Project

For this project, the objective is to design a composite structure to withstand the load shown on Figure 1.

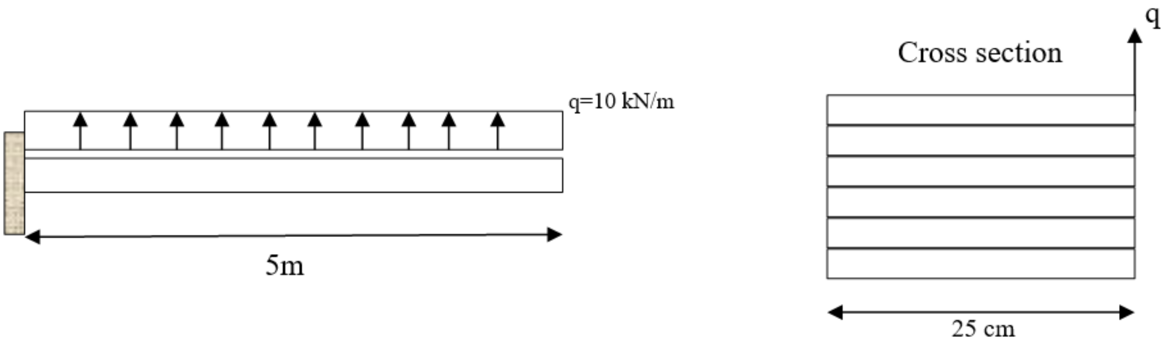

Figure 1: Loading Diagram

The material used will be a graphite/epoxy composite, with the following material properties and strengths:

Table 1: Mechanical Properties

E_1	155 GPa
E_2	12.10 GPa
G_{12}	4.4 GPa
ν_{12}	0.248
X_C	-1250 MPa
X_T	1500 MPa
Y_C	-200 MPa
Y_T	50 MPa
τ	100 MPa

An additional constraint is that the layup will be symmetric, with a total of 6 layers. For this project, we will determine the orientation and thickness for each ply. We will attempt to optimize our design for weight, which is equivalent to volume, and since two of our dimensions are already defined, this is essentially optimizing the thickness.
Our first step will be to turn the load q into a vector that we can use for our analysis, in particular, we will be looking at the moments generated by this load.
We assume that the x-direction is along the length of the beam, with the positive x-direction pointing away from the fixed support. The load q will act in the positive z-direction, and the y-direction points out of the page.
We need to acknowledge that the load q is offset from the center of the beam and that the load is normalized by length. To obtain our moments, we use the equations:

$$M_x = \frac{((q \cdot 5\text{ m}) \cdot .5 \cdot 5\text{ m})}{25\text{ cm}} = 500000\text{ Nm/m} \tag{1}$$

$$M_{xy} = \frac{(-(q \cdot 5\text{ m}) \cdot .5 \cdot 25\text{ cm})}{25\text{ cm}} = -25000\text{ Nm/m} \tag{2}$$

At this point, I began a Monte Carlo Optimization. The inputs that I will randomize will be $\theta_1, \theta_2, \theta_3, t_1, t_2,$ and t_3. Even though the structure will have 6 plies, I only need to vary the parameters of 3 of them due to symmetry.
The angles are bounded by [-90,90] and the thickness bounds were later honed in after multiple iterations. It started off as .01 to 1 mm layer thicknesses, and eventually became 5 to 10 mm layer thicknesses due to too many failures.

My process was to typically have around 100 randomly generated structures for evaluation, which drives refinement in the ranges.
The failure criteria that I am using is the max-stress criteria combined with first-ply-fail theory, meaning that the failure is not progressive, and has a hard zero-tolerance policy. These are the equations for the maximum stress failure criteria:

$$X_T \geq \sigma_1 \geq X_C \tag{3}$$
$$Y_T \geq \sigma_2 \geq Y_C \tag{4}$$
$$|\sigma_{12}| < |\tau| \tag{5}$$

