Appendix B: MATLAB Script Files

The following script files have been used to generate the solutions and plots for the examples shown in the text.

```
%      VIBS2 brings up a menu of the available exercises designed
%        for the Vibrations II course at the University of
%        Cincinnati.
%
%**********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%*********************************************************************
%

echo off
clear
while 1
      vibs2= ['v2_001'
                'v2_002'
                'v2_009'
                'v2_011'
                'v2_015'
                'v2_016'
                'v2_017'
                'v2_020'
                'v2_052'
                'v2_055'
                'v2_056'
                'v2_057'
                'v2_090'
                'v2_095'
                'v2_099'];
      clc
      help v2list
      n = input('Select a VIBS2 program number: ');
      if ((n <= 0) | (n > 17))
                clear
            break
      end
      vibs2 = vibs2(n,:);
      eval(vibs2)
      clear
end
clc
```

```
%
%        Vibrations 2 Scripts
%
%        1)  v2_001   - SDOF System
%        2)  v2_002   - SDOF System, Under-Critical-Over Damped
%        3)  v2_009   - SDOF System, Transfer Function (3-D)
%        4)  v2_011   - SDOF System
%        5)  v2_015   - Changing Stiffness, SDOF System
%        6)  v2_016   - Changing Damping, SDOF System
%        7)  v2_017   - Changing Mass, SDOF System
%        8)  v2_020   - SDOF System
%        9)  v2_052   - SDOF System
%        10) v2_055   - 2DOF System-Proportional Damping
%        11) v2_056   - 2DOF System-Non-Proportional Damping
%        12) v2_057   - 2DOF System-Different Plot Formats
%        13) v2_090   - MDOF(3) System
%        14) v2_095   - MDOF(2-3) Addition of MKC Subsystem
%        15) v2_099   - MDOF(3) System, FRF by Inversion
%
%        0)  EXIT
%
%
```

```
% v2_001.m
%
% This is a script file to solve a sdof system
% given the mass, damping and stiffness terms
% in dimensionless units.  The output includes
% poles, residues (modal coefficients) and both
% time and frequency domain plots of the impulse
% frequency response functions.

%*********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%********************************************************************
%
clg, clear
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
mass=input('Mass value: (10)');if isempty(mass), mass=10;end;
stiff=input('Stiffness value: (16000)');if isempty(stiff),stiff=16000;end;
damp=input('Damping value: (10)'); if isempty(damp),damp=10;end;
f=linspace(0,150,500);
H=1.0./((-mass.*f.*f+stiff)+j.*damp.*f);
plot(f,real(H))
pause
plot(f,imag(H))
pause
clear f,H;
pause
a=[mass,damp,stiff]
b=[0,0,1]
[r,p,k]=residue(b,a)
residue = r(1)
lambda=p(1)
whos
pause(2)
t=linspace(0,5,500);
xt=residue./2*exp(lambda.*t) + residue'./2*exp(lambda'.*t);
plot(t,xt)
xlabel('Time (Sec.)'),ylabel('Amplitude'),grid
if plt==1,print -deps v2_001a.eps, end;
pause
clear t,xt;
f=linspace(0,150,500);
xf=residue./(j.*f-lambda) + residue'./(j.*f-lambda');
plot(f,abs(xf))
xlabel('Frequency (Hz)'),ylabel('Magnitude'),grid
if plt==1,print -deps v2_001b.eps, end;
pause
semilogy(f,abs(xf))
xlabel('Frequency (Hz)'),ylabel('Log Magnitude'),grid
```

```
if plt==1,print -deps v2_001c.eps, end;
pause
plot(f,360./(2.*pi).*angle(xf))
xlabel('Frequency (Hz)'),ylabel('Phase (Deg)'),grid
if plt==1,print -deps v2_001d.eps, end;
pause
plot(f,360./(2.*pi).*angle(xf))
xlabel('Frequency (Hz)'),ylabel('Phase (Deg)'),grid
if plt==1,print -deps v2_001e.eps, end;
pause
plot(f,real(xf))
xlabel('Frequency (Hz)'),ylabel('Real'),grid
if plt==1,print -deps v2_001f.eps, end;
pause
plot(f,imag(xf))
xlabel('Frequency (Hz)'),ylabel('Imaginary'),grid
if plt==1,print -deps v2_001g.eps, end;
```

```
% v2_002.m
%
% This is a script file to illustrate the response
% of an underdamped, critically damped and overdamped
% SDOF system to an initial displacement.
%
% SDOF System
% Underdamped, critically damped overdamped response
% Figures for UC-SDRL-CN-20-263-662, Chapter 2

%**********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%*********************************************************************
%
clear,clg
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end
mag=10;
V=[-15,15,-15,15];
%
%     Overdamped Solution
%
lambda1=-2;
lambda2=-8;
t=linspace(0,4.0,201);
x=mag.*exp(lambda1.*t)+mag.*exp(lambda2.*t);
plot(t,x)
xlabel('Time(sec.)'),ylabel('Displacement'),grid
title('Overdamped SDOF System')
if plt==1,print -deps v2_002a.eps, end;
pause
clg
axis(V);
plot(real(lambda1),imag(lambda1),'*',real(lambda2),imag(lambda2),'*')
axis;
xlabel('Real Part of Complex Frequency')
ylabel('Imaginary Part of Complex Frequency'),grid
title('Overdamped SDOF System')
if plt==1,print -deps v2_002b.eps, end;
pause
clear x
lambda1=-5;
lambda2=-5;
x=mag.*exp(lambda1.*t)+mag.*exp(lambda2.*t);
plot(t,x)
xlabel('Time(sec.)'),ylabel('Magnitude'),grid
title('Critically Damped SDOF System')
if plt==1,print -deps v2_002c.eps, end;
pause
```

```
clg
axis(V);
plot(real(lambda1),imag(lambda1),'*',real(lambda2),imag(lambda2),'*')
axis;
xlabel('Real Part of Complex Frequency')
ylabel('Imaginary Part of Complex Frequency'),grid
title('Critically Damped SDOF System')
if plt==1,print -deps v2_002d.eps, end;
pause
clg
clear x
lambda1=-5+j*10;
lambda2=-5-j*10;
x=mag.*exp(lambda1.*t)+mag.*exp(lambda2.*t);
plot(t,x)
xlabel('Time(sec.)'),ylabel('Magnitude'),grid
title('Underdamped SDOF System')
if plt==1,print -deps v2_002e.eps, end;
pause
clg
axis(V);
plot(real(lambda1),imag(lambda1),'*',real(lambda2),imag(lambda2),'*')
axis;
xlabel('Real Part of Complex Frequency')
ylabel('Imaginary Part of Complex Frequency'),grid
title('Underdamped SDOF System')
if plt==1,print -deps v2_002f.eps, end;
pause
```

```
% v2_009.m
%
% This is a script file to solve a sdof system
% given the mass, damping and stiffness terms
% in dimensionless units.  The output is a three
% dimensional plot in the s domain (complex independent
% variable.
