Eigenvalues and Eigenvectors : Basis for Modal Analysis

September 22, 1998

In the first model problem, we have consider vibration of an elastic string:

$$m\frac{\partial^2 u}{\partial t^2} = T\frac{\partial^2 u}{\partial x^2} - ku + f \qquad in \qquad \Omega = (0, L)$$
$$\frac{\partial u}{\partial t}(0, x) = v_0(x) \quad , \quad u(0, x) = u_0(x)$$
$$u(t,0) = u(t, L) = 0$$

Here a distributed elastic foundation is assumed to be connected on the elastic string spanned by a tensile force T, and u_0 and v_0 are the initial displacement and velocity at the initial time $t = 0$. Applying the weighted residual method with the finite number of trial functions and test functions $\{\phi_j(x), \psi_i(x)\}$, a discrete problem

$$\mathbf{M}\frac{d^2\mathbf{u}}{dt^2} + \mathbf{K}\mathbf{u} = \mathbf{f}$$
$$\mathbf{u}(0) = \mathbf{u}_0 \quad , \quad \frac{d\mathbf{u}}{dt}(0) = \mathbf{v}_0$$

where

$$m_{ij} = \langle m\phi_j, \psi_i \rangle = \int_0^L m\phi_j(x)\psi_i(x)dx$$
$$k_{ij} = \left\langle T\frac{d\phi_j}{dx}, \frac{d\psi_i}{dx} \right\rangle + \langle k\phi_j, \psi_i \rangle = \int_0^L \left\{ T\frac{d\phi_j}{dx}\frac{d\psi_i}{dx} + k\phi_j(x)\psi_i(x) \right\} dx$$
$$f_i = \langle f, \psi_i \rangle = \int_0^L f(t, x)\psi_i(x)dx$$
$$u_{0i} = \langle u_0, \psi_i \rangle = \int_0^L u_0(x)\psi_i(x)dx \quad , \quad v_{0i} = \langle v_0, \psi_i \rangle = \int_0^L v_0(x)\psi_i(x)dx$$

We shall now consider a homogeneous problem that the right hand side, the applied distributed force f is zero:

$$\mathbf{M}\frac{d^2\mathbf{u}}{dt^2}+\mathbf{K}\mathbf{u}=\mathbf{0}$$

and we shall consider a stationary problem with harmonic motion such that

$$\mathbf{u}(t)=e^{i\omega t}\mathbf{x} \quad , \quad i=\sqrt{-1}$$

where ω is a frequency of the harmonic motion in time, and ***x*** is independent of time *t*. Substitution of the harmonic motion into the equation of motion, we have

$$-\lambda\,\mathbf{M}\mathbf{x}+\mathbf{K}\mathbf{x}=\mathbf{0} \quad , \quad \lambda=\omega^2$$

that is

$$\mathbf{K}\mathbf{x}=\lambda\,\mathbf{M}\mathbf{x} \quad , \quad \lambda=\omega^2 .$$

When *m* = *n* in the discrete problem, this is called a *generalized eigenvalue problem*, where λ is an *eigenvalue* and **x** is an associated *eigenvector* of **K** and **M**. If we can decompose the mass matrix into the following form

$$\mathbf{M}=\mathbf{L}\mathbf{L}^T$$

where **L** is non-singular in the sense that its inverse $\mathbf{L}^{-1}$ exists, then

$$\mathbf{K}\mathbf{x}=\lambda\,\mathbf{M}\mathbf{x}=\lambda\,\mathbf{L}\mathbf{L}^T\mathbf{x} \quad \Leftrightarrow \quad \mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-T}\mathbf{L}^T\mathbf{x}=\lambda\,\mathbf{L}^{-1}\mathbf{M}\mathbf{x}=\lambda\,\mathbf{L}^T\mathbf{x}$$

Putting

$$\mathbf{y}=\mathbf{L}^T\mathbf{x}$$

we can convert the generalized eigenvalue problem to the form of the standard eigenvalue problem

$$\mathbf{L}^{-1}\mathbf{K}\left(\mathbf{L}^{-1}\right)^T\mathbf{y}=\lambda\,\mathbf{y} \quad , \quad \mathbf{A}\mathbf{y}=\lambda\,\mathbf{y} .$$

