Composite Materials and Design

Assignment Project Report by **Kartik Krishna (2014A4TS246U)**

1. Objectives:

To use MATLAB code to determine the optimum thickness (weight) required to ensure no failure at all points within each lamina using Tsai-Wu failure criterion for following laminate configurations and loading.

Case 1: [0/45/-45/90]s
Case 2: [45/0/-45/90]s
Case 3: [45/-45/0/90]s
Case 4: [45/-45/90/0]s

Loading Conditions:
Nx = Ny = 5000;
Nxy = Mxy = My = 0;
ΔT (°C) = -200;
ΔC = 0;

2. Functions and scripts used in Code (and what they do):

a) **epsilon_kappa()** - Function which takes material parameters and computes the ABD matrix and **returns the epsilon0 and kappa** values.
b) **QSalpha()** - Function which takes material parameters and r**eturns a Qbar matrix** (for given parameter theta), **Sbar matrix** and **alphak** (which is the thermal properties vector transformed to a global coordinate system)
c) **Strength_Ratio()** - Function which takes all the problem parameters and returns the **TsaiWu numbers** for all the layers. The layers are arranged as **[<Tsai Wu numbers for all top surfaces>, <Tsai Wu numbers for all bottom surfaces>]** which for our problem is an **8x2 array.**
d) **transforms()** - a Function which takes an angle and **computes both T1 and T2 and returns them.**

3. How the Code works:

The main code is called from **`Project_Main.m`**

a) First, all the F-values (F1, F22, etc.) are calculated for the Tsai-Wu failure criteria using the given values for compressive and tensile strengths.
b) We also start with an initial guess thickness
c) The code then enters a loop for a set of iterations
d) In the loop, the Tsai-Wu numbers are calculated using the **`Strength_Ratio()`** function for all the layers and arranged in an 8x2 array
e) The **largest value** of the Tsai-Wu array is chosen because this corresponds to the **weakest** point in the material
f) The idea is to use a small initial thickness and to progressively increase the thickness until **the weakest layer is strong enough to handle the loading conditions.**
g) If the Tsai-Wu number is greater than 1.01, we increase the thickness, if the Tsai-Wu is less than .99, we decrease the thickness by 10% of itself.
h) If Tsai-Wu is between .99 and 1.01 or in other words almost 1.00, we have found our optimum thickness.

4. Constants Used:

2014A4TS**246**Umod3 = 0
Therefore, values corresponding to 0 were used

```
E1 = 38.6 * 10^9; %Pa (Axial Modulus)
E2 = 8.27 * 10^9 ; %Pa (Transverse Modulus)
v12 = .26; %Dimensionless (Poisson's Ratio)
G12 = 4.14 * 10^9; %Pa (Shear Modulus)
a = 8.6 * 10^-6; %m/C (Axial CTE)
t = 22.1 * 10^-6; %m/C (Transverse CTE)

thetas = [0,45,-45,90,90,-45,45,0] %Deg (Angles of stacking)
N = length(thetas); %no. of layers
lt = 1e-3; %m (Layer Thickness) INITIAL GUESS

H = lt * N/2;
deltaT = -200;
Nx = 5000; Ny = 5000; %Nm^-1
Mx = -3000; %N
My = 0;

xt = 1062e6;
xc = -610e6;
```

```
yt = 31e6;
yc = -118e6;
s = 72e6;
```

