AE 6311: Advanced Structural Dynamics
Homework 3

James Grisham

April 4, 2015

Problem 1

Problem Statement

Beam vibration be Ritz-super-element:

(a) Develop a 10-DOF deam Ritz-super-element for a uniform beam using the following data: $L = 100$ in., $E = 10^7$ psi, $\rho = 0.1/386.4$ lb-s^2/in^4, $A = 4$ in^2, $I = 16/12$ in^4. Use monomials for the basis function and use the nodal DOF given in Figure A.

(b) Perform vibration analysis using the model developed in part (a) and compare the first 3 natural frequencies with those computed by FEM for the following BC's:

  (1) C-F

  (2) C-S

  (3) C-C

  (4) S-S

(c) Compare the first three mode shapes of the C-S model computed by the two methods by plotting them on the same figure using different colors and line types. Identify which is which using a legend.

(d) Solve the C-S beam using $\{c\}$-based model also. You should get the same results as the C-S beam results of part (b).

Solution

The ten degree of freedom Ritz-super-element was developed using MATLAB. The length of the beam had to be shortened to 25 inches so that the inverse of the transformation matrix $\mathbf{A}$ was not ill-conditioned. A general MATLAB function was written named `beam_ritz.m`. The code can be seen in the appendix.

Plotting output is optional. Some sample output is shown in Figure 1.

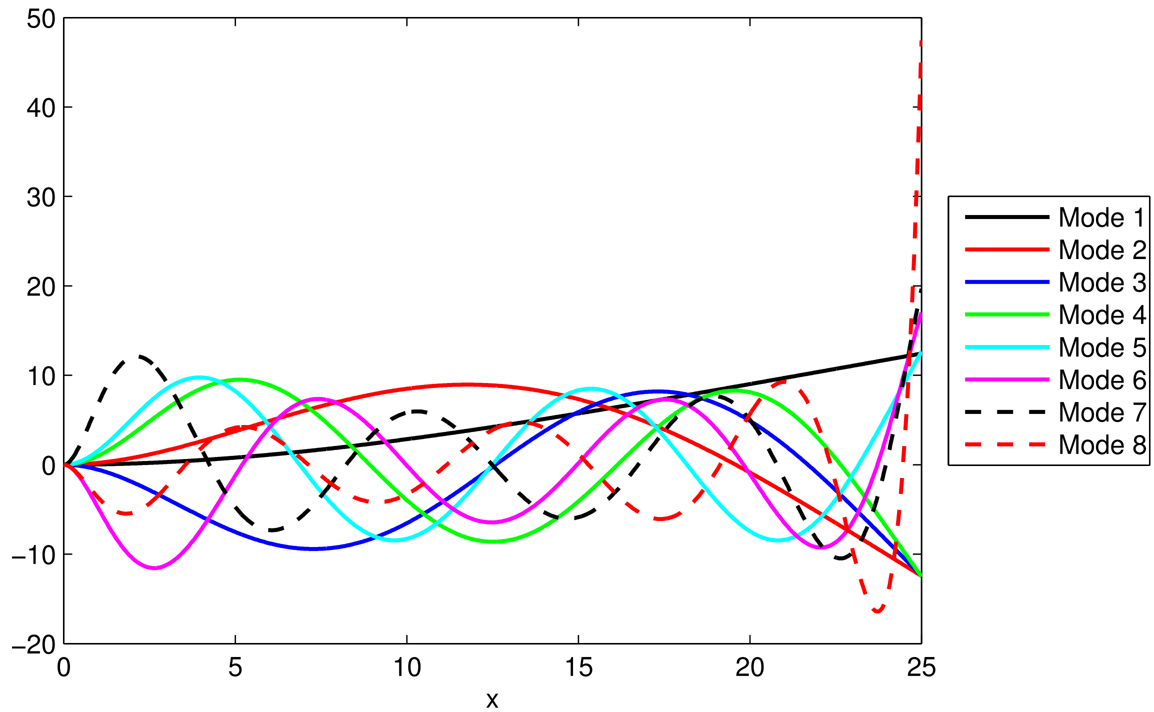

Figure 1: Sample output for a clamped-free beam.

The results from the function were compared with results from FEM using the provided `FEA_Beam_Vibration_2015A` function. A comparison between the natural frequencies was written out to a file. These files are shown below:

Clamped-free.

```
Ritz          FEM           diff
6.384526E+02 6.384534E+02 8.313472E-04
4.001114E+03 4.001315E+03 2.009247E-01
1.120325E+04 1.120754E+04 4.296264E+00
```

Clamped-supported.

```
Ritz          FEM           diff
2.799702E+03 2.799772E+03 7.015927E-02
9.072832E+03 9.075188E+03 2.355847E+00
1.894161E+04 1.895080E+04 9.192744E+00
```

Clamped-clamped.

```
Ritz          FEM           diff
4.062634E+03 4.062848E+03 2.142946E-01
1.119882E+04 1.120324E+04 4.426208E+00
2.199540E+04 2.198694E+04 8.459277E+00
```

Supported-supported.

```
Ritz         FEM          diff
1.792164E+03 1.792182E+03 1.841343E-02
7.168655E+03 7.169821E+03 1.166148E+00
1.613342E+04 1.614253E+04 9.112735E+00
```

A comparison between the first three mode shapes is shown below:

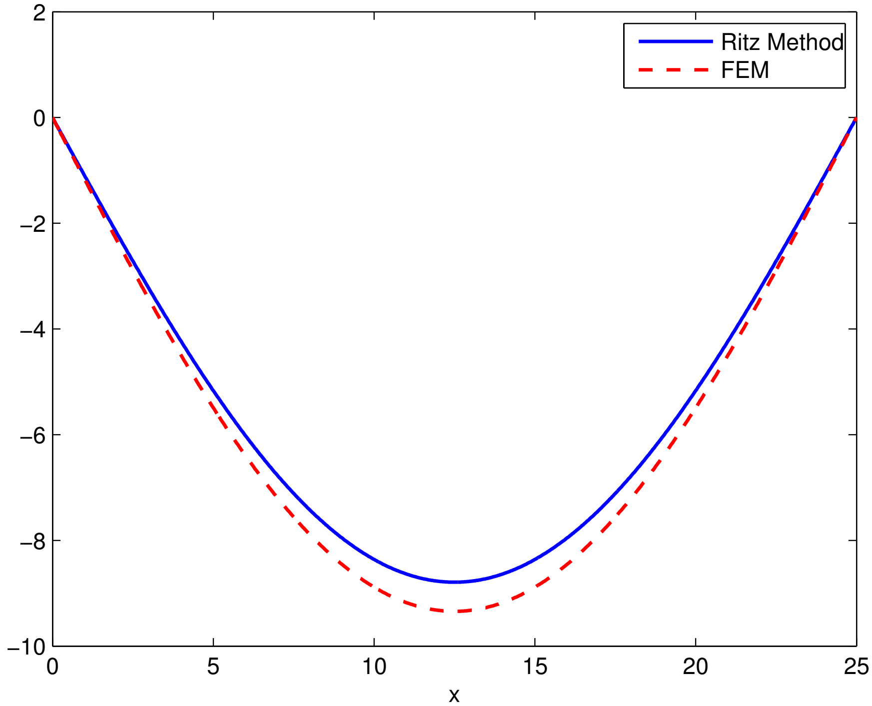

Figure 2: Mode 1.