%
% SDOF System, Laplace Domain (3D) plot
% Figures for UC-SDRL-CN-20-263-662, Chapter 2

%**********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%*********************************************************************
%
clear,clg
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end
mass=10;
a=input('Real Part of Pole: (-0.0625)'); if isempty(a),a=-0.0625;end
b=input('Imaginary Part of Pole: (0.51)'); if isempty(b),b=0.51;end
rot=input('Rotation View Angle: (60)'); if isempty(rot),rot=60;end
lambda(1)=a+j*b;
lambda(2)=a-j*b;
%
%        Set up mesh for only quadrants two and three
%
[sigma,omega]=meshdom(-0.2:0.005:0,-1:0.02:1);
s=sigma+j*omega;
H=(1.0./mass).*(1.0./((s-lambda(1)).*(s-lambda(2))));
view=[rot,30];
mesh(real(H),view)
%     title('Transfer Function (SDOF):  Real Part')
if plt==1,print -f1 -deps v2_009a,end;
pause
mesh(imag(H),view)
%     title('Transfer Function (SDOF):  Imaginary Part')
if plt==1,print -f1 -deps v2_009b,end;
pause
mesh(abs(H),view)
%     title('Transfer Function (SDOF):  Magnitude')
if plt==1,print -f1 -deps v2_009c,end;
pause
mesh(angle(H),view)
%     title('Transfer Function (SDOF):  Phase')
if plt==1,print -f1 -deps v2_009d,end;
pause
mesh(log(abs(H)),view)
%     title('Transfer Function (SDOF):  Log Magnitude')
```

```
if plt==1,print -f1 -deps v2_009e,end;
```

```
% v2_011.m
%
% This is a script file to solve a sdof system
% given the mass, damping and stiffness terms
% in dimensionless units.  The output includes
% poles, residues (modal coefficients) and both
% time and frequency domain plots of the impulse
% frequency response functions.
%
% SDOF System
% Figures for UC-SDRL-CN-20-263-662, Chapter 2

%*********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%********************************************************************
%
clear,clg
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
mass=input('Mass value:  (10)'); if isempty(mass),mass=10;end
stiff=input('Stiffness value:  (4)'); if isempty(stiff),stiff=4;end
damp=input('Damping value:  (2)'); if isempty(damp),damp=2;end
mass,stiff,damp
a=[mass,damp,stiff];
b=[0,0,1];
[r,p,k]=residue(b,a);
residu=r(1);
lambda=p(1);
t=linspace(0,60,500);
xt=residu./2*exp(lambda.*t) + residu'./2*exp(lambda'.*t);
axis([0,60,-0.1,0.1])
plot(t,xt)
xlabel('Time (Sec.)'),ylabel ('Amplitude'),grid
if plt==1,print -f1 -deps v2_011a,end;
pause
clg
clear t,xt;
f=linspace(0,3,500);
xf=residu./(j.*f-lambda) + residu'./(j.*f-lambda');
axis([0,3,-1.0,1.0])
plot(f,real(xf))
xlabel('Frequency (Rad/Sec)'),ylabel('Real'),grid
if plt==1,print -f1 -deps v2_011b,end;
pause
clg
plot(f,imag(xf))
xlabel('Frequency (Rad/Sec)'),ylabel('Imaginary'),grid
if plt==1,print -f1 -deps v2_011c,end;
axis([1,2,3,4]);axis;
```

```
% v2_015.m
%
% This is a script file to solve a sdof system
% given the mass, damping and stiffness terms
% in dimensionless units when the stiffness is varied.
% The output includes
% poles, residues (modal coefficients) and
% frequency domain plots of the
% frequency response functions.
%
% SDOF System, Change of Stiffness
% Figures for UC-SDRL-CN-20-263-662, Chapter 2

%*********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%********************************************************************
%
clg,clear
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
pi=3.14159;
mass=input('Mass value: (10)');if isempty(mass),mass=10;end
istiff=input('Initial Stiffness value: (1000)');if isempty(istiff),istiff=1000;end
damp=input('Damping value: (10)');if isempty(damp),damp=10;end
f=linspace(0,50,300);
hold off
for i=1:10
stiff=istiff + (i-1).*200
a=[mass,damp,stiff];
b=[0,0,1];
[r,p,k]=residue(b,a);
residu= r(1);
lambda=p(1);
xf=residu./(j.*f-lambda) + residu'./(j.*f-lambda');
semilogy(f,abs(xf))
hold on
end
xlabel('Frequency (Rad/Sec)'),ylabel('Log Magnitude'),grid
if plt==1,print -f1 -deps v2_015a,end;
pause
hold off
clg;
for i=1:10
stiff=istiff + (i-1).*200
a=[mass,damp,stiff];
b=[0,0,1];
[r,p,k]=residue(b,a);
residu= r(1);
lambda=p(1);
```

```
xf=residu./(j.*f-lambda) + residu’./(j.*f-lambda’);
scale=360.0/(2.0*pi);
plot(f,scale.*angle(xf))
hold on
end
xlabel(’Frequency (Rad/Sec)’),ylabel(’Phase (Deg)’),grid
if plt==1,print -f1 -deps v2_015b,end;
hold off
```

```
% v2_016.m
%
% This is a script file to solve a sdof system
% given the mass, damping and stiffness terms
% in dimensionless units when the damping is varied.
