clear; clc; format SHORTG;

%Coordinates of Airfoil and conversion to 39.5 cm

C = table2array(readtable('Clark Y Airfoil Coordinates.xlsx'));

x\_offset = 4.8;

y\_offset = 2.7;

C = [C(:,3) C(:,4)];

for i = 1:size(C,1)-1

theta(i) = atand((C(i+1,2)-C(i,2))/(C(i+1,1)-C(i,1)));

end

%Coordinates with conversion to 39.5 cm with offset

C1 = [(C(:,1)-x\_offset),C(:,2)-y\_offset];

C1(C1(:,1)<0,:) = [];

C1U = C1(1:(size(C1,1)/2),:);

U = [[6,2.9];[6.3,2.9];[13.5,3.5];[13.5,3.5];[20.5,3.35];[20.5,3.35];

[24.1,2.85];[25.35,2.65];[29.85,1.9];[29.85,1.9];[33.9,1.1];[37.49,0.3]]

U = [U(:,1)-4.8,U(:,2)-2.7];

CU = C(1:(size(C,1)/2),:);

CU = [CU(:,1)-x\_offset,CU(:,2)-y\_offset];

j = find(CU(:,1) == max(CU(CU(:,1) < x\_offset-.5))) % this doesn't work...

alfa = theta(14);

% Linear interpolation of Truss Intersect Points

for i=1:size(U,1)

RUx(i) = U(i,1);

x0y0U = C1U(C1U == max(C1U(RUx(i) > C1U(:,1),1)),1:2);

x1y1U = C1U(C1U == min(C1U(RUx(i) < C1U(:,1),1)),1:2);

RUy(i) = ((x1y1U(2)-x0y0U(2))/(x1y1U(1)-x0y0U(1)))\*(RUx(i)-x0y0U(1))+ x0y0U(2);

end

% Constants:

a = 0.004; % Edge of square beam of wing rib

I = (a^4)/12; % Second Moment of Inertia

E = 6690e6;

% Distances for the cantilever beam on reference frame alpha

for i = 1:size(U,1)

if i == 1

SUBP(i) = sqrt((U(i+1,1)-U(i,1))^2+(U(i+1,2)-U(i,2))^2);

elseif i == size(U,1)

SUBP(i) = SUBP(i-1) + sqrt((CU(end,1)-U(i,1))^2+(CU(end,2)-U(i,2))^2)

else

SUBP(i) = SUBP(i-1) + sqrt((U(i+1,1)-U(i,1))^2+(U(i+1,2)-U(i,2))^2);

end

end

L = [(SUBP(1)/2)\*10 SUBP(2) SUBP(4) SUBP(6) SUBP(7) SUBP(8) SUBP(10) SUBP(11)]

Lx1 = L.\*cosd(alfa);

Ly1 = L.\*sind(alfa);

SUBP = [Lx1' Ly1'];

L = L./100;

Ux1 = [RUx(2) RUx(3) RUx(5) RUx(7) RUx(8) RUx(9) RUx(11) RUx(12)];

Uy1 = [RUy(2) RUy(3) RUy(5) RUy(7) RUy(8) RUy(9) RUy(11) RUy(12)];

TIP = [Ux1',Uy1'];

dy = Uy1 - Ly1;

dy = dy./100;

syms F [1 size(L,2)]

for i = 1:size(L,2)

for j = 1:size(L,2)

if j < i

K(i,j) = ((F(i)\*L(j)^2)\*(3\*L(i)-L(j)))/(6\*E\*I);

elseif j > i

K(i,j) = ((F(i)\*L(i)^2)\*(3\*L(j)-L(i)))/(6\*E\*I);

else

K(i,j) = ((F(i)\*L(j)^3)/(3\*E\*I));

end

end

end

K\_sum = sum(K);

for i = 1:size(L,2)

eqn(i) = K\_sum(i) == dy(i);

end

F\_sol = solve(eqn,F);

F\_sol\_cell = struct2cell(F\_sol);

F\_sol\_cat = cat(2,F\_sol\_cell{:});

F\_sol\_double = double(F\_sol\_cat)

Fx = -1.\*F\_sol\_double.\*sind(alfa);

Fy = F\_sol\_double.\*cosd(alfa);

F1 = [Fx' Fy'];

dy\_RDM6 = [-4.431e-4 -1.08e-2 -3.178e-2 -4.499e-2 -4.990e-2 -6.871e-2 -8.695e-2 -1.04e-1];

dy\_RDM6 = dy\_RDM6./cosd(alfa);

per\_error = abs((abs(dy\_RDM6-dy)./dy\_RDM6).\*100)

max\_error = max(abs(dy-dy\_RDM6))