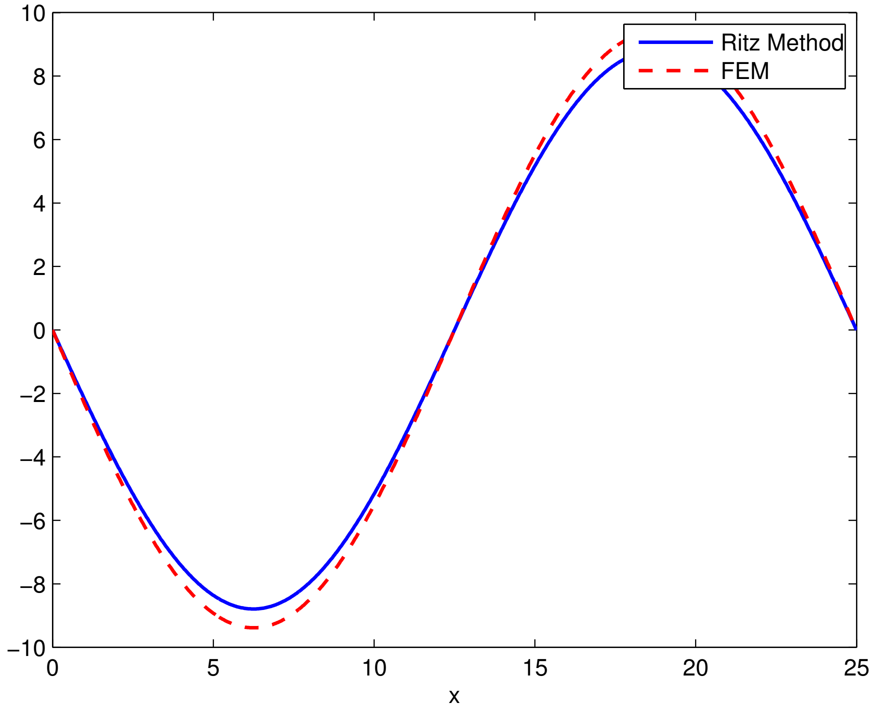

Figure 3: Mode 2.

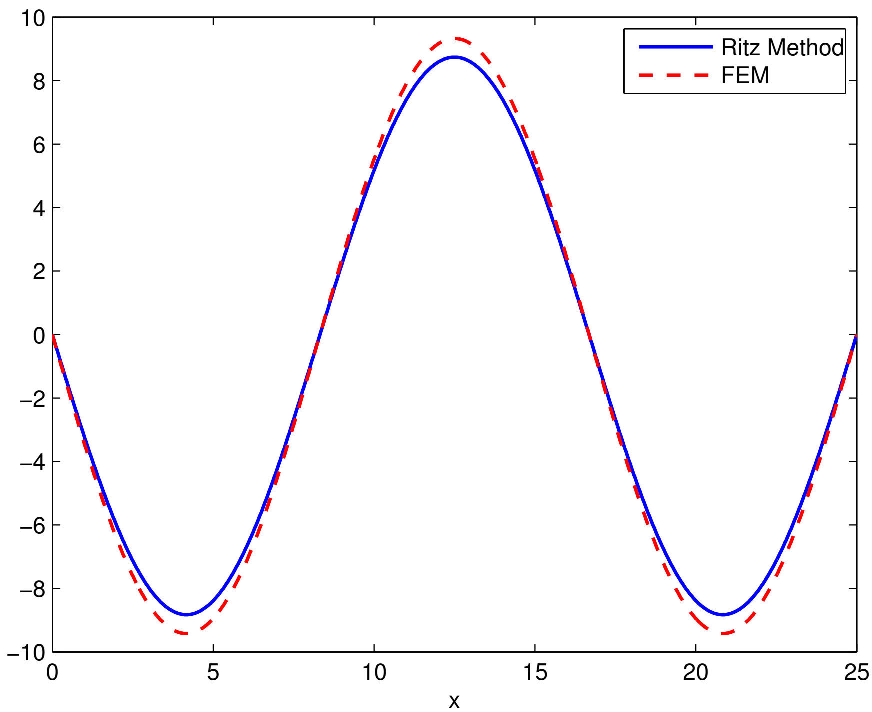

Figure 4: Mode 3.

The sample output for the code along with the comparison between the {u} and {c} coordinates is shown below.

```
hw3prob1 running in /home/James/Desktop/School/Courses/UTA/AE_6311_Advanced_Structural_Dynamics/ma

Applying boundary conditions for clamped-free beam.
Saving figure 1 to ../Images/hw3prob1_ex

Comparison between natural frequencies for different coordinates:

{u}          {c}          difference
------------------------------------------
6.384526E+02 6.384526E+02 8.523102E-10
4.001114E+03 4.001114E+03 5.238689E-10
1.120325E+04 1.120325E+04 4.401591E-08
2.199287E+04 2.199287E+04 2.007963E-05
3.651559E+04 3.651559E+04 6.542820E-04
6.463980E+04 6.463980E+04 7.117439E-03
9.735290E+04 9.735289E+04 7.699433E-03
4.705376E+05 4.705379E+05 3.376772E-01
------------------------------------------

Applying boundary conditions for clamped-clamped beam.

Comparison between natural frequencies for different coordinates:

{u}          {c}          difference
------------------------------------------
4.062634E+03 4.062634E+03 2.388197E-08
1.119882E+04 1.119882E+04 8.805000E-08
2.199540E+04 2.199540E+04 4.951198E-06
3.652130E+04 3.652130E+04 1.078016E-03
6.417781E+04 6.417781E+04 5.056503E-04
9.657149E+04 9.657156E+04 7.511296E-02
------------------------------------------

Applying boundary conditions for supported-supported beam.

Comparison between natural frequencies for different coordinates:

{u}          {c}          difference
------------------------------------------
1.792164E+03 1.792164E+03 1.990543E-08
7.168655E+03 7.168655E+03 1.859553E-08
1.613342E+04 1.613342E+04 2.404357E-06
2.871303E+04 2.871304E+04 3.299791E-03
4.846001E+04 4.846001E+04 2.022306E-03
7.327961E+04 7.328006E+04 4.535312E-01
2.474807E+05 2.474806E+05 9.834092E-02
```

```
3.737277E+05 3.737417E+05 1.405169E+01
------------------------------------------

Applying boundary conditions for clamped-supported beam.

Comparison between natural frequencies for different coordinates:

{u}          {c}          difference
------------------------------------------
2.799702E+03 2.799702E+03 2.254274E-08
9.072832E+03 9.072832E+03 1.307671E-07
1.894161E+04 1.894161E+04 7.493109E-07
3.253298E+04 3.253297E+04 5.442493E-03
5.475655E+04 5.475655E+04 4.319568E-03
8.624124E+04 8.624062E+04 6.232433E-01
2.987697E+05 2.987672E+05 2.487715E+00
------------------------------------------

Saving image to ../Images/hw3prob1_mode1
Saving image to ../Images/hw3prob1_mode2
Saving image to ../Images/hw3prob1_mode3

Done.

Data files are in ./data
```