% The output includes
% poles, residues (modal coefficients) and
% frequency domain plots of the
% frequency response functions.
%
% SDOF System, Change of Damping
% Figures for UC-SDRL-CN-20-263-662, Chapter 2

%**********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%*********************************************************************
%
clg,clear
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
pi=3.14159;
mass=input('Mass value: (10)');if isempty(mass),mass=10;end
stiff=input('Stiffness value: (1600)');if isempty(stiff),stiff=1600;end
idamp=input('Initial Damping value: (2)');if isempty(idamp),idamp=2;end
f=linspace(0,50,300);
hold off;
for i=1:10
damp=idamp + (i-1).*2
a=[mass,damp,stiff];
b=[0,0,1];
[r,p,k]=residue(b,a);
residu= r(1);
lambda=p(1);
xf=residu./(j.*f-lambda) + residu'./(j.*f-lambda');
semilogy(f,abs(xf))
hold on;
end
xlabel('Frequency (Rad/Sec)'),ylabel('Log Magnitude'),grid
title('Effect of Changing Damping')
if plt==1,print -f1 -deps v2_016a,end;
pause
hold off;
clg;
for i=1:10
damp=idamp + (i-1).*2
a=[mass,damp,stiff];
b=[0,0,1];
[r,p,k]=residue(b,a);
residu= r(1);
```

```
lambda=p(1);
xf=residu./(j.*f-lambda) + residu’./(j.*f-lambda’);
scale=360.0/(2.0*pi);
plot(f,scale.*angle(xf))
hold on;
end
xlabel(’Frequency (Rad/Sec)’),ylabel(’Phase (Deg)’),grid
if plt==1,print -f1 -deps v2_016b,end;
hold off;
```

```
% v2_017.m
%
% This is a script file to solve a sdof system
% given the mass, damping and stiffness terms
% in dimensionless units when the mass is varied.
% The output includes
% poles, residues (modal coefficients) and
% frequency domain plots of the
% frequency response functions.
%
% SDOF System, Change of Mass
% Figures for UC-SDRL-CN-20-263-662, Chapter 2

%*********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%********************************************************************
%
clg,clear
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
pi=3.14159;
imass=input('Initial Mass value: (2)');if isempty(imass),imass=2;end
stiff=input('Stiffness value: (1600)');if isempty(stiff),stiff=1600;end
damp=input('Damping value: (10)');if isempty(damp),damp=10;end
f=linspace(0,50,300);
hold off
for i=1:10
mass=2 + (i-1).*1
a=[mass,damp,stiff];
b=[0,0,1];
[r,p,k]=residue(b,a);
residu= r(1);
lambda=p(1);
xf=residu./(j.*f-lambda) + residu'./(j.*f-lambda');
semilogy(f,abs(xf))
hold on
end
xlabel('Frequency (Rad/Sec)'),ylabel('Log Magnitude'),grid
if plt==1,print -f1 -deps v2_017a,end;
pause
hold off
for i=1:10
mass=2 + (i-1).*1
a=[mass,damp,stiff];
b=[0,0,1];
[r,p,k]=residue(b,a);
residu= r(1);
lambda=p(1);
xf=residu./(j.*f-lambda) + residu'./(j.*f-lambda');
```

```
scale=360.0/(2.0*pi);
plot(f,scale.*angle(xf))
hold on
end
xlabel('Frequency (Rad/Sec)'),ylabel('Log Magnitude'),grid
if plt==1,print -f1 -deps v2_017b,end;
hold off
```

```
% v2_020.m
%
% This is a script file to solve a sdof system
% given the mass, damping and stiffness terms
% in dimensionless units.  The output is a three
% dimensional plot in the s domain (complex independent
% variable.
%
% SDOF System, Laplace Domain (3D) plot
% Figures for UC-SDRL-CN-20-263-662, Chapter 2

%*********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%********************************************************************
%
clear,clg
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end
mass=10;
a=input('Real Part of Pole: (-0.0625)'); if isempty(a),a=-0.0625;end
b=input('Imaginary Part of Pole: (0.51)'); if isempty(b),b=0.51;end
rot=input('Rotation View Angle: (60)'); if isempty(rot),rot=60;end
lambda(1)=a+j*b;
lambda(2)=a-j*b;
%
%        Set up mesh for only quadrants two and three
%
[sigma,omega]=meshdom(-0.2:0.005:0,-1:0.02:1);
s=sigma+j*omega;
H=(1.0./mass).*(1.0./((s-lambda(1)).*(s-lambda(2))));
view=[rot,30];
mesh(real(H),view)
%     title('Transfer Function (SDOF):  Real Part')
if plt==1,print -f1 -deps v2_020a,end;
pause
mesh(imag(H),view)
%     title('Transfer Function (SDOF):  Imaginary Part')
if plt==1,print -f1 -deps v2_020b,end;
pause
mesh(abs(H),view)
%     title('Transfer Function (SDOF):  Magnitude')
if plt==1,print -f1 -deps v2_020c,end;
pause
mesh(angle(H),view)
%     title('Transfer Function (SDOF):  Phase')
if plt==1,print -f1 -deps v2_020d,end;
pause
mesh(log(abs(H)),view)
%     title('Transfer Function (SDOF):  Log Magnitude')
```

```
if plt==1,print -f1 -deps v2_020e,end;
```

```
%  v2_052.m
%
% This is a script file to solve a sdof system
% given the mass, damping and stiffness terms
% in dimensionless units.  The output includes
% poles, residues (modal coefficients) and both
% time and frequency domain plots of the impulse
% frequency response functions.
