AE 6311: Advanced Structural Dynamics
Homework 4

James Grisham

May 1, 2015

Problem 1

Problem Statement

For the 3-dof system with $\mathbf{K}$, $\mathbf{M}$, and $\mathbf{F}$ shown below, find the forcing frequency for $|U_2| = |U_3|$.

Solution

The method used to solve this problem is described in problem 2. The answer is

$$\boxed{\Omega = 5.47723 \text{ rad/s}} \tag{1}$$

Problem 2

Problem Statement

Develop a general method to solve the following problem: find the forcing frequency so that $|U_r(\Omega)| = a|U_s(\Omega)|$, where r and s are any two dof in the system, and a is a non-zero positive constant. The system is undamped with mass matrix $\mathbf{M}$, stiffness matrix $\mathbf{K}$ and sinusoidal load amplitude vector $\mathbf{F}$. Validate your solution by applying it to problem 1 and a 3-node beam model with load applied at the tip.

Solution

This type of problem can be solved using transfer functions. In general the equations of motion for a harmonically excited MDOF system can be written as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F}e^{i\Omega t} \tag{2}$$

Assuming a solution of the form

$$\mathbf{u}(t) = \mathbf{U}e^{i\Omega t} \tag{3}$$

and inserting it into (2),

$$\left(-\Omega^2\mathbf{M} + i\Omega\mathbf{C} + \mathbf{K}\right)\mathbf{U} = \mathbf{F} \tag{4}$$

or

$$\mathbf{Z}\mathbf{U} = \mathbf{F} \tag{5}$$

where $\mathbf{Z}$ is the impedance matrix. Now, the amplitudes of the steady-state responses can be solved for by

$$\mathbf{U} = \mathbf{Z}^{-1}\mathbf{F} \equiv \mathbf{H}\mathbf{F} \tag{6}$$

where $\mathbf{H}$ can be thought of as a matrix of transfer functions. Or,

$$U_i = \sum_{j=1}^{N} H_{ij} F_j \tag{7}$$

For one value,

$$|U_1| = |H_{11}F_1 + H_{12}F_2 + \ldots + H_{1N}F_N| \tag{8}$$

However, only one value of F is nonzero for the cases given. Letting the one nonzero value be m,

$$|H_{rm}(\Omega)| = a|H_{sm}(\Omega)| \tag{9}$$

Now, the value of Ω can be determined by solving the polynomial in Ω given in (9). This approach is general and should work for any method that yields stiffness, mass and damping matrices.

For the problem in part A,

$$H_{21} = \frac{100. - 10.\Omega^2}{-1.\Omega^6 + 70.\Omega^4 - 1025.\Omega^2 + 2500.} \tag{10a}$$

$$H_{31} = -\frac{100.}{1.\Omega^6 - 70.\Omega^4 + 1025.\Omega^2 - 2500.} \tag{10b}$$

Solving for Ω yields

$$\boxed{\Omega = 5.47723 \text{ rad/s}} \tag{11}$$

For the 3-node beam model, the stiffness and mass matrices were determined using Ritz method.

For a beam with the following inputs,

```
% Inputs from Hw3
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
```

The method yields

$$\boxed{\Omega = 407622.99 \text{ rad/s}} \tag{12}$$

The equation to be solve was formed symbolically, then converted to a numeric function handle using `matlabFunction`. I then tried to use `fzero`, but sometimes got negative frequencies. Then, I tried `fminbnd` and made sure that the solution was positive. A plot of the objective function seems to show multiple solutions. The solution found in this case is marked by the green dot.

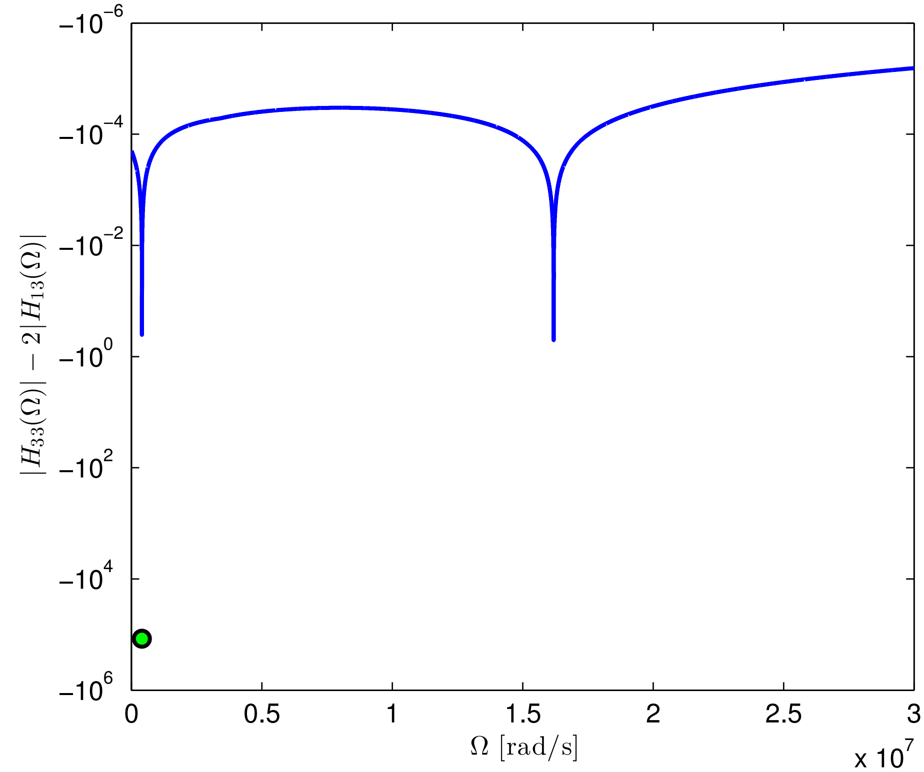


Figure 1: Plot of the objective function that was used to solve for Ω in the beam problem.

Problem 3

Problem Statement

Compute and plot the frequency responses in 3 forms (Bode, Nyquist and CO-QUAD) for the beam of problem 4-2. Assume the system is proportionally damped. The modal damping ratios for the first two modes are 0.01. The applied load is a unit force at the tip. Plot the tip and midspan responses.

Solution

For proportional damping,

$$\mathbf{C} = a_0\mathbf{M} + a_1\mathbf{K}$$

Using this relation along with the other typical expressions from mode superposition, the modal damping ratio for the r-th mode can be related to the coefficients a_0 and a_1.

$$\zeta_r = \frac{1}{2}\left(\frac{a_0}{\omega_r} + a_1\omega_r\right)$$

Solving the system of equations for the constants yields

$$a_0 = \frac{2\omega_1\omega_2(\zeta_2\omega_1 - \zeta_1\omega_2)}{\omega_1^2 - \omega_2^2} \tag{13a}$$

$$a_1 = \frac{2\zeta_1\omega_1 - 2\zeta_2\omega_2}{\omega_1^2 - \omega_2^2} \tag{13b}$$

Because the loading is not harmonic, the typical approach for determining the frequency response functions won't work. Because of this, I used the Fast Fourier Transorm, which is an implementation of the discrete Fourier transform (DFT). The DFT of an N point signal, $x[n]$ is

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

A MATLAB function was written for taking the FFT and plotting the results. The frequency responses of the tip and midspan translational responses are shown below.