Problem 2

Problem Statement

Beam vibration be Ritz-super-element:

(a) Develop a 10-DOF deam Ritz-super-element for a uniform beam using the following data: $L = 100$ in., $E = 10^7$ psi, $\rho = 0.1/386.4$ lb-s^2/in^4, $A = 4$ in^2, $I = 16/12$ in^4. Use poly-cosine for the basis function and use the nodal DOF given in Figure A (i.e., $\psi = [1\ x\ x^2\ x^3\ \cos(r\pi x/L)\ \ldots\ \cos(N\pi x/L)]$).

(b) Perform vibration analysis using the model developed in part (a) and compare the first 3 natural frequencies with those computed by FEM for the following BC's:

(1) C-F

(2) C-S

(3) C-C

(4) S-S

(c) Compare the first three mode shapes of the C-S model computed by the two methods by plotting them on the same figure using different colors and line types. Identify which is which using a legend.

(d) Solve the C-S beam using $\{c\}$-based model also. You should get the same results as the C-S beam results of part (b).

Solution

The results from the function were compared with results from FEM using the provided `FEA_Beam_Vibration_2015A` function. A comparison between the natural frequencies was written out to a file. These files are shown below:

Clamped-free.

```
Ritz         FEM          diff
6.384756E+02 6.384534E+02 2.219424E-02
4.002963E+03 4.001315E+03 1.647728E+00
1.121472E+04 1.120754E+04 7.175504E+00
```

Clamped-supported.

```
Ritz         FEM          diff
2.802103E+03 2.799772E+03 2.331142E+00
9.073387E+03 9.075188E+03 1.801456E+00
1.905367E+04 1.895080E+04 1.028733E+02
```

Clamped-clamped.

```
Ritz         FEM          diff
4.069381E+03 4.062848E+03 6.533083E+00
1.119881E+04 1.120324E+04 4.432861E+00
2.216084E+04 2.198694E+04 1.739013E+02
```

Supported-supported.

```
Ritz         FEM          diff
1.792859E+03 1.792182E+03 6.767185E-01
7.168667E+03 7.169821E+03 1.153494E+00
1.619765E+04 1.614253E+04 5.511255E+01
```

A comparison between the first three mode shapes is shown below:

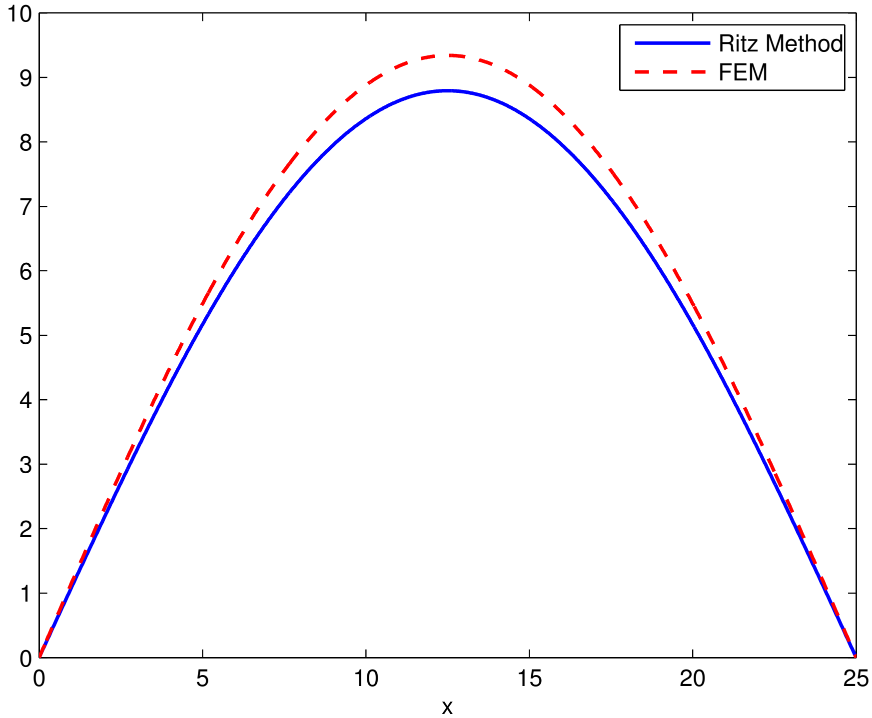

Figure 5: Mode 1.

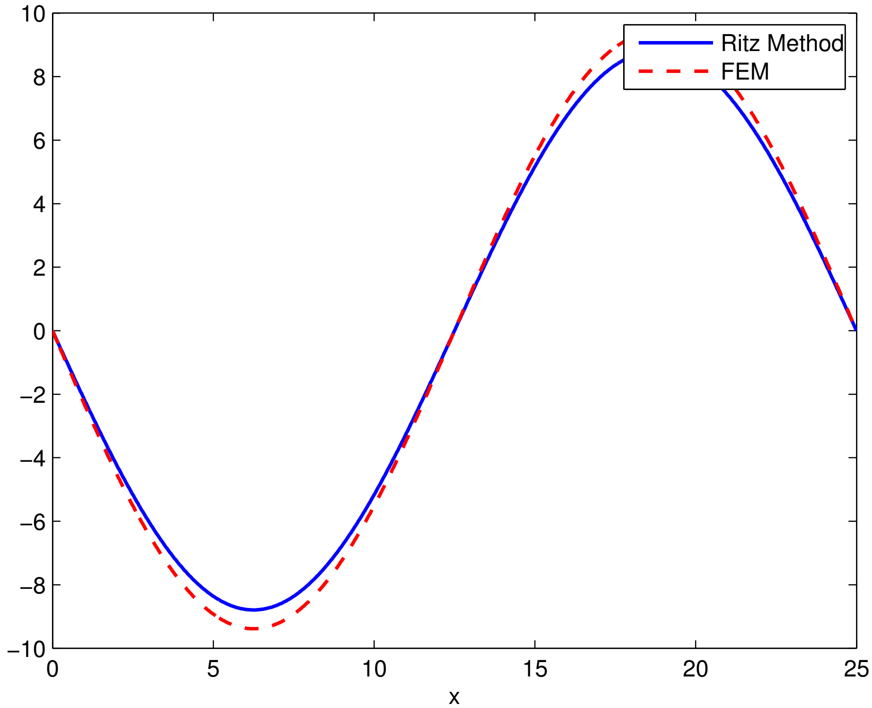

Figure 6: Mode 2.