5. Results:

***NOTE: THE THICKNESS BEING FOUND HERE IS THE LAYER THICKNESS SO THE TOTAL IS (LAYER THICKNESS x 8)**

```
>> Case I
thetas =

    0   45  -45   90   90  -45   45    0

Iteration 1, Thickness 1.0000 mm, Max Tsai Wu 1.961
Iteration 2, Thickness 1.1000 mm, Max Tsai Wu 1.622
Iteration 3, Thickness 1.2100 mm, Max Tsai Wu 1.368
Iteration 4, Thickness 1.3310 mm, Max Tsai Wu 1.176
Iteration 5, Thickness 1.4641 mm, Max Tsai Wu 1.031
Iteration 6, Thickness 1.6105 mm, Max Tsai Wu 0.919
Iteration 7, Thickness 1.4495 mm, Max Tsai Wu 1.044
Iteration 8, Thickness 1.5944 mm, Max Tsai Wu 0.929
Iteration 9, Thickness 1.4350 mm, Max Tsai Wu 1.058
Iteration 10, Thickness 1.5785 mm, Max Tsai Wu 0.940
Iteration 11, Thickness 1.4206 mm, Max Tsai Wu 1.073
Iteration 12, Thickness 1.5627 mm, Max Tsai Wu 0.951
Iteration 13, Thickness 1.4064 mm, Max Tsai Wu 1.087
Iteration 14, Thickness 1.5470 mm, Max Tsai Wu 0.962
Iteration 15, Thickness 1.3923 mm, Max Tsai Wu 1.103
Iteration 16, Thickness 1.5316 mm, Max Tsai Wu 0.974
Iteration 17, Thickness 1.3784 mm, Max Tsai Wu 1.118
Iteration 18, Thickness 1.5163 mm, Max Tsai Wu 0.986
Iteration 19, Thickness 1.3646 mm, Max Tsai Wu 1.134
Iteration 20, Thickness 1.5011 mm, Max Tsai Wu 0.999

Optimized Thickness is 1.501 mm

>> Case II
thetas =

   45    0  -45   90   90  -45    0   45

Iteration 1, Thickness 1.0000 mm, Max Tsai Wu 4.430
```

```
Iteration 2, Thickness 1.1000 mm, Max Tsai Wu 3.437
Iteration 3, Thickness 1.2100 mm, Max Tsai Wu 2.715
Iteration 4, Thickness 1.3310 mm, Max Tsai Wu 2.185
Iteration 5, Thickness 1.4641 mm, Max Tsai Wu 1.792
Iteration 6, Thickness 1.6105 mm, Max Tsai Wu 1.499
Iteration 7, Thickness 1.7716 mm, Max Tsai Wu 1.278
Iteration 8, Thickness 1.9487 mm, Max Tsai Wu 1.111
Iteration 9, Thickness 2.1436 mm, Max Tsai Wu 0.982
Iteration 10, Thickness 1.9292 mm, Max Tsai Wu 1.126
Iteration 11, Thickness 2.1222 mm, Max Tsai Wu 0.994

Optimized Thickness is 2.122 mm
```

>> Case III
thetas =

45 -45   0   90   90   0 -45   45

```
Iteration 1, Thickness 1.0000 mm, Max Tsai Wu 5.632
Iteration 2, Thickness 1.1000 mm, Max Tsai Wu 4.260
Iteration 3, Thickness 1.2100 mm, Max Tsai Wu 3.279
Iteration 4, Thickness 1.3310 mm, Max Tsai Wu 2.572
Iteration 5, Thickness 1.4641 mm, Max Tsai Wu 2.058
Iteration 6, Thickness 1.6105 mm, Max Tsai Wu 1.682
Iteration 7, Thickness 1.7716 mm, Max Tsai Wu 1.404
Iteration 8, Thickness 1.9487 mm, Max Tsai Wu 1.197
Iteration 9, Thickness 2.1436 mm, Max Tsai Wu 1.041
Iteration 10, Thickness 2.3579 mm, Max Tsai Wu 0.923
Iteration 11, Thickness 2.1222 mm, Max Tsai Wu 1.056
Iteration 12, Thickness 2.3344 mm, Max Tsai Wu 0.934
Iteration 13, Thickness 2.1009 mm, Max Tsai Wu 1.071
Iteration 14, Thickness 2.3110 mm, Max Tsai Wu 0.946
Iteration 15, Thickness 2.0799 mm, Max Tsai Wu 1.086
Iteration 16, Thickness 2.2879 mm, Max Tsai Wu 0.957
Iteration 17, Thickness 2.0591 mm, Max Tsai Wu 1.102
Iteration 18, Thickness 2.2650 mm, Max Tsai Wu 0.969
Iteration 19, Thickness 2.0385 mm, Max Tsai Wu 1.118
Iteration 20, Thickness 2.2424 mm, Max Tsai Wu 0.982
Iteration 21, Thickness 2.0181 mm, Max Tsai Wu 1.135
Iteration 22, Thickness 2.2200 mm, Max Tsai Wu 0.994

Optimized Thickness is 2.220 mm
```

```
>> Case IV
thetas =

   45  -45   90    0    0   90  -45   45

Iteration 1, Thickness 1.0000 mm, Max Tsai Wu 7.617
Iteration 2, Thickness 1.1000 mm, Max Tsai Wu 5.703
Iteration 3, Thickness 1.2100 mm, Max Tsai Wu 4.336
Iteration 4, Thickness 1.3310 mm, Max Tsai Wu 3.353
Iteration 5, Thickness 1.4641 mm, Max Tsai Wu 2.641
Iteration 6, Thickness 1.6105 mm, Max Tsai Wu 2.121
Iteration 7, Thickness 1.7716 mm, Max Tsai Wu 1.737
Iteration 8, Thickness 1.9487 mm, Max Tsai Wu 1.452
Iteration 9, Thickness 2.1436 mm, Max Tsai Wu 1.239
Iteration 10, Thickness 2.3579 mm, Max Tsai Wu 1.077
Iteration 11, Thickness 2.5937 mm, Max Tsai Wu 0.954
Iteration 12, Thickness 2.3344 mm, Max Tsai Wu 1.092
Iteration 13, Thickness 2.5678 mm, Max Tsai Wu 0.965
Iteration 14, Thickness 2.3110 mm, Max Tsai Wu 1.108
Iteration 15, Thickness 2.5421 mm, Max Tsai Wu 0.977
Iteration 16, Thickness 2.2879 mm, Max Tsai Wu 1.123
Iteration 17, Thickness 2.5167 mm, Max Tsai Wu 0.989
Iteration 18, Thickness 2.2650 mm, Max Tsai Wu 1.140
Iteration 19, Thickness 2.4915 mm, Max Tsai Wu 1.002

Optimized Thickness is 2.492 mm
```