To obtain the σ's necessary to evaluate failure, I started with generating a ABD matrix for reach random design. The equations used to calculate each of the ABD matrices are

$$A_{ij} = \sum_{k=1}^{n} \bar{Q}_{ij}^{k} \left(z_k - z_{k-1}\right) \tag{6}$$
$$B_{ij} = \frac{1}{2} \sum_{k=1}^{n} \bar{Q}_{ij}^{k} \left(z_k^2 - z_{k-1}^2\right) \tag{7}$$
$$D_{ij} = \frac{1}{3} \sum_{k=1}^{n} \bar{Q}_{ij}^{k} \left(z_k^3 - z_{k-1}^3\right) \tag{8}$$

Here, z is the distance away from the center-line of the composite layup, with a positive value pointing downwards. The indices of z begin at the top (negative value) with z_0 and end at z_n at the bottom (positive value) where n is the number of plies (6 total).
The $\bar{Q}$ matrix is equal to

$$\left[\bar{Q}\right] = \left[T\right]^{-1} \left[Q\right] \left[T\right]^{-T} \tag{9}$$

Where $[Q]$ is the reduced stiffness matrix that has the form

$$\begin{bmatrix} \frac{E_1}{1-(\nu_{12}\nu_{21})} & \frac{\nu_{21}E_1}{1-(\nu_{21}\nu_{12})} & 0 \\ \frac{\nu_{12}E_2}{1-(\nu_{12}\nu_{21})} & \frac{E_2}{1-(\nu_{12}\nu_{21})} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \tag{10}$$

This matrix is symmetric because of the relationship

$$\frac{\nu_{12}}{E_1} = \frac{\nu_{21}}{E_2} \tag{11}$$

The $[T]$ matrix is a transformation matrix:

$$\begin{bmatrix} \cos^2\theta & \sin^2\theta & 2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -\sin\theta\cos\theta \\ -\sin\theta\cos\theta & \sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \tag{12}$$

After creating the ABD matrix, I initiated a force/moment vector

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 500000 \\ 0 \\ -25000 \end{bmatrix} \text{Nm/m} \tag{13}$$

I obtain the principal stresses by first obtaining the strains and curvatures using the equation

$$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 500000 \\ 0 \\ -25000 \end{Bmatrix} = \begin{bmatrix} A & B \\ B & D \end{bmatrix} \begin{Bmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{14}$$

Once I have the strains and curvatures, I obtain the principal stresses using these relations:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = [\bar{Q}] \begin{Bmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} + \frac{z_k - z_{k-1}}{2} [\bar{Q}] \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{15}$$

Finally, to get stress values that I could use to critique with the maximum stress criteria, I applied a transformation matrix to bring everything back to the local coordinates. This happened for each ply.
The general equation that I will be using is:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_{12} \end{bmatrix} = [T] \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} \tag{16}$$