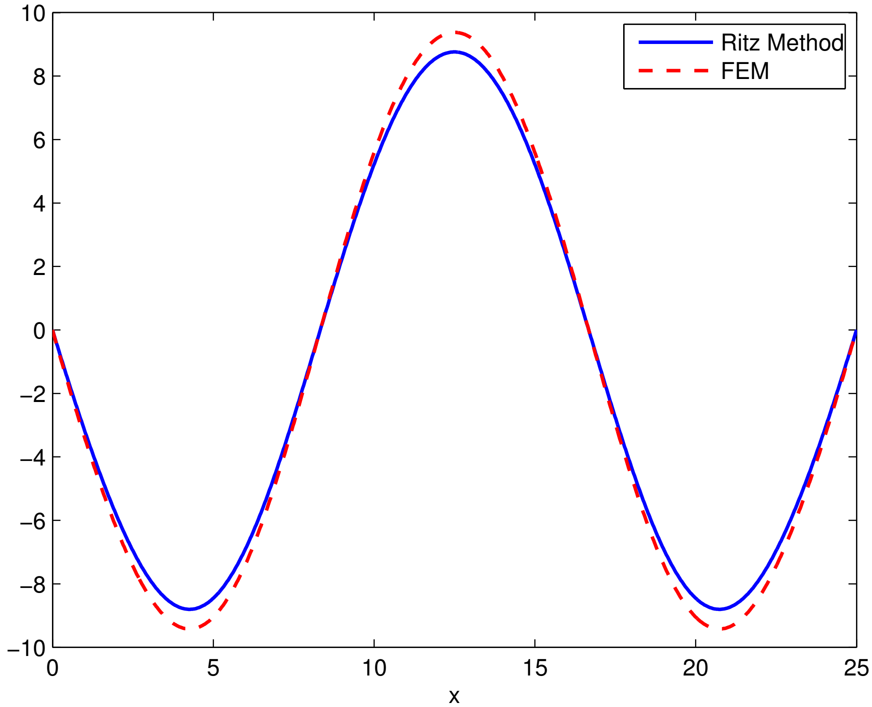

Figure 7: Mode 3.

The comparison between u and c coordinates is shown below:

```
hw3prob2 running in /home/James/Desktop/School/Courses/UTA/AE_6311_Advanced_Structural_Dynamics/ma

Applying boundary conditions for clamped-free beam.
Saving figure 1 to ../Images/hw3prob2_ex

Comparison between natural frequencies for different coordinates:

{u}          {c}          difference
------------------------------------------
6.384756E+02 6.384756E+02 2.604565E-10
4.002963E+03 4.002963E+03 2.996785E-10
1.121472E+04 1.121472E+04 4.616595E-09
2.202195E+04 2.202195E+04 3.626701E-08
3.638876E+04 3.638876E+04 9.184514E-07
5.488336E+04 5.488336E+04 8.860734E-07
8.212326E+04 8.212326E+04 2.903180E-04
2.058005E+05 2.058005E+05 1.558023E-03
------------------------------------------


Applying boundary conditions for clamped-clamped beam.

Comparison between natural frequencies for different coordinates:
```

```
{u}          {c}          difference
-----------------------------------------
4.069381E+03 4.069381E+03 6.684786E-11
1.119881E+04 1.119881E+04 1.129592E-09
2.216084E+04 2.216084E+04 1.158696E-08
3.629685E+04 3.629685E+04 2.553861E-09
5.558426E+04 5.558426E+04 1.721492E-08
8.113310E+04 8.113310E+04 1.797162E-08
-----------------------------------------

Applying boundary conditions for supported-supported beam.

Comparison between natural frequencies for different coordinates:

{u}          {c}          difference
-----------------------------------------
1.792859E+03 1.792859E+03 5.084758E-09
7.168667E+03 7.168667E+03 9.586074E-10
1.619765E+04 1.619765E+04 5.202310E-09
2.868147E+04 2.868147E+04 7.428025E-08
4.562395E+04 4.562395E+04 2.169691E-08
6.587318E+04 6.587318E+04 8.758667E-06
1.098235E+05 1.098235E+05 3.604509E-08
2.641872E+05 2.641872E+05 1.062311E-03
-----------------------------------------

Applying boundary conditions for clamped-supported beam.

Comparison between natural frequencies for different coordinates:

{u}          {c}          difference
-----------------------------------------
2.802103E+03 2.802103E+03 2.160050E-10
9.073387E+03 9.073387E+03 6.748451E-10
1.905367E+04 1.905367E+04 7.668859E-09
3.238041E+04 3.238041E+04 3.423338E-09
5.050509E+04 5.050509E+04 1.534499E-08
7.235792E+04 7.235792E+04 8.601346E-07
1.519068E+05 1.519068E+05 7.107010E-06
-----------------------------------------

Saving image to ../Images/hw3prob2_mode1
Saving image to ../Images/hw3prob2_mode2
Saving image to ../Images/hw3prob2_mode3

Done.
```

```
Data file
```

Problem 3

Problem Statement

Find the beam shape of a linearly tapered beam to maximize the lowest natural frequency, under total beam volume (mass) constraint. The beam is made of isotropic material with Young's modulus E, mass density ρ. The length of the beam is L. The tip mass is w_1/g. The beam has a rectangular cross section with unit thickness. The beam height varies linearly from x_1 to x_2 at the top. The vibration analysis is to be carried out by Ritz-super element using a 10-dof model. Use a monomial basis if the first letter in your last name is in A–M. Use poly-cosine basis if the first letter in your last name is in N–Z. Use the following numerical data:

$L = 60$ in., $E = 10 \times 10^7$ psi, $\rho = 0.1/386.4$ lb-s^2/in^4. The tip weight w_1 is 50 lbs. The allowable volume is 1000 in^3. Plot the optimal beam shape. Plot contours of the objective function and constraint boundaries in the design space. Plot design histories on the contour plots and use symbols to identify the initial and final designs.

Solution

This problem was solved using the same method as before. The only difference is that numerical intgration and differentiation was used instead of symbolic because of speed limitations. The symbolic computations take too long. The results are shown in the below figures.

