (props['E'] \* props['area'])  # Simplified elongation formula

            elongations[material].append(elongation)

    for material, elong in elongations.items():

        plt.plot(loads, elong, *label*=material)

    plt.xlabel("Load (GPa\*cm²)")

    plt.ylabel("Elongation")

    plt.legend()

    plt.title("Load vs Elongation for Different Materials")

    plt.show()

# Plot load-elongation curves

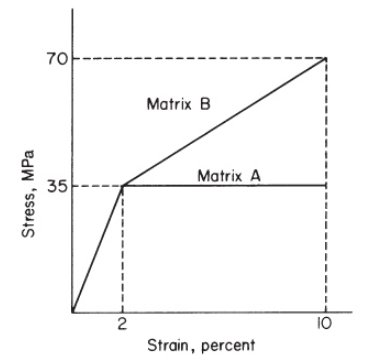
plot\_load\_elongation(materials, limiting\_load)

A screen shot of a computer

Description automatically generated

**3.9.** Two unidirectional composites are fabricated with glass fibers (*Vf* = 50%) and matrices *A* and *B*, whose stress–strain curves are shown in [Figure 3.49](https://jigsaw.vitalsource.com/books/9781119389972/epub/OPS/c03.xhtml?create=true#c03-fig-0049). The glass fibers are elastic up to failure and have an elastic modulus of 70 GPa and ultimate tensile strength of 2.8 GPa. If the composites are stressed parallel to the fibers, calculate

1. the composite stress at 1% and 4% strains for each composite, and
2. minimum and critical fiber volume fractions for each composite.



[**Figure 3.49**](https://jigsaw.vitalsource.com/books/9781119389972/epub/OPS/c03.xhtml?create=true#R_c03-fig-0049). Stress-strain curves for matrices A and B (Exercise Problem 3.9).

# Material properties

E\_f = 70  # GPa for the glass fiber

sigma\_u\_f = 2800  # MPa (2.8 GPa converted to MPa)

V\_f = 0.5  # Fiber volume fraction (50%)

V\_m = 1 - V\_f  # Matrix volume fraction

# Matrix stress from the graph at 1% strain and 4% strain for A and B

stress\_matrix\_A\_1 = 17.5  # MPa

stress\_matrix\_A\_4 = 35  # MPa

stress\_matrix\_B\_1 = 35  # MPa

stress\_matrix\_B\_4 = 70  # MPa

# Fiber stress at 1% and 4% strain using Hooke's Law (E\_f \* strain)

strain\_1 = 0.01

strain\_4 = 0.04

fiber\_stress\_1 = E\_f \* strain\_1 \* 1000  # Convert to MPa

fiber\_stress\_4 = E\_f \* strain\_4 \* 1000  # Convert to MPa

# Composite stress calculations using rule of mixtures

composite\_A\_1 = V\_f \* fiber\_stress\_1 + V\_m \* stress\_matrix\_A\_1

composite\_A\_4 = V\_f \* fiber\_stress\_4 + V\_m \* stress\_matrix\_A\_4

composite\_B\_1 = V\_f \* fiber\_stress\_1 + V\_m \* stress\_matrix\_B\_1

composite\_B\_4 = V\_f \* fiber\_stress\_4 + V\_m \* stress\_matrix\_B\_4

# Print the results

print(*f*"Composite A stress at 1% strain: {composite\_A\_1*:.2f*} MPa")

print(*f*"Composite A stress at 4% strain: {composite\_A\_4*:.2f*} MPa")