In this case, if **M** and **K** are symmetric, then **A** is also symmetric. That is, the transformation $\mathbf{y} = \mathbf{L}^T\mathbf{x}$ of the eigenvector, yields a symmetric eigenvalue problem. However, $\mathbf{M}^{-1}\mathbf{K}$ need not be symmetric even for both symmetric **M** and **K**, and then the eigenvalue problem

$$\mathbf{M}^{-1}\mathbf{K}\mathbf{x} = \lambda\,\mathbf{x}$$

must be solved only by a un-symmetric eigenvalue solver.

Suppose that we have n number of eigenvalues, $\lambda_1,....,\lambda_n$, and n number of linearly independent eigenvectors, $\mathbf{x}_1,....,\mathbf{x}_n$, of the generalized eigenvalue problem:

$$\mathbf{K}\mathbf{x}_i = \lambda_i\mathbf{M}\mathbf{x}_i \quad , \quad i = 1,2,.....,n\,.$$

Applying the Gram-Schmidt orthogonalization process with respect to a given matrix **M** of n number of linearly independent vectors $\mathbf{z}_1,.....,\mathbf{z}_n$:

$$\mathbf{x}_1 = \frac{\mathbf{z}_1}{\sqrt{(\mathbf{z}_1\,,\,\mathbf{M}\mathbf{z}_1)}},$$

$$\bar{\mathbf{z}}_i = \mathbf{z}_i - \sum_{j=1}^{i-1}(\mathbf{x}_j, \mathbf{M}\mathbf{z}_i)\mathbf{x}_j \quad , \quad \mathbf{x}_i = \frac{\bar{\mathbf{z}}_i}{\sqrt{(\bar{\mathbf{z}}_i\,,\,\mathbf{M}\bar{\mathbf{z}}_i)}} \quad , \quad i = 2,....,n$$

n number of linearly independent eigenvectors $\mathbf{x}_1,....,\mathbf{x}_n$, can be orthonomalized with respect to the matrix **M** (i.e. $(\mathbf{x}_i\,,\,\mathbf{M}\mathbf{x}_j) = \delta_{ij}$), and then they can span the n dimensional space $\mathbf{R}^n$, that is, any n component vector can be represented by a linear combination of the eigenvectors. Here $(\mathbf{z},\mathbf{y}) = \mathbf{z}^T\mathbf{y}$ is the dot product of two n component vectors.

Suppose that we assume the solution form of the discrete vibration problem:

$$\mathbf{u}(t) = \sum_{j=1}^{n} U_j(t)\mathbf{x}_j\,,$$

then we have

$$\sum_{j=1}^{n}\left(\mathbf{M}\mathbf{x}_j\frac{d^2U_j}{dt^2}+\mathbf{K}\mathbf{x}_jU_j\right)=\mathbf{f} \quad \Leftrightarrow \quad \sum_{j=1}^{n}\left(\mathbf{M}\mathbf{x}_j\frac{d^2U_j}{dt^2}+\lambda_j\mathbf{M}\mathbf{x}_jU_j\right)=\mathbf{f}$$

$$\sum_{j=1}^{n}\mathbf{x}_jU_j(0)=\mathbf{u}_0 \quad , \quad \sum_{j=1}^{n}\mathbf{x}_j\frac{dU_j}{dt}(0)=\mathbf{v}_0$$

Multiplying another eigenvector $\mathbf{x}_i$ to these equations, and applying the orthogonality condition

$$\mathbf{x}_i\mathbf{M}\mathbf{x}_j=\delta_{ij}=\begin{cases}1 & if \quad i=j\\ 0 & if \quad i\neq j\end{cases},$$

we can derive n number of single ordinary differential equations from the discrete problem:

$$\frac{d^2U_j}{dt^2}+\lambda_jU_j=(\mathbf{x}_j,\mathbf{f})$$

$$U_j(0)=(\mathbf{x}_j,\mathbf{u}_0) \quad , \quad \frac{dU_j}{dt}(0)=(\mathbf{x}_j,\mathbf{v}_0)$$

These single second order ordinary differential equations can be solved analytically by using either MATLAB or MATHEMATICA, and then we can obtain the analytical solution $\mathbf{u}$ of the discrete problem. It is, however, noted that if the number n becomes large, finding all the eigenvalues and their associated eigenvectors becomes unrealistic. In such a case, we shall approximate the solution $\mathbf{u}$ by using few number of eigenvectos. In general, if the eigenvalues are reordered from the small absolute values to larger ones, then the dynamic motion is largely dominated by eigenvectors associated with smaller eigenvalues.