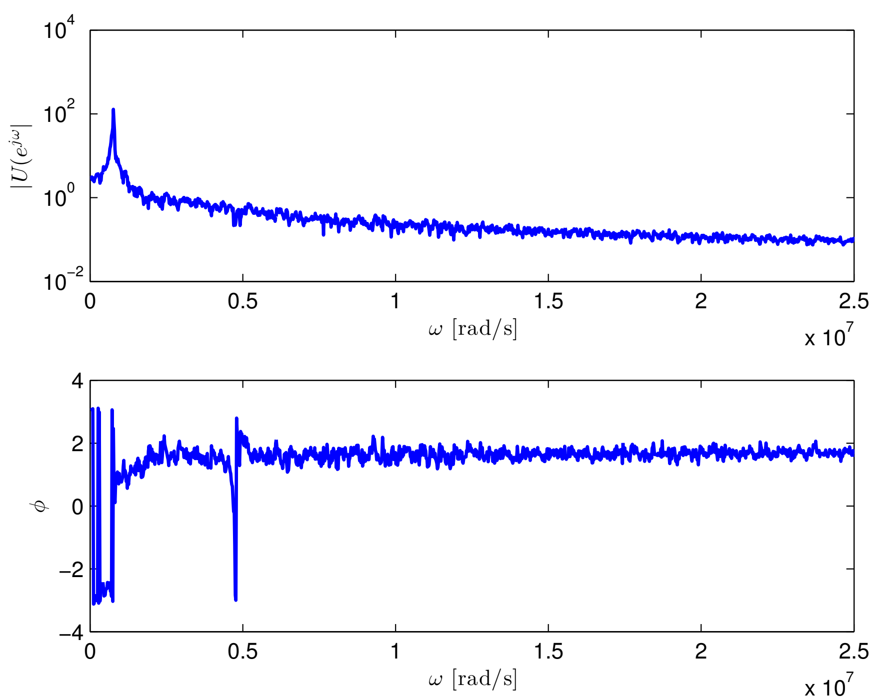


Figure 2: Bode plot for tip.

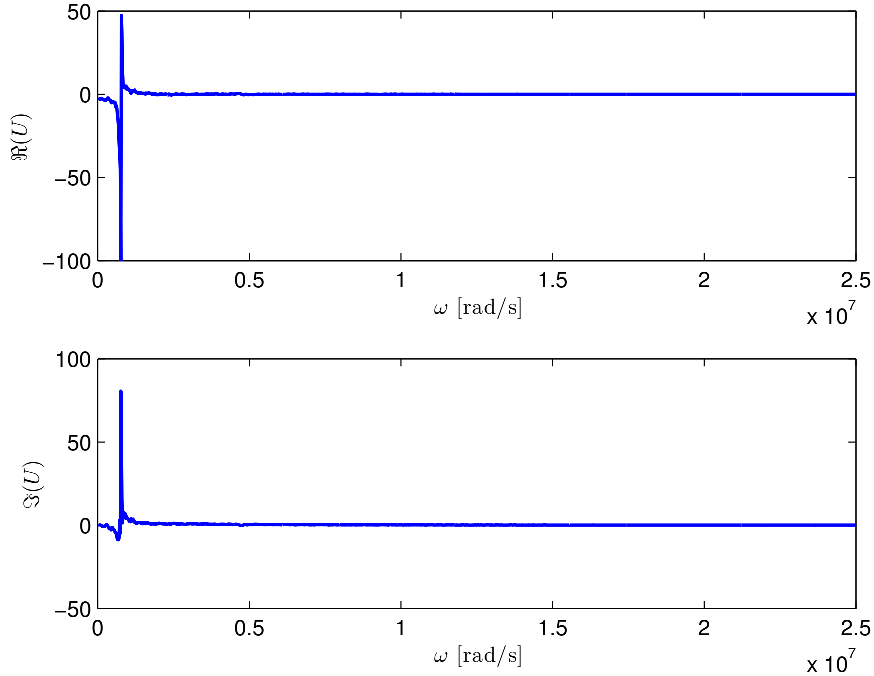


Figure 3: CO-QUAD plot for tip.

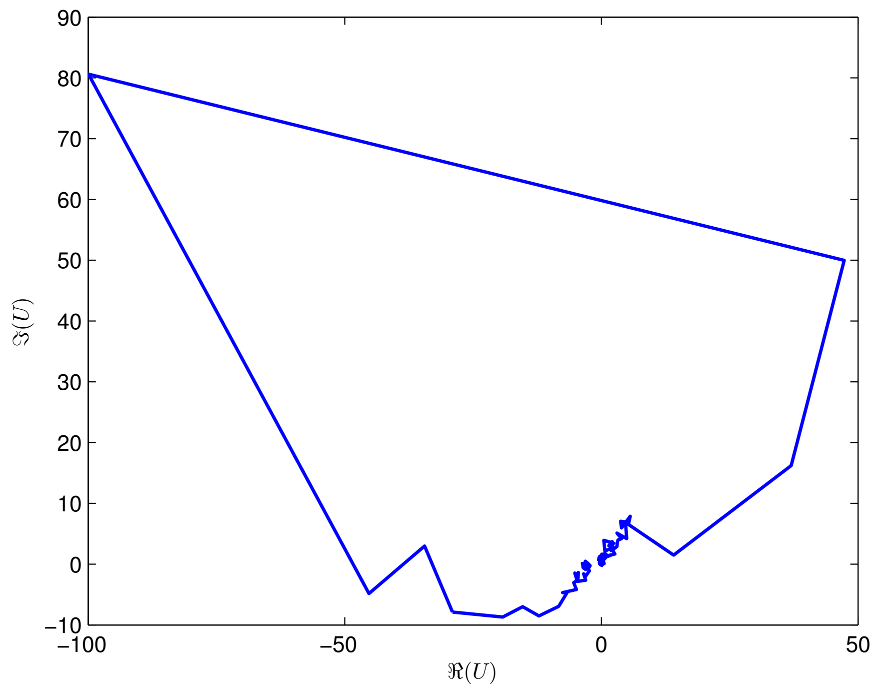


Figure 4: Nyquist plot for tip.

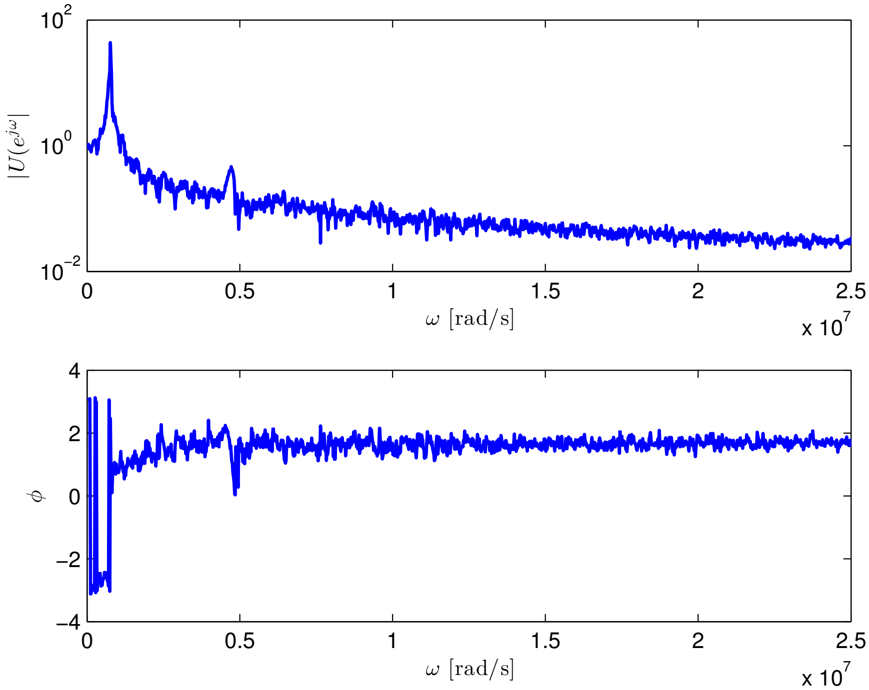


Figure 5: Bode plot for midspan.