A lower bound was placed on the design variables. Ideally, the thickness at the right hand side would be zero. In this case, the lower bound was set to 0.01.

$$X_{\text{opt}} = \begin{Bmatrix} 33.3233 \\ 0.0100 \end{Bmatrix} \tag{1}$$

The optimal natural frequency was

$$\omega_1 = 202.76 \text{ rad/s} \tag{2}$$

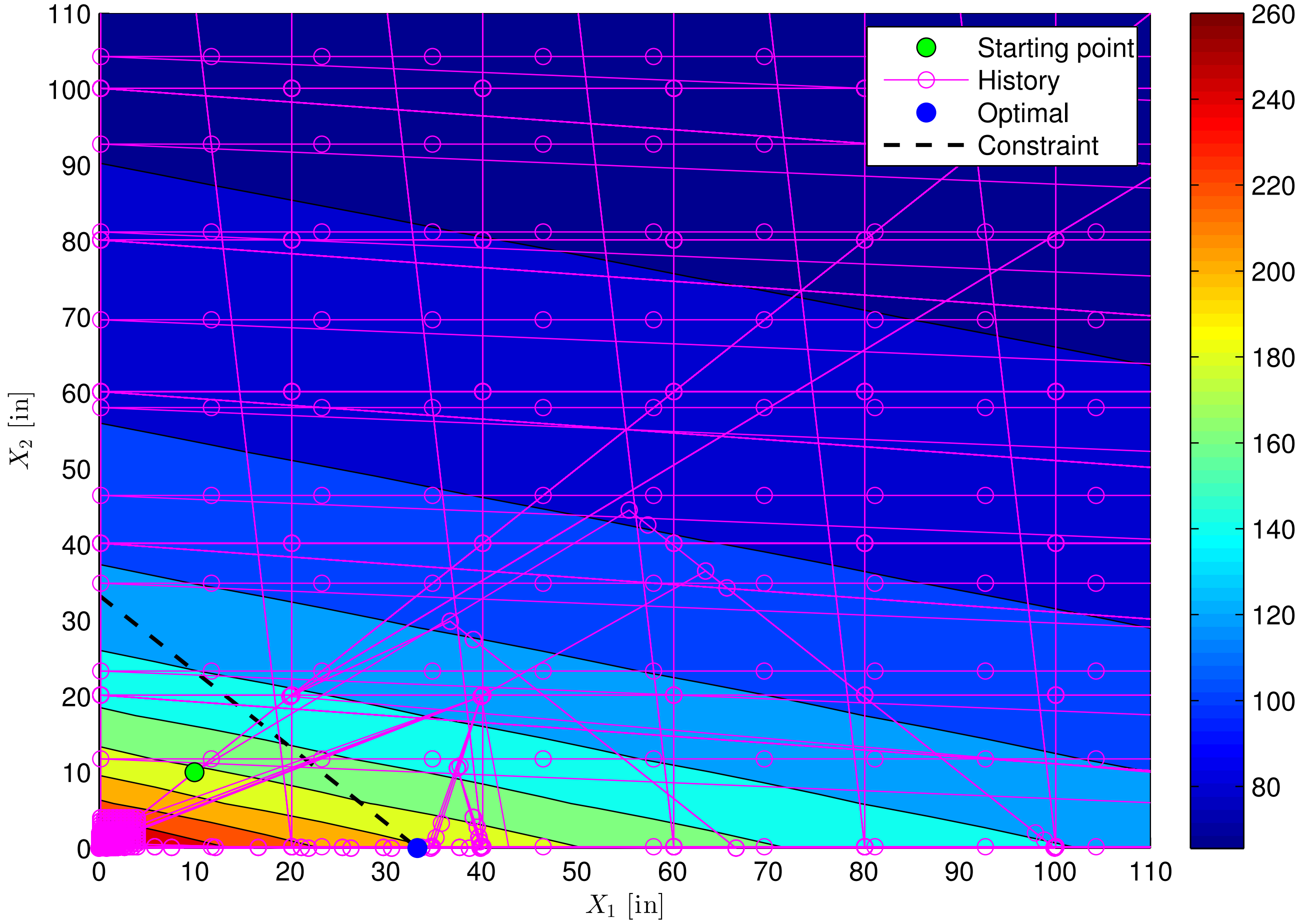

Figure 8: Contours of first natural frequency.

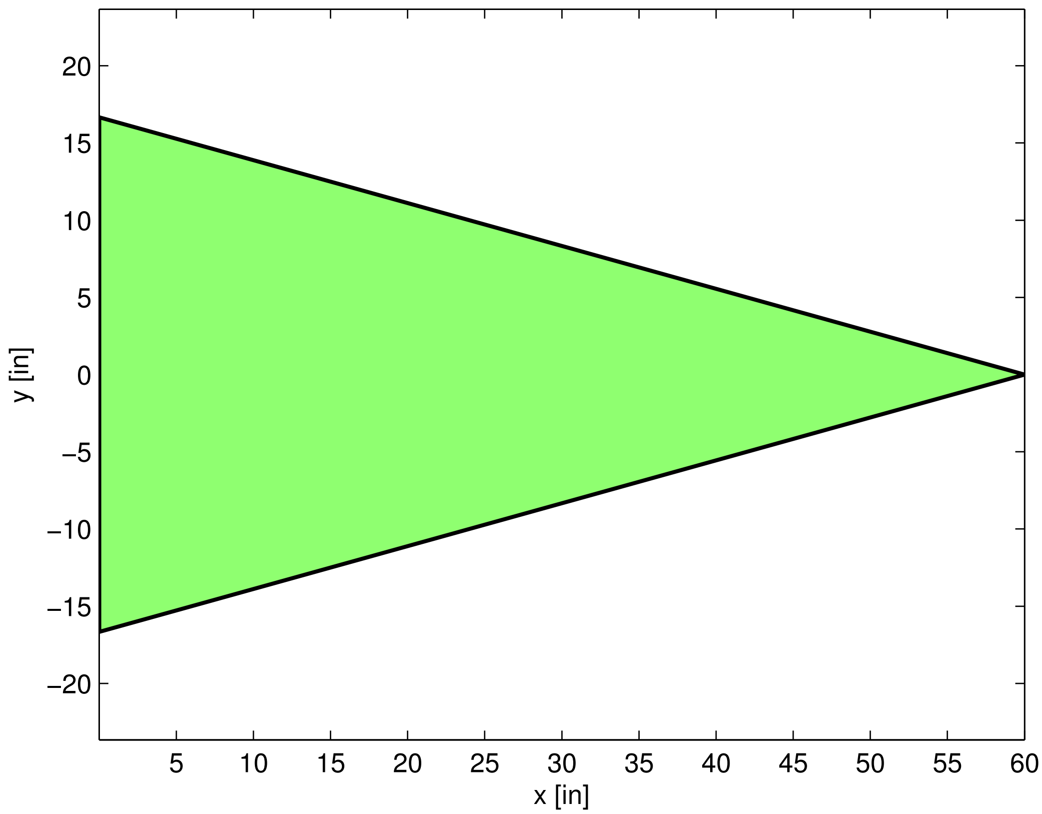

Figure 9: Optimal beam (mass at tip is not included in drawing).

