**Problem 2.**

Solution checked with attached MATLAB script

>> HW7.m

Failure: tsai\_hill = 1.643259

1.0e+03 \*

-0.5000 0.0648 1.0000

-0.1450 -0.1398 0.0450

-0.0572 -0.0479 0.0572

Within Tsai-Wu Failure Envelope

Within Hashin Failure Envelope

Clearly, Tsai-Hill is the least conservative in the 4th quadrant given by the stress state; it’s the only one predicting failure. Within this quadrant, Tsai-Wu is the least conservative as it allows compression in the 2-direction to delay tensile failure in the 1-direction.

clear all

clc

format compact

theta = 10;

s = sind(theta);

c = cosd(theta);

T = [ c^2 s^2 2\*s\*c ;...

s^2 c^2 -2\*s\*c ;...

-c\*s c\*s c^2-s^2 ];

slp = 1000;

sln = 500;

stp = 45;

stn = 145;

slt = 57.2;

stt = 32.4;

E1 = 163e3;

E2 = 11.3e3;

G12 = 5.5e3;

nu12 = 3.13;

stress\_xy = [75 -150 -10]';

stress\_12 = T \* stress\_xy;

%% Tsai-Hill criteria

if stress\_12(1) > 0 && stress\_12(2) > 0

tsai\_hill = (stress\_12(1)^2 / slp^2) - ( stress\_12(1)\*stress\_12(2) / slp^2) + (stress\_12(2)^2 / stp^2) + (stress\_12(3)^2 / slt^2);

elseif stress\_12(1) < 0 && stress\_12(2) > 0

tsai\_hill = (stress\_12(1)^2 / sln^2) - ( stress\_12(1)\*stress\_12(2) / sln^2) + (stress\_12(2)^2 / stp^2) + (stress\_12(3)^2 / slt^2);

elseif stress\_12(1) > 0 && stress\_12(2) < 0

tsai\_hill = (stress\_12(1)^2 / slp^2) - ( stress\_12(1)\*stress\_12(2) / slp^2) + (stress\_12(2)^2 / stn^2) + (stress\_12(3)^2 / slt^2);

elseif stress\_12(1) < 0 && stress\_12(2) < 0

tsai\_hill = (stress\_12(1)^2 / sln^2) - ( stress\_12(1)\*stress\_12(2) / sln^2) + (stress\_12(2)^2 / stn^2) + (stress\_12(3)^2 / slt^2);

end

% Check the value of tsai\_hill against failure (> 1)

if tsai\_hill > 1

% Print a message and the relevant limits

fprintf(' Failure: tsai\_hill = %f\n', tsai\_hill)

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

end

%% Tsai-Wu criteria

F1 = 1/slp + 1/sln;

F11 = -1/(slp\*sln);

F2 = 1/stp + 1/stn;

F22 = -1/(stp\*stn);

F66 = 1/slt^2;

tsai\_wu = F1\*stress\_12(1) + F2 \* stress\_12(2) + F11\*stress\_12(1)^2 + F22\*stress\_12(2)^2 + F66\*stress\_12(3)^2;

% Check the value of tsai\_hill against failure (> 1)

if tsai\_wu > 1

% Print a message and the relevant limits

fprintf(' Failure: tsai\_wu = %f\n Stress State:\n', tsai\_wu)

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

else

fprintf(' Within Tsai-Wu Failure Envelope\n')

end

%% Hashin criteria

if stress\_12(1) > 0

hash\_fiber = stress\_12(1)/slp;

else

hash\_fiber = stress\_12(1)/-sln;

end

if stress\_12(2) > 0

hash\_matrix = (stress\_12(2)/slp)^2 + (stress\_12(3)/slt)^2;

else

hash\_matrix = (stress\_12(2)/(2\*stt))^2 + (stress\_12(2)/stp)\*((stn/(2\*stt))^2-1) + (stress\_12(3)/slt)^2;

end

% Check the value of tsai\_wu against failure for fiber and matrix

if hash\_fiber > 1

if stress\_12(1) > 0

% Print a message and the relevant limits

fprintf(' Tensile Fiber Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

else

% Print a message and the relevant limits

fprintf(' Compressive Fiber Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

end

elseif hash\_matrix > 1

if stress\_12(2) > 0

% Print a message and the relevant limits

fprintf(' Tensile Matrix Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

else

% Print a message and the relevant limits

fprintf(' Compressive Matrix Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

end

else

fprintf(' Within Hashin Failure Envelope\n')

end

**Problem 3.**

function [ ] = lamina\_failure( )

%lamina\_failure takes a string excelfilename and opens and reads the data

% in the excel file. Compares data using the Maximum Stress and Maximum

% Strain criteria to determine failure.

