NANYANG TECHNOLOGICAL UNIVERSITY
School of Mechanical and Aerospace Engineering

MA 4704

AEROELASTICITY

Name of Student: Seah Yong Le Stanley

Matric No: U2121954F **Date: 18 October 2024**

Name of Supervisor: Prof Ng Bing Feng

Grade: ________________________

Contents

1 TASK 1

a) What is a possible aircraft model that has this wing?

Since the length of the wing is 30m, the possible aircrafts are likely to be Airbus A350 or Boeing 787. Comparing the two aircrafts, the Airbus A350 has a Rolls Royce Trent XWB engine whereas the 787 has a Rolls Royce Trent 1000 engine. Therefore, looking at the diagram given in the CA, the aircraft model has an engine that fits the description of the Rolls Royce Trent 1000, hence the possible Aircraft Model is the likely Boeing 787-8 Dreamliner

b) What is the value of y for the engine location?

Considering the distance of engine to the middle of the plane, as well as the length of fuselage, the value of y is 6.845 m

$$y = 9.73 - 0.5 * 5.77 = 6.845\ m$$

c) What is the mass of the engine for this aircraft?

The mass of one engine is 6120 kg (Rolls-Royce Trent 1000)

d) What are the assumptions for the assembly to only undergo pure bending vibration?

1) Homogenous linear isotropic material properties
2) Centre of Gravity of wing and Elastic axis coincides so bending and torsion is decoupled
3) External forces act through the shear centre, hence there is only bending and no torsion

e) What are some of the methods to increase the resonant frequency of the wing?

$$\kappa_1 = \frac{\omega_1}{\sqrt{\frac{EI}{\rho A l^4}}}$$

Based on the above formula derived in lecture slides, one way to increase the resonant frequency is to use stiffer materials that have a higher Young's Modulus E, as well as materials that have a lower density, resulting in lower mass. Another way to increase resonant frequency is to increase the Wing thickness to increase I

2 TASK 2

2.1 Deriving Mass and Stiffness Matrix

Key Terms:

$$Youngs'\ Modulus, E = 2 \times 10^{11}\ N\ m^{-2}$$

$$Density, \rho = 1600\ kg\ m^{-3}$$

Based on Lecture Slides,

$$K_{wing} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \dot{\xi_i}(t)\dot{\xi_j}(t)\int_0^l \rho A(y)\phi_i(y)\phi_j(y)dy$$

$$K_{engine} = \frac{1}{2}\mu m\dot{v}^2(l,t) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{engine}\phi_i(6.845)\phi_j(6.845)\dot{\xi_i}(t)\dot{\xi_j}(t)$$

$$Total\ KE = \bar{K} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\int_0^l \rho A\phi_i(y)\phi_j(y)dy + m_{engine}\phi_i(6.845)\phi_j(6.845)]\,\dot{\xi_i}(t)\dot{\xi_j}(t)$$

$$P = \frac{1}{2}\int_0^l EI(y)\left(\frac{\delta^2 v}{\delta y^2}\right)dy = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \dot{\xi_i}(t)\dot{\xi_j}(t)\int_0^l EI(y)\phi_i''(y)\phi_j''(y)dy$$

$$Mass\ Matrix = M = \int_0^l \rho A(y)\phi_i(y)\phi_j(y)dy + m_{engine}\phi_i(6.845)\phi_j(6.845)$$

$$Stiffness\ Matrix = \int_0^l EI(y)\phi_i''(y)\phi_j''(y)dy$$

Based on previously defined Shape Function in Slides,

$$\phi_i(y) = \frac{\left(\frac{y}{l}\right)^{i+1}\left[2 + i - i\left(\frac{y}{l}\right)\right]}{i(i+1)(i+2)}$$

2.2 Mass and Stiffness Matrix Obtained

Based on N = 7,

$$Mass\ Matrix = \begin{pmatrix} 2047 & 422 & 142 & 61 & 30 & 16 & 10 \\ 422 & 90 & 31 & 13.5 & 6.8 & 3.8 & 2.2 \\ 142 & 31 & 10.9 & 4.8 & 2.4 & 1.4 & 0.8 \\ 61 & 13.5 & 4.8 & 2.1 & 1.1 & 0.6 & 0.4 \\ 30 & 6.8 & 2.4 & 1.1 & 0.6 & 0.3 & 0.2 \\ 16 & 3.8 & 1.4 & 0.6 & 0.3 & 0.2 & 0.1 \\ 9 & 2.2 & 0.8 & 0.4 & 0.2 & 0.1 & 0.07 \end{pmatrix}$$