Code listing

```
function varargout = beam_ritz(L, E, rho, A, I, N, BC, b, plt, pnm, ...
    dofn)
% BEAM_RITZ is a function that is used for performing analysis of
%   Euler-Bernoulli beam elements using the Ritz method.  The inputs to the
%   function are explained below.
%
%   Example function call: [Lambda,Phi,fig_handle] = BEAM_RITZ(L,E,rho,A,I)
%
%       L    - length of the beam.
%       E    - modulus of elasticity for the beam.
%       rho  - beam density.
%       A    - cross-sectional area of the beam.
%       I    - Moment of inertia of the beam.
%       N    - number of DOFs used (optional, default is 10).
%       BC   - boundary conditions (optional, default is C-F).  Options are
%              C-F, C-S, C-C, and S-S, where C stands for clamped, F stands
%              for free, and S stands for supported.
%       b    - type of basis used (optional). Possibilities are monomial
%              (default) or polycosine.
%       plt  - optional plotting, set to 'true' for plot output.
%       pnm  - plot name used for saving (optional).
```

```
%       dofn - numbering for degrees of freedom (optional).  Assumed
%              numbering: [1 2 N-1 N 3 4 5 6 ... N-3 N-2] (i.e., the outer
%              nodes are numbered first).
%
%   The outputs are: natural frequencies, numerical mode shapes, and K and
%   M matrices.

% Author       : James Grisham
% Date         : 03/23/2015
% Revision date:

% Checking inputs
if nargin < 5 || nargin > 11
    error('Incorrect arguments.');
elseif nargin == 5
    N = 10;
    BC = 'C-F';
    dofn = zeros(1,N);
    dofn(1:4) = [1 2 N-1 N];
    for n = 5:N
        dofn(n) = n-2;
    end
    plt = 'true';
    pnm = [];
    b = 'monomial';
elseif nargin == 6
    BC = 'C-F';
    dofn = zeros(1,N);
    dofn(1:4) = [1 2 N-1 N];
    for n = 5:N
        dofn(n) = n-2;
    end
    plt = 'true';
    pnm = [];
    b = 'monomial';
elseif nargin == 7
    dofn = zeros(1,N);
    dofn(1:4) = [1 2 N-1 N];
    for n = 5:N
        dofn(n) = n-2;
    end
    plt = 'true';
    pnm = [];
    b = 'monomial';
elseif nargin == 8
    plt = 'true';
    pnm = [];
    dofn = zeros(1,N);
    dofn(1:4) = [1 2 N-1 N];
    for n = 5:N
        dofn(n) = n-2;
    end
elseif nargin == 9
    pnm = [];
```

```
    dofn = zeros(1,N);
    dofn(1:4) = [1 2 N-1 N];
    for n = 5:N
        dofn(n) = n-2;
    end
elseif nargin == 10
    dofn = zeros(1,N);
    dofn(1:4) = [1 2 N-1 N];
    for n = 5:N
        dofn(n) = n-2;
    end
end

% Checking to make sure that the DOFs are even
if mod(N,2)
    error('Number of degrees of freedom must be even.')
end

% Setting up basis functions and forming matrices
syms x
if strcmp(b,'monomial')
    Psi = x.^(0:N-1);
elseif strcmp(b,'polycosine')
    Psi = x.^(0:3);
    for r = 1:N-4
        Psi(r+4) = cos(r*pi*x/L);
    end
else
    error('Basis function option %s not recognized.  Exiting.',b)
end

% Forming mass and stiffness matrices
d2Psi = diff(Psi,x,2);
Mc = int(Psi.'*Psi*rho*A,x,0,L);
Kc = int(d2Psi.'*d2Psi*E*I,x,0,L);

% Applying boundary conditions
if strcmp(BC,'C-F')
    fprintf('Applying boundary conditions for clamped-free beam.\n')
    A = sym(zeros(2,N));
    A(1,:) = subs(Psi,x,0);
    A(2,:) = subs(diff(Psi,x),x,0);
    bcid = [1 2];
elseif strcmp(BC,'C-S')
    fprintf('Applying boundary conditions for clamped-supported beam.\n')
    A = sym(zeros(3,N));
    A(1,:) = subs(Psi,x,0);
    A(2,:) = subs(diff(Psi,x),x,0);
    A(3,:) = subs(Psi,x,L);
    bcid = [1 2 3];
elseif strcmp(BC,'C-C')
    fprintf('Applying boundary conditions for clamped-clamped beam.\n')
    A = sym(zeros(4,N));
    A(1,:) = subs(Psi,x,0);
```

```
    A(2,:) = subs(diff(Psi,x),x,0);
    A(3,:) = subs(Psi,x,L);
    A(4,:) = subs(diff(Psi,x),x,L);
    bcid = [1 2 3 4];
elseif strcmp(BC,'S-S')
    fprintf('Applying boundary conditions for supported-supported beam.\n')
    A = sym(zeros(2,N));
    A(1,:) = subs(Psi,x,0);
    A(2,:) = subs(Psi,x,L);
    bcid = [1 3];
else
    error('Boundary condition %s not recognized.',BC);
end

T = null(A);
KcB = double(T.'*Kc*T);
McB = double(T.'*Mc*T);

% Solving the eigensystem {u}-coordinate system
[Phi,Lambda] = eig(KcB,McB);

% Figuring out coordinates of dofs, assuming a uniform grid
x_u = linspace(0,L,N/2);
index = 1;
x_c = zeros(1,N);
for d = 1/2*(dofn(1:2:end)-1)+1
    x_c(index) = x_u(d);
    x_c(index+1) = x_u(d);
    index = index + 2;
end

% Computing the matrix that is used to relate {u} and {c} coordinates
dPsi = diff(Psi,x);
At = zeros(size(Kc));
for idx = 1:numel(x_c)
