EAE 135 Project 2

Jackson Wright, Kaitaro Hori, Sydney Makar

University of California, Davis

March 3rd, 2025

1 Mathematical Derivation

1. Symbolically determine P1 and P2.

1. $P_{Thrust} = 1250\ kN$

$P_{lift} = 1400\ kN$

$P_{weight} = 500\ kN$

$$P_2 = P_{lift,x_2} + P_{thr,x_2} + \cancel{P_{weight,x_2}}^{\,0}$$

$$= P_{lift} \cdot \cos\alpha + 0$$

$$= 1400 \cdot \cos(5^\circ)$$

$$= 1394.6723\ kN$$

$$P_1 = P_{lift,x_1} + P_{thr,x_1} + \cancel{P_{weight,x_1}}^{\,0}$$

$$= P_{thr} + P_{lift} \sin\alpha$$

$$= -1250 + 1400 \sin(5^\circ)$$

$$= -1127.982\ kN$$

※ Neglect the axial component for weight

2. Symbolically determine the axial displacement, axial strain, and axial stress of the fuselage due to the axial load.

2.

Govern. Eqn:

$$\frac{d}{dx_1}\left[S\,\frac{du_1}{dx_1}\right] = -P_1(x_1)$$

$$\frac{d}{dx_1}\left[S\,\frac{du_1}{dx_1}\right] = 0$$

- $u_1 = 0$ at $x_1 = 0$
- $S\frac{du_1}{dx_1} = (122.02 - P_{thruster})$ at $x_1 = L$

$$S\frac{du_1}{dx_1} = C_1 = 122.02 - 1250$$

$$\Rightarrow C_1 = -1128$$

$$\frac{du_1}{dx_1} = \frac{-1128}{S}$$

$$u_1(x_1) = \frac{-1128x}{S} + C_2$$

$u_2(0) = 0 \therefore C_2 = 0$

$$u_1(x_1) = \frac{-1128x}{S}$$

$$\varepsilon_1 = \frac{du_1}{dx_1} = \frac{-1128}{S}$$

$$\sigma_1 = E\cdot\left(-\frac{1128}{S}\right)$$

As shown in the above derivation, axial displacement is derived by integrating twice the governing equation of an Euler-Bernoulli beam under axial loading. The constants of integration are determined using the boundary conditions $u_1 = 0$ at $x_1 = 0$ and $S\frac{u_1}{dx_1} = P_1$ at $x = L$.

3. Numerically determine the axial stiffness (S) and the bending stiffness (H_{33}^C).

3.

$$S = \int_A E\,dA = E_{Al}A_{Al} + \sum E_{CFRP}A_{CFRP}$$

$$= 5.4642 \text{ GPa m}^2$$

$$H_{33}^C\,Al = E_{Al} \cdot \pi(1.495^2 - 1.490^2)$$

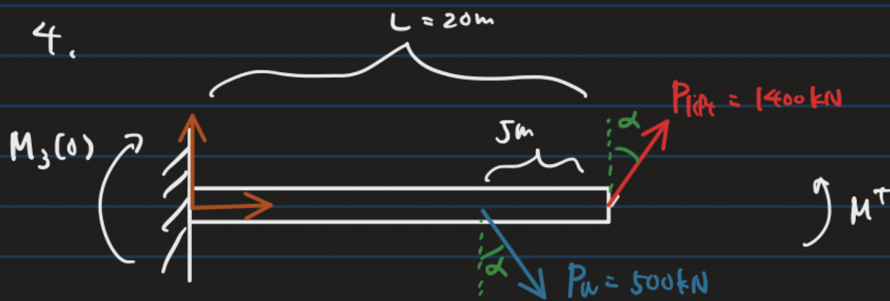
$$= 3.235$$

$$\Rightarrow \quad H_{33}^C = 6.102 \text{ GPa m}^2$$

H_{33}^C for Aluminum is calculated as shown above. To calculate H_{33}^C for the CFRP, concepts from classical laminate beam theory were employed. Using a rotation matrix and Reuter's matrix, $\bar{Q}$ is computed. Taking the inverse of $\bar{Q}$ gives $\bar{S}$. $(E_{x1x1})_{ith}$ is then determined for each layer of the composite as as $\frac{1}{(\bar{S}_{11})_{ith}}$. In practice, these values were computed in MATLAB, calling a function that calculates $\bar{Q}$ for each layer of the composite, then converting these values into $\bar{S}$ then $(E_{x1x1})_{ith}$. The code for this is in the appendix.

4. Symbolically determine the bending moment contribution ($M_{3_{max}}$); use it for the calculation of the axial stress due to bending.

4.



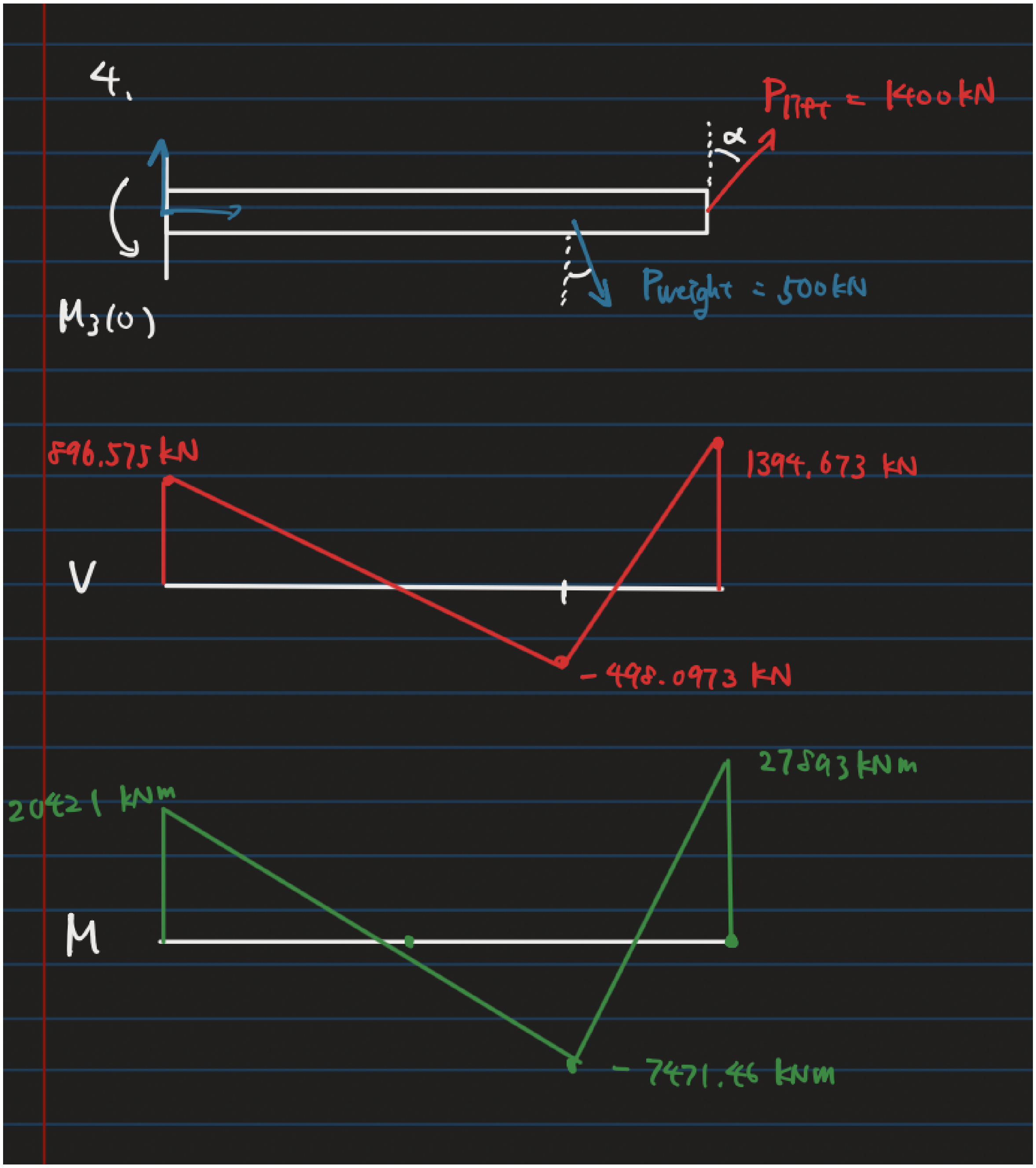
$$\sum M_3 = -M_3(0) - P_w \cos(\alpha)\cdot(L-5) + P_{lift}\cos(\alpha)\cdot L = 0$$