$$Stiffness\ Matrix = \begin{pmatrix} 1994473 & 190151 & 35104 & 9371 & 3202 & 1309 & 613 \\ 190151 & 35104 & 9371 & 3202 & 1309 & 613 & 318 \\ 35104 & 9371 & 3202 & 1309 & 613 & 318 & 180 \\ 9371 & 3202 & 1309 & 613 & 318 & 180 & 109 \\ 3202 & 13039 & 613 & 318 & 180 & 109 & 70 \\ 1309 & 613 & 318 & 180 & 109 & 70 & 47 \\ 613 & 318 & 180 & 109 & 70 & 47 & 33 \end{pmatrix}$$

2.3 Results

n	1st Natural Freq (Hz)	2nd Natural Freq (Hz)	3rd Natural Freq (Hz)	4th Natural Freq (Hz)	Percentage change in 4th Natural Freq (%)
4	2.60883	9.179996	25.91908	91.82206	-
5	2.60883	8.714356	20.10399	47.77597	47.96896
6	2.60443	8.713737	18.78183	35.72593	25.22197
7	2.60249	8.657349	18.77981	32.83250	8.098963
8	2.60230	8.621849	18.52677	32.83004	0.007493

N = 7 used

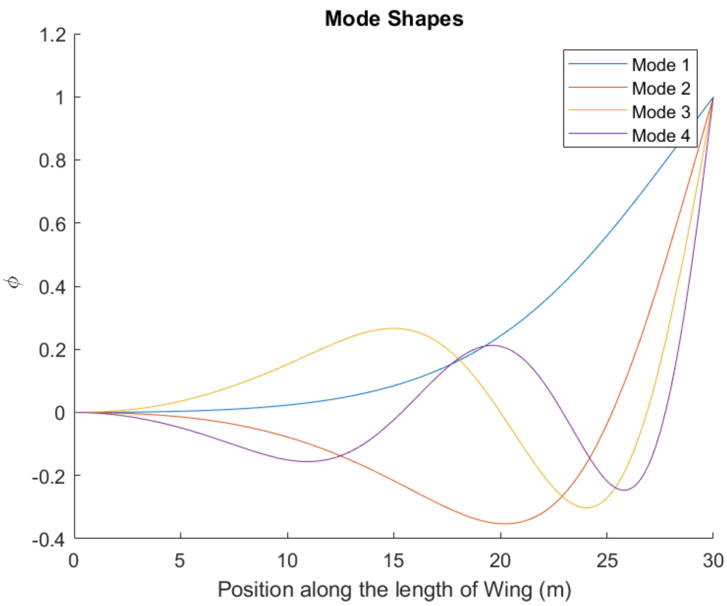

Figure 1: Mode Shapes 1 to 4

As n increases, the 4th natural frequency generally decreases as it converges to a certain point, whereas the differences are not as pronounced in the 1st to 3rd natural frequency. The graph for the different mode shapes generally coincides with the shapes as shown in the lecture, where the mode shapes were derived from the equation:

$$\phi_i(y) = \sum_{k=1}^{n} \left(\{\bar{\xi}\}\right)_k \phi_k(y)$$

3 TASK 3

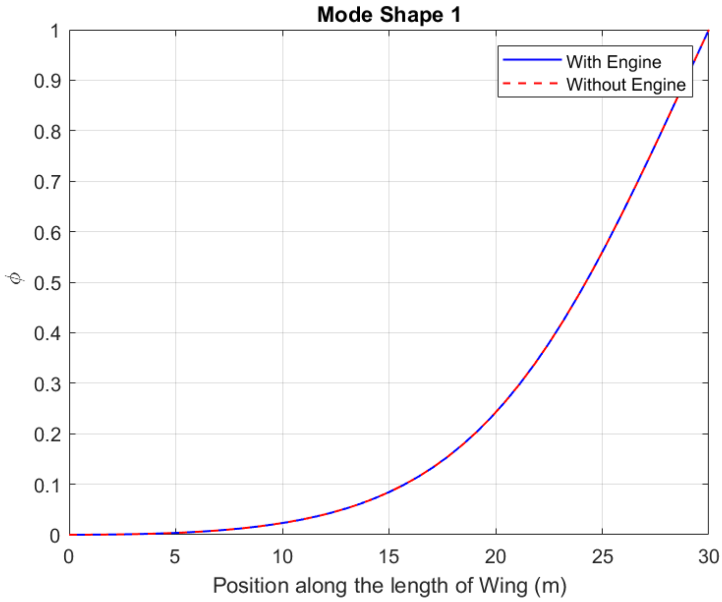

Figure 2: Mode Shape 1 with and without Engine

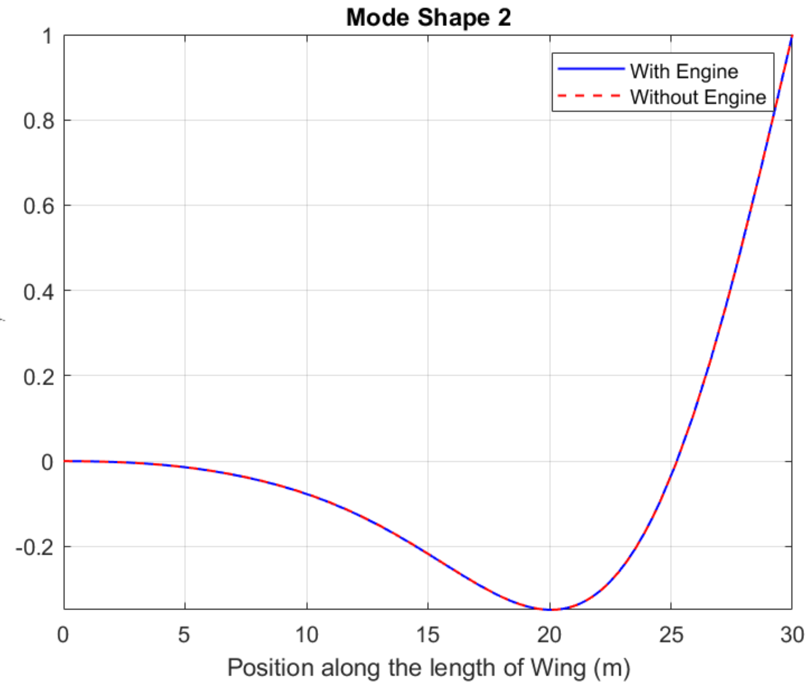

Figure 3: Mode Shape 2 with and without Engine

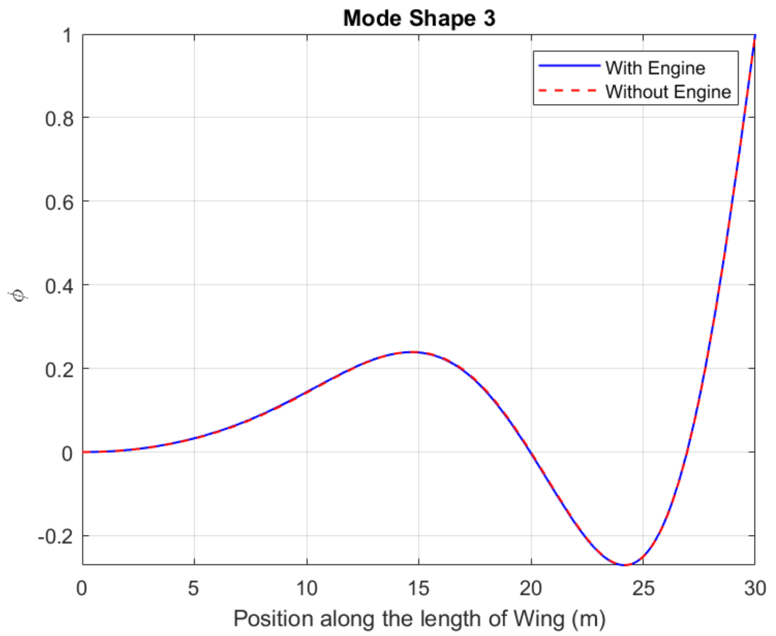

Figure 4: Mode Shape 3 with and without Engine

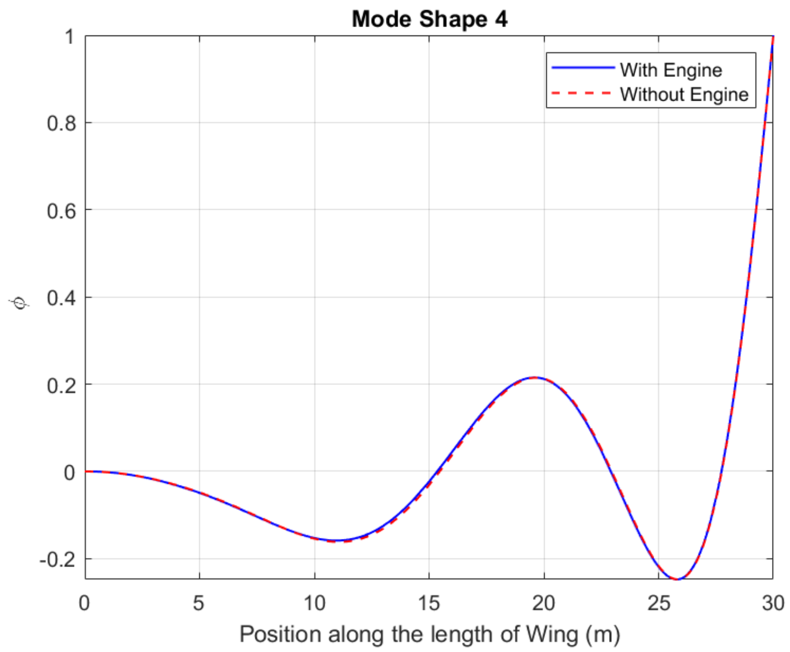

Figure 5: Mode Shape 4 with and without Engine

Looking at the plots for the different mode shapes with and without an engine, the mode shapes do not have much difference for all mode shapes, where only the graph for Mode Shape 4 shows a slight