    if mod(idx,2)
        At(idx,:) = subs(Psi,x,x_c(idx));
    else
        At(idx,:) = subs(dPsi,x,x_c(idx));
    end
end

% Recovering {c}
Phi_c = T*Phi;

% Computing eigenfunctions
Psi_x = (Psi*Phi_c).';

% Plotting eigenfunctions
if strcmp(plt,'true') || strcmp(plt,'True') || strcmp(plt,'T')

    % Determining possible linestyles
    colors = {'k','r','b','g','c','m'};
    styles = {'-','--','.','o','-.'};
```

```
    lstyles = cell(1,N);
    index = 1;
    break_out = 'false';
    for s = 1:numel(styles)
        for c = 1:numel(colors)
            lstyles{index} = strcat(styles{s},colors{c});
            index = index + 1;
            if index == N
                break_out = 'true';
                break
            end
        end
        if strcmp(break_out,'true')
            break
        end
    end

    % Adding more colors if necessary
    while index < N && isempty(lstyles{index})
        idc = floor(rand*numel(colors))+1;
        ids = floor(rand*numel(styles))+1;
        lstyles{index} = strcat(styles{ids},colors{idc});
        index = index + 1;
    end

    % Converting from symbolic to numeric for plotting
    x_n = linspace(0,L,1000);
    figure('Position',[10 60 660 360]);
    hold on;
    ph = zeros(1,numel(Psi_x));
    legend_label = cell(1,numel(Psi_x));
    eigfcns = zeros(numel(Psi_x),1000);
    for n = 1:numel(Psi_x)
        f_n = subs(Psi_x(n),x,x_n);
        eigfcns(n,:) = f_n;
        ph(n) = plot(x_n, f_n, lstyles{n}, 'LineWidth',1.5);
        legend_label{n} = ['Mode ', num2str(n)];
    end
    xlabel('x')
    legend(ph,legend_label,'Location','EastOutside')
    set(gca,'box','on')

    % Saving plot
    if ~isempty(pnm)
        fprintf('Saving figure %d to %s\n',gcf,pnm)
        set(gcf,'PaperPositionMode','Auto')
        print(gcf,'-depsc',pnm)
    end

end

% Solving in other coordinates
AtI = inv(At);
[nr ~] = size(Kc);
```

```
keep = 1:nr;
keep(bcid) = [];
Ku = AtI.'*Kc*AtI;
Mu = AtI.'*Mc*AtI;
Ku = Ku(keep,keep);
Mu = Mu(keep,keep);
KuB = double(Ku);
MuB = double(Mu);
[~,Lu] = eig(KuB,MuB);
wu = sqrt(diag(Lu));
wc = sqrt(diag(Lambda));

% Comparing natural frequencies in different coordinates
fprintf(['\nComparison between natural frequencies for', ...
    ' different coordinates:\n\n'])
fprintf('%-12s %-12s %-12s\n','{u}','{c}','difference')
fprintf('------------------------------------------\n')
for n = 1:numel(wc)
    fprintf('%10E %10E %10E\n',wu(n),wc(n),abs(wc(n)-wu(n)))
end
fprintf('------------------------------------------\n\n')

% Outputing natural frequencies, mode shapes, M and K matrices
varargout{1} = sqrt(diag(Lambda));
varargout{2} = eigfcns;
varargout{3} = sqrt(diag(Lu));
varargout{4} = Ku;
varargout{5} = Mu;

end
```

```
%% Clearing and setting up

clc,clear,close all
[path, ~, ~] = fileparts(mfilename('fullpath'));
fprintf('\n%s running in %s ...\n\n',mfilename, path);
cd(path)
addpath(genpath(pwd))

% Making sure directory for data files exists
if ~exist('data','dir')
    fprintf('Creating new directory named data which contains output.\n\n')
    mkdir('data')
end

%% Inputs

L = 25;             % in
E = 1e7;            % psi
rho = 0.1/386.4;    % lb-s^2/in^4
A = 4;              % in^2
I = 16/12;          % in^4
N = 10;
```

```
n_modes = 3;       % number of modes to plot

%% C-F clamped free

% Calling beam_ritz
plt_nm = '../Images/hw3prob1_ex';
W_ritz = beam_ritz(L,E,rho,A,I,N,'C-F','monomial','true',plt_nm);

% Calling finite element code
ids = [1 2];
nn = 10;
ne = nn-1;
xnodes = linspace(0,L,N);
beamI = I*ones(1,ne);
beamA = A*ones(1,ne);
mode_plot = [1 2 3];
mode_animate = 0;
[W_fem, ~, ~, ~, ~, ~, ~, ~, ~, ~] = ...
    FEA_Beam_Vibration_2015A(xnodes,E,rho,beamI,beamA,ids,mode_plot, ...
    mode_animate);

% Writing output
write_wn('./data/cf.dat',W_ritz(1:3),W_fem(1:3));

%% C-C clamped clamped

% Calling beam_ritz
W_ritz = beam_ritz(L,E,rho,A,I,N,'C-C');

% Calling finite element code
ids = [1 2 2*nn-1 2*nn];
[W_fem, ~, ~, ~, ~, ~, ~, ~, ~, ~] = ...
    FEA_Beam_Vibration_2015A(xnodes,E,rho,beamI,beamA,ids,mode_plot, ...
    mode_animate);

% Writing output
write_wn('./data/cc.dat',W_ritz(1:3),W_fem(1:3));

%% S-S supported supported

% Calling beam_ritz
[W_ritz, Phi_ritz, fh] = beam_ritz(L,E,rho,A,I,N,'S-S');

% Calling finite element code
ids = [1 2*nn-1];
[W_fem, ~, ~, ~, ~, ~, ~, ~, ~, ~] = ...
    FEA_Beam_Vibration_2015A(xnodes,E,rho,beamI,beamA,ids,mode_plot, ...
    mode_animate);

% Writing output
write_wn('./data/ss.dat',W_ritz(1:3),W_fem(1:3));

%% C-S clamped supported