$$M_3(0) = -500\cos(5)\cdot 15 + 1400\cos(5)\cdot 20$$

$$M_3(0) = 20421.9913 \text{ kNm}$$

$$= M_{3_{Max}}$$

5. Draw the shear and bending moment diagram



2 Report

2.1 Plots

1. Plot the axial stress distribution due to the axial load (compression).

   As we derived from the Deliverable 1, the axial stress due to the axial load is;

$$\sigma_1 = E * (\frac{-1128}{S}) \tag{1}$$

   As we derived $M_3(0) = 22422\ kN * m$ and $S = 5.4672\ GPa$ from the Deliverable 1, we plug those values and E_{11}^i of each layer into the Equation (1)

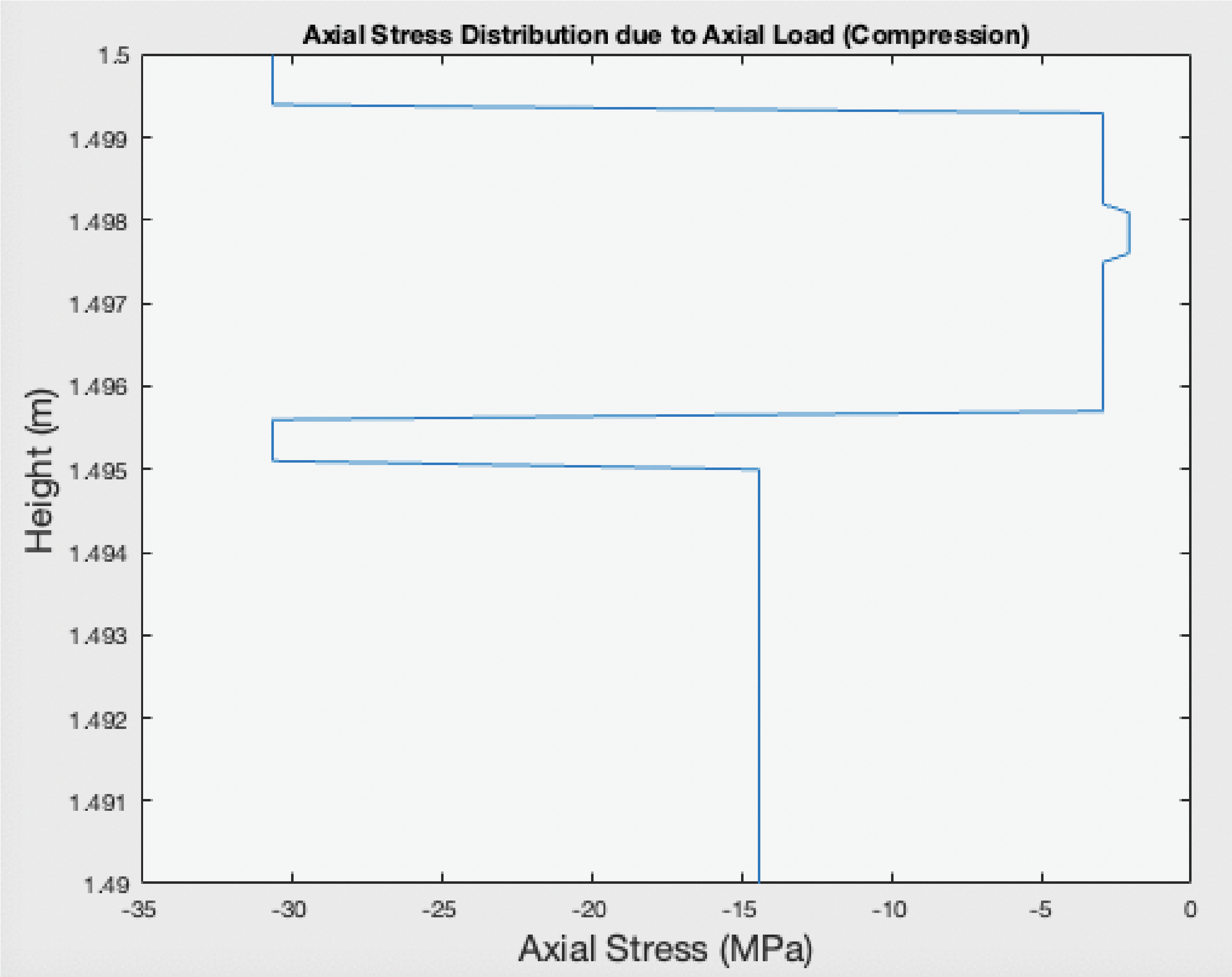


Figure 1: Position Along Beam Height vs. Top Axial Stress

2. Plot the axial stress distribution due to the bending at the wall (compression) of Position Along Beam Height vs. axial stress due to bending

$$\sigma_1(0, x_2, x_3) = -\frac{(E^i_{x_1x_1}) * x_{2i} * M_3(0)}{H^c_{33}} \quad (2)$$

Under bending, a cantilever has the highest bending moment at its root. The Equation (2) gives us the axial stress distribution along the x_2 locations in a rectangular cross-section.