%
% SDOF Damped System
% Figures for UC-SDRL-CN-20-263-662, Chapter 2

%*********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%********************************************************************
%
clear,clg
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
mass=input('Mass value: (10)'); if isempty(mass), mass=10;end;
stiff=input('Stiffness value: (4)'); if isempty(stiff),stiff=4;end;
damp=input('Damping value: (2)'); if isempty(damp),damp=2;end;
a=[mass,damp,stiff];
b=[0,0,1];
[r,p,k]=residue(b,a);
residu = r(1)
lambda=p(1)
t=linspace(0,60,500);
xt=residu./2*exp(lambda.*t) + residu'./2*exp(lambda'.*t);
axis([0,60,-0.1,0.1])
plot(t,xt)
xlabel('Time (Sec.)'),ylabel ('Amplitude'),grid
if plt==1,print -f1 -deps v2_052a,end;
pause
clg; clear t; clear xt;
f=linspace(0,3,500);
xf=residu./(j.*f-lambda) + residu'./(j.*f-lambda');
axis([0,3,-1.0,1.0])
plot(f,real(xf))
xlabel('Frequency (Rad/Sec)'),ylabel('Real'),grid
if plt==1,print -f1 -deps v2_052b,end;
pause
clg
plot(f,imag(xf))
xlabel('Frequency (Rad/Sec)'),ylabel('Imaginary'),grid
if plt==1,print -f1 -deps v2_052c,end;
axis([1,2,3,4]);axis;
```

```
% v2_055.m
%
% This is a script file to solve a 2 DOF system
% given the mass, damping and stiffness matrices
% in dimensionless units.  The output includes poles,
% residues(modal coefficients) and time and frequency
% domain plots of impulse and frequency response
% functions.
%
% 2 DOF Proportionally Damped System
% Figures for UC-SDRL-CN-20-263-662, Chapter 5

%*********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%********************************************************************
%
clear,clg
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
pi=3.14159265;
alpha=0.0
beta=0.5
mass=[5,0;0,10];
stiff=[4,-2;-2,6];
damp=alpha*mass + beta*stiff;
null=[0,0;0,0];
%    Form 2N x 2N state space equation.
a=[null,mass;mass,damp];
b=[-mass,null;null,stiff];
[x,d]=eig(-inv(a)*b);
% Sort Modal Frequencies
orig_lambda=diag(d);
[Y,I]=sort(imag(orig_lambda));
lambda=orig_lambda(I);
xx=x(:,I);
%     Normalize x matrix
for ii=1:4
xx(1:4,ii)=xx(1:4,ii)./xx(3,ii);
end
%     Extract modal vectors from state-space formulation
psi(1:2,1)=xx(3:4,1);
psi(1:2,2)=xx(3:4,2);
%      Calculate modal mass matrix
mm=psi.'*mass*psi;
%      Calculate modal scaling value (Q)
Q(1)=1./(2*j*imag(lambda(1))*mm(1,1));
Q(2)=1./(2*j*imag(lambda(2))*mm(2,2));
%      Calculate residue matrices
A1=Q(1).*psi(1:2,1)*psi(1:2,1).';
```

```
A2=Q(2).*psi(1:2,2)*psi(1:2,2).';
%      Formulate H(1,2) FRF as Default
resp=input('Enter response (row) DOF: (1)');if isempty(resp),resp=1;end
inp=input('Enter input (column) DOF: (2)');if isempty(inp),inp=2;end
residu(1) = A1(resp,inp);
residu(2) = A2(resp,inp);
A1,A2
pause
magA1=abs(A1);
phaseA1=angle(A1).*360.0./(2.0*pi);
magA2=abs(A2);
phaseA2=angle(A2).*360.0./(2.0*pi);
magA1,phaseA1
magA2,phaseA2
pause
lambda,residu
pause
%      Calculate desired plots
t=linspace(0,30,500);
xt1=residu(1).*exp(lambda(1).*t) + residu(1)'.*exp(lambda(1)'.*t);
xt2=residu(2).*exp(lambda(2).*t) + residu(2)'.*exp(lambda(2)'.*t);
xt=xt1+xt2;
axis([0,30,-0.10,0.10])
plot(t,xt)
xlabel('Time (Sec.)'),ylabel('Amplitude'),grid
if plt==1,print -f1 -deps v2_055c,end;
pause
clg
plot(t,xt1)
xlabel('Time (Sec.)'),ylabel('Amplitude'),grid
if plt==1,print -f1 -deps v2_055f,end;
pause
clg
plot(t,xt2)
xlabel('Time (Sec.)'),ylabel('Amplitude'),grid
if plt==1,print -f1 -deps v2_055i,end;
pause
clg
clear t; clear xt; clear xt1; clear xt2;
f=linspace(0,2.5,500);
xf1=residu(1)./(j.*f-lambda(1)) + residu(1)'./(j.*f-lambda(1)');
xf2=residu(2)./(j.*f-lambda(2)) + residu(2)'./(j.*f-lambda(2)');
xf=xf1+xf2;
axis([0,2.5,-0.6,0.6])
plot(f,real(xf))
xlabel('Frequency (Rad/Sec)'),ylabel('Real'),grid
if plt==1,print -f1 -deps v2_055a,end;
pause
clg
plot(f,imag(xf))
xlabel('Frequency (Rad/Sec)'),ylabel('Imaginary'),grid
if plt==1,print -f1 -deps v2_055b,end;
pause
clg
plot(f,real(xf1))
```

```
xlabel('Frequency (Rad/Sec)'),ylabel('Real'),grid
if plt==1,print -f1 -deps v2_055d,end;
pause
clg
plot(f,imag(xf1))
xlabel('Frequency (Rad/Sec)'),ylabel('Imaginary'),grid
if plt==1,print -f1 -deps v2_055e,end;
pause
clg
axis([0,2.5,-0.2,0.2])
plot(f,real(xf2))
xlabel('Frequency (Rad/Sec)'),ylabel('Real'),grid
if plt==1,print -f1 -deps v2_055g,end;
pause
clg
plot(f,imag(xf2))
xlabel('Frequency (Rad/Sec)'),ylabel('Imaginary'),grid
if plt==1,print -f1 -deps v2_055h,end;
axis([1 2 3 4]);axis;
```

```
% v2_056.m
%
% This is a script file to solve a 2 DOF system
% given the mass, damping and stiffness matrices
% in dimensionless units.  The output includes poles,
% residues(modal coefficients) and time and frequency
% domain plots of impulse and frequency response
% functions.