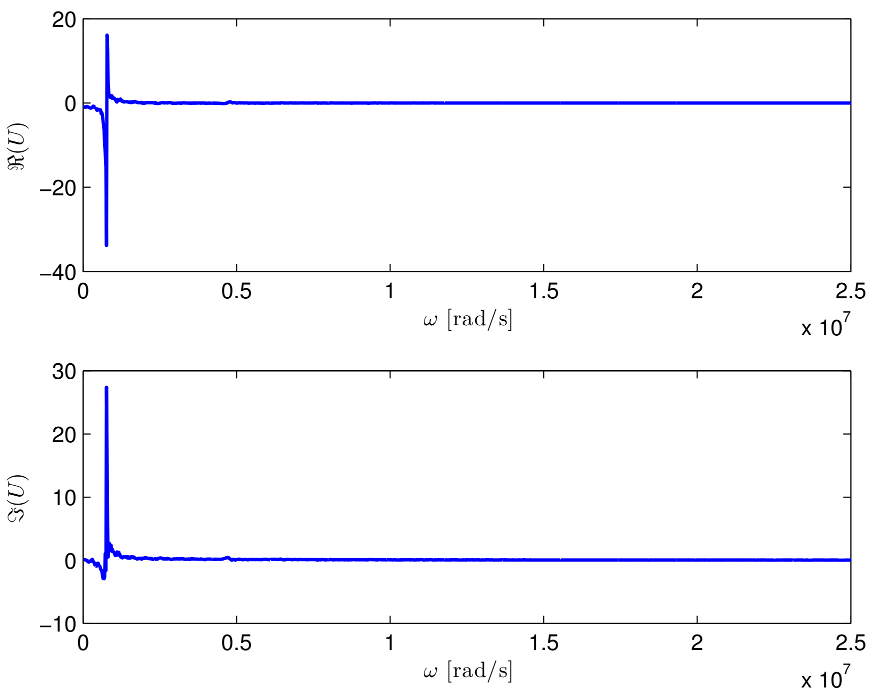


Figure 6: CO-QUAD plot for midspan.

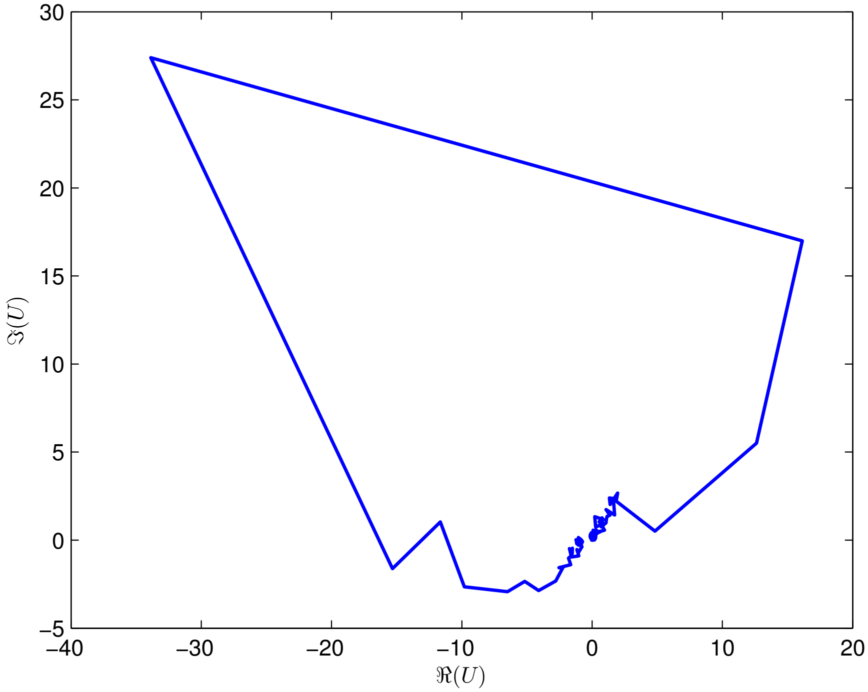


Figure 7: Nyquist plot for midspan.

Problem 4

Problem Statement

Find the transient response of the system subject to a unit step force applied at the tip. Assume zero initial conditions.

Solution

This problem can be solved using Laplace transforms. I believe the system is undamped. The stiffness and mass matrices were formed using the Ritz method. The inputs to the code were:

```
L = 10;            % in
E = 1e4;           % psi
rho = 0.1/386.4;   % lb-sˆ2/inˆ4
A = 4;             % inˆ2
I = 16/12;         % inˆ4
N = 6;
```

The system of ODEs can be written as

$$\sum_{j=1}^{N} M_{ij}\ddot{u}_j(t) + \sum_{j=1}^{N} K_{ij}u_j(t) = F_i(t) \tag{14}$$

Then, the index i is iterated over to form N differential equations. After doing so, the Laplace transform is applied to both sides. Assuming zero initial conditions,

$$\sum_{j=1}^{N} M_{ij} U_j(s) + \sum_{j=1}^{N} K_{ij} U_j(s) = \mathcal{L}[F_i] \tag{15}$$

The resulting algebraic system of equations is then solved for the U_j terms. Now, the inverse Laplace transform can be used to determine the responses as a function of time. This entire process was accomplished using Mathematica. The results are shown below.

Homework 4 Problem 4

In[156]:=
```
ClearAll["Global`*"]
```

Inputs from MATLAB Ritz method code

In[157]:=
```
K = {{1939.809524, -4335.238095, -1365.333333, -7314.285714},
   {-4335.238095, 12 647.619048, 3413.333333, 12 190.476190},
   {-1365.333333, 3413.333333, 2730.666667, 0.000000},
   {-7314.285714, 12 190.476190, 0.000000, 48 761.904762}};
```

In[158]:=
```
M = {{0.001563, -0.000851, 0.000657, 0.001195},
   {-0.000851, 0.000598, -0.000657, -0.000896}, {0.000657, -0.000657,
    0.004207, 0.000000}, {0.001195, -0.000896, 0.000000, 0.009560}};
```

In[159]:=
```
F = {1, 0, 0, 0};
```

Forming the ODEs

In[160]:=
```
uvpp = {u1''[t], u2''[t], u3''[t], u4''[t]};
```

In[161]:=
```
uv = {u1[t], u2[t], u3[t], u4[t]};
```

In[162]:=
```
eqs = Table[Sum[M[[i]][[j]] uvpp[[j]], {j, 1, 4}] +
    Sum[K[[i]][[j]] uv[[j]], {j, 1, 4}] == F[[i]], {i, 1, 4}]
```

Out[162]=
```
{1939.81 u1[t] - 4335.24 u2[t] - 1365.33 u3[t] - 7314.29 u4[t] +
   0.001563 u1″[t] - 0.000851 u2″[t] + 0.000657 u3″[t] + 0.001195 u4″[t] == 1,
 -4335.24 u1[t] + 12 647.6 u2[t] + 3413.33 u3[t] + 12 190.5 u4[t] -
   0.000851 u1″[t] + 0.000598 u2″[t] - 0.000657 u3″[t] - 0.000896 u4″[t] == 0,
 0. - 1365.33 u1[t] + 3413.33 u2[t] + 2730.67 u3[t] + 0.000657 u1″[t] -
   0.000657 u2″[t] + 0.004207 u3″[t] == 0, 0. - 7314.29 u1[t] + 12 190.5 u2[t] +
   48 761.9 u4[t] + 0.001195 u1″[t] - 0.000896 u2″[t] + 0.00956 u4″[t] == 0}
```