As we calculated above, $H^c_{33} = 6.102\ GPa$. Plugging in derived H^c_{33}, $M_3(0)$, and Young's Moduli for each layer into the Equation (2), we obtained a graph below;

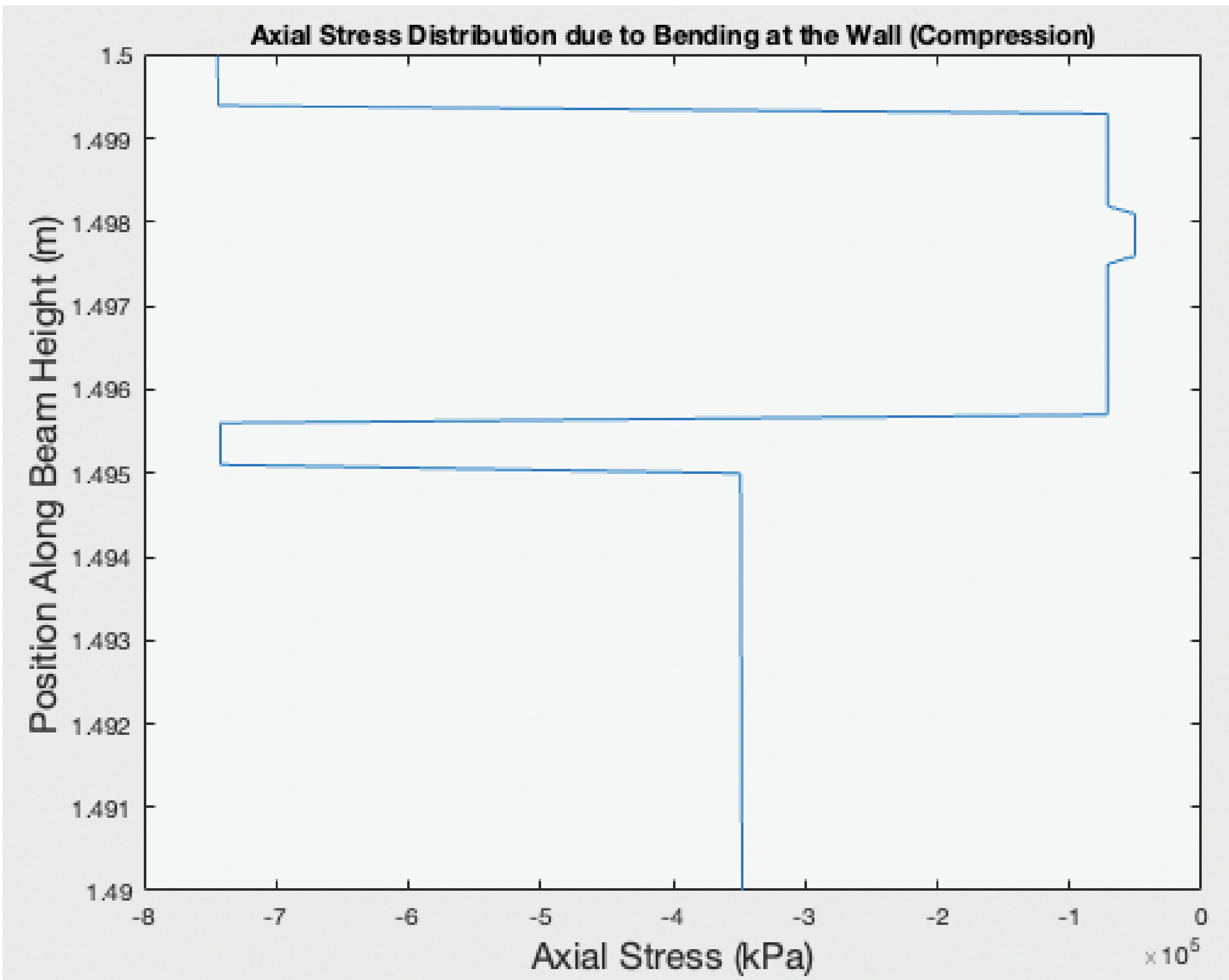


Figure 2: Position Along Beam Height vs. Axial Stress due to Bending (compression)

3. Plot the axial stress distribution due to bending at the wall (tension), as Position Along Beam Height vs. axial stress due to bending

   We use the same method as the compression and the Equation (2). Axial stress in tension is essentially the opposite (negative) of axial stress in compression. Therefore, the graph shows the opposite sign for each plot.