% get some info from the file being read

status = xlsfinfo('HW7.xlsx');

% Check the status variable to make sure the file is an Excel sheet

% If the status isn't the same as the string given

if strcmp(status,'Microsoft Excel Spreadsheet') == 0

% display an error message

disp('Error: File not an Excel sheet.')

% and return nothing

return

% If the string does match, continue the function

else

% Read data from the excel file

data = xlsread('HW7.xlsx');

% Extract individual components from the data matrix

stress\_xy = data(1:3,1);

strengths = data(:,2);

mat\_props = data(1:4,3);

theta = data(1,4);

E1 = mat\_props(1);

E2 = mat\_props(2);

G12 = mat\_props(3);

nu12 = mat\_props(4);

slp = strengths(1);

sln = strengths(2);

stp = strengths(3);

stn = strengths(4);

slt = strengths(5);

stt = strengths(6);

% Calculate nu\_21 from nu\_12

nu21 = nu12 \* E2 / E1;

% Calculate sin and cos

s = sind(theta);

c = cosd(theta);

% Transformation matrix

T = [ c^2 s^2 2\*s\*c ;...

s^2 c^2 -2\*s\*c ;...

-c\*s c\*s c^2-s^2 ];

% Calculate stress state

stress\_12 = T \* stress\_xy;

%% Maximum Stress criteria

fprintf('\nMaximum Stress Criteria:\n')

% Check the Stresses in each direction

% Check against positive and negative 1-direction

if stress\_12(1) < -sln

% Print a message and the relevant limits

fprintf(' Fails in 1-direction compression\n')

show = [ -sln stress\_12(1) slp ];

disp(show)

elseif stress\_12(1) > slp

% Print a message and the relevant limits

fprintf(' Fails in 1-direction tension\n')

show = [ -sln stress\_12(1) slp ];

disp(show)

end

% Check against positive and negative 2-direction

if stress\_12(2) < -stn

% Print a message and the relevant limits

fprintf(' Fails in 2-direction compression\n')

show = [ -stn stress\_12(2) stp ];

disp(show)

elseif stress\_12(2) > stp

% Print a message and the relevant limits

fprintf(' Fails in 2-direction tension\n')

show = [ -stn stress\_12(2) stp ];

disp(show)

end

% Check in 1-2-direction

if abs(stress\_12(3)) > slt

% Print a message and the relevant limits

fprintf(' Fails in shear\n')

show = [ -slt stress\_12(3) slt ];

disp(show)

end

% If it passes all the others, it must pass through this to be within

% the failure envelope

if stress\_12(1) >= -sln && stress\_12(1) <= slp &&...

stress\_12(2) >= -stn && stress\_12(2) <= stp &&...

abs(stress\_12(3)) <= slt

% Print a message and the relevant limits

fprintf(' Within failure envelope\n')

end

%% Maximum Strain criteria

% Calculate strains

strain\_12 = [ E1^-1 -nu21/E2 0; -nu12/E1 E2^-1 0; 0 0 G12^-1 ] \* stress\_12;

% Calculate max strains

elp = slp/E1;

eln = sln/E1;

etp = stp/E2;

etn = stn/E2;

elt = slt/G12;

% Check the Strains in each direction

fprintf('\nMaximum Strain Criteria:\n')

% Check against positive and negative 1-direction

if strain\_12(1) < -eln

% Print a message and the relevant limits

fprintf(' Fails in 1-direction compression\n')

show = [ -eln strain\_12(1) elp ];

disp(show)

elseif strain\_12(1) > elp

% Print a message and the relevant limits

fprintf(' Fails in 1-direction tension\n')

show = [ -eln strain\_12(1) elp ];

disp(show)

end

% Check against positive and negative 2-direction

if strain\_12(2) < -etn

% Print a message and the relevant limits

fprintf(' Fails in 2-direction compression\n')

show = [ -etn strain\_12(2) etp ];

disp(show)

elseif strain\_12(2) > etp

% Print a message and the relevant limits

fprintf(' Fails in 2-direction tension\n')

show = [ -etn strain\_12(2) etp ];

disp(show)

end

% Check in 1-2-direction

if abs(strain\_12(3)) > elt

% Print a message and the relevant limits

fprintf(' Fails in shear\n')

show = [ -elt strain\_12(3) elt ];

disp(show)

end

% If it passes all the others, it must pass through this to be within

% the failure envelope

if strain\_12(1) >= -eln && strain\_12(1) <= elp &&...