difference between the two graphs. This demonstrates that the mass of the engine has little to no effect on the natural vibrations of the wing, meaning that the engine has little influence on the wing's overall vibrational behaviour.

4 TASK 4

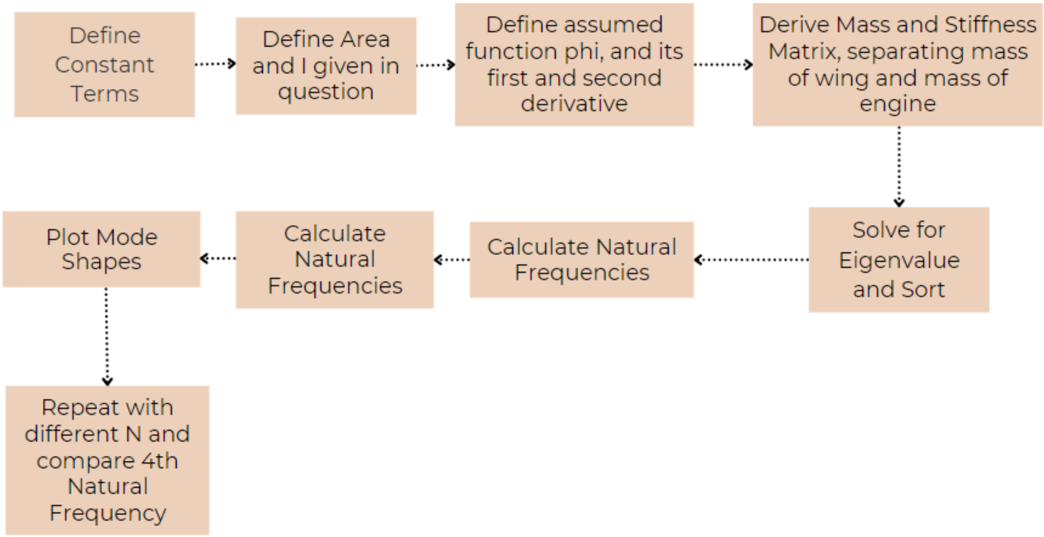

Figure 6: Flow Chart of Code Structure

5 TASK 5

5.1 Deriving Basic Function 1

As per the lecture, a second derivative of the dimensionless basic function can be modelled as follows, aligning with the assumption that the moment vanishes at the free end $(y = l)$:

$$\phi_i''(y) = \frac{1}{L^3}(L-y)\left[cos\left(\frac{i\pi y}{L}\right)\right]$$