Indeed, if an approximation of the solution $\mathbf{u}$ of the static problem:

$$\mathbf{K}\mathbf{u}=\mathbf{f}$$

is considered, and its approximate solution $\mathbf{u}_m$ is defined by using the first m number of

eigenvectors associated with the *m* number of eigenvalues from the small side:

$$\mathbf{u}_m = \sum_{j=1}^{m} U_j \mathbf{x}_j$$

then the norm of error of the is defined, and it is bounded by

$$e_m = \sqrt{(\mathbf{u}_m - \mathbf{u})^T (\mathbf{u}_m - \mathbf{u})} = \sqrt{\sum_{j=m+1}^{n} U_j \mathbf{x}_j{}^T \sum_{i=m+1}^{n} U_i \mathbf{x}_i}$$
$$= \sqrt{\sum_{j=m+1}^{n} \frac{(\mathbf{K} U_j \mathbf{x}_j)^T}{\lambda_j} \sum_{i=m+1}^{n} \frac{\mathbf{K} U_i \mathbf{x}_i}{\lambda_i}} \le \left(\sum_{j=1}^{n} \frac{1}{|\lambda_j|} \right) \|\mathbf{f}\|$$

Here we have used the fact that

$$\mathbf{K}\mathbf{x}_j = \lambda_j \mathbf{x}_j \quad , \quad j = 1,.....,n \quad and \quad \mathbf{K} \sum_{j=1}^{n} U_j \mathbf{x}_j = \mathbf{f}$$ [1].

Example 1 : Static Equilibrium Problem As an example of approximation by using a finite number of eigenvectors, we shall consider a similar differential equation with the static equilibrium equation of an elastic string spanned on a distributed spring with a tensile force:

$$-\frac{d}{dx}\left(a(x)\frac{du}{dx} \right) + k(x)u = f(x) \quad in \quad (0, L)$$
$$u(0) = u(L) = 0$$

Decomposing the interval Ω=(0,*L*) into *n* number of equal size subinterval by introducing the equally distributed nodes defined by

---

[1] Multiplying the eigenvector $\mathbf{x}_i$ to this equation, we have $\lambda_i U_i = \mathbf{x}_i{}^T \mathbf{f}$ that is $U_i = \frac{\mathbf{x}_i{}^T \mathbf{f}}{\lambda_i}$, $i = 1,...,n$. This yields that the coefficient U_i is inversely proportional to the eigenvalue λ_I, and the solution of the static equilibrium: $\mathbf{u} = \sum_{i=1}^{n} \frac{\mathbf{x}_i{}^T \mathbf{f}}{\lambda_i} \mathbf{x}_i$.

$$x_i = (i-1)\frac{L}{n-1} \quad , \quad i = 1,.....,n+1,$$

we shall apply the central finite difference approximation:

$$-\frac{\left(a\frac{du}{dx}\right)_{i+\frac{1}{2}} - \left(a\frac{du}{dx}\right)_{i-\frac{1}{2}}}{\Delta x} + k_i u_i = f_i \quad , \quad i = 2,....,n$$

and

$$-\frac{a_{i+\frac{1}{2}}u_{i+1} - \left(a_{i+\frac{1}{2}} + a_{i-\frac{1}{2}}\right)u_i + a_{i-\frac{1}{2}}u_{i-1}}{\Delta x^2} + k_i u_i = f_i \quad , \quad i = 2,....,n$$