strain\_12(2) >= -etn && strain\_12(2) <= etp &&...

abs(strain\_12(3)) <= elt

% Print a message and the relevant limits

fprintf(' Within failure envelope\n')

show2 = [ [ -eln; -etn; -elt ] strain\_12 [ elp; etp; elt ] ];

disp(show2)

end

%% Tsai-Hill criteria

fprintf('\nTsai-Hill Criteria:\n')

if stress\_12(1) > 0 && stress\_12(2) > 0

tsai\_hill = (stress\_12(1)^2 / slp^2) - ( stress\_12(1)\*stress\_12(2) / slp^2) + (stress\_12(2)^2 / stp^2) + (stress\_12(3)^2 / slt^2);

elseif stress\_12(1) < 0 && stress\_12(2) > 0

tsai\_hill = (stress\_12(1)^2 / sln^2) - ( stress\_12(1)\*stress\_12(2) / sln^2) + (stress\_12(2)^2 / stp^2) + (stress\_12(3)^2 / slt^2);

elseif stress\_12(1) > 0 && stress\_12(2) < 0

tsai\_hill = (stress\_12(1)^2 / slp^2) - ( stress\_12(1)\*stress\_12(2) / slp^2) + (stress\_12(2)^2 / stn^2) + (stress\_12(3)^2 / slt^2);

elseif stress\_12(1) < 0 && stress\_12(2) < 0

tsai\_hill = (stress\_12(1)^2 / sln^2) - ( stress\_12(1)\*stress\_12(2) / sln^2) + (stress\_12(2)^2 / stn^2) + (stress\_12(3)^2 / slt^2);

end

% Check the value of tsai\_hill against failure (> 1)

if tsai\_hill > 1

% Print a message and the relevant limits

fprintf(' Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

end

%% Tsai-Wu criteria

fprintf('\nTsai-Wu Criteria:\n')

F1 = 1/slp + 1/sln;

F11 = -1/(slp\*sln);

F2 = 1/stp + 1/stn;

F22 = -1/(stp\*stn);

F66 = 1/slt^2;

tsai\_wu = F1\*stress\_12(1) + F2 \* stress\_12(2) + F11\*stress\_12(1)^2 + F22\*stress\_12(2)^2 + F66\*stress\_12(3)^2;

% Check the value of tsai\_hill against failure (> 1)

if tsai\_wu > 1

% Print a message and the relevant limits

fprintf(' Failure\n Stress State:\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

else

fprintf(' Within Failure Envelope\n')

end

%% Hashin criteria

fprintf('\nHashins Criteria:\n')

if stress\_12(1) > 0

hash\_fiber = stress\_12(1)/slp;

else

hash\_fiber = stress\_12(1)/-sln;

end

if stress\_12(2) > 0

hash\_matrix = (stress\_12(2)/slp)^2 + (stress\_12(3)/slt)^2;

else

hash\_matrix = (stress\_12(2)/(2\*stt))^2 + (stress\_12(2)/stp)\*((stn/(2\*stt))^2-1) + (stress\_12(3)/slt)^2;

end

% Check the value of tsai\_wu against failure for fiber and matrix

if hash\_fiber > 1

if stress\_12(1) > 0

% Print a message and the relevant limits

fprintf(' Tensile Fiber Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

else

% Print a message and the relevant limits

fprintf(' Compressive Fiber Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

end

elseif hash\_matrix > 1

if stress\_12(2) > 0

% Print a message and the relevant limits

fprintf(' Tensile Matrix Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

else

% Print a message and the relevant limits

fprintf(' Compressive Matrix Failure\n')

show = [ [ -sln; -stn; -slt ] stress\_12 [ slp; stp; slt ] ];

disp(show)

end

else

fprintf(' Within Failure Envelope\n')

end

end