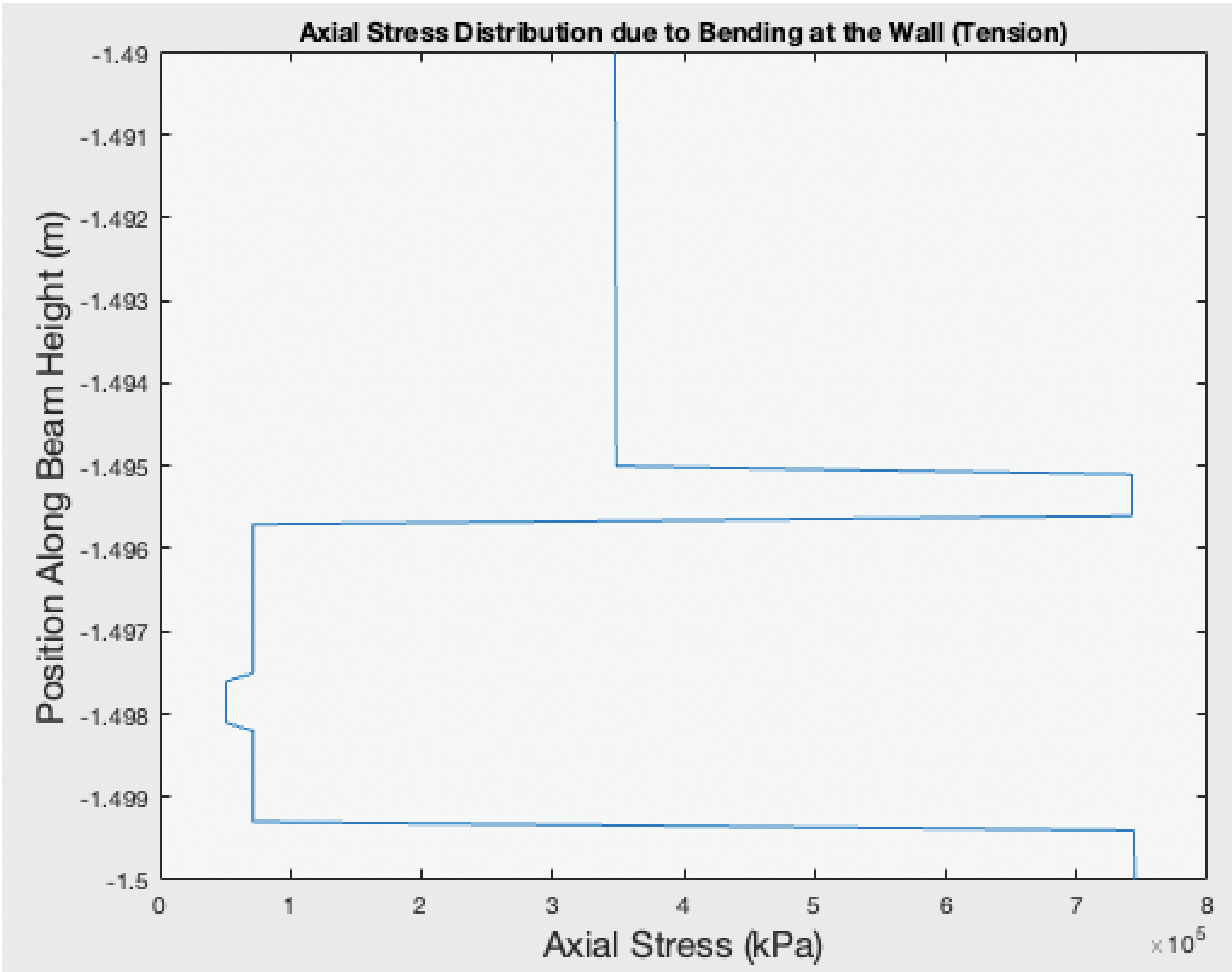


Figure 3: Position Along Beam Height vs. Axial Stress due to Bending (tension)

4. Plot the total axial stress distribution at the wall (Compression), as Position Along Beam Height versus total axial stress.

   The total axial stress can be obtained from combining Equation (1) and Equation (2). Therefore, simply summing the axial stress due to bending and the axial stress due to axial loading gives the total axial stress. Since the axial stress due to bending compresses the top of the beam and puts the bottom of the beam in tension, the total axial stress is slightly greater at the top of the beam that at the bottom. This is shown by the differences in magnitudes at each layer between Figure 4 and Figure 5.