At each instance where a ply fails, I mark it as a failure and record the failure type (tension in the X-direction, shear, etc). I also let it keep testing in all of the other directions and plies after a failure is detected to get a better understanding of where the failures occur. I append each failure type and ply number to a string for tracking.
After collecting all the data for a run, I compile everything to a spreadsheet that is time-stamped, which allows me to keep track of how trends change.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	trial	stack_1	stack_2	stack_3	stack_4	stack_5	stack_6	thickness_1	thickness_2	thickness_3	thickness_4	thickness_5	thickness_6	height	failure	failure_mode
2	1	0	15	-30	-30	15	0	0.007	0.009	0.01	0.01	0.009	0.007	0.052	0	
3	2	-90	0	-45	-45	0	-90	0.008	0.01	0.007	0.007	0.01	0.008	0.05	1	YC1 XC2 XT5 YT6
4	3	0	0	-45	-45	0	0	0.008	0.009	0.005	0.005	0.009	0.008	0.044	1	XC1
5	4	60	30	0	0	30	60	0.006	0.005	0.008	0.008	0.005	0.006	0.038	1	YC1 TAU1 XC2 YC2 TAU2 XC3 XT4 XT5 YT5 TAU5 YT6 TAU6
6	5	75	-75	-45	-45	-75	75	0.006	0.005	0.009	0.009	0.005	0.006	0.04	1	YC1 TAU1 YC2 TAU2 XC3 YC3 TAU3 XT4 YT4 TAU4 YT5 TAU5 YT6 TAU6
7	6	90	60	0	0	60	90	0.006	0.01	0.006	0.006	0.01	0.006	0.044	1	YC1 TAU1 YC2 TAU2 XC3 XT4 YT5 TAU5 YT6 TAU6
8	7	-30	-15	15	15	-15	-30	0.009	0.01	0.008	0.008	0.01	0.009	0.054	1	TAU1 YT6 TAU6
9	8	75	0	0	0	0	75	0.006	0.006	0.006	0.006	0.006	0.006	0.036	1	YC1 XC2 XT5 YT5 YT6
10	9	-90	0	0	0	0	-90	0.007	0.005	0.008	0.008	0.005	0.007	0.04	1	YC1 XC2 XT5 YT5 YT6
11	10	-15	0	-30	-30	0	-15	0.01	0.006	0.008	0.008	0.006	0.01	0.048	1	TAU1 TAU6
12	11	-45	90	30	30	90	-45	0.008	0.005	0.007	0.007	0.005	0.008	0.04	1	XC1 YC1 TAU1 XT2 YC2 XC3 XT4 XC5 YT5 YT6 TAU6
13	12	75	-45	-60	-60	-45	75	0.008	0.009	0.009	0.009	0.009	0.008	0.052	1	YC1 TAU1 YC2 TAU2 YT4 YT5 TAU5 YT6 TAU6
14	13	0	-45	0	0	-45	0	0.008	0.008	0.005	0.005	0.008	0.008	0.042	1	XC1 YT1 XT6
15	14	-90	0	0	0	0	-90	0.006	0.007	0.01	0.01	0.007	0.006	0.046	1	YC1 XC2 XT5 YT6
16	15	-90	-45	60	60	-45	-90	0.006	0.007	0.01	0.01	0.007	0.006	0.046	1	YC1 TAU1 XC2 YC2 TAU2 YT4 XT5 YT5 TAU5 YT6 TAU6
17	16	-60	0	75	75	0	-60	0.008	0.005	0.005	0.005	0.005	0.008	0.036	1	YC1 TAU1 XC2 YT4 XT5 YT6 TAU6
18	17	-75	-30	-30	-30	-30	-75	0.006	0.01	0.01	0.01	0.01	0.006	0.052	1	YC1 YC2 TAU2 YT4 YT5 TAU5 YT6
19	18	-45	45	-90	-90	45	-45	0.008	0.007	0.008	0.008	0.007	0.008	0.046	1	TAU1 XC2 TAU2 XC4 YT4 YT5 TAU5 YT6 TAU6
20	19	-45	0	0	0	0	-45	0.009	0.007	0.008	0.008	0.007	0.009	0.048	1	TAU1 XC2 YT2 XT5 YT6 TAU6
21	20	90	-30	60	60	-30	90	0.01	0.008	0.008	0.008	0.008	0.01	0.052	1	YC1 TAU1 XC2 TAU2 YT4 XT5 YT5 TAU5 YT6 TAU6
22	21	-90	0	0	0	0	-90	0.005	0.01	0.006	0.006	0.01	0.005	0.042	1	YC1 XC2 XT5 YT6
23	22	-45	-60	0	0	-60	-45	0.005	0.006	0.008	0.008	0.006	0.005	0.038	1	XC1 YC1 TAU1 YC2 TAU2 XC3 XT4 YT5 TAU5 XT6 YT6 TAU6
24	23	0	75	-90	-90	75	0	0.007	0.006	0.008	0.008	0.006	0.007	0.042	1	XC1 YT5 XT6
25	24	0	0	0	0	0	0	0.005	0.009	0.008	0.008	0.009	0.005	0.044	1	XC1
26	25	0	-15	0	0	-15	0	0.008	0.005	0.01	0.01	0.005	0.008	0.046	0	
27	26	-30	0	75	75	0	-30	0.005	0.006	0.007	0.007	0.006	0.005	0.036	1	TAU1 XC2 YT2 XT5 YT6 TAU6
28	27	0	0	-45	-45	0	0	0.005	0.006	0.008	0.008	0.006	0.005	0.038	1	XC1 XC2 XT6
29	28	-45	90	30	30	90	-45	0.008	0.006	0.009	0.009	0.006	0.008	0.046	1	YC1 TAU1 YC2 XC3 XT4 YT5 YT6 TAU6