In[163]:=
```
Leqs = Table[LaplaceTransform[eqs[[i]], t, s], {i, 1, 4}]
```

Out[163]=
```
{1939.81 LaplaceTransform[u1[t], t, s] - 4335.24 LaplaceTransform[u2[t], t, s] -
   1365.33 LaplaceTransform[u3[t], t, s] - 7314.29 LaplaceTransform[u4[t], t, s] +
   0.001563 (s^2 LaplaceTransform[u1[t], t, s] - s u1[0] - u1′[0]) -
   0.000851 (s^2 LaplaceTransform[u2[t], t, s] - s u2[0] - u2′[0]) +
   0.000657 (s^2 LaplaceTransform[u3[t], t, s] - s u3[0] - u3′[0]) +
   0.001195 (s^2 LaplaceTransform[u4[t], t, s] - s u4[0] - u4′[0]) == 1/s,
 -4335.24 LaplaceTransform[u1[t], t, s] + 12 647.6 LaplaceTransform[u2[t], t, s] +
   3413.33 LaplaceTransform[u3[t], t, s] + 12 190.5 LaplaceTransform[u4[t], t, s] -
   0.000851 (s^2 LaplaceTransform[u1[t], t, s] - s u1[0] - u1′[0]) +
   0.000598 (s^2 LaplaceTransform[u2[t], t, s] - s u2[0] - u2′[0]) -
   0.000657 (s^2 LaplaceTransform[u3[t], t, s] - s u3[0] - u3′[0]) -
   0.000896 (s^2 LaplaceTransform[u4[t], t, s] - s u4[0] - u4′[0]) == 0,
 0. - 1365.33 LaplaceTransform[u1[t], t, s] + 3413.33 LaplaceTransform[u2[t], t, s] +
   2730.67 LaplaceTransform[u3[t], t, s] +
   0.000657 (s^2 LaplaceTransform[u1[t], t, s] - s u1[0] - u1′[0]) -
   0.000657 (s^2 LaplaceTransform[u2[t], t, s] - s u2[0] - u2′[0]) +
   0.004207 (s^2 LaplaceTransform[u3[t], t, s] - s u3[0] - u3′[0]) == 0,
 0. - 7314.29 LaplaceTransform[u1[t], t, s] + 12 190.5 LaplaceTransform[u2[t], t, s] +
   48 761.9 LaplaceTransform[u4[t], t, s] +
   0.001195 (s^2 LaplaceTransform[u1[t], t, s] - s u1[0] - u1′[0]) -
   0.000896 (s^2 LaplaceTransform[u2[t], t, s] - s u2[0] - u2′[0]) +
   0.00956 (s^2 LaplaceTransform[u4[t], t, s] - s u4[0] - u4′[0]) == 0}
```

In[164]:=
```
Ls = Solve[Leqs, Table[LaplaceTransform[uv[[i]], t, s], {i, 1, 4}]];
```

In[165]:=
```
ReplaceAll[Ls,
  Thread[Rule[{u1[0], u1'[0], u2[0], u2'[0], u3[0], u3'[0], u4[0], u4'[0]},
    {0, 0, 0, 0, 0, 0, 0, 0}]]];
```

In[166]:=
```
Lsf = Ls // Flatten;
```

In[167]:=
```
soln = Table[InverseLaplaceTransform[Lsf[[i]], s, t], {i, 1, 4}];
```

In[168]:=
```
soln =
  ReplaceAll[soln, Thread[Rule[{u1[0], u1'[0], u2[0], u2'[0], u3[0], u3'[0], u4[0],
       u4'[0]}, {0, 0, 0, 0, 0, 0, 0, 0}]]] // FullSimplify;
```

In[169]:=
```
resp1 = u1[t] /. soln[[1]]
```

Out[169]=

$$0.025 - 0.0121339\, e^{(0.-126.173\, i)\, t} - 0.0121339\, e^{(0.+126.173\, i)\, t} - 0.000313014\, e^{(0.-794.983\, i)\, t} - 0.000313014\, e^{(0.+794.983\, i)\, t} + 8.93183\times 10^{-18}\, e^{(0.+805.653\, i)\, t} + 8.93183\times 10^{-18}\, e^{(0.-805.653\, i)\, t} - 1.68649\times 10^{-16}\, e^{(0.-2258.45\, i)\, t} - 1.68649\times 10^{-16}\, e^{(0.+2258.45\, i)\, t} - 0.0000440956\, e^{(0.+2272.76\, i)\, t} - 0.0000440956\, e^{(0.-2272.76\, i)\, t} - 9.01835\times 10^{-6}\, e^{(0.-10\,056.5\, i)\, t} - 9.01835\times 10^{-6}\, e^{(0.+10\,056.5\, i)\, t}$$

In[170]:=
```
resp2 = u2[t] /. soln[[2]]
```

Out[170]=

$$0.00375 - 0.00167029\,e^{(0.-126.173\,i)\,t} - 0.00167029\,e^{(0.+126.173\,i)\,t} - 6.24539\times10^{-17}\,e^{(0.-553.858\,i)\,t} - 6.24539\times10^{-17}\,e^{(0.+553.858\,i)\,t} - 0.000149651\,e^{(0.+794.983\,i)\,t} - 0.000149651\,e^{(0.-794.983\,i)\,t} - 3.78529\times10^{-15}\,e^{(0.+805.653\,i)\,t} - 3.78529\times10^{-15}\,e^{(0.-805.653\,i)\,t} + 2.66053\times10^{-15}\,e^{(0.+1820.39\,i)\,t} + 2.66053\times10^{-15}\,e^{(0.-1820.39\,i)\,t} - 1.64967\times10^{-13}\,e^{(0.+2258.45\,i)\,t} - 1.64967\times10^{-13}\,e^{(0.-2258.45\,i)\,t} - 0.0000346096\,e^{(0.+2272.76\,i)\,t} - 0.0000346096\,e^{(0.-2272.76\,i)\,t} - 1.32853\times10^{-15}\,e^{(0.+6497.6\,i)\,t} - 1.32853\times10^{-15}\,e^{(0.-6497.6\,i)\,t} - 0.000020448\,e^{(0.+10\,056.5\,i)\,t} - 0.000020448\,e^{(0.-10\,056.5\,i)\,t}$$

In[171]:=
```
resp3 = u3[t] /. soln[[3]]
```

Out[171]=

$$0.0078125 - 0.00408604\,e^{(0.-126.173\,i)\,t} - 0.00408604\,e^{(0.+126.173\,i)\,t} + 0.000183506\,e^{(0.-553.858\,i)\,t} + 0.000183506\,e^{(0.+553.858\,i)\,t} + 0.311533\,e^{(0.-794.983\,i)\,t} + 0.311533\,e^{(0.+794.983\,i)\,t} - 0.293343\,e^{(0.-805.653\,i)\,t} - 4.72177\times10^{11}\,e^{(0.+805.653\,i)\,t} + 4.72177\times10^{11}\,e^{(0.+805.653\,i)\,t} + 0.000763663\,e^{(0.-1820.39\,i)\,t} + 0.000763663\,e^{(0.+1820.39\,i)\,t} - 0.172762\,e^{(0.-2258.45\,i)\,t} - 0.172762\,e^{(0.+2258.45\,i)\,t} + 0.162857\,e^{(0.-2272.76\,i)\,t} + 0.162857\,e^{(0.+2272.76\,i)\,t} - 0.000198068\,e^{(0.-6497.6\,i)\,t} - 0.000198068\,e^{(0.+6497.6\,i)\,t} + 0.000440639\,e^{(0.-10\,056.5\,i)\,t} + 0.000440639\,e^{(0.+10\,056.5\,i)\,t}$$

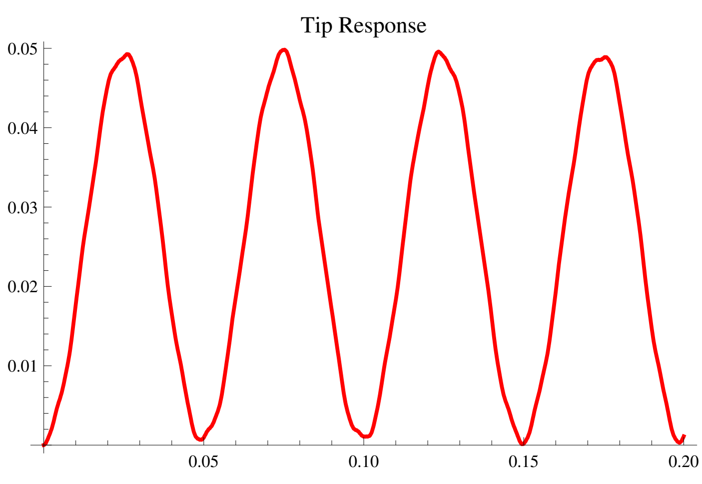
In[172]:=
```
resp4 = u4[t] /. soln[[4]]
```