Integrating with respect to y,

$$\phi_i'(y) = \int \frac{1}{L^3}(L-y)\left[cos\left(\frac{i\pi y}{L}\right)\right]dy$$

$$= \frac{1}{L^3}\int L\left[cos\left(\frac{i\pi y}{L}\right)\right]\,dy - \frac{1}{L^3}\int y\left[cos\left(\frac{i\pi y}{L}\right)\right]dy$$

$$= \frac{1}{L^3}\frac{L^2}{i\pi}\sin\left(\frac{i\pi y}{L}\right) - \frac{1}{L^3}\left\{\frac{yL}{i\pi}\sin\left(\frac{i\pi y}{L}\right) - \int\frac{L}{i\pi}\sin\left(\frac{i\pi y}{L}\right)dy\right\}$$

$$= \frac{1}{i\pi L}\sin\left(\frac{i\pi y}{L}\right) - \frac{y}{i\pi L^2}\sin\left(\frac{i\pi y}{L}\right) - \frac{L^2}{i^2\pi^2L^3}\cos\left(\frac{i\pi y}{L}\right)$$

$$= \frac{1}{i\pi L}\sin\left(\frac{i\pi y}{L}\right) - \frac{1}{i^2\pi^2L}\cos\left(\frac{i\pi y}{L}\right) - \frac{y}{i\pi L^2}\sin\left(\frac{i\pi y}{L}\right) + C_1$$

Integrating with respect to y again,

$$\phi_i(y) = \int\left[\frac{1}{i\pi L}\sin\left(\frac{i\pi y}{L}\right) - \frac{1}{i^2\pi^2L}\cos\left(\frac{i\pi y}{L}\right) - \frac{y}{i\pi L^2}\sin\left(\frac{i\pi y}{L}\right) + C_1\right]dy$$

$$= -\frac{1}{i^2\pi^2}\cos\left(\frac{i\pi y}{L}\right) - \frac{1}{i^3\pi^3}\sin\left(\frac{i\pi y}{L}\right) - \frac{1}{i\pi L^2}\left[-\frac{yL}{i\pi}cos\left(\frac{i\pi y}{L}\right) - \int -\frac{L}{i\pi}cos\left(\frac{i\pi y}{L}\right)dy\right] + C_1y + C_2$$

$$= -\frac{1}{i^2\pi^2}\cos\left(\frac{i\pi y}{L}\right) - \frac{1}{i^3\pi^3}\sin\left(\frac{i\pi y}{L}\right) + \frac{y}{i^2\pi^2L}\cos\left(\frac{i\pi y}{L}\right) - \frac{1}{i^3\pi^3}\sin\left(\frac{i\pi y}{L}\right) + C_1y + C_2$$

$$= \frac{\frac{y}{L}-1}{i^2\pi^2}\cos\left(\frac{i\pi y}{L}\right) - \frac{2}{i^3\pi^3}\sin\left(\frac{i\pi y}{L}\right) + C_1y + C_2$$

Setting boundary condition for zero slope1,

$$\phi_i'(0) = -\frac{1}{i^2\pi^2L} + C_1 = 0$$

$$\therefore C_1 = \frac{1}{i^2\pi^2L}$$

Setting boundary conditions for zero displacement,

$$\phi_i(0) = -\frac{1}{i^2\pi^2} + C_2 = 0$$

$$\therefore C_2 = \frac{1}{i^2\pi^2}$$

Hence,

$$\phi_i(y) = \frac{\frac{y}{L} - 1}{i^2\pi^2}\cos\left(\frac{i\pi y}{L}\right) - \frac{2}{i^3\pi^3}\sin\left(\frac{i\pi y}{L}\right) + \frac{1}{i^2\pi^2 L}y + \frac{1}{i^2\pi^2}$$

Where the function is a complete set and are linearly independent.

5.3 Results

n	1st Natural Freq (Hz)	2nd Natural Freq (Hz)	3rd Natural Freq (Hz)	4th Natural Freq (Hz)	Percentage change in 4th Natural Freq (%)
4	2.796257	11.591342	30.49493	81.23606	-
5	2.713160	10.205093	25.43372	54.21679	33.2
6	2.668341	9.5986680	22.42990	45.22325	16.5
7	2.648757	9.2340313	20.97694	39.73373	12.1
8	2.635511	9.0555414	20.03825	36.83257	7.302
9	2.628752	8.9347930	19.54369	34.97038	5.056
10	2.623302	8.8688325	19.20742	33.90049	3.059
11	2.620192	8.8180586	19.01655	33.197681	2.073