```

```
% Calling beam_ritz
W_ritz = beam_ritz(L,E,rho,A,I,N,'C-S');

% Calling finite element code
ids = [1 2 2*nn-1];
[W_fem, Phi_g, Phi, Kff, Mff, Le, Kgg, Mgg, IDF, OUT] = ...
    FEA_Beam_Vibration_2015A(xnodes,E,rho,beamI,beamA,ids,mode_plot, ...
    mode_animate);

% Writing output
write_wn('./data/cs.dat',W_ritz(1:3),W_fem(1:3));

%% Plotting

tol = 100;
imgpath = '../Images/';
fname = 'hw3prob1_mode';

for n = 1:n_modes

    % Normalizing
    [~,ii] = max(abs(Phi_g(:,n)));
    [pp,iy] = max(abs(Phi_ritz(n,:)));
    y_ritz = Phi_ritz(n,:);
    y_fem = sign(y_ritz(iy)*Phi_g(ii,n))*y_ritz*abs(Phi_g(ii,n))/ ...
        max(abs(y_ritz));
    [~,fmax] = max(y_fem);
    [~,rmax] = max(y_ritz);
    if abs(fmax-rmax)>tol
        y_fem = -y_fem;
    end

    % Plotting
    figure
    hold on
    ph1=plot(linspace(0,L,numel(y_ritz)),y_ritz,'LineWidth',1.5);
    ph2=plot(linspace(0,L,numel(y_ritz)),y_fem,'--r','LineWidth',1.5);
    xlabel('x')
    legend([ph1,ph2],'Ritz Method','FEM')
    set(gca,'box','on')
    set(gcf,'PaperPositionMode','Auto')
    fprintf('Saving image to %s\n',[imgpath,fname,num2str(n)]);
    print(gcf,'-depsc',[imgpath,fname,num2str(n)])
end

fprintf('\nDone.\n\n')
fprintf('Data files are in ./data\n\n')
```

```
%% Clearing and setting up

clc,clear,close all
[path, ~, ~] = fileparts(mfilename('fullpath'));
fprintf('\n%s running in %s ...\n\n',mfilename, path);
```

```
cd(path)
addpath(genpath(pwd))
addpath(genpath('/home/James/Desktop/School/Courses/UTA/AE_6311_Advanced_Structural_Dynamics'))

% Making sure directory for data files exists
if ~exist('data','dir')
    fprintf('Creating new directory named data which contains output.\n\n')
    mkdir('data')
end

%% Inputs

L = 25;             % in
E = 1e7;            % psi
rho = 0.1/386.4;    % lb-s^2/in^4s
A = 4;              % in^2
I = 16/12;          % in^4
N = 10;
n_modes = 3;        % number of modes to plot

%% C-F clamped free

% Calling beam_ritz
plt_nm = '../Images/hw3prob2_ex';
W_ritz = beam_ritz(L,E,rho,A,I,N,'C-F','polycosine','true',plt_nm);

% Calling finite element code
ids = [1 2];
nn = 10;
ne = nn-1;
xnodes = linspace(0,L,N);
beamI = I*ones(1,ne);
beamA = A*ones(1,ne);
mode_plot = [1 2 3];
mode_animate = 0;
[W_fem, ~, ~, ~, ~, ~, ~, ~, ~, ~] = ...
    FEA_Beam_Vibration_2015A(xnodes,E,rho,beamI,beamA,ids,mode_plot, ...
    mode_animate);

% Writing output
write_wn('./data/cf2.dat',W_ritz(1:3),W_fem(1:3));

%% C-C clamped clamped

% Calling beam_ritz
W_ritz = beam_ritz(L,E,rho,A,I,N,'C-C','polycosine');

% Calling finite element code
ids = [1 2 2*nn-1 2*nn];
[W_fem, ~, ~, ~, ~, ~, ~, ~, ~, ~] = ...
    FEA_Beam_Vibration_2015A(xnodes,E,rho,beamI,beamA,ids,mode_plot, ...
    mode_animate);

% Writing output
```

```
write_wn('./data/cc2.dat',W_ritz(1:3),W_fem(1:3));

%% S-S supported supported

% Calling beam_ritz
[W_ritz, Phi_ritz, fh] = beam_ritz(L,E,rho,A,I,N,'S-S','polycosine');

% Calling finite element code
ids = [1 2*nn-1];
[W_fem, ~, ~, ~, ~, ~, ~, ~, ~, ~] = ...
    FEA_Beam_Vibration_2015A(xnodes,E,rho,beamI,beamA,ids,mode_plot, ...
    mode_animate);

% Writing output
write_wn('./data/ss2.dat',W_ritz(1:3),W_fem(1:3));

%% C-S clamped supported

% Calling beam_ritz
W_ritz = beam_ritz(L,E,rho,A,I,N,'C-S','polycosine');

% Calling finite element code
ids = [1 2 2*nn-1];
[W_fem, Phi_g, Phi, Kff, Mff, Le, Kgg, Mgg, IDF, OUT] = ...
    FEA_Beam_Vibration_2015A(xnodes,E,rho,beamI,beamA,ids,mode_plot, ...
    mode_animate);

% Writing output
write_wn('./data/cs2.dat',W_ritz(1:3),W_fem(1:3));

%% Plotting

tol = 100;
imgpath = '../Images/';
fname = 'hw3prob2_mode';

for n = 1:n_modes

    % Normalizing
    [~,ii] = max(abs(Phi_g(:,n)));
    [pp,iy] = max(abs(Phi_ritz(n,:)));
    y_ritz = Phi_ritz(n,:);
    y_fem = sign(y_ritz(iy)*Phi_g(ii,n))*y_ritz*abs(Phi_g(ii,n))/ ...
        max(abs(y_ritz));
    [~,fmax] = max(y_fem);
    [~,rmax] = max(y_ritz);
    if abs(fmax-rmax)>tol
        y_fem = -y_fem;
    end

    % Plotting
    figure
    hold on
    ph1=plot(linspace(0,L,numel(y_ritz)),y_ritz,'LineWidth',1.5);
```

```
    ph2=plot(linspace(0,L,numel(y_ritz)),y_fem,'--r','LineWidth',1.5);
    xlabel('x')
    legend([ph1,ph2],'Ritz Method','FEM')
    set(gca,'box','on')
    set(gcf,'PaperPositionMode','Auto')
    fprintf('Saving image to %s\n',[imgpath,fname,num2str(n)]);
    print(gcf,'-depsc',[imgpath,fname,num2str(n)])
end

fprintf('\nDone.\n\n')
fprintf('Data files are in ./data\n\n')
```

```
%% Clearing workspace

clc,clear,close all

%% Inputs

L = 60;            % in
E = 10e7;          % psi
rho = 0.1/386.5;   % lb-s^2/in^4
w = 50;            % lbs
g = 32.2;          % ft/s^2
I = 16/12;         % in^4
m_t = w/g;         % slugs
Vmax = 1000;       % in^3

% Global variables for histories
global Xhist;
global Fhist;

%% Performing optimization

% Inequality constraint and initial guess
Aeq = [L/2 L/2];
beq = Vmax;
X0 = [10 10];
lb = [0.01 0.01];
ub = 1e7*ones(1,2);

% Setting options
opts = optimset('Display','iter','MaxIter',500,'MaxFunEvals',1000,...
    'Algorithm','active-set');

% Running optimization
[Xopt,wopt] = fmincon(@(x) ritz_obj_func(x,m_t,L,E,rho,I,10, ...
    'C-F'),X0, [], [],Aeq,beq,lb,ub,[],opts);

%% Plotting contours of the objective function with history and constraint

t1 = linspace(0.1,110,20);
t2 = linspace(0.1,110,20);
[T1 T2] = meshgrid(t1,t2);
```

```
OBJ = zeros(numel(t1),numel(t2));
for ii = 1:numel(t2)
    for jj = 1:numel(t1)
        OBJ(ii,jj) = -ritz_obj_func([t1(jj) t2(ii)],m_t,L,E,rho,I,10, ...
            'C-F');
    end
end

% Creating figure
figure('Position',[400 280 720 480])
hold on

% Plotting
Xhisr = real(Xhist);
contourf(T1,T2,OBJ);
confunc = 2*Vmax/L-t1;
ph4=plot(t1,confunc,'--k','LineWidth',1.5);
ph2=plot(Xhisr(:,1),Xhisr(:,2),'-om');
ph1=plot(X0(1),X0(2),'ok','MarkerFaceColor','g','MarkerSize',7);
ph3=plot(Xopt(1),Xopt(2),'ob','MarkerFaceColor','b','MarkerSize',7);
colorbar

% Improving aesthetics
legend([ph1,ph2,ph3,ph4],'Starting point','History','Optimal','Constraint')
set(gca,'XLim',[0 110],'YLim',[0 110])
xlabel('$X_1$ [in]','Interpreter','LaTeX')
ylabel('$X_2$ [in]','Interpreter','LaTeX')

% Saving plot
imgpath = ['/home/James/Desktop/School/Courses/UTA/', ...
    'AE_6311_Advanced_Structural_Dynamics/Images/'];
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc',[imgpath,'hw3prob3'])

% Plotting beam
figure
axis('equal')
xpts = [0 0 L L 0];
ypts = [-Xopt(1)/2 Xopt(1)/2 Xopt(2)/2 -Xopt(2)/2 -Xopt(1)/2];
patch(xpts,ypts,10,'LineWidth',1.5)
xlabel('x [in]')
ylabel('y [in]')
set(gca,'Box','on')
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc',[imgpath,'hw3prob3_beam'])
```