Knowing that the boundary condition yields

$$u_1 = u_{n+1} = 0$$

we can derive a matrix equation

$$\mathbf{Ku} = \mathbf{f}$$

where

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \frac{a_{1.5}}{\Delta x^2} + \frac{a_{2.5}}{\Delta x^2} + k_2 & -\frac{a_{2.5}}{\Delta x^2} & 0 & 0 & \dots & 0 \\ 0 & -\frac{a_{2.5}}{\Delta x^2} & \frac{a_{2.5}}{\Delta x^2} + \frac{a_{3.5}}{\Delta x^2} + k_3 & -\frac{a_{3.5}}{\Delta x^2} & 0 & \dots & 0 \\ 0 & 0 & -\frac{a_{3.5}}{\Delta x^2} & \frac{a_{3.5}}{\Delta x^2} + \frac{a_{4.5}}{\Delta x^2} + k_4 & -\frac{a_{4.5}}{\Delta x^2} & \dots & 0 \\ 0 & 0 & 0 & -\frac{a_{4.5}}{\Delta x^2} & \frac{a_{4.5}}{\Delta x^2} + \frac{a_{5.5}}{\Delta x^2} + k_5 & \dots & 0 \\ : & : & : & : & : & . & : \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$\mathbf{u}^T = \{u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad \ldots \quad u_{n+1}\}$$

$$\mathbf{f}^T = \{0 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad \ldots \quad 0\}$$

In this example, let us assume

$$a(x) = 1 + 0.8\sin\left(2\pi \frac{x}{L}\right) \quad , \quad k(x) = 1 + e^{-\frac{x}{L}} \quad , \quad f(x) = \sin\left(\pi \frac{x}{L}\right)$$

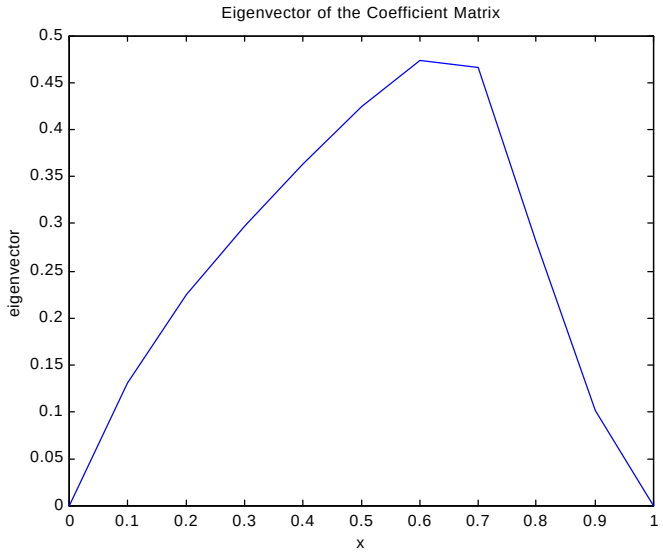
and n = 10. Using the following MATLAB script program:

```
% finite difference method to solve
% -d( a du/dx )/dx +ku = f  in  (0,L)
% u(0) = u(L) = 0
% MEAM 501    1998 Fall
%
L=1;
n=10;
dx=L/n;
% set up nodal values of a, k, and f
a=[];
k=[];
f=[];
x=[];
for i=1:n+1
   xi=(i-1)*dx;
   x(i)=xi;
   a(i)=1+0.8*sin(2*pi*xi/L);
   k(i)=1+exp(-xi/L);
   f(i)=sin(pi*xi/L);
end
% set up the coefficient matrix K
A=[];
for i=2:n
   am=0.5*(a(i-1)+a(i))/dx^2;
   ap=0.5*(a(i)+a(i+1))/dx^2;
```

```
   ad=am+ap+k(i);
   A(i,i-1)=-am;
   A(i,i)=ad;
   A(i,i+1)=-ap;
end
% set up the boundary condition
A(2,1)=0;
A(n,n+1)=0;
A(1,1)=1;
A(n+1,n+1)=1;
f(1)=0;
f(n+1)=0;
A
f
% solution of this problem
u=f/A
plot(x,u)
% eigenvalues and eigenvectors of the coefficient matrix A
[X,L]=eig(A)
% reordering the eigenvalues and eigenvectors