%
% 2 DOF Non-Proportionally Damped System
% Figures for UC-SDRL-CN-20-263-662, Chapter 5

%**********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%*********************************************************************
%
clear,clg
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
pi=3.14159265;
mass=[5,0;0,10];
stiff=[4,-2;-2,6];
damp=[6,-4;-4,5];
null=[0,0;0,0];
%    Form 2N x 2N state space equation.
a=[null,mass;mass,damp];
b=[-mass,null;null,stiff];
[x,d]=eig(-inv(a)*b);
% Sort Modal Frequencies
orig_lambda=diag(d);
[Y,I]=sort(imag(orig_lambda));
lambda=orig_lambda(I);
xx=x(:,I);
xxx=input('Hit any key to continue');
%     Compute 'modal a' and 'modal b' matrix
ma=xx.'*a*xx;
mb=xx.'*b*xx;
%     Extract modal vectors from state-space formulation
psi(1:2,1)=xx(3:4,1);
psi(1:2,2)=xx(3:4,2);
psi(1:2,3)=xx(3:4,3);
psi(1:2,4)=xx(3:4,4);
%      Calculate residue matrices
A1=psi(1:2,1)*psi(1:2,1).'./ma(1,1);
A2=psi(1:2,2)*psi(1:2,2).'./ma(2,2);
A3=psi(1:2,3)*psi(1:2,3).'./ma(3,3);
A4=psi(1:2,4)*psi(1:2,4).'./ma(4,4);
%      Formulate H(1,2) FRF as Default
resp=input('Enter response (row) DOF: (1)');if isempty(resp),resp=1;end;
```

```
inp=input('Enter input (column) DOF: (2)');if isempty(inp),inp=2;end;
residu(1) = A1(resp,inp);
residu(2) = A2(resp,inp);
residu(3) = A3(resp,inp);
residu(4) = A4(resp,inp);
A1,A2,A3,A4
xxx=input('Hit any key to continue');
lambda,residu
xxx=input('Hit any key to continue');
%      Calculate desired plots
t=linspace(0,30,500);
xt1=residu(1).*exp(lambda(1).*t) + residu(2).*exp(lambda(2).*t);
xt2=residu(3).*exp(lambda(3).*t) + residu(4).*exp(lambda(4).*t);
xt=xt1+xt2;
axis([0,30,-0.10,0.10])
plot(t,xt)
xlabel('Time (Sec.)'),ylabel('Amplitude'),grid
if plt==1,print -f1 -deps v2_056c,end;
pause
clg
plot(t,xt1)
xlabel('Time (Sec.)'),ylabel('Amplitude'),grid
if plt==1,print -f1 -deps v2_056i,end;
pause
clg
plot(t,xt2)
xlabel('Time (Sec.)'),ylabel('Amplitude'),grid
if plt==1,print -f1 -deps v2_056f,end;
pause
clg
clear t; clear xt; clear xt1; clear xt2;
f=linspace(0,2.5,500);
xf1=residu(1)./(j.*f-lambda(1)) + residu(2)./(j.*f-lambda(2));
xf2=residu(3)./(j.*f-lambda(3)) + residu(4)./(j.*f-lambda(4));
xf=xf1+xf2;
axis([0,2.5,-0.6,0.6])
plot(f,real(xf))
xlabel('Frequency (Rad/Sec)'),ylabel('Real'),grid
if plt==1,print -f1 -deps v2_056a,end;
pause
clg
plot(f,imag(xf))
xlabel('Frequency (Rad/Sec)'),ylabel('Imaginary'),grid
if plt==1,print -f1 -deps v2_056b,end;
pause
clg
plot(f,real(xf1))
xlabel('Frequency (Rad/Sec)'),ylabel('Real'),grid
if plt==1,print -f1 -deps v2_056g,end;
pause
clg
plot(f,imag(xf1))
xlabel('Frequency (Rad/Sec)'),ylabel('Imaginary'),grid
if plt==1,print -f1 -deps v2_056h,end;
pause
```

```
clg
axis([0,2.5,-0.6,0.6])
plot(f,real(xf2))
xlabel('Frequency (Rad/Sec)'),ylabel('Real'),grid
if plt==1,print -f1 -deps v2_056d,end;
pause
clg
plot(f,imag(xf2))
xlabel('Frequency (Rad/Sec)'),ylabel('Imaginary'),grid
if plt==1,print -f1 -deps v2_056e,end;
axis([1 2 3 4]);axis;
```

```
% v2_057.m
%
% This is a script file to solve a 2 DOF system
% given the mass, damping and stiffness matrices
% in dimensionless units and plot any desired
% frequency response function.
