```
clear; clc; format SHORTG;

%Coordinates of Aerofoil 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),:);

% Upper beam coordinates from real-life scale drawing of wing rib
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];

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
% SUBP calculations
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;

% TIP calcuations
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; % in the reference frame 1 (y1)
dy = dy./(100*cosd(alfa)); % in m and reference frame alfa (y alfa)

% 8-by-8 matrix representing superpositioning of cantilever beam
% deflections
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

% Solving for the forces
F_sol = solve(eqn,F);

% Conversion to display what the forces are in numeric form
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'];

dy1 = [dy.*sind(alfa);dy.*cosd(alfa)];

% in m and in reference frame 1
dy_RDM6 = [-4.602e-2 -1.121 -3.301 -4.672 -5.182 -7.136 -9.030 -1.08e1].*1e-2;
```

```
% Percentage error calculations
per_error = abs((abs(dy_RDM6-dy1(2,:))./dy_RDM6).*100)
[max_error, idx] = max(abs(dy1(2,:)-dy_RDM6))
```