Out[172]=

$$0.0028125 - 0.00142568\,e^{(0.-126.173\,i)\,t} - 0.00142568\,e^{(0.+126.173\,i)\,t} + 6.90236\times10^{-7}\,e^{(0.-553.858\,i)\,t} + 6.90236\times10^{-7}\,e^{(0.+553.858\,i)\,t} + 0.0269247\,e^{(0.-794.983\,i)\,t} + 0.0269247\,e^{(0.+794.983\,i)\,t} - 0.0276788\,e^{(0.-805.653\,i)\,t} - 0.0276788\,e^{(0.+805.653\,i)\,t} - 2.79182\times10^{-7}\,e^{(0.-1820.39\,i)\,t} - 2.79182\times10^{-7}\,e^{(0.+1820.39\,i)\,t} - 5.75616\times10^{6}\,e^{(0.-2258.45\,i)\,t} + 0.607464\,e^{(0.+2258.45\,i)\,t} + 5.75616\times10^{6}\,e^{(0.-2258.45\,i)\,t} - 0.616053\,e^{(0.+2272.76\,i)\,t} - 0.616053\,e^{(0.-2272.76\,i)\,t} + 1.5705\times10^{-6}\,e^{(0.-6497.6\,i)\,t} + 1.5705\times10^{-6}\,e^{(0.+6497.6\,i)\,t} + 0.0107646\,e^{(0.-10\,056.5\,i)\,t} + 0.0107646\,e^{(0.+10\,056.5\,i)\,t}$$

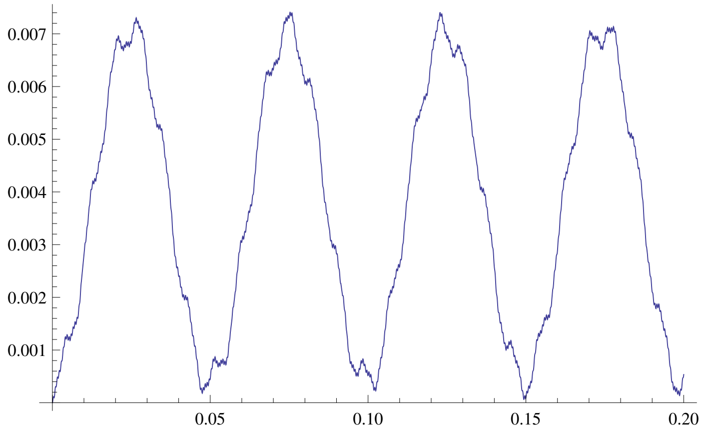
In[173]:=
```
Plot[Re[resp1], {t, 0, 0.2}, PlotStyle → {Thick, Red}, PlotLabel → "Tip Response"]
```

Out[173]=



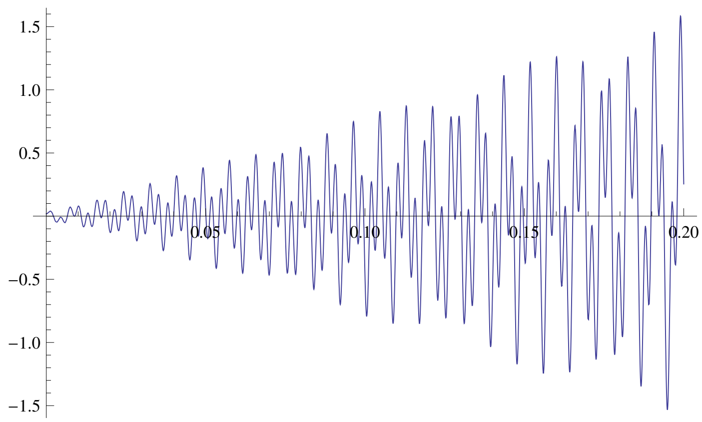
In[174]:=
```
Plot[Re[resp2], {t, 0, .2}]
```

Out[174]=



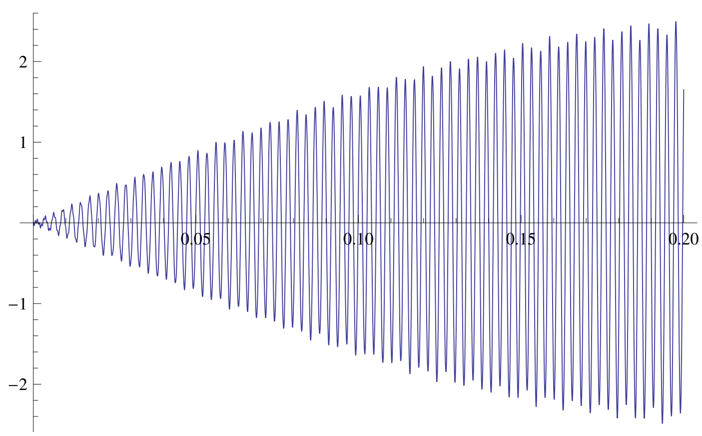
In[175]:=
```
Plot[Re[resp3], {t, 0, .2}]
```

Out[175]=



In[176]:=
```
Plot[Re[resp4], {t, 0, .2}]
```

Out[176]=



In[177]:=
```
A = K + M;
```

In[178]:=
```
Ainv = Inverse[A];
```

In[179]:=
```
Uj = Ainv.{LaplaceTransform[1, t, s], 0, 0, 0};
```

In[180]:=
```
Table[InverseLaplaceTransform[Uj[[i]], s, t], {i, 1, 4}]
```

Out[180]=
```
{0.0249985, 0.00374979, 0.00781198, 0.00281232}
```

Problem 5

Problem Statement

Solve problem 4-4 by mode superposition. Plot the tip responses of ode45 and this solution on the same graph. Also show mode truncation effect in the modal solution. Show how it is improved by mode acceleration solutions.

Solution

A MATLAB function was developed for solving for the response of MDOF systems by mode superposition. It is available in the appendix. The plot of the tip response is shown below:

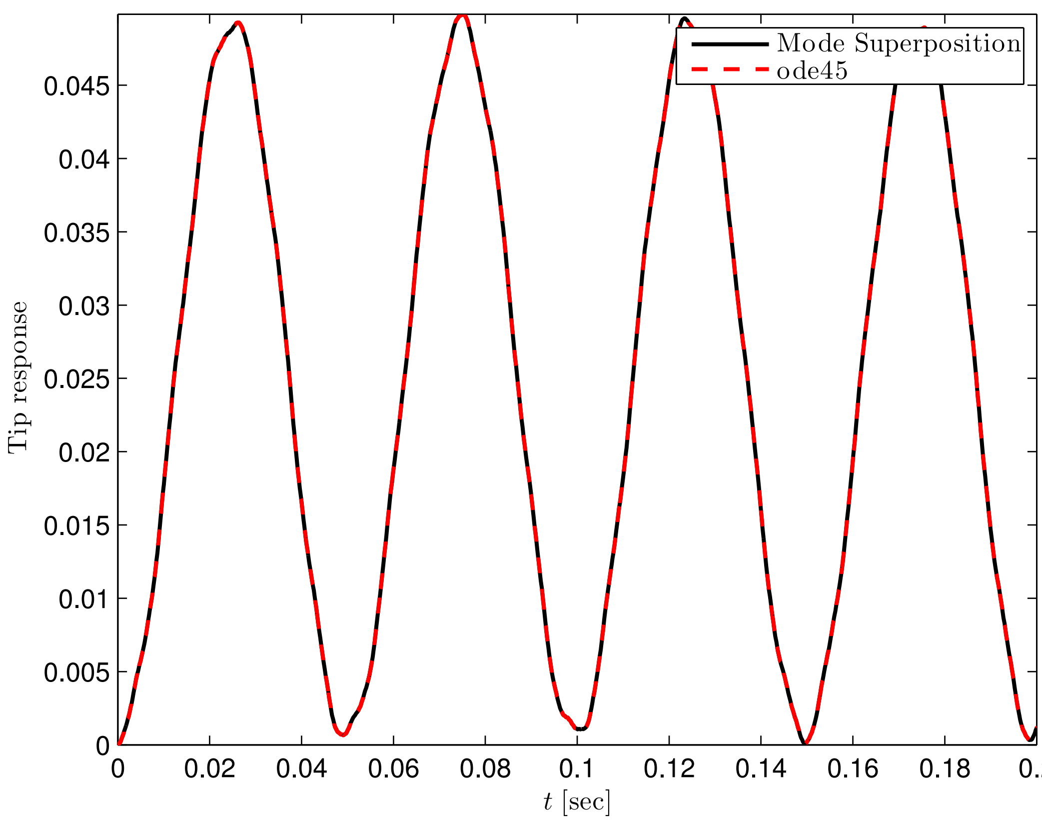


Figure 8: Tip response using mode superposition."