%
% Plot Formats for FRF and IRF Measurements
% Figures for UC-SDRL-CN-20-263-662, Appendix A

%*******************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%******************************************************************
%
clear,clg
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
pi=3.14159265;
axis('normal');
axis([1 2 3 4]),axis;
pi=3.14159265;
alpha=5;
mass=[8,0;0,10];
stiff=[600000,-200000;-200000,200000];
damp=alpha.*mass;
null=[0,0;0,0];
%    Form 2N x 2N state space equation.
a=[null,mass;mass,damp];
b=[-mass,null;null,stiff];
[x,d]=eig(b,-a);
% Sort Modal Frequencies
orig_lambda=diag(d);
[Y,I]=sort(imag(orig_lambda));
lambda=orig_lambda(I);
xx=x(:,I);
xx,lambda
xxx=input('Hit any key to continue');
%     Compute 'modal a' and 'modal b' matrix
ma=xx.'*a*xx;
mb=xx.'*b*xx;
%     Extract modal vectors from state-space formulation
psi(1:2,1)=xx(3:4,1);
psi(1:2,2)=xx(3:4,2);
psi(1:2,3)=xx(3:4,3);
psi(1:2,4)=xx(3:4,4);
%      Calculate residue matrices
A1=psi(1:2,1)*psi(1:2,1).'./ma(1,1);
A2=psi(1:2,2)*psi(1:2,2).'./ma(2,2);
A3=psi(1:2,3)*psi(1:2,3).'./ma(3,3);
```

```
A4=psi(1:2,4)*psi(1:2,4).'./ma(4,4);
resp=input('Enter response (row) DOF: (1)');if isempty(resp),resp=1;end
inp=input('Enter input (column) DOF: (2)');if isempty(inp),inp=2;end
residu(1) = A1(resp,inp);
residu(2) = A2(resp,inp);
residu(3) = A3(resp,inp);
residu(4) = A4(resp,inp);
xxx=input('Hit any key to continue');
lambda,residu
xxx=input('Hit any key to continue');
%      Calculate desired plots
t=linspace(0,1,501);
xt1=residu(1)./2*exp(lambda(1).*t);
xt2=residu(2)./2*exp(lambda(2).*t);
xt3=residu(3)./2*exp(lambda(3).*t);
xt4=residu(4)./2*exp(lambda(4).*t);
xt=xt1+xt2+xt3+xt4;
plot(t,xt)
%       Reset plot axis for same + and - limits
V=axis;
ymax=max([abs(V(3)),abs(V(4))]);V(3)=-ymax;V(4)=+ymax;axis(V);
%
plot(t,xt)
xlabel('Time (Sec.)'),ylabel ('Amplitude'),grid
if plt==1,meta v2_057a,end;
pause
axis([1 2 3 4]),axis;
clear t; clear xt;
f=linspace(0,100,501);
om=f.*2.*pi;
H1=residu(1)./(j.*om-lambda(1));
H2=residu(2)./(j.*om-lambda(2));
H3=residu(3)./(j.*om-lambda(3));
H4=residu(4)./(j.*om-lambda(4));
H=H1+H2+H3+H4;
plot(f,real(H))
V=axis;
ymax=max([abs(V(3)),abs(V(4))]);V(3)=-ymax;V(4)=+ymax;axis(V);
plot(f,real(H))
xlabel('Frequency (Hertz)'),ylabel('Amplitude (Real)'),grid
if plt==1,meta v2_057b,end;
pause
axis([1 2 3 4]),axis;
plot(f,imag(H))
V=axis;
ymax=max([abs(V(3)),abs(V(4))]);V(3)=-ymax;V(4)=+ymax;axis(V);
plot(f,imag(H))
xlabel('Frequency (Hertz)'),ylabel('Amplitude (Imag)'),grid
if plt==1,meta v2_057c,end;
pause
axis([1 2 3 4]),axis;
plot(f,abs(H))
xlabel('Frequency (Hertz)'),ylabel('Magnitude'),grid
if plt==1,meta v2_057d,end;
pause
```

```
axis([1 2 3 4]),axis;
plot(f,20.*log10(abs(H)))
xlabel('Frequency (Hertz)'),ylabel('Log Magnitude (dB)'),grid
if plt==1,meta v2_057e,end;
pause
plot(f,360./(2.*pi).*angle(H))
V=axis;
ymax=max([abs(V(3)),abs(V(4))]);V(3)=-ymax;V(4)=+ymax;axis(V);
plot(f,360./(2.*pi).*angle(H))
xlabel('Frequency (Hertz)'),ylabel('Phase (Degrees)'),grid
if plt==1,meta v2_057f,end;
pause
axis([1 2 3 4]),axis;
axis('square')
polar(angle(H),abs(H))
xlabel('Amplitude (Real)'),ylabel('Amplitude (Imag)'),grid
if plt==1,meta v2_057g,end;
```

```
% v2_090.m
%
% This is a script file to solve a 3 DOF system
% given the mass, damping and stiffness matrices
% in dimensionless units and plot any desired
% frequency response function.

%**********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%*********************************************************************
%
clear,clg
pi=3.14159265;
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
mass=[10,0,0;0,14,0;0,0,12];
stiff=[5000,-3000,0;-3000,5500,-2500;0,-2500,2500];
damp=[50,-30,0;-30,55,-25;0,-25,25];
null=[0,0,0;0,0,0;0,0,0];
%    Form 2N x 2N state space equation.