N = 10 used

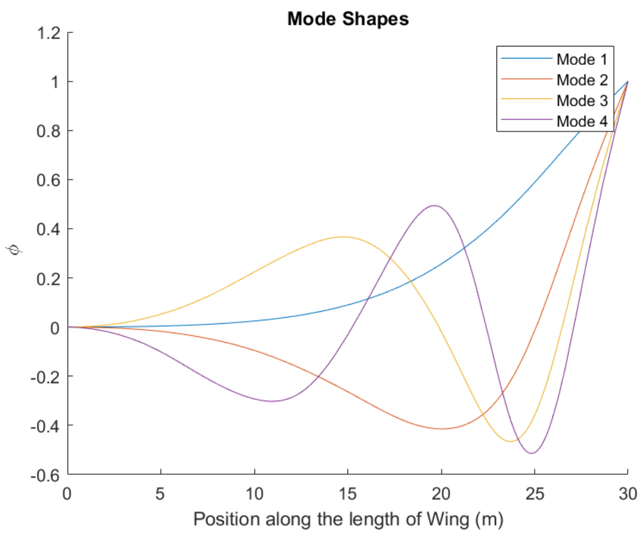

Figure 6: Mode Shapes with Alternative Cosine Function

Based on the natural frequencies obtained, both functions converge to a similar value, with the modal shapes looking similar as well. However, one of the main differences between the two different functions were the computing time taken in calculating and plotting the modal shapes. The original polynomial function had a shorter computing time of less than 1 minute, whereas the cosine function had computing time of more than 15 minutes. Furthermore, the computing time increased drastically whenever N was increased. Comparing between the two functions, the polynomial function took a lesser number of mode shapes compared to the cosine function.

Hence, the polynomial function used within the Raleigh-Ritz method allowed for faster and simpler calculations for approximate modal analysis, especially when the number of mode shapes required are higher.

6 DECLARATION OF HONESTY

Help gotten:

1. Lecture 5 Video on Bending Vibration of Beams – for assumptions for the assembly to only undergo pure bending vibration
2. Lecture 6 Slides on Ritz Method on deriving the mass and stiffness matrixes (for the alternate function)
3. Lecture on CA 2/3 explaining the code and steps
4. Prof Ng Bing Feng
5. Tu Xiayang – for comparing our results and mode shapes to make sure our results are similar and hopefully correct

7 REFERENCES

Boeing Commercial Airplanes. (n.d.). 787 airplane characteristics for airport planning. https://www.boeing.com/content/dam/boeing/boeingdotcom/commercial/airports/acaps/787.pdf

8 APPENDIX

Task 2:

```
clear all
close all

% Constants
L = 30;
rho = 1600;
E = 2*10^11;

n = 7 ;

syms y

% Define A and Second moment of Area
A = -60.78*(y/L)^3 + 121.16*(y/L)^2-79.624*(y/L) + 19.344 ;
I = 2.7462*exp(-7.958*(y/L));

% Define Assumed Functions
func1 = sym(zeros(1,n));
func_d = sym(zeros(1,n));
func_dd = sym(zeros(1,n));

for i = 1 :n
    func1(i) = ((y ./ L).^(i + 1)).*(2 + i - i.*(y/L)) .* (1 / i / (i+1) / (i+2));
    func_d(i) = diff (func1(i) , y);
    func_dd(i) = diff (func_d(i) , y);
end

% Define Mass and Stiffness Matrix
M1 = zeros(n,n);
M2 = zeros(n,n);
M = zeros(n,n);
K = zeros(n,n);

for i = 1 : n
    for j= 1 : n
        % Calculate M1
        m1 = rho .* A .* func1(i) .* func1(j);
        M1(i,j) = int(m1, y, [0 L]);

        % Calculate M2
        M2(i,j) = 6120 * ((6.845/L).^(i + 1)).*(2+i-i.*(6.845/L)) .* (1 / i /
(i+1) / (i+2)).*((6.845/L).^(j+1)).*(2+j-j.*(6.845/L)).* (1 / j / (j+1) / (j+2)) ;

        % Total mass matrix M
        M(i,j) = double(M1(i,j) + M2(i,j));

        % Calculate Stiffness Matrix
        k1= E * I * func_dd(i) * func_dd(j);
        K(i,j) =  double(int(k1, y, [0 L]));
    end
end

[V, d] = eig(K, M);

% Sort eigenvalues and eigenvectors
[d_sorted, Index] = sort(diag(d), 'ascend');
```

```
for j = 1:n
    V_sort(:,j)=V(:,Index(j));
end

% Obtain natural frequencies
nat_freq = zeros(1,n);
for k = 1:n
    nat_freq(1,k) = real(sqrt(d_sorted(k))/2/pi);
end
nat_freq;
fprintf(['First Four Natural Frequencies (Hz), ...' ...
    'overall: \n %9.2E \n %9.2E \n %9.2E \n %9.2E\n'],...
    nat_freq(1),nat_freq(2),nat_freq(3), nat_freq(4));


% Calculate and plot each mode shape
for j = 1:4
    mode_shape = 0;
    for i = 1:n
        mode_shape = mode_shape + ((y/L)^(i+1))*(2+i-i*(y/L)) /
(i*(i+1)*(i+2)) .*V_sort(i,j);
    end

    figure(1)
    hold on
    mode_shape = mode_shape ./ subs(mode_shape,L);
    fplot(mode_shape,[0,L])
end

title('Mode Shapes')
xlabel('Position along the length of Wing (m)')
ylabel('$ ~\mathrm{\phi}$', 'interpreter','latex')
legend('Mode 1', 'Mode 2', 'Mode 3', 'Mode 4')
hold off;
```