This result agrees well with the results obtained using the Laplace transform:

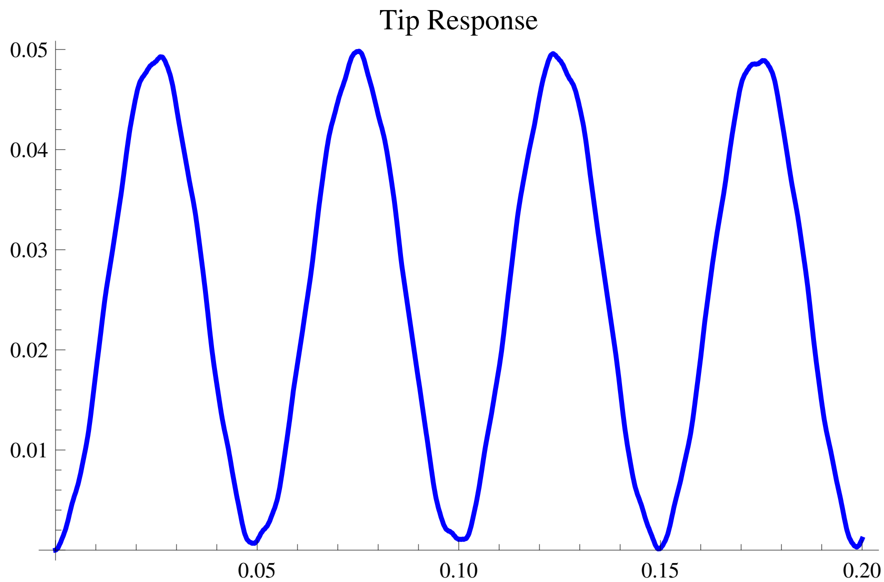


Figure 9: Tip response using the Laplace transform.

A separate MATLAB function was also developed for numerical solution of general n-dof systems. This was done using the following formulation:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F} \tag{16}$$

$$\frac{d\mathbf{u}}{dt} = \mathbf{v}$$
$$\frac{d^2\mathbf{u}}{dt} = \dot{\mathbf{v}}$$

Then, the system can be written as

$$\dot{\mathbf{v}} = \mathbf{A}\mathbf{v} + \mathbf{f} \tag{17}$$

where $\mathbf{A}$ is the system matrix given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \tag{18}$$

and $\mathbf{f}$ is given by

$$\mathbf{f} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F} \end{Bmatrix} \tag{19}$$

The mode truncation solution is shown below:

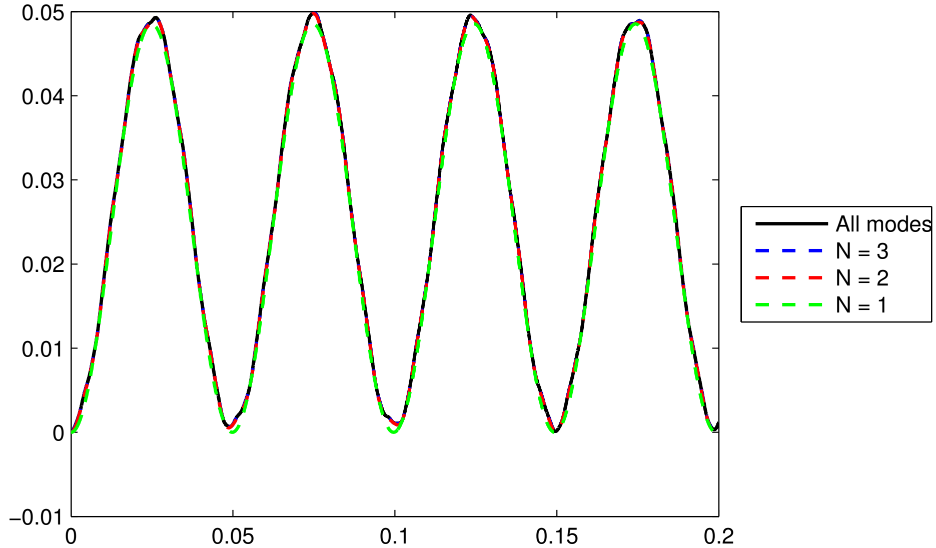


Figure 10: Differences between response for different number of modes truncated.

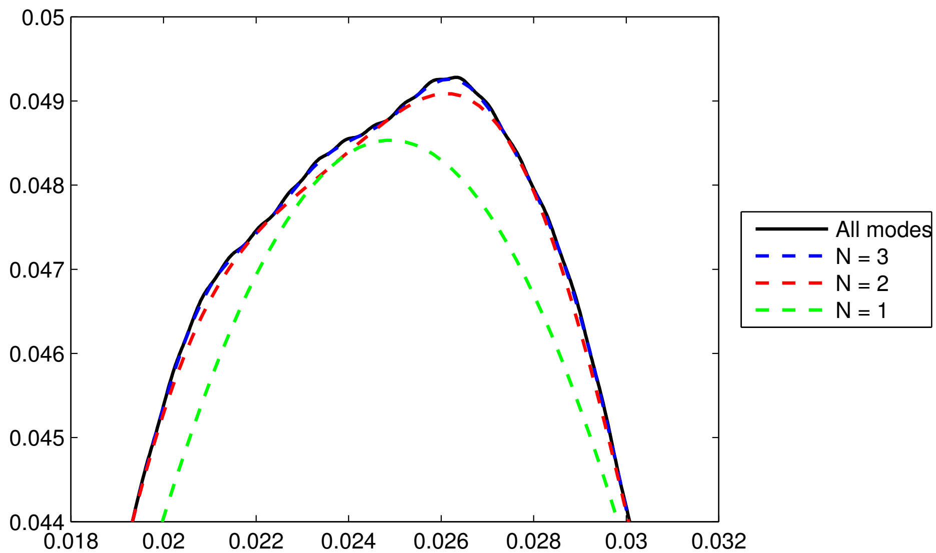


Figure 11: Differences between response for different number of modes truncated (close-up).

The mode acceleration solutions improve the approximation as shown in the below plots, especially the close-up comparison.

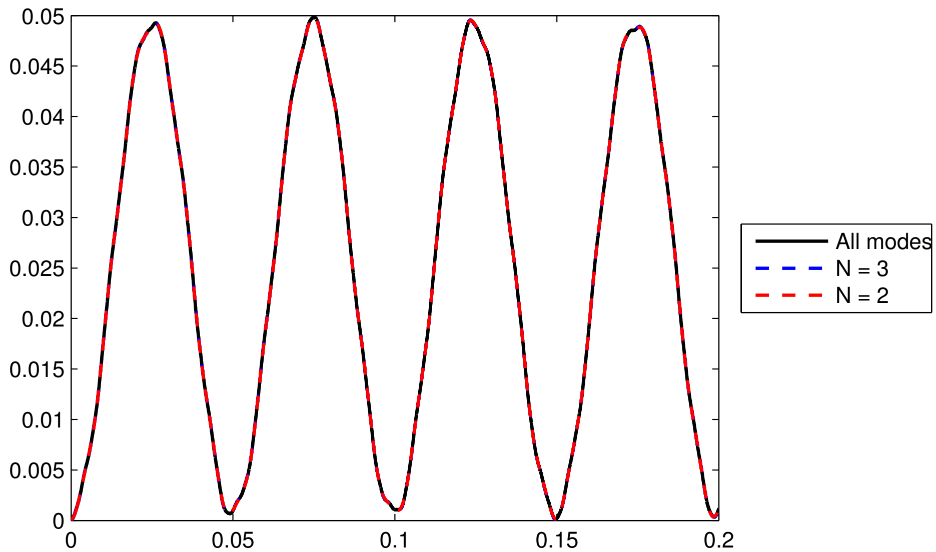


Figure 12: Mode acceleration solution.