a=[null,mass;mass,damp];
b=[-mass,null;null,stiff];
[x,d]=eig(b,-a);
% Sort Modal Frequencies
orig_lambda=diag(d);
[Y,I]=sort(imag(orig_lambda));
lambda=orig_lambda(I);
xx=x(:,I);
%     Normalize x matrix to real vectors if possible
for ii=1:6
xx(1:6,ii)=xx(1:6,ii)./xx(4,ii);
end
%     Compute 'modal a' and 'modal b' matrix
ma=xx.'*a*xx;
mb=xx.'*b*xx;
%     Extract modal vectors from state-space formulation
psi(1:3,1)=xx(4:6,1);
psi(1:3,2)=xx(4:6,2);
psi(1:3,3)=xx(4:6,3);
psi(1:3,4)=xx(4:6,4);
psi(1:3,5)=xx(4:6,5);
psi(1:3,6)=xx(4:6,6);
lambda
xxx=input('Hit any key to continue');
psi
xxx=input('Hit any key to continue');
ma
xxx=input('Hit any key to continue');
%      Calculate residue matrices
```

```
A1=psi(1:3,1)*psi(1:3,1).'./ma(1,1);
A2=psi(1:3,2)*psi(1:3,2).'./ma(2,2);
A3=psi(1:3,3)*psi(1:3,3).'./ma(3,3);
A4=psi(1:3,4)*psi(1:3,4).'./ma(4,4);
A5=psi(1:3,5)*psi(1:3,5).'./ma(5,5);
A6=psi(1:3,6)*psi(1:3,6).'./ma(6,6);
resp=input('Enter response (row) DOF: (1)');if isempty(resp),resp=1;end
inp=input('Enter input (column) DOF: (1)');if isempty(inp),inp=1;end
residu(1) = A1(resp,inp);
residu(2) = A2(resp,inp);
residu(3) = A3(resp,inp);
residu(4) = A4(resp,inp);
residu(5) = A5(resp,inp);
residu(6) = A6(resp,inp);
%      Calculate desired plots
om=linspace(0,50,501);
H1=residu(1)./(j.*om-lambda(1));
H2=residu(2)./(j.*om-lambda(2));
H3=residu(3)./(j.*om-lambda(3));
H4=residu(4)./(j.*om-lambda(4));
H5=residu(5)./(j.*om-lambda(5));
H6=residu(6)./(j.*om-lambda(6));
H=H1+H2+H3+H4+H5+H6;
om(151),H(151)
om(251),H(251)
xxx=input('Hit any key to continue');
subplot(211),semilogy(om,abs(H),om(151),abs(H(151)),'*',om(251),abs(H(251)),'*')
xlabel('Frequency (Rad/Sec)'),ylabel('Magnitude'),grid
subplot(212),plot(om,angle(H),om(151),angle(H(151)),'*',om(251),angle(H(251)),'*')
xlabel('Frequency (Rad/Sec)'),ylabel('Phase'),grid
title(['FRF   Input: ',num2str(inp),'   Response: ',num2str(resp)])
pause
if plt==1,print -f1 -deps v2_090a,end;
```

```
% v2_095.m
%
% This is a script file to solve a 2 DOF system
% given the mass, damping and stiffness matrices
% in dimensionless units and plot any desired
% frequency response function.  A third DOF is
% then added to act as a spring-mass-damper to
% reduce the magnitude of the first mode of the
% original system.  The 3 DOF system is then
% solved and any frequency response functions
% can be plotted.

%*********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%********************************************************************
%
clear, clg
pi=3.14159265;
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
alpha=0.001
mass=[8,0;0,10];
stiff=[6000,-2000;-2000,2000];
damp=alpha.*stiff;
null=[0,0;0,0];
%    Form 2N x 2N state space equation.
a=[null,mass;mass,damp];
b=[-mass,null;null,stiff];
[x,d]=eig(b,-a);
% Sort Modal Frequencies
orig_lambda=diag(d);
[Y,I]=sort(imag(orig_lambda));
lambda=orig_lambda(I);
xx=x(:,I);
%     Normalize x matrix to real vectors if possible
for ii=1:4
xx(1:4,ii)=xx(1:4,ii)./xx(3,ii);
end
%     Compute 'modal a' and 'modal b' matrix
ma=xx.'*a*xx;
mb=xx.'*b*xx;
%     Extract modal vectors from state-space formulation
psi(1:2,1)=xx(3:4,1);
psi(1:2,2)=xx(3:4,2);
psi(1:2,3)=xx(3:4,3);
psi(1:2,4)=xx(3:4,4);
lambda
xxx=input('Hit any key to continue');
psi
```

```
xxx=input('Hit any key to continue');
ma
xxx=input('Hit any key to continue');
%       Calculate residue matrices
A1=psi(1:2,1)*psi(1:2,1).'./ma(1,1);
A2=psi(1:2,2)*psi(1:2,2).'./ma(2,2);
A3=psi(1:2,3)*psi(1:2,3).'./ma(3,3);
A4=psi(1:2,4)*psi(1:2,4).'./ma(4,4);
resp=input('Enter response (row) DOF: (1)');if isempty(resp),resp=1;end
inp=input('Enter input (column) DOF: (1)');if isempty(inp),inp=1;end
residu(1) = A1(resp,inp);
residu(2) = A2(resp,inp);
residu(3) = A3(resp,inp);
residu(4) = A4(resp,inp);
A1,A2,A3,A4
xxx=input('Hit any key to continue');
lambda,residu
xxx=input('Hit any key to continue');
%       Calculate desired plots
om=linspace(0,50,500);
H1=residu(1)./(j.*om-lambda(1));
H2=residu(2)./(j.*om-lambda(2));
H3=residu(3)./(j.*om-lambda(3));
H4=residu(4)./(j.*om-lambda(4));
HH=H1+H2+H3+H4;
fig1=figure(1);
subplot(211),semilogy(om,abs(HH))
xlabel('Frequency (Rad/Sec)'),ylabel('Magnitude'),grid
title('Frequency Response Function')
subplot(212),plot(om,angle(HH))
xlabel('Frequency (Rad/Sec)'),ylabel('Phase'),grid
pause
if plt==1,print -f1 -deps v2_095a,end;
%
% now solve 3 dof system (with spring-mass-damper added)
%
pi=3.14159265;
alpha=0.001
mass=[8,0,0;0,10,0;0,0,2.50];
stiff=[6000,-2000,0;-2000,2300,-300;0,-300,300];
damp=alpha.*stiff;
null=[0,0,0;0,0,0;0,0,0];
%     Form 2N x 2N state space equation.