... Test | 03-08-01-27-44 | 03-08-01-33-00 | 03-08-01-40-22 | 03-08-01-54-18 | 03-08-02-00-57 | 03-08-02-11-47 | ...

Figure 2: Screenshot of spreadsheet

Now here is a table that highlights the results of my optimization

Table 2: Optimization Highlights

Trial	Best Result Orientation	Best Result Height (m)
1	-	-
2	-	-
3	$[-7^\circ/77^\circ/1^\circ/]_s$	0.056
4	$[-8^\circ/41^\circ/64^\circ/]_s$	0.05
5	$[-15^\circ/0^\circ/0^\circ/]_s$	0.048
6	$[0^\circ/-15^\circ/0^\circ/]_s$	0.046
7	$[10^\circ/-14^\circ/48^\circ/]_s$	0.044
8	$[0^\circ/0^\circ/-60^\circ/]_s$	0.0442
9	$[0^\circ/0^\circ/0^\circ/]_s$	0.0436
10	$[-15^\circ/0^\circ/15^\circ/]_s$	0.0436

Notes/Changelog:

Trial 1: All results had failures in 5 or 6 plies

Trial 2: Changes thickness bounds to .1 to 10 mm, best result failed in only one ply

Trial 3: Thickness bounds changed to 1 to 10 mm. Only one result did not fail.

Trial 4: Thickness bounds changed to 5 to 10 mm. Again, only one result did not fail, but had a handful of cases that almost passed (only failed in one direction and in one ply).

Trial 5: Thickness bounds were left unchanged, but I discretize the available angles to more 'standard' angles, namely ±[0,15,30,45,60,75,90]. I noticed that cases with a lot of 0° plies seemed to be less failure-prone

Trial 6: After noticing that 0° plies seemed to be more successful, I increased the frequencies of 0s by changing the available angles to be ±[0,0,0,15,30,45,60,75,90].

Trial 7: I switched back to completely random angles varying from −90° to 90°. Instead of randomly generating 100 combinations, I generated 1000.

Trial 8: I switched back to the 0-biased allowable angles from Trial 6, but this time, I allowed the random thickness to generate a number by the .1 mm. Again, I had 1000 cases, which took 26 minutes to generate and analyze. For the first time, the best result's height did not decrease.

Trial 9: Out of curiosity, I decided to see what would happen if I forced all the angles to be equal to 0. I only ran 100 cases and it took just two minutes.

Trial 10: Finally, I decided to call it quits. The final trial had the thickness bound from 5 to 10 mm with randomization down to the .1 mm. The angles were randomly selected from ±[0,0,0,15,30,45,60,75,90].

To conclude, I say that my final optimized beam has these properties:

Table 3: Optimized Beam Properties

Parameter	Value
θ_1	-15°
θ_2	0°
θ_3	15°
t_1	5.7 mm
t_2	7.5 mm
t_3	8.6 mm
Total Thickness	43.6 mm