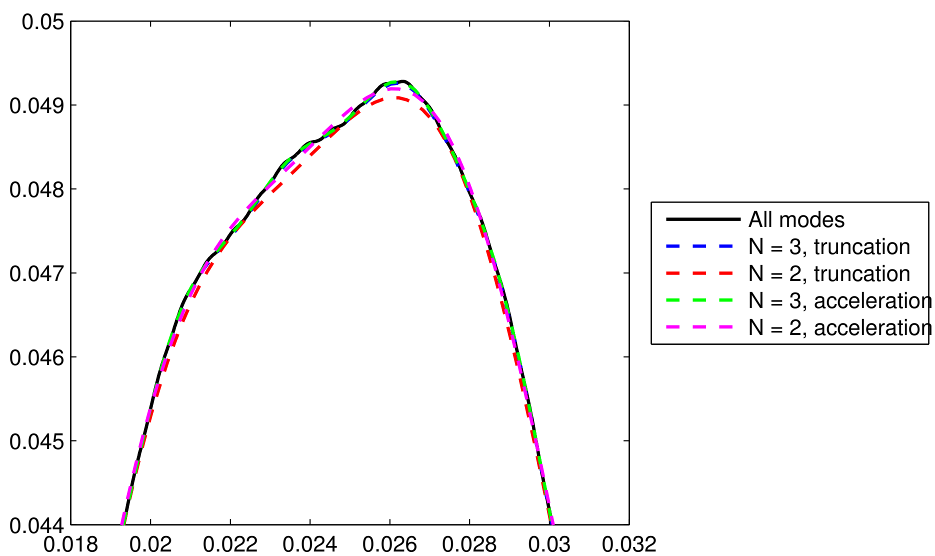


Figure 13: Comparison between mode truncation and mode acceleration methods.

Code listing

```
function [t,u] = mdof_response_num(K,M,C,F,t_final)

% This only works for step inputs, but could be generalized further
% pretty easily.

% Getting size of inputs
[n ~] = size(K);

```

```
% Initializing
A = zeros(2*n);
f = zeros(2*n,1);

% Forming A matrix
A(1:n,n+1:2*n) = eye(n);
Minv = inv(M);
A(n+1:2*n,1:n) = -Minv*K;
A(n+1:2*n,n+1:2*n) = -Minv*C;

% Forming f
f(n+1:2*n) = Minv*F;

% Setting zero initial conditions
Y0 = zeros(2*n,1);

% Solving ode
[t,u] = ode45(@odefunc,[0 t_final],Y0);

    function dy = odefunc(t,y)
        dy = zeros(n,1);
        dy = A*y + f;
    end

end
```

```
function [t_vec,u] = mdof_response_symb(K,M,Zeta,F,forcing_type,t_final,Omega)

% Inputs
% forcing_type = 'sine';
% forcing_type = 'step';

% Finding eigenvalues
[Phi,Lambda] = eig(K,M);

% Determining natural frequencies
omega = sqrt(diag(Lambda));

% Computing generalized matrices (proportionally damped)
GM = Phi.'*M*Phi;
GK = Phi.'*K*Phi;
GC = zeros(size(GM));
for idx = 1:numel(GM(:,1))
    GC(idx,idx) = 2*Zeta(idx)*omega(idx)*GM(idx);
end
GF = Phi.'*F;

% Solving differential equation symbolically
syms q(t) t
Dq = diff(q);
D2q = diff(q,2);
if strcmp(forcing_type,'sine')
    for idx = 1:numel(GM(:,1))
```

```
        modal_soln(idx,1) = vpa(dsolve(GM(idx,idx)*D2q + GC(idx,idx)*Dq + ...
            GK(idx,idx)*q == GF(idx)*sin(Omega*t), q(0)==0,Dq(0)==0),8);
    end
elseif strcmp(forcing_type,'step')
    for idx = 1:numel(GM(:,1))
        modal_soln(idx,1) = vpa(dsolve(GM(idx,idx)*D2q + GC(idx,idx)*Dq + ...
            GK(idx,idx)*q == GF(idx), q(0)==0,Dq(0)==0),8);
    end
end

% Converting back to physical coordinates
soln = vpa(Phi*modal_soln,8);
for s = 1:numel(soln)
    disp(vpa(simplify(soln(s)),4));
end

% Substituting time vector
t_vec = linspace(0,t_final,1000);
u = zeros(numel(soln),numel(t_vec));
for idx = 1:numel(soln)
    u(idx,:) = subs(soln(idx),t,t_vec);
end
% u = double(u);

end
```

```
function [f,U] = freq_resp_plots(t,u,img_path,rootname)

xmax = 2.5e7;

% Taking the FFT of the data
f_s = 1/(t(2) - t(1));
N = numel(t);
n = linspace(0,N-1,N);
f = n*f_s/N;
U = fft(u,N);
if mod(N/2,2) ~= 0
    f = f(2:(N+1)/2);
    U = U(2:(N+1)/2);
else
    f = f(2:N/2);
    U = U(2:N/2);
end
omega = 2*pi*f;

% BODE
figure
subplot(2,1,1)
plot(omega,abs(U),'LineWidth',1.5)
xlabel('$\omega$ [rad/s]','Interpreter','LaTeX')
ylabel('$|U(e^{j\omega}|$','Interpreter','LaTeX')
set(gca,'YScale','Log','XLim',[0 xmax])
subplot(2,1,2)
```

```
plot(omega,atan2(imag(U),real(U)),'LineWidth',1.5)
xlabel('$\omega$ [rad/s]','Interpreter','LaTeX')
ylabel('$\phi$','Interpreter','LaTeX')
set(gca,'XLim',[0 xmax])
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc',[img_path,'/',rootname,'_bode'])

% CO-QUAD
figure
subplot(2,1,1)
plot(omega,real(U),'LineWidth',1.5)
xlabel('$\omega$ [rad/s]','Interpreter','LaTeX')
ylabel('$\Re(U)$','Interpreter','LaTeX')
set(gca,'XLim',[0 xmax])
subplot(2,1,2)
plot(omega,imag(U),'LineWidth',1.5)
xlabel('$\omega$ [rad/s]','Interpreter','LaTeX')
ylabel('$\Im(U)$','Interpreter','LaTeX')
set(gca,'XLim',[0 xmax])
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc',[img_path,'/',rootname,'_coquad'])

% Nyquist
figure
plot(real(U),imag(U),'LineWidth',1.5);
xlabel('$\Re(U)$','Interpreter','LaTeX')
ylabel('$\Im(U)$','Interpreter','LaTeX')
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc',[img_path,'/',rootname,'_nyquist'])

end
```

```
cd %% Clearing workspace

clc,clear,close all

%% Calling test for part I

hw4prob2_test

%% Part II

clc,clear,close all

% Inputs from Hw3
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 = 6;
a = 2;