6. Appendix A (MATLAB Codes):

____
_

1. Project_Main.m

```
clear all; clc;
%% Constants
E1 = 38.6 * 10^9; %Pa (Axial Modulus)
E2 = 8.27 * 10^9 ; %Pa (Transverse Modulus)
v12 = .26; %Dimensionless (Poisson's Ratio)
G12 = 4.14 * 10^9; %Pa (Shear Modulus)
```

```
a = 8.6 * 10^-6; %m/C (Axial CTE)
t = 22.1 * 10^-6; %m/C (Transverse CTE)

thetas = [45, -45, 90, 0, 0, 90, -45, 45] %Deg (Angles of stacking)
N = length(thetas); %no. of layers
lt = 1e-3; %m (Layer Thickness) INITIAL GUESS

H = lt * N/2;
deltaT = -200;
Nx = 5000; Ny = 5000; %Nm^-1
Mx = -3000; %N
My = 0;

xt = 1062e6;
xc = -610e6;
yt = 31e6;
yc = -118e6;
s = 72e6;

F1 = (1/xt + 1/xc);
F2 = (1/yt + 1/yc);
F11 = -1/(xt*xc);
F22 = -1/(yt*yc);
F66 = 1/s^2;

for i = 1:100
    [tsai_top, tsai_bot] = Strength_Ratio(E1, E2, v12, G12, a, t,
thetas, lt, deltaT, Nx, Ny, Mx, My, F1, F2, F11, F22, F66);

    tsai = max(max([tsai_top, tsai_bot]));

    if tsai > 1.01
        fprintf('Iteration %d, Thickness %.4f mm, Max Tsai Wu
%.3f\n',i, lt*1000, tsai)
        lt= lt+lt/10; %Increse thickness if TsaiWu is large
    elseif tsai < .99
        fprintf('Iteration %d, Thickness %.4f mm, Max Tsai Wu
%.3f\n',i, lt*1000, tsai)
        lt= lt-lt/10; %Reduce thickness if TsaiWu is small
    else
        fprintf('Iteration %d, Thickness %.4f mm, Max Tsai Wu
%.3f\n',i, lt*1000, tsai)
        break;
```

```
    end
end

if (.99<tsai && tsai<1.01)
    fprintf('\nOptimized Thickness is %.3f mm \n',lt*1000)
else
    disp('more iterations needed')
end
```