Task 3:

```
clear all
close all

% Constants
L = 30;
rho = 1600;
E = 2*10^11;
y_engine = 6.845;

n = 7 ;

syms y

% Define A and Second moment of Area
A = -60.78*(y/L)^3 + 121.16*(y/L)^2-79.624*(y/L) + 19.344 ;
I = 2.7462*exp(-7.958*(y/L));

% Define Assumed Functions
func1 = sym(zeros(1,n));
func_d = sym(zeros(1,n));
func_dd = sym(zeros(1,n));

for i = 1 :n
    func1(i) = ((y ./ L).^(i + 1)).*(2 + i - i.*(y/L)) .* (1 / i / (i+1) / (i+2));
    func_d(i) = diff (func1(i) , y);
    func_dd(i) = diff (func_d(i) , y);
end

% Define Mass and Stiffness Matrix
M1 = zeros(n,n);
M2 = zeros(n,n);
M = zeros(n,n);
K = zeros(n,n);
m = zeros (n,n);


for i = 1 : n
    for j= 1 : n
        % Calculate M1
        m1 = rho .* A .* func1(i) .* func1(j);
        M1(i,j) = int(m1, y, [0 L]);

        % Calculate M2
        M2(i,j) = 6120 * ((6.845/L).^(i + 1)).*(2+i-i.*(6.845/L)) .* (1 / i /
(i+1) / (i+2)).*((6.845/L).^(j+1)).*(2+j-j.*(6.845/L)).* (1 / j / (j+1) / (j+2)) ;

        % Total mass matrix M
        M(i,j) = double(M1(i,j) + M2(i,j));
        m(i,j) = double (M1(i,j)) ;

        % Calculate Stiffness Matrix
        k1= E * I * func_dd(i) * func_dd(j);
        K(i,j) =  double(int(k1, y, [0 L]));
    end
end

[V, d] = eig(K, M);
```

```
% Sort eigenvalues and eigenvectors
[d_sorted, Index] = sort(diag(d), 'ascend');
for j = 1:n
    V_sort(:,j)=V(:,Index(j));
end

[V2, d2] = eig(K, m);

% Sort eigenvalues and eigenvectors
[d_sorted2, Index2] = sort(diag(d2), 'ascend');
for j = 1:n
    V_sort2(:,j)=V2(:,Index2(j));
end

difference = V_sort - V_sort2

% Obtain natural frequencies
nat_freq = zeros(1,n);
for k = 1:n
    nat_freq(1,k) = real(sqrt(d_sorted(k))/2/pi);
end
nat_freq;
fprintf(['First Four Natural Frequencies (Hz) with engine, ...' ...
    'overall: \n %9.2E \n %9.2E \n %9.2E \n %9.2E\n'],...
    nat_freq(1),nat_freq(2),nat_freq(3), nat_freq(4));

% Obtain natural frequencies
nat_freq2 = zeros(1,n);
for k = 1:n
    nat_freq2(1,k) = real(sqrt(d_sorted2(k))/2/pi);
end
nat_freq;
fprintf(['First Four Natural Frequencies (Hz) without engine, ...' ...
    'overall: \n %9.2E \n %9.2E \n %9.2E \n %9.2E\n'],...
    nat_freq2(1),nat_freq2(2),nat_freq2(3), nat_freq2(4));

% Calculate and plot each mode shape
for j = 1:4
    mode_shape = 0;
    mode_shape2 = 0;
    for i = 1:n
        mode_shape = mode_shape + ((y/L)^(i+1))*(2+i-i*(y/L)) /
(i*(i+1)*(i+2)) .*V_sort(i,j) ;
        mode_shape2 = mode_shape2 + ((y/L)^(i+1))*(2+i-i*(y/L)) /
(i*(i+1)*(i+2)) .*V_sort2(i,j);
    end

    % Normalize mode shapes
    mode_shape = mode_shape ./ subs(mode_shape,L);
    mode_shape2 = mode_shape2 ./ subs(mode_shape2,L);

    % Create a new figure for each mode
    figure;

    % Plot mode shape with engine
    fplot(matlabFunction(mode_shape), [0, L],'-b','LineWidth', 1); % Blue for with
engine
    hold on;
```

```
    % Plot mode shape without engine
    fplot(matlabFunction(mode_shape2), [0, L], "--r", 'LineWidth', 1); % Red for
without engine

    % Formatting the plot
    title(['Mode Shape ' num2str(j)]);
    xlabel('Position along the length of Wing (m)');
    ylabel('$ ~\mathrm{\phi}$', 'interpreter','latex');
    legend('With Engine', 'Without Engine');
    grid on;

    % Release the hold for the next figure
    hold off;
end
```