```

```
% Calling beam_ritz
plt_nm = '../Images/basis';
[~,~,~,Ku,Mu] = beam_ritz(L,E,rho,A,I,N,'C-F','monomial','true',plt_nm);
close all

% Forming the impedance matrix
syms Omega
Z = Ku - Omega*Mu;
H = inv(Z);

% Enforcing |U_5| = 2 |U_3|
m = 3; % the force is applied at node 3
symb = abs(H(3,m)) - a*abs(H(1,m));
num = matlabFunction(symb);
omega_soln = fminbnd(num,0.0,1e7);
fprintf('Omega = %.2f rad/s\n',omega_soln)

Mn = subs(Mu,Omega,omega_soln);
Kn = subs(Ku,Omega,omega_soln);

[p,l] = eig(Kn,Mn);

% Plotting
Omega_v = linspace(0,3e7,50000);
Obj = num(Omega_v);
hold on
plot(Omega_v,Obj,'LineWidth',1.5)
plot(omega_soln,num(omega_soln),'ok','LineWidth',1.5,'MarkerFaceColor','g')
set(gca,'YScale','Log')
xlabel('$\Omega$ [rad/s]','Interpreter','LaTeX')
ylabel('$|H_{33}(\Omega)| - 2 |H_{13}(\Omega)|$','Interpreter','LaTeX')
set(gca,'Box','on')
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc','../Images/hw4prob2.eps')
```

```
%% Clearing workspace

clc,clear,close all
addpath(genpath(pwd))

%% Inputs

L = 10;             % in
E = 1e4;            % psi
rho = 0.1/386.4;    % lb-s^2/in^4
A = 4;              % in^2
I = 16/12;          % in^4
N = 6;
a = 2;

% Modal damping
zeta1 = 0.01;
zeta2 = zeta1;
```

```

%% Calling beam_ritz

plt_nm = '../Images/basis';
[~,~,~,Ku,Mu] = beam_ritz(L,E,rho,A,I,N,'C-F','monomial','true',plt_nm);
close all

%% Eigenvalue problem

% Numerically evaluating
Kn = double(vpa(Ku,6));
Mn = double(vpa(Mu,6));
[Phi,Lambda] = eig(Kn,Mn);
omega = sqrt(diag(Lambda));

% Forming generalized stiffness and mass matrices
GM = Phi.'*Mn*Phi;
GK = Phi.'*Kn*Phi;

% Determining a0 and a1 values
a0 = 2*omega(1)*omega(2)*(zeta2*omega(1) - zeta1*omega(2))/ ...
    (omega(1)^2 - omega(2)^2);
a1 = (2*zeta1*omega(1) - 2*zeta2*omega(2))/(omega(1)^2 - omega(2)^2);

% Computing C matrix
C = a0*Mn + a1*Kn;

%% Using ODE45 to solve the system
[t,u] = mdof_response_num(Kn,Mn,C,[1;0;0;0],2);

%% Plotting responses

figure

% Tip response
subplot(2,1,1);
plot(t,u(:,1),'LineWidth',1.5)
xlabel('t [sec]','Interpreter','LaTeX')
ylabel('Tip response','Interpreter','LaTeX')

% Mid span response
subplot(2,1,2);
plot(t,u(:,3),'LineWidth',1.5);
xlabel('t [sec]','Interpreter','LaTeX')
ylabel('Midspan response','Interpreter','LaTeX')

% Saving figure
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc','../Images/hw4prob3_responses')

%% Computing and plotting frequency responses

[~,~] = freq_resp_plots(t,u(:,1),'../Images','tip');
```

```
[~,~] = freq_resp_plots(t,u(:,3),'../Images','midspan');
```

```
%% Clearing workspace

clc,clear,close all
addpath(genpath(pwd))

%% Inputs

L = 10;             % in
E = 1e4;            % psi
rho = 0.1/386.4;    % lb-s^2/in^4
A = 4;              % in^2
I = 16/12;          % in^4
N = 6;
a = 2;

%% Calling beam_ritz

plt_nm = '../Images/basis';
[~,~,~,Ku,Mu] = beam_ritz(L,E,rho,A,I,N,'C-F','monomial','true',plt_nm);
close all

K = double(vpa(Ku,6));
M = double(vpa(Mu,6));

%% Calling function that solves problem using mode superposition

[t,u] = mdof_response_symb(K,M,[0;0;0;0],[1;0;0;0],'step',0.2,0);

% Plotting results for the tip response
hold on
plot(t,u(1,:),'-k','LineWidth',1.5)


%% Calling function that solves the problem using ODE45

[t_n,u_n] = mdof_response_num(K,M,zeros(size(K)),[1;0;0;0],0.2);

plot(t_n,u_n(:,1),'--r','LineWidth',1.5)

xlabel('$t$ [sec]','Interpreter','LaTeX')
ylabel('Tip response','Interpreter','LaTeX')
lh = legend('Mode Superposition','ode45');
set(lh,'Interpreter','LaTeX')
set(gca,'Box','on','YLim',[0,max(u_n(:,1))])
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc','../Images/hw4prob5_modesuperposition.eps')

%% Solving the problem using mode truncation

[~,u3] = mode_displacement(K,M,[0;0;0;0],[1;0;0;0],'step',0.2,3,0);
[~,u2] = mode_displacement(K,M,[0;0;0;0],[1;0;0;0],'step',0.2,2,0);
```

```
[t,u1] = mode_displacement(K,M,[0;0;0;0],[1;0;0;0],'step',0.2,1,0);

figure('Position',[30 440 630 350])
hold on
plot(t_n,u_n(:,1),'-k','LineWidth',1.5)
plot(t,u3(1,:),'--b','LineWidth',1.5)
plot(t,u2(1,:),'--r','LineWidth',1.5)
plot(t,u1(1,:),'--g','LineWidth',1.5)
set(gca,'Box','On')
lh = legend('All modes','N = 3','N = 2','N = 1');
set(lh,'Location','EastOutside')
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc','../Images/hw4prob5_mode_trunc')

figure('Position',[50 135 630 350])
hold on
plot(t_n,u_n(:,1),'-k','LineWidth',1.5)
plot(t,u3(1,:),'--b','LineWidth',1.5)
plot(t,u2(1,:),'--r','LineWidth',1.5)
plot(t,u1(1,:),'--g','LineWidth',1.5)
set(gca,'XLim',[0.018 0.032],'YLim',[0.0440 0.05],'Box','On')
lh = legend('All modes','N = 3','N = 2','N = 1');
set(lh,'Location','EastOutside')
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc','../Images/hw4prob5_mode_trunc_close')

%% Solving the problem using mode acceleration

[~,u3a] = mode_acceleration(K,M,[0;0;0;0],[1;0;0;0],'step',0.2,3,0);
[t,u2a] = mode_acceleration(K,M,[0;0;0;0],[1;0;0;0],'step',0.2,2,0);
% [t,u1] = mode_displacement(K,M,[0;0;0;0],[1;0;0;0],'step',0.2,1,0);

figure('Position',[30 440 630 350])
hold on
plot(t_n,u_n(:,1),'-k','LineWidth',1.5)
plot(t,u3a(1,:),'--b','LineWidth',1.5)
plot(t,u2a(1,:),'--r','LineWidth',1.5)
set(gca,'Box','On')
lh = legend('All modes','N = 3','N = 2');
set(lh,'Location','EastOutside')
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc','../Images/hw4prob5_mode_acc')

% Comparing
figure('Position',[30 440 630 350])
hold on
plot(t_n,u_n(:,1),'-k','LineWidth',1.5)
plot(t,u3(1,:),'--b','LineWidth',1.5)
plot(t,u2(1,:),'--r','LineWidth',1.5)
plot(t,u3a(1,:),'--g','LineWidth',1.5)
plot(t,u2a(1,:),'--m','LineWidth',1.5)
set(gca,'XLim',[0.018 0.032],'YLim',[0.0440 0.05],'Box','On')
lh = legend('All modes','N = 3, truncation','N = 2, truncation', ...
    'N = 3, acceleration', 'N = 2, acceleration');
```

```
set(lh,'Location','EastOutside')
set(gcf,'PaperPositionMode','Auto')
print(gcf,'-depsc','../Images/hw4prob5_comparison')
```