____
_

____
_

2. **Strength_Ratio.m**

```
function [sr_top, sr_bot] = Strength_Ratio(E1, E2, v12, G12, a, t,
thetas, lt, deltaT, Nx, Ny, Mx, My, F1, F2, F11, F22, F66)
%Function which calculates TSAI WU values for Project_Main
%NOTE: IN THIS FUNCTION ALL SR STANDS FOR TSAI WU

N = length(thetas);
H = lt * N/2;
sr_top = zeros(N,1);
sr_bot = zeros(N,1);

[e0, kappa] = epsilon_kappa(E1, E2, v12, G12, a, t, thetas, lt,
deltaT, Nx, Ny, Mx, My);

for k = 1:N

    thetak = thetas(k);
    [Qk, ~, alphak] = QSalpha(E1, E2, v12, G12, thetak, a, t);

    %Getting transfrom ready
    [T1, ~] = transforms(thetak);
    %In an earlier iteration of the code, I had mixed up zk and zk1
for top and bottom
    %check at top
    zk1 =  -H + (k-1)*lt; %z_k-1
    eps_top = e0 + zk1*kappa;%top layer epsilon
    top_stress_g = Qk * (eps_top-alphak*deltaT); %top stress global
```

```
    top_stress_l = T1 * top_stress_g; %top stress local

    % check at bottom
    zk = -H + (k)*lt; %z_k
    eps_bot = e0 + zk*kappa;
    bottom_stress_g = Qk * (eps_bot-alphak*deltaT);% Sigma x, y, z
    bottom_stress_l = T1 * bottom_stress_g; %Sigma 1, 2, 3

    s1t = top_stress_l(1);
    s2t = top_stress_l(2);
    s12t = top_stress_l(3);
    s1b = bottom_stress_l(1);
    s2b = bottom_stress_l(2);
    s12b = bottom_stress_l(3);

    Eqn1 = F1*s1t + F2*s2t + F11*(s1t)^2 + F22*(s2t)^2 +
F66*(s12t)^2;
    Eqn2 = F1*s1b + F2*s2b + F11*(s1b)^2 + F22*(s2b)^2 +
F66*(s12b)^2;

    sr_top(k) = Eqn1; %THESE ARE TSAI WU VALUES BEING STACKED IN
ARRAYS
    sr_bot(k) = Eqn2;

end
______________________________________________________________________
_

______________________________________________________________________
_
```

3. **epsilon_kappa.m**

```
function [e0, kappa] = epsilon_kappa(E1, E2, v12, G12, a, t, thetas,
lt, deltaT, Nx, Ny, Mx, My)
%% MAIN

A = zeros(3);
B = zeros(3);
D = zeros(3);
Nt = 0; Mt = 0;
N = length(thetas);
```

```
H = lt * N/2;

for k = 1:N
    thetak = thetas(k);
    [Qk, Sb, alphak] = QSalpha(E1, E2, v12, G12, thetak, a, t);
    zk = -H + (k)*lt; %z_k
    zk1 = -H + (k-1)*lt; %z_k-1

    Ak = Qk*(zk-zk1);
    A = A + Ak;

    Ntk = deltaT*(Qk * alphak)*lt; %Calculate Nthermal and Mthermal
    Mtk = deltaT*((Qk*alphak)*(zk^2-zk1^2));

    Nt = Nt + Ntk;
    Mt = Mt + Mtk;

    Bk = Qk*((zk^2)-(zk1^2))/2;
    B = B + Bk;

    Dk = Qk*((zk^3)-(zk1^3))/3;
    D = D + Dk;
end

ABD = [A, B; B, D];

final = [Nt(1) + Nx; Nt(2) + Ny; Nt(3); Mt(1) + Mx; Mt(2)+ My;
Mt(3)];
ek = ABD\final;
e0 = ek(1:3);
kappa = ek(4:6);
```

____
_

____
_

4. **QSalpha.m**

____
_

```
function [Qb, Sb, alphak] = QSalpha(E1, E2, v12, G12, theta, aCTE,
tCTE)

%% MAIN

%Creating a Reduced Compliance Matrix

S(1,1) = 1/E1;
S(2,2) = 1/E2;
S(3,3) = 1/G12;
S(1,2) = -v12/E1;
S(2,1) = -v12/E1;

%Creating the Reduced Stiffness Matrix

Q = S^-1;

m = cosd(theta);
n = sind(theta);

%Transformation Matrices

T1 = [m^2, n^2, 2*m*n; n^2, m^2, -2*m*n; -m*n, m*n, (m^2-n^2)];
T2 = [m^2, n^2, m*n; n^2, m^2, -m*n; -2*m*n, 2*m*n, (m^2-n^2)];

%Creating Global Q and S Matrices

Qb = T1\Q*T2;
Sb = Qb^-1;

%Creating the Alpha Vector in xy system

alphak = T2\[aCTE; tCTE; 0];
end
______________________________________________________________________
_

______________________________________________________________________
_
```

5. **transforms.m**

```
function [T1, T2] = transforms(theta)

%function which gives a T1 and T2 given a theta

m = cosd(theta);
n = sind(theta);

%Transformation Matrices

T1 = [m^2, n^2, 2*m*n; n^2, m^2, -2*m*n; -m*n, m*n, (m^2-n^2)];
T2 = [m^2, n^2, m*n; n^2, m^2, -m*n; -2*m*n, 2*m*n, (m^2-n^2)];
```