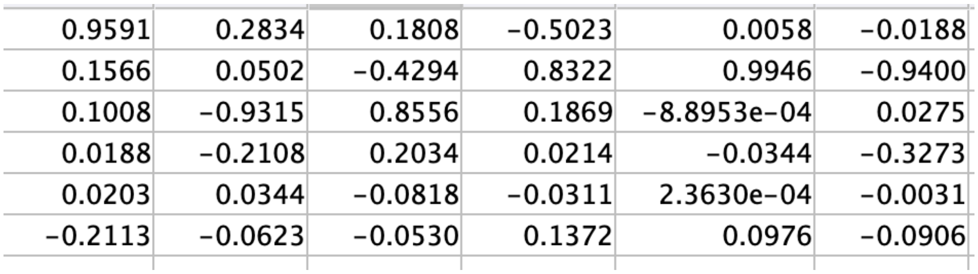
The results of the code are:

Vectors:



| | | | | | |
|---|---|---|---|---|---|
| 0.9591 | 0.2834 | 0.1808 | -0.5023 | 0.0058 | -0.0188 |
| 0.1566 | 0.0502 | -0.4294 | 0.8322 | 0.9946 | -0.9400 |
| 0.1008 | -0.9315 | 0.8556 | 0.1869 | -8.8953e-04 | 0.0275 |
| 0.0188 | -0.2108 | 0.2034 | 0.0214 | -0.0344 | -0.3273 |
| 0.0203 | 0.0344 | -0.0818 | -0.0311 | 2.3630e-04 | -0.0031 |
| -0.2113 | -0.0623 | -0.0530 | 0.1372 | 0.0976 | -0.0906 |

Nature frequency:

```
17267.8276269149    0         0         0         0         0
0         16655.4506897704    0         0         0         0
0         0         3499.16557548451    0         0         0
0         0         0         2040.80367696303    0         0
0         0         0         0         147.551476248265    0
0         0         0         0         0         156.140564346498
```

```
From low to high:

0.00583910332442152
0.994615026224670
-0.000889530904402633
-0.0343804628693519
0.000236303862484630
0.0975909378406063

-0.0188007141900911
-0.939971427735358
0.0274924747046342
-0.327305960005226
-0.00307886003087313
-0.0905855462404336

-0.502338275926197     0.180822993645488      0.283407662678247      0.959117273755345
0.832162797195946      -0.429435530663129     0.0501504353378320     0.156571345987759
0.186874075090766      0.855571384694054      -0.931474129971031     0.100786073748199
0.0213587244771269     0.203409984348149      -0.210843661029658     0.0187634255262462
-0.0311495094466494    -0.0818418168371135    0.0343624257544725     0.0203498963098960
0.137160229335727      -0.0530289633138930    -0.0623309719583162    -0.211318365791563

with
```