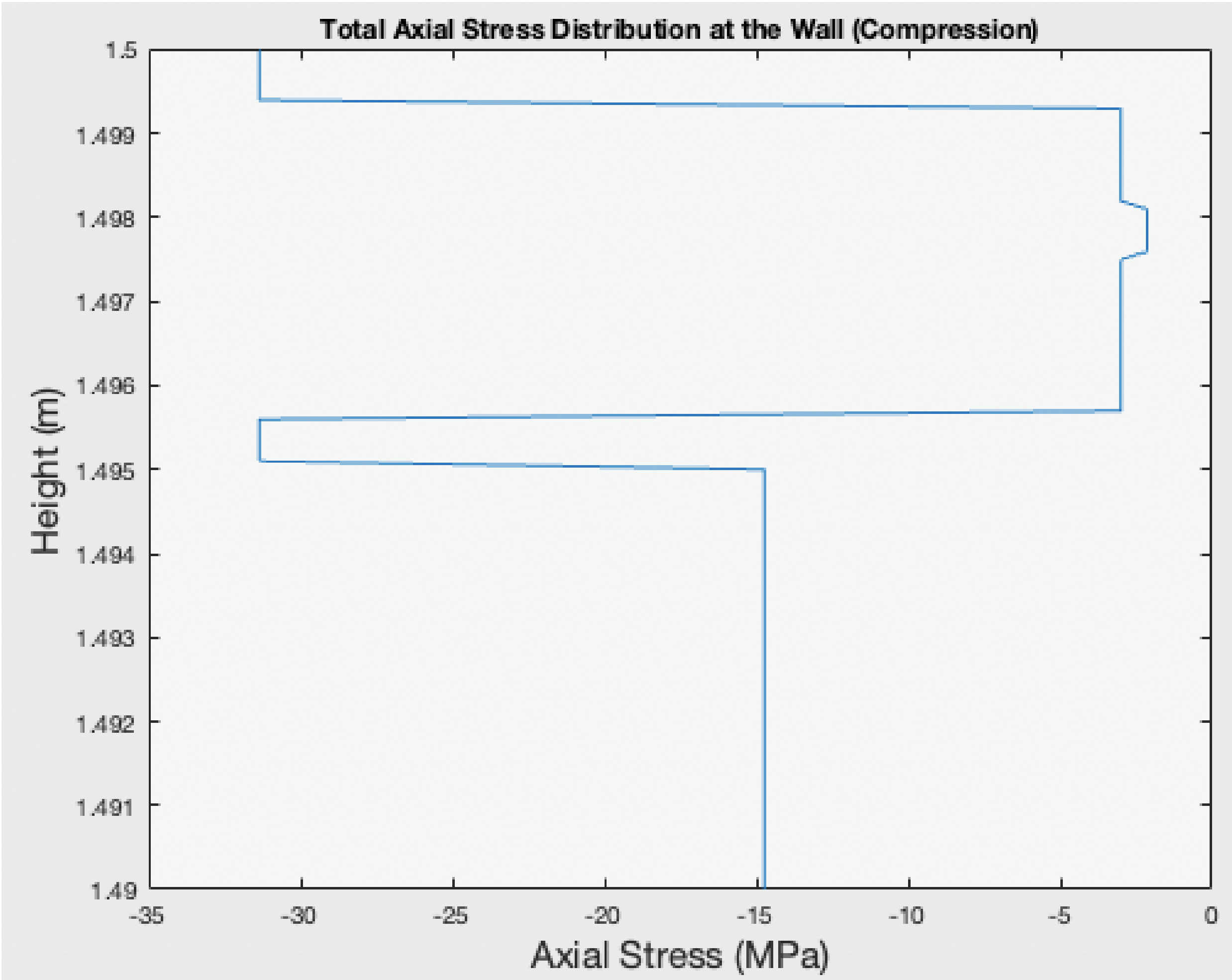


Figure 4: Position Along Beam Height vs. Total Axial Stress

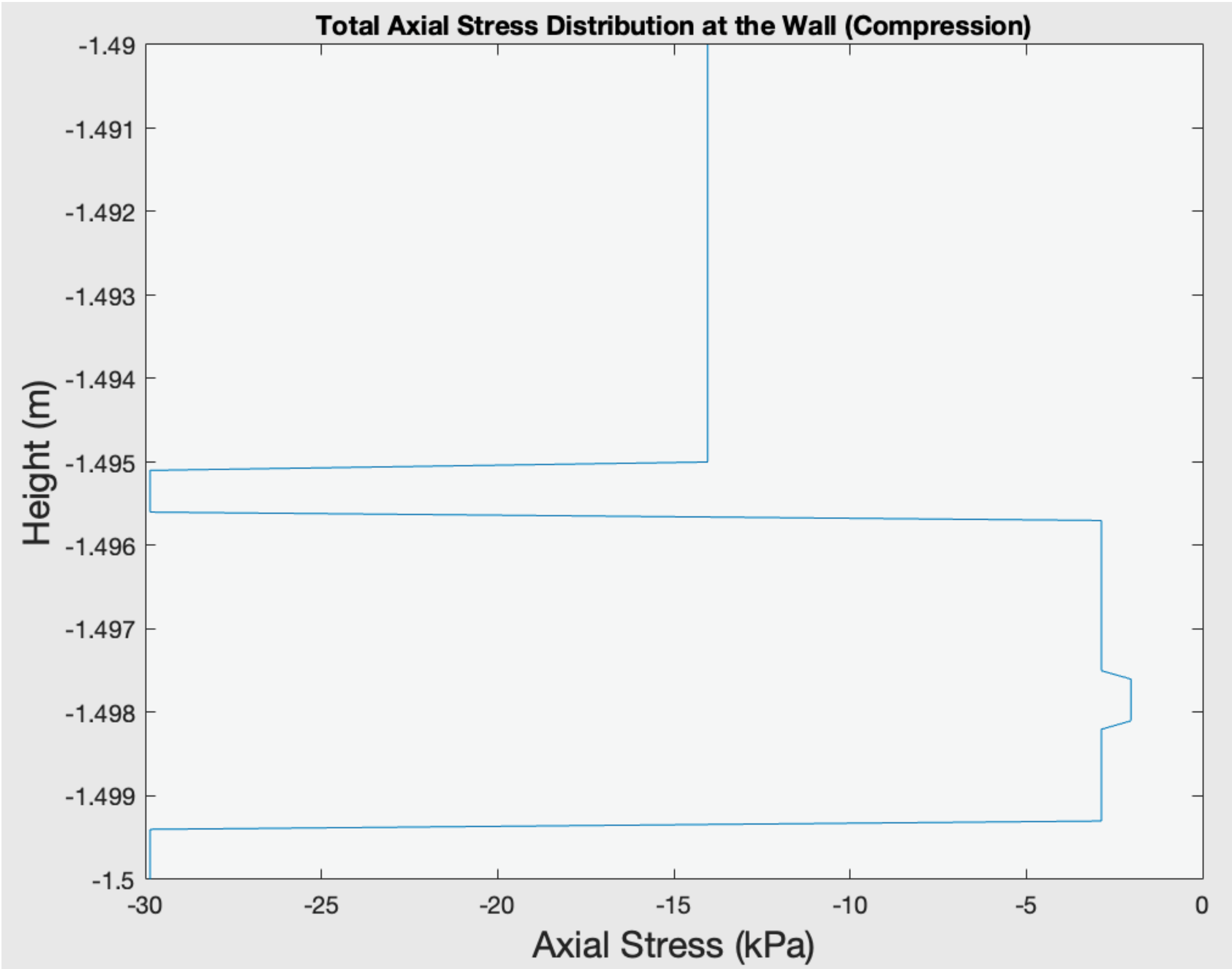


Figure 5: Position Along Beam Height vs. Total Axial Stress

5. Plot the safety factor along the wall of Wall Position (X2) vs Safety Factor (SF)
   As safety factor can be derived as;

$$SF = \frac{\sigma_{strength}}{\sigma_1} \tag{3}$$

   Use the total axial stress obtained above in Equation (3).

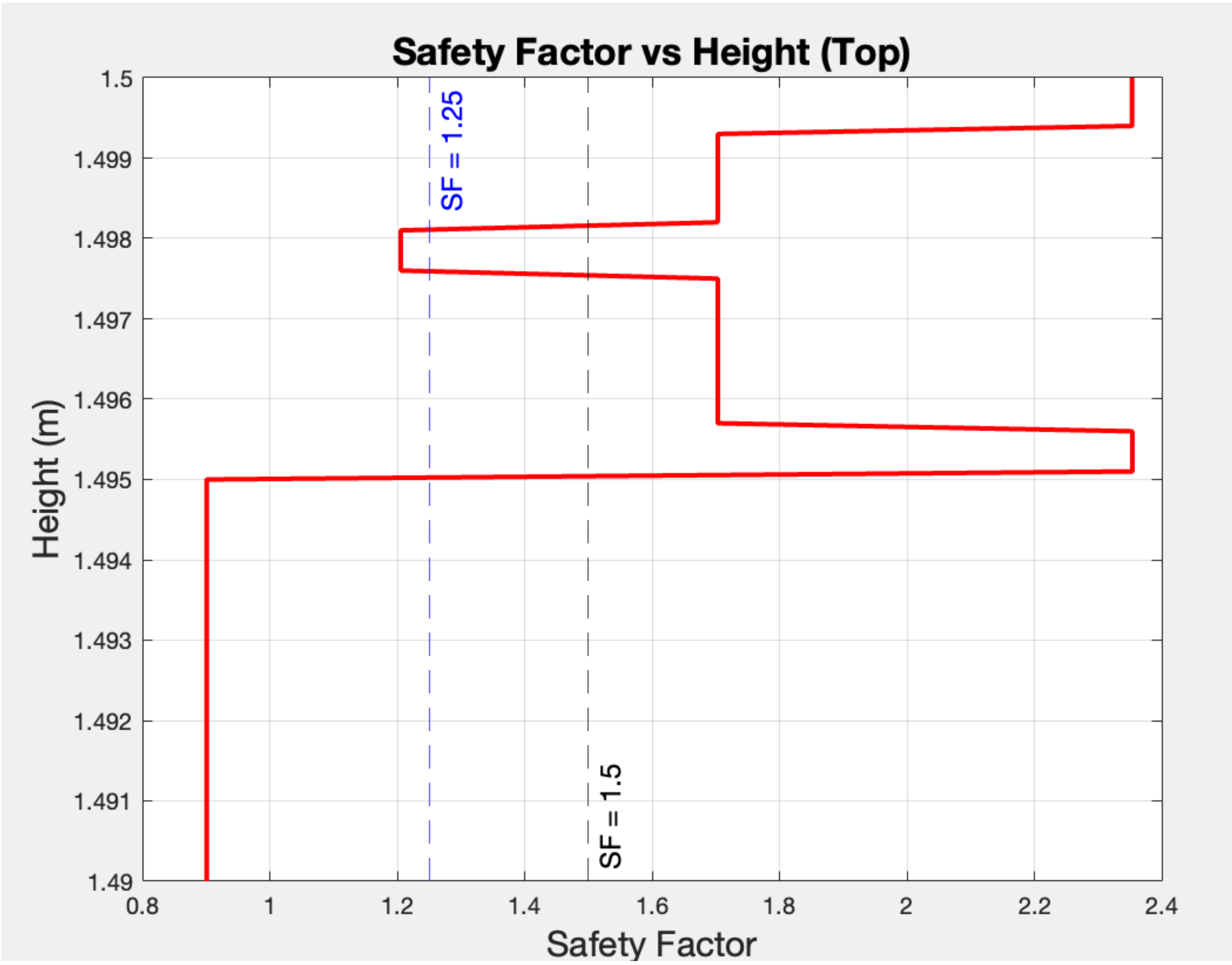


Figure 6: Wall Position (X_2) vs Safety Factor (SF)