a=[null,mass;mass,damp];
b=[-mass,null;null,stiff];
[x,d]=eig(b,-a);
x,d
xxx=input('Hit any key to continue');
%      Pick frequency roots
lambda=diag(d);
%      Normalize x matrix to real vectors if possible
for ii=1:6
x(1:6,ii)=x(1:6,ii)./x(4,ii);
end
%      Compute 'modal a' and 'modal b' matrix
```

```
ma=x.'*a*x;
mb=x.'*b*x;
%     Extract modal vectors from state-space formulation
psi(1:3,1)=x(4:6,1);
psi(1:3,2)=x(4:6,2);
psi(1:3,3)=x(4:6,3);
psi(1:3,4)=x(4:6,4);
psi(1:3,5)=x(4:6,5);
psi(1:3,6)=x(4:6,6);
%      Calculate residue matrices
A1=psi(1:3,1)*psi(1:3,1).'./ma(1,1);
A2=psi(1:3,2)*psi(1:3,2).'./ma(2,2);
A3=psi(1:3,3)*psi(1:3,3).'./ma(3,3);
A4=psi(1:3,4)*psi(1:3,4).'./ma(4,4);
A5=psi(1:3,5)*psi(1:3,5).'./ma(5,5);
A6=psi(1:3,6)*psi(1:3,6).'./ma(6,6);
residu(1) = A1(resp,inp);
residu(2) = A2(resp,inp);
residu(3) = A3(resp,inp);
residu(4) = A4(resp,inp);
residu(5) = A5(resp,inp);
residu(6) = A6(resp,inp);
A1,A2,A3,A4,A5,A6
xxx=input('Hit any key to continue');
lambda,residu
xxx=input('Hit any key to continue');
%      Calculate desired plots
om=linspace(0,50,500);
H1=residu(1)./(j.*om-lambda(1));
H2=residu(2)./(j.*om-lambda(2));
H3=residu(3)./(j.*om-lambda(3));
H4=residu(4)./(j.*om-lambda(4));
H5=residu(5)./(j.*om-lambda(5));
H6=residu(6)./(j.*om-lambda(6));
H=H1+H2+H3+H4+H5+H6;
fig2=figure(2);
subplot(211),semilogy(om,abs(H),om,abs(HH))
xlabel('Frequency (Rad/Sec)'),ylabel('Magnitude'),grid
title('Frequency Response Function')
subplot(212),plot(om,angle(H),om,angle(HH))
xlabel('Frequency (Rad/Sec)'),ylabel('Phase'),grid
pause
if plt==1,print -f2 -deps v2_095b,end;
% Now compute final H22 using impedance method
M3=2.5;
C3=.3;
K3=300;
H33=-M3.*om.*om+j.*C3.*om+K3;
H33=H33./((-M3.*om.*om).*(K3+j.*om.*C3));
HHH=HH.*H33./(HH+H33);
fig3=figure(3);
subplot(211),semilogy(om,abs(H),om,abs(HHH))
xlabel('Frequency (Rad/Sec)'),ylabel('Magnitude'),grid
title('Frequency Response Function')
subplot(212),plot(om,angle(H),om,angle(HHH))
```

```
xlabel('Frequency (Rad/Sec)'),ylabel('Phase'),grid
pause
if plt==1,print -f3 -deps v2_095c,end;
```

```
% v2_099.m
%
% This is a script file to solve a 3 DOF system
% given the mass, damping and stiffness matrices
% in dimensionless units and plot any desired
% frequency response function.
%
% Solution for FRF by Inverse of [B]

%**********************************************************************
% Author: Randall J. Allemang
% Date:     18-Apr-94
% Structural Dynamics Research Lab
% University of Cincinnati
% Cincinnati, Ohio  45221-0072
% TEL:  513-556-2725
% FAX:  513-556-3390
% E-MAIL: randy.allemang@uc.edu
%*********************************************************************
%
clear,clf,close all;
pi=3.14159265;
plt=input('Store plots to file (Yes=1): (0)');if isempty(plt),plt=0;end;
%
%  solve 3 dof system
%
mass=[10,0,0;0,14,0;0,0,12];
stiff=[5000,-3000,0;-3000,5500,-2500;0,-2500,2500];
damp=[50,-30,0;-30,55,-25;0,-25,25];
null=[0,0,0;0,0,0;0,0,0];
%    Form 2N x 2N state space equation.
a=[null,mass;mass,damp];
b=[-mass,null;null,stiff];
[x,d]=eig(b,-a);
% Sort Modal Frequencies
orig_lambda=diag(d);
[Y,I]=sort(imag(orig_lambda));
lambda=orig_lambda(I);
xx=x(:,I);
%     Normalize x matrix to real vectors if possible
for ii=1:6
xx(1:6,ii)=xx(1:6,ii)./xx(4,ii);
end
%     Compute 'modal a' and 'modal b' matrix
ma=xx.'*a*xx;
mb=xx.'*b*xx;
%     Extract modal vectors from state-space formulation
psi(1:3,1)=xx(4:6,1);
psi(1:3,2)=xx(4:6,2);
psi(1:3,3)=xx(4:6,3);
psi(1:3,4)=xx(4:6,4);
psi(1:3,5)=xx(4:6,5);
psi(1:3,6)=xx(4:6,6);
lambda
xxx=input('Hit any key to continue');
```

```
psi
xxx=input('Hit any key to continue');
ma
xxx=input('Hit any key to continue');
clear a,clear b,clear x,clear d;

omega=linspace(0,100,501);
for ii=1:501;
FRF(:,:,ii)=inv(-mass.*omega(ii).*omega(ii) + damp.*j.*omega(ii) + stiff);
end;

for Ni=1:3;
for No=1:3;
H=squeeze(FRF(No,Ni,:));
figure;
subplot(211),semilogy(omega,abs(H))
xlabel('Frequency (Rad/Sec)'),ylabel('Magnitude'),grid
scale=360.0/(2.0*pi);
subplot(212),plot(omega,scale.*angle(H))
xlabel('Frequency (Rad/Sec)'),ylabel('Phase'),grid
title('Frequency Response Function: H(x,x)');
pause
end
end
if plt==1,print -f1 -deps v2_099a,end;
pause
```