147.551476248265, 156.140564346498, 2040.80367696303, 3499.16557548451, 16655.4506897704, 17267.8276269149

```
----------------

% Name: Jiahui Lu
% UID: 204945099
clear;
density = 2698.9; % kg/m^3
```

```
E = 68e9; G = 25e9; % Gpa
M = density * 0.05 * 0.05 * 0.1; % Mass in kg
CenterOfMass = [0.05 0.225 0.025]; % Center of the mass
% Moment of inertia for x, y, z paralleling axis
Ix = M * ((0.05)^2 + (0.05)^2) / 12;
Iy = M * ((0.05)^2 + (0.1)^2) / 12;
Iz = M * ((0.05)^2 + (0.1)^2) / 12;
delta = [0 0 0 1 1 1; 0 0 0 1 1 1;0 0 0 1 1 1; 1 1 1 0 0 0; 1 1 1 0 0 0; 1 1
1 0 0 0];
K = zeros(6, 6);
%% Stiffness Matrix: First element - the wire
N = zeros(6,6);
n2 = [0 0 1];
n3 = [0 1 0];
n1 = [-1 0 0]; % n1 = n2 X n3
% Demensional data
h = 0.15; % height
w = 0.0025; % width
t = 0.0025; % thickness
A = w * t;
% Moment of inertia
I1 = w*(t^3)/12;
I2 = t*(w^3)/12;
% vector L
L = [0.0025/2 0.05 + 0.15 0.05 - 0.0025/2];
% The calculation for J
val = 0;
for n=1:2:100
    val = val + (tanh(n*pi*t/(2*w)) / (n^5));
end
J = ((w^3)*t / 3) * (1 - ((192*w / ((pi^5)*t)) * val));
% Fill the N matrix
N(1:3, 1) = n1.';
N(1:3, 2) = n2.';
N(1:3, 3) = n3.';
N(4:6, 4) = n1.';
N(4:6, 5) = n2.';
N(4:6, 6) = n3.';
N(4:6, 1) = (cross(L, n1)).';
N(4:6, 2) = (cross(L, n2)).';
N(4:6, 3) = (cross(L, n3)).';
% N dot delta
NDelta = zeros(6,6);
NDelta(1:6, 1:3) = N(1:6, 4:6);
NDelta(1:6, 4:6) = N(1:6, 1:3);
C = zeros(6, 6);
% Fill the C matrix
C(1, 1) = h / (E*I1);
C(1, 5) = -(h^2) / (2*E*I1);
C(2, 2) = h / (E*I2);
C(2, 4) = (h^2) / (2*E*I2);
C(3, 3) = h / (G*J);
C(4, 2) = (h^2) / (2*E*I2);
C(4, 4) = (h^3) / (3*E*I2);
C(5, 1) = -(h^2) / (2*E*I1);
C(5, 5) = (h^3) / (3*E*I1);
C(6, 6) = h / (E*A);
```

```
S = inv(C);
K1 = NDelta * S * inv(N); % K1 = (NDelta / C) / N

%% Stiffness Matrix: Second element - the flat
N = zeros(6,6);
n2 = [1 0 0];
n3 = [0 1 0];
n1 = [0 0 1]; % n1 = n2 X n3
% Demensional data
h = 0.15; % height
w = 0.05; % width
t = 0.0025; % thickness
A = w * t;
% Moment of inertia
I1 = w*(t^3)/12;
I2 = t*(w^3)/12;
% vector L
L = [0.1 - 0.0025/2 0.05 + 0.15 0.05/2];
% The calculation for J
val = 0;
for n=1:2:100
    val = val + (tanh(n*pi*w/(2*t)) / (n^5));
end
J = ((t^3)*w / 3) * (1 - ((192*t / ((pi^5)*w)) * val));
% Fill the N matrix
N(1:3, 1) = n1.';
N(1:3, 2) = n2.';
N(1:3, 3) = n3.';
N(4:6, 4) = n1.';
N(4:6, 5) = n2.';
N(4:6, 6) = n3.';
N(4:6, 1) = (cross(L, n1)).';
N(4:6, 2) = (cross(L, n2)).';
N(4:6, 3) = (cross(L, n3)).';
% N dot delta
NDelta = zeros(6,6);
NDelta(1:6, 1:3) = N(1:6, 4:6);
NDelta(1:6, 4:6) = N(1:6, 1:3);
C = zeros(6, 6);
% Fill the C matrix
C(1, 1) = h / (E*I1);
C(1, 5) = -(h^2) / (2*E*I1);
C(2, 2) = h / (E*I2);
C(2, 4) = (h^2) / (2*E*I2);
C(3, 3) = h / (G*J);
C(4, 2) = (h^2) / (2*E*I2);
C(4, 4) = (h^3) / (3*E*I2);
C(5, 1) = -(h^2) / (2*E*I1);
C(5, 5) = (h^3) / (3*E*I1);
C(6, 6) = h / (E*A);

S = inv(C);
K2 = NDelta * S * inv(N); % K1 = (NDelta / C) / N

K = K1 + K2;

%% Mass Matrix
```

```
N = zeros(6,6);
n1 = [1 0 0];
n2 = [0 1 0];
n3 = [0 0 1];
% Fill the N matrix
N(1:3, 1) = n1.';
N(1:3, 2) = n2.';
N(1:3, 3) = n3.';
N(4:6, 4) = n1.';
N(4:6, 5) = n2.';
N(4:6, 6) = n3.';
N(4:6, 1) = (cross(CenterOfMass, n1)).';
N(4:6, 2) = (cross(CenterOfMass, n2)).';
N(4:6, 3) = (cross(CenterOfMass, n3)).';
% N dot delta
NDelta = zeros(6,6);
NDelta(1:6, 1:3) = N(1:6, 4:6);
NDelta(1:6, 4:6) = N(1:6, 1:3);
% Diag matrix In
In = zeros(6, 6);
In(1, 1) = Ix;
In(2, 2) = Iy;
In(3, 3) = Iz;
In(4, 4) = M;
In(5, 5) = M;
In(6, 6) = M;

M = NDelta * In * inv(N);

[a, b] = eig(inv(M) * K);
res = sqrt(b);
```