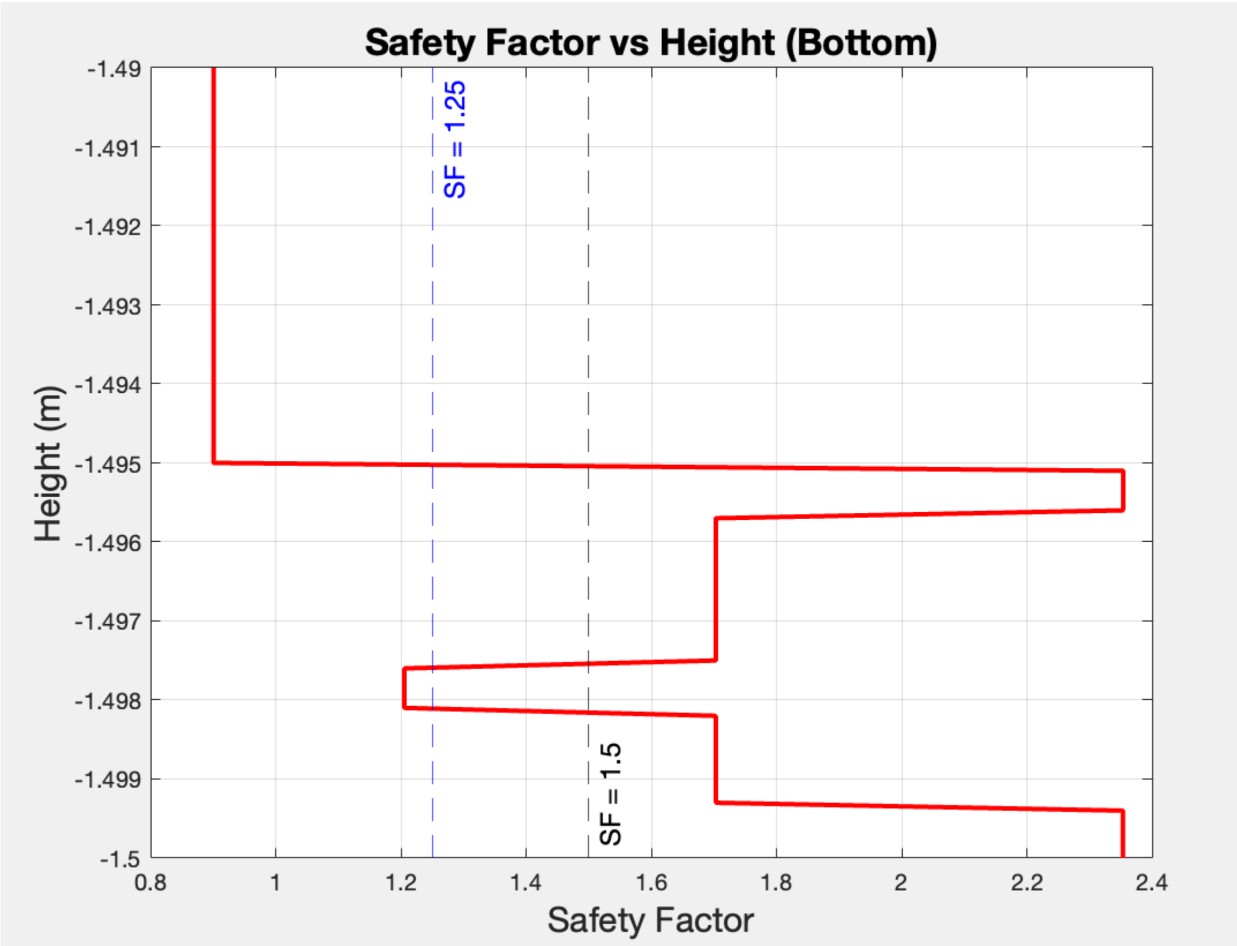


Figure 7: Wall Position (X_2) vs Safety Factor (SF)

2.2 Discussion

6. Compare the safety factor of this design for the aircraft application and for the spacecraft application. Discuss why these are different, and what parameters contribute to how safety factors are determined.

   - Axial compressive strength for 0 deg. = axial tensile strength X_C = -184 ksi = -1268.6358 MPa for composites, -226 ksi = -1558.21576 MPa for aluminum.
   - Axial compressive strength for 90 deg. = transverse compressive strength Y_C = 24.4 ksi = 168.232 MPa for composites, 28.9 ksi = 199.259 MPa for aluminum.
   - Axial compressive strength for +45, -45 deg. = transverse compressive strength Y_C = 24.4 ksi = 168.232 MPa for composites, 28.9 ksi = 199.259 MPa for aluminum.

   As the entire fuselage is compressed, we used axial compressive strength from the Figure 8, Tables with composite materials properties. All the values above are the maximum compressive axial strength for each layer of material needed to calculate safety factor. As the fuselage at its point of connection is in compression $\forall r$, the axial tensile strengths were not needed.

Figure 6 and 7 show the sign of possible fracture within the diameter of 1.495 m, which is a layer of aluminum. Also, about the middle of the CFRP laminate shows a sign of possible failure as the safety factor line is below 1.25, which is the safety factor for spacecraft.

7. What would your team change about the fuselage design so that the fuselage would pass the safety factor requirements for both aircraft and spacecraft?

   Each layer of the CFRP laminate could be increased to past the safety factor requirements. This would increase the outer diameter of the fuselage and make it less susceptible to fracture. However, it would also increase the weight of the fuselage increasing transverse loading. As shown in the safety factor graphs above, at both top and bottom side of the fuselage, aluminum layer most likely fails under the diameter of 1.495 m. Therefore, to reinforce the structure of the fuselage, we need consider the increase of aluminum layer thickness.

3 Recorded Presentation

https://video.ucdavis.edu/media/Project2/1_rz51ydqx

Contributions of each member:

- Deliverable 1: Kaitaro and Jackson worked together
- Deliverable 2: All three members worked together
- Deliverable 3: Sydney

4 Appendix

4.1 MATLAB Code

Below is the MATLAB code used for the plots and derivations:

```
E11 = 148.24;%GPa
E22 = 10.07; %GPa
G12 = 5.58; %Gpa
v12 = 0.30;
v21 = v12*E22/E11;
Rc = 1.5; %m
Ra = 1.495; %m
Ri = 1.490; %m
EcAl = 69.69; %GPa
EtAl = 68.95; %GPa

```

```
Q = [(E11/(1-v12*v21)), (v21*E11)/(1-v12*v21), 0;
    (v12*E22/(1-v12*v21)), (E22/(1-v12*v21)), 0;
    0, 0, G12];



function Qbar = calculateQbar(beta,Q)

beta = deg2rad(beta);

T = [cos(beta)^2, sin(beta)^2, 2*sin(beta)*cos(beta);
    sin(beta)^2, cos(beta)^2, -2*sin(beta)*cos(beta);
    -sin(beta)*cos(beta), sin(beta)*cos(beta), cos(beta)^2-sin(beta)^2];

R = [1 0 0 ; 0 1 0; 0 0 2];

Qbar = inv(T)*Q*R*T*inv(R);

end

Qbar0 = calculateQbar(0,Q);

Qbar90 = calculateQbar(90,Q);

Qbar45 = calculateQbar(45,Q);

QbarNeg45 = calculateQbar(-45,Q);


% Checks
% Qbar90(1,1) - Q(2,2)
% Qbar45(1,1) - Qbar45(2,2)
% QbarNeg45(1,1) - QbarNeg45(2,2)
% Qbar45(3,3) - Q(3,3)
% QbarNeg45(3,3) - Q(3,3)

Sbar0 = inv(Qbar0);
Sbar90 = inv(Qbar90);
Sbar45 = inv(Qbar45);
SbarNeg45 = inv(QbarNeg45);

% Axial Stiffness Ex1x1 of Each Layer
E0 = 1/(Sbar0(1,1));
E90 = 1/(Sbar90(1,1));
E45 = 1/(Sbar45(1,1));
ENeg45 = 1/(SbarNeg45(1,1));

% Bending Stiffness
Ex1x1 = [E0, E45, ENeg45, E90, E90, ENeg45, E45, E0];
H33cCFRP = 0;
ri = 1.495; %m
layerThickness = (Rc-Ra)/8;
H33cCFRPArray = [];
for n = 1:8
```

```
    E = Ex1x1(n);
    riplus = ri + layerThickness;
    H33cCFRPArray(end+1) = E*pi*(riplus^4-ri^4)/4;
    ri = ri + layerThickness;
end

H33cAl = EcAl*pi*(Ra^4-Ri^4)/4;

H33c = H33cAl + sum(H33cCFRPArray);


% Stress Distribution due to Bending (Compression)
r = 1.490;
rArray = [];
sigma1Array = [];

M30 = 20422; %kNm

while r <= 1.5
    if r< 1.495
        E = EcAl;
    elseif r < (1.495 + layerThickness)
        E = Ex1x1(1);
    elseif r < (1.495 + 2*layerThickness)
        E = Ex1x1(2);
    elseif r < (1.495 + 3*layerThickness)
        E = Ex1x1(3);
    elseif r < (1.495 + 4*layerThickness)
        E = Ex1x1(3);
    elseif r < (1.495 + 5*layerThickness)
        E = Ex1x1(5);
    elseif r < (1.495 + 6*layerThickness)
        E = Ex1x1(6);
    elseif r < (1.495 + 7*layerThickness)
        E = Ex1x1(7);
    else
        E = Ex1x1(8);
    end

    sigma1 = -E*r*M30/H33c;
    rArray(end+1) = r;
    sigma1Array(end+1) = sigma1;
    r = r + 0.0001;

end

% Axial stress due to bending at the wall (compression)
figure(1)
plot(sigma1Array,rArray)
xlabel('Axial Stress (kPa)','FontSize',15);
ylabel('Position Along Beam Height (m)','FontSize',15);
title('Axial Stress Distribution due to Bending at the Wall (Compression)'
    )