Code

```
%% MAE 166C Project; Willy Teav 104917860

clearvars; close all ;clc

E1 = 155e9;
E2 = 12.1e9;
G12 = 4.4e9;
nu12 = .248;
XT = 1500e6;
XC = -1250e6;
YT = 50e6;
YC = -200e6;
TAU = 100e6;

q = 10e3; %N/m
L = 5; %m
w = 25e-2; %m

N_x = 0; N_y = 0; N_xy = 0; M_y = 0;
M_x = (q*L^2)/(2*w);
M_xy = -(q*L)/2;

NM_vec = [N_x; N_y; N_xy; M_x; M_y; M_xy];

% pseudo code
% 3 angles and 3 thicknesses
% randomly pick angles and thicknesses
% calculate volume and whether or not it passes
% log all in excel sheet
% try again with new values

% initializations
angle1 = 45; angle2 = 45; angle3 = 45;
t1 = .1e-3; t2 = .1e-3; t3 = .1e-3;

sign = [-1,1];

sheetname = datestr(now,'mm-dd-HH-MM-SS');
num_trials = 100;
wb = waitbar(0,'progress');
angles = [0,0,0,15,30,45,60,75,90];
%angles = [0,0];
tic
for trial = 1:num_trials
    failure = 0;
    failure_layer = 0;
    failure_mode = "";
    angle1 = angles(randi(length(angles)))*sign(randi(2));
    angle2 = angles(randi(length(angles)))*sign(randi(2));
    angle3 = angles(randi(length(angles)))*sign(randi(2));
%     angle1 = randi([0,90])*sign(randi(2));
%     angle2 = randi([0,90])*sign(randi(2));
%     angle3 = randi([0,90])*sign(randi(2));
    t1 = randi([50,100])/1e4; t2 = randi([50,100])/1e4; t3 = randi([50,100])/1e4;
    %5 to 10 mm layer thickness
    stack = [angle1, angle2, angle3, angle3, angle2, angle1];
```

```
 thickness = [t1, t2, t3, t3, t2, t1];
 z = make_z(stack,thickness);

 A_mtx = A(stack,z,E1,E2,G12,nu12);
 B_mtx = B(stack,z,E1,E2,G12,nu12);
 D_mtx = D(stack,z,E1,E2,G12,nu12);

 ABD_mtx = [A_mtx,B_mtx;B_mtx,D_mtx];

 syms epsx epsy gammaxy kx ky kxy

 eq = NM_vec == ABD_mtx*[epsx; epsy; gammaxy; kx; ky; kxy];
 vals = (vpasolve(eq));
 epsx = double(vals.epsx);
 epsy = double(vals.epsy);
 gammaxy = double(vals.gammaxy);
 kx = double(vals.kx);
 ky = double(vals.ky);
 kxy = double(vals.kxy);

for i = 1:length(stack)
    sigma_middle = Qbar(stack(i),E1,E2,G12,nu12)*[epsx;epsy;gammaxy] + ...
    .5*(z(i)+z(i+1))*Qbar(stack(i),E1,E2,G12,nu12)*[kx;ky;kxy];
    sigmax(i) = sigma_middle(1);
    sigmay(i) = sigma_middle(2);
    sigmaxy(i) = sigma_middle(3);
    strainx(i) = epsx + .5*(z(i)+z(i+1))*kx;
    strainy(i) = epsy + .5*(z(i)+z(i+1))*ky;
    strainxy(i) = gammaxy + .5*(z(i)+z(i+1))*kxy;
end

for layer = 1:length(stack)
 sigma_vec = T(stack(layer))*[sigmax(layer);sigmay(layer);sigmaxy(layer)];
 sigma_1 = sigma_vec(1);
 sigma_2 = sigma_vec(2);
 sigma_12 = sigma_vec(3);

 if sigma_1 > XT || sigma_1 < XC
     %sigma_1
     if sigma_1 > 0
         failure_mode = append(failure_mode," XT",num2str(layer));
     else
         failure_mode = append(failure_mode," XC",num2str(layer));
     end
     %failure_layer = layer;
     failure = 1;
 end

 if sigma_2 > YT || sigma_2 <YC
     %sigma_2
     if sigma_2 > 0
         failure_mode = append(failure_mode," YT",num2str(layer));
     else
         failure_mode = append(failure_mode," YC",num2str(layer));
     end
     %failure_layer = layer;
     failure = 1;
 end
```

```
    if abs(sigma_12) > abs(TAU)
        %sigma_12
        failure_mode = append(failure_mode," TAU",num2str(layer));