Task 5:

```
clear all
close all

% Constants
L = 30;
rho = 1600;
E = 2*10^11;

n = 10;

syms y

% Define A and Second moment of Area
A = -60.78*(y/L)^3 + 121.16*(y/L)^2-79.624*(y/L) + 19.344 ;
I = 2.7462*exp(-7.958*(y/L));

% Define Assumed Functions
func1 = sym(zeros(1,n));
func_d = sym(zeros(1,n));
func_dd = sym(zeros(1,n));

for i = 1 :n
    func1(i) = (((y/L-1)/(i^2*pi^2))*cos(i*pi*y/L)-
(2/(i^3*pi^3))*sin(i*pi*y/L)+y/(i^2*pi^2*L)+1/(i^2*pi^2));
    func_d(i) = diff (func1(i) , y);
    func_dd(i) = diff (func_d(i) , y);
end

% Define Mass and Stiffness Matrix
M1 = zeros(n,n);
M2 = zeros(n,n);
M = zeros(n,n);
K = zeros(n,n);

for i = 1 : n
    for j= 1 : n
        % Calculate M1
        m1 = rho .* A .* func1(i) .* func1(j);
        M1(i,j) = int(m1, y, [0 L]);

        % Calculate M2
        M2(i,j) = 6120 * (((6.845/L-1)/(i^2*pi^2))*cos(i*pi*6.845 /L)-
(2/(i^3*pi^3))*sin(i*pi*6.845 /L)+6.845 /(i^2*pi^2*L)+1/(i^2*pi^2)) * (((6.845/L-
1)/(j^2*pi^2))*cos(j*pi*6.845 /L)-(2/(j^3*pi^3))*sin(j*pi*6.845 /L)+6.845
/(j^2*pi^2*L)+1/(j^2*pi^2)) ;

        % Total mass matrix M
        M(i,j) = double(M1(i,j) + M2(i,j));

        % Calculate Stiffness Matrix
        k1= E * I * func_dd(i) * func_dd(j);
        K(i,j) =  double(int(k1, y, [0 L]));
    end
end

[V, d] = eig(K, M);
```

```matlab
% Sort eigenvalues and eigenvectors
[d_sorted, Index] = sort(diag(d), 'ascend');
for j = 1:n
    V_sort(:,j)=V(:,Index(j));
end

% Obtain natural frequencies
nat_freq = zeros(1,n);
for k = 1:n
    nat_freq(1,k) = real(sqrt(d_sorted(k))/2/pi);
end
nat_freq;
fprintf(['First Four Natural Frequencies (Hz), ...' ...
    'overall: \n %9.2E \n %9.2E \n %9.2E \n %9.2E\n'],...
    nat_freq(1),nat_freq(2),nat_freq(3), nat_freq(4));


% Calculate and plot each mode shape
for j = 1:4
    mode_shape = 0;
    for i = 1:n
        mode_shape = mode_shape + (((y/L-1)/(i^2*pi^2))*cos(i*pi*y/L)-
(2/(i^3*pi^3))*sin(i*pi*y/L)+y/(i^2*pi^2*L)+1/(i^2*pi^2)) .*V_sort(i,j);
    end

    figure(1)
    hold on
    mode_shape = mode_shape ./ subs(mode_shape,L);
    fplot(mode_shape,[0,L])
end

title('Mode Shapes')
xlabel('Position along the length of Wing (m)')
ylabel('$ ~\mathrm{\phi}$', 'interpreter','latex')
legend('Mode 1', 'Mode 2', 'Mode 3', 'Mode 4')
hold off;
```