```
clear all;
%% Background Information
stu_no = 1008247531;
           %TUVWXYZ

TU = 82;
V = 4;
WX = 75;
YZ = 31;

AF = 250 - (2*TU);
BF = 10 + (8*V);
CF = WX;
DF = -20 - YZ;

t = 0.125;

% Initially tried using For loops but not enough computational power, so
% went on with manual trials after 5. Looked at the local stresses for
each plies and tried to optimize them.

%theta = [5 0 45 45 0 5]; % In Degrees - For six it doesn't converge for
any solution - breaks off in all arrangements

% after several tries on optiming the stress vectors, Finally recieved
the combination written down here for theta

theta = [5 0 50 2 50 0 5]; % Works for Both Tsai-hill and Tsai-wu with
equally distributed loads around the center -any further distribution
around this works nicely
theta = theta.*(pi/180); % Conversion to Radians - I my system Default is
radians

E1 = 125;
E2 = 9.8;
G12 = 5.5;
nu_12 = 0.24;

sl_up = 900;
sl_down = 800;
st_up = 55;
st_down = 170;
tau = 90;

S = [1/E1 -nu_12/E1 0; -nu_12/E1 1/E2 0; 0 0 1/G12];

Q_local = S^-1;

T = @(x) [(cos(x))^2 (sin(x))^2 2*cos(x)*sin(x); (sin(x))^2 (cos(x))^2 -
2*cos(x)*sin(x); -cos(x)*sin(x) cos(x)*sin(x) (cos(x))^2 - (sin(x))^2];

%% 1.) Calculating ABD Matrix
A = zeros(3);
B = zeros(3);
```

```
D = zeros(3);

for k = 1:4
   z_k = (k-2)*0.125;
   z_k1 = z_k - 0.125;
   Q_global =
((inv(T(theta(k))))*(Q_local)*(inv(transpose(T(theta(k))))));
   A = A + 0.125*Q_global;
   B = B + (0.125*(z_k + z_k1))*Q_global;
   D = D + (0.125*(z_k^2 + z_k1^2 + z_k*z_k1))*Q_global;
end

abd = [A B;B D];

fm = [AF;BF;CF;DF;0;0];
e_mat = abd\fm;

%% 2.) Using epsilon matrix for calculations of Stresses

tsai_hill = zeros(length(theta),1);
tsai_wu = zeros(length(theta),1);
sigma_all = zeros(length(theta),3);
strain_global_all = zeros(length(theta),3);
sigma_global_all = zeros(length(theta),3);

for p = 1:length(theta)

curr_stack = p;
center = t*(length(theta)/2); % Calculating center from bottom
dist = t*((2*curr_stack-1)/2); % Calculating Dist from bottom
epsilon = e_mat(1:3) + (abs(center-dist))*e_mat(4:6);
qbar = ((inv(T(theta(p))))*(Q_local)*(inv(transpose(T(theta(p))))));
sigma_global = qbar*epsilon;

sigma_global_all(p,:) = sigma_global;
strain_global_all(p,:) = epsilon; %capturing sigma global for each ply

F11 = abs(1/(sl_up*sl_down));
F22 = abs(1/(st_down*st_up));
F66 = abs(1/(tau^2));
F1 = (1/sl_up - abs(1/sl_down));
F2 = (1/st_up - abs(1/st_down));
F12 = 0.5*sqrt(F11*F22);

    sigma = T(theta(curr_stack))*sigma_global;
    sigma_all(p,:) = sigma;

    if sigma(2) >= 0
        st = 55;
    else
        st = 170;
    end

    if sigma(1) >= 0
```

```
        sl = 900;
    else
        sl = 800;
    end


    tsai_hill(p) = (sigma(1)^2)/(sl^2) - (sigma(1)*sigma(2))/(sl^2) +
(sigma(2)^2)/(st^2) + (sigma(3)^2)/(tau^2);

    tsai_wu(p) = F11*sigma(1)^2 + F22*sigma(2)^2 + F66*sigma(3)^2 +
F1*sigma(1) + F2*sigma(2) + 2*F12*sigma(1)*sigma(2);

end

transpose(fm)
transpose(strain_global_all)
transpose(sigma_global_all)
transpose(tsai_hill)
transpose(tsai_wu)
```

--Code Ends------------------------