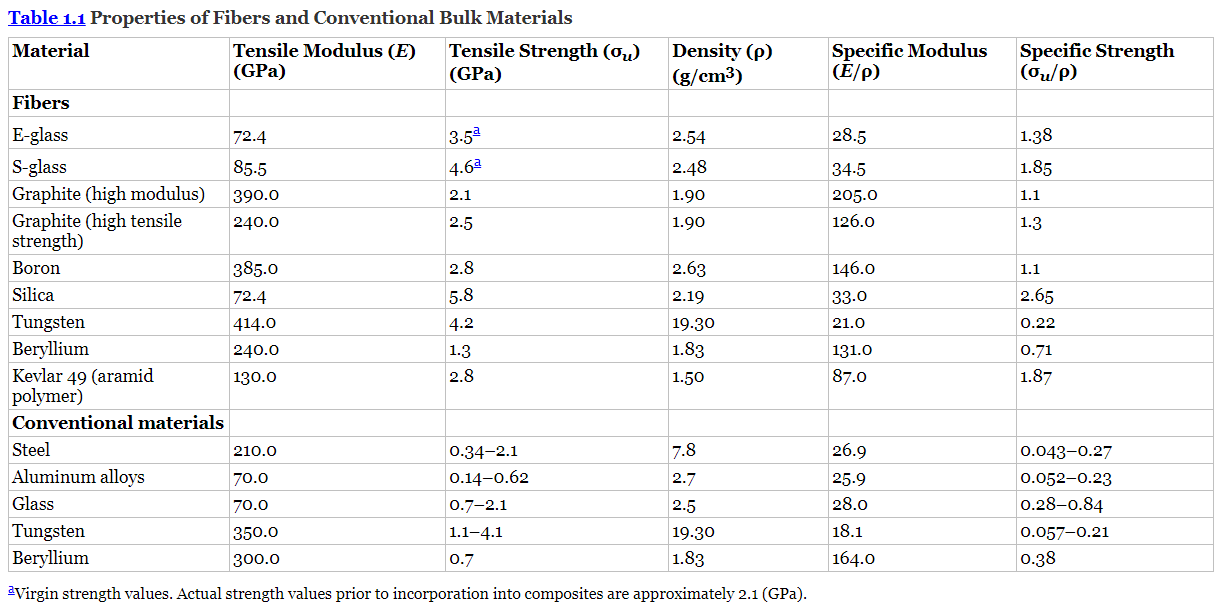
print(*f*"Composite B stress at 1% strain: {composite\_B\_1*:.2f*} MPa")

print(*f*"Composite B stress at 4% strain: {composite\_B\_4*:.2f*} MPa")

A screenshot of a computer screen

Description automatically generated

**3.11.** A tension link is made of an aluminum alloy. If aluminum is replaced by a unidirectional graphite–epoxy composite, calculate the required fiber volume fraction in the composite so that its longitudinal modulus matches the modulus of aluminum alloy. What is the percentage weight saving in this material replacement? Use the properties given in [Table 1.1](https://jigsaw.vitalsource.com/books/9781119389972/epub/OPS/c01.xhtml#c01-tbl-0001). The elastic modulus of epoxy is 3.5 GPa, and its density is 1.2 g/cm3.



# Given Data

E\_aluminum = 70  # GPa

rho\_aluminum = 2.7  # g/cm³

E\_f = 390  # GPa (Graphite Fiber)

rho\_f = 1.9  # g/cm³

E\_m = 3.5  # GPa (Epoxy Matrix)

rho\_m = 1.2  # g/cm³

# Step 1: Calculate the required fiber volume fraction (V\_f)

E\_c = E\_aluminum  # We want the composite modulus to match aluminum

# Using Rule of Mixtures: E\_c = V\_f \* E\_f + (1 - V\_f) \* E\_m

# Solve for V\_f

V\_f = (E\_c - E\_m) / (E\_f - E\_m)

# Matrix Volume Fraction (V\_m)

V\_m = 1 - V\_f

# Step 2: Calculate the composite density

rho\_c = V\_f \* rho\_f + V\_m \* rho\_m

# Step 3: Calculate the percentage weight saving

weight\_saving = (rho\_aluminum - rho\_c) / rho\_aluminum \* 100

# Display Results

print(*f*"Required Fiber Volume Fraction (V\_f): {V\_f*:.4f*}")

print(*f*"Composite Density (rho\_c): {rho\_c*:.4f*} g/cm³")

print(*f*"Percentage Weight Saving: {weight\_saving*:.2f*}%")

A black background with white text

Description automatically generated