```

```
% Axial stress due to bending at the wall (tension)
figure(2)
plot(-sigma1Array,-rArray)
xlabel('Axial Stress (kPa)','FontSize',15);
ylabel('Position Along Beam Height (m)','FontSize',15);
title('Axial Stress Distribution due to Bending at the Wall (Tension)')




% Axial Stiffness
S = 0;
ri = 1.495;
for n = 1:8
    E = Ex1x1(n);
    riplus = ri + layerThickness;
    S = S + E*pi*(riplus^2-ri^2);
    ri = ri + layerThickness;
end
S = S + EcAl*pi*(Ra^2-Ri^2);

% Axial Stress Disribution due to Axial Load
r = 1.490;
rArray = [];
sigmaDueToAxialArray = [];

M30 = 20422;

while r <= 1.5
    if r< 1.495
        E = EcAl;
    elseif r < (1.495 + layerThickness)
        E = Ex1x1(1);
    elseif r < (1.495 + 2*layerThickness)
        E = Ex1x1(2);
    elseif r < (1.495 + 3*layerThickness)
        E = Ex1x1(3);
    elseif r < (1.495 + 4*layerThickness)
        E = Ex1x1(3);
    elseif r < (1.495 + 5*layerThickness)
        E = Ex1x1(5);
    elseif r < (1.495 + 6*layerThickness)
        E = Ex1x1(6);
    elseif r < (1.495 + 7*layerThickness)
        E = Ex1x1(7);
    else
        E = Ex1x1(8);
    end

    sigmaDueToAxial = -E*1128/(5.4642*10^3);
    rArray(end+1) = r;
    sigmaDueToAxialArray(end+1) = sigmaDueToAxial;
    r = r + 0.0001;
end
```

```
figure(3)
plot(sigmaDueToAxialArray,rArray)
xlabel('Axial Stress (MPa)','FontSize',15);
ylabel('Height (m)','FontSize',15);
title('Axial Stress Distribution due to Axial Load (Compression)')


% Total Axial Stress Distribution
totStressArray = 10^(-6)*sigma1Array + sigmaDueToAxialArray;
totrArray = rArray;


figure(4)
plot(totStressArray,totrArray)
xlabel('Axial Stress (MPa)','FontSize',15);
ylabel('Height (m)','FontSize',15);
title('Total Axial Stress Distribution at the Wall (Compression)')

% Material strengths (MPa)

X_C_composites = 1268.6358;
X_C_aluminum = 1558.21568;
Y_C_composites = 168.23;
Y_C_aluminum = 199.26;

% Define layer sequence (0 / +45 / -45 / 90 / 90 / -45 / +45 / 0)
layerSequence = [X_C_composites, Y_C_composites, Y_C_composites, Y_C_composites, ...
                 Y_C_composites, Y_C_composites, Y_C_composites, X_C_composites];

% Initialize arrays
materialStrengthArrayTOP = [];
materialStrengthArrayBOTTOM = [];

% Top compartment (Compression: 1.495 to 1.5)
r = 1.490;
materialStrengthTOP = 0;
rArray_a = [];
while r <= 1.5
    if r < 1.495
        materialStrengthTOP = X_C_aluminum;
    elseif r < (1.495 + 1*layerThickness)
        materialStrengthTOP = layerSequence(1);  % 0
    elseif r < (1.495 + 2*layerThickness)
        materialStrengthTOP = layerSequence(2);  % +45
    elseif r < (1.495 + 3*layerThickness)
        materialStrengthTOP = layerSequence(3);  % -45
    elseif r < (1.495 + 4*layerThickness)
        materialStrengthTOP = layerSequence(4);  % 90
    elseif r < (1.495 + 5*layerThickness)
        materialStrengthTOP = layerSequence(5);  % 90
    elseif r < (1.495 + 6*layerThickness)
        materialStrengthTOP = layerSequence(6);  % -45
```

```
    elseif r < (1.495 + 7*layerThickness)
        materialStrengthTOP = layerSequence(7);  % +45
    else
        materialStrengthTOP = layerSequence(8);  % 0
    end

    rArray_a(end+1) = r;
    materialStrengthArrayTOP(end+1) = materialStrengthTOP;
    r = r+0.0001;

end

r_b = -1.490;
materialStrengthBOTTOM = 0;
rArray_b = [];

% Bottom compartment (Tension: -1.495 to -1.5)
while r_b >= -1.5

    if r_b > -1.495
        materialStrengthBOTTOM = X_C_aluminum;  % Aluminum (Outer Layer)
    elseif r_b > (-1.495 - 1*layerThickness)
        materialStrengthBOTTOM = layerSequence(1);  % 0
    elseif r_b > (-1.495 - 2*layerThickness)
        materialStrengthBOTTOM = layerSequence(2);  % +45
    elseif r_b > (-1.495 - 3*layerThickness)
        materialStrengthBOTTOM = layerSequence(3);  % -45
    elseif r_b > (-1.495 - 4*layerThickness)
        materialStrengthBOTTOM = layerSequence(4);  % 90
    elseif r_b > (-1.495 - 5*layerThickness)
        materialStrengthBOTTOM = layerSequence(5);  % 90
    elseif r_b > (-1.495 - 6*layerThickness)
        materialStrengthBOTTOM = layerSequence(6);  % -45
    elseif r_b > (-1.495 - 7*layerThickness)
        materialStrengthBOTTOM = layerSequence(7);  % +45
    else
        materialStrengthBOTTOM = layerSequence(8);  % 0   (Inner)
    end

    rArray_b(end+1) = r_b;
    materialStrengthArrayBOTTOM(end+1) = materialStrengthBOTTOM;
    r_b = r_b-0.0001;
end

figure(5)
totStressArray2 = -10^(-6)*sigma1Array + sigmaDueToAxialArray
plot(totStressArray2,-totrArray)
xlabel('Axial Stress (kPa)','FontSize',15);
ylabel('Height (m)','FontSize',15);
title('Total Axial Stress Distribution at the Wall (Compression)')

% Compute Safety Factor for Tensile and Compressive Regions
safetyFactorArrayTOP = abs(10^2*totStressArray2) ./
   materialStrengthArrayTOP;
```

```
safetyFactorArrayBOTTOM = abs(10^2*totStressArray2) ./ materialStrengthArrayBOTTOM;

% Plot Safety Factor vs Height for Top Compartment
figure(6);
plot(safetyFactorArrayTOP, rArray_a,'r', 'LineWidth', 2);
ylabel('Height (m)', 'FontSize', 15);
xlabel('Safety Factor', 'FontSize', 15);
title('Safety Factor vs Height (Top)', 'FontSize', 16);
grid on;

hold on;

% Add vertical lines at SF = 1.5 and SF = 1.25
xline(1.5, '--k', 'SF = 1.5', 'LabelVerticalAlignment', 'bottom', 'FontSize', 12);
xline(1.25, '--b', 'SF = 1.25', 'LabelVerticalAlignment', 'top', 'FontSize', 12);

hold off;

figure(7);
plot(safetyFactorArrayBOTTOM, rArray_b,'r', 'LineWidth', 2);
ylabel('Height (m)', 'FontSize', 15);
xlabel('Safety Factor', 'FontSize', 15);
title('Safety Factor vs Height (Bottom)', 'FontSize', 16);
grid on;

hold on;

% Add vertical lines at SF = 1.5 and SF = 1.25
xline(1.5, '--k', 'SF = 1.5', 'LabelVerticalAlignment', 'bottom', 'FontSize', 12);
xline(1.25, '--b', 'SF = 1.25', 'LabelVerticalAlignment', 'top', 'FontSize', 12);

hold off;
```

Listing 1: External MATLAB Script

4.2 Property Table

Table– Typical properties of unidirectional composites

based on Mechanics of Fibrous Composites, 1997 by C. T. Herakovich
and Introduction to Design and Analysis with Advanced Composite Materials, 1997, by S. R. Swanson

Material	AS4/epoxy (Vf = 0.62)	T300/epoxy (Vf = 0.62)	Kevlar/epoxy (Vf = 0.55)	S2 glass/epoxy (Vf = 0.60)
Density (lb/in^3)	0.055	0.056	0.05	0.072
Ply thickness (in.)	0.005	0.005	0.005	0.009
E_{11} (Msi)	21.5	19.2	11.0	6.31
E_{22} (Msi)	1.46	1.56	0.8	1.67
v12	0.30	0.24	0.34	0.27
G_{12} (Msi)	0.81	0.82	0.3	0.50
Axial tensile strength (ksi)	310	219	200	250
Transverse tensile strength (ksi)	7.75	6.3	4.0	6.0
Axial compressive strength (ksi)	-184	-226	-34	-120
Transverse compressive strength (ksi)	-24.4	-28.9	-7.70	-17.1
Axial tensile max strain	0.0140	0.0124	0.0185	0.0295
Axial compressive max strain	-0.010	-0.010	-0.0031	-0.0173

Vf = fiber volume ratio (ratio of the volume of the reinforcement with respect to entire volume)

3

Figure 8: Table we used to calculate safety factor