        %failure_layer = layer;
        failure = 1;
    end
   end

%outputs
   trial;
   stack;
   thickness;
   height = sum(thickness);
   failure;
   failure_mode;
   %failure_layer

   tab = table(trial,stack,thickness,height,failure,failure_mode);
   %sheetname = datestr(now,'mm-dd-HH-MM-SS');

   if trial == 1
   %xlswrite('MAE_166C_Optimization.xlsx',["Trial","stack","thickness"...
   %    ,"height","failure","failure_mode"],'Test','A1')

   %xlswrite('MAE_166C_Optimization.xlsx',["Silly Goose"],'Test','P1:S1')

%    xlswrite('MAE_166C_Optimization.xlsx',tab.Properties.VariableNames,...
%        'Test','A1')

    writetable(tab,'MAE_166C_Optimization.xlsx','Sheet',sheetname,'Range','A1')
   end

   excel_range = append('A',num2str(trial+1));

%    xlswrite('MAE_166C_Optimization.xlsx',[num2cell(trial),num2cell(stack),...
%        num2cell(thickness),num2cell(height),num2cell(failure),...
%        failure_mode],'Test',excel_range)

   xlswrite('MAE_166C_Optimization.xlsx',tab.Variables,sheetname,excel_range)

%wb = waitbar(trial/num_trials,'progress');
waitbar(trial/num_trials,wb,'progress')
end

winopen('MAE_166C_Optimization.xlsx')
toc
%% debug
% thickness = [1, 1, 1, 2, 2, 2];
% stack = [45,45,90, 90, 45, 45];
% z = make_z(stack,thickness)


%% Functions
function T = T(theta)
T = [cosd(theta)^2, sind(theta)^2, 2*sind(theta)*cosd(theta);
              sind(theta)^2, cosd(theta)^2, -2*sind(theta)*cosd(theta);
              -sind(theta)*cosd(theta), sind(theta)*cosd(theta), ...
```

```
                cosd(theta)^2 - sind(theta)^2];
end

function Q = Q(E1,E2,G12,nu12)
nu21 = (E2/E1)*nu12;
Q = [E1/(1-(nu12*nu21)) , (nu21*E1)/(1 - (nu21*nu12)), 0;
    (nu12*E2)/(1 - (nu12*nu21)), E2/(1-(nu12*nu21)), 0;
    0, 0, G12];
end

function Qbar = Qbar(theta,E1,E2,G12,nu12)
Qbar = T(theta)^-1 * Q(E1,E2,G12,nu12) * T(theta)'^-1;
end

function alphabar = alphabar(theta,alpha1,alpha2)
alphabar = [alpha1*cosd(theta)^2+alpha2*sind(theta)^2;...
    alpha1*sind(theta)^2+alpha2*cosd(theta)^2;...
    2*(alpha1-alpha2)*cosd(theta)*sind(theta)];
end

function z = make_z(array,thickness)
N = length(array);
%height = thickness*N;
%z = linspace(-height/2,height/2,N+1);
height = sum(thickness);
z = zeros(1,N);
z(1) = -height/2;
for i = 2:N+1
    z(i) = z(i-1) + thickness(i-1);
end
end

function A = A(array,z_array,E1,E2,G12,nu12)
%let array be an array of angles
N = length(array);
A = zeros(3);
for i = 2:N+1
    A = A + Qbar(array(i-1),E1,E2,G12,nu12)*(z_array(i)-z_array(i-1));
end
end

function B = B(array,z_array,E1,E2,G12,nu12)
%let array be an array of angles
N = length(array);
B = zeros(3);
if ~isequal(array(:),flip(array(:)))
    %fixed a rounding error that causes symmetric arrays to not turn B = 0
for i = 2:N+1
    B = B + Qbar(array(i-1),E1,E2,G12,nu12)*(z_array(i)^2-z_array(i-1)^2);
end
B = B/2;
end
end

function D = D(array,z_array,E1,E2,G12,nu12)
%let array be an array of angles
N = length(array);
D = zeros(3);
```

```
for i = 2:N+1
    D = D + Qbar(array(i-1),E1,E2,G12,nu12)*(z_array(i)^3-z_array(i-1)^3);
end
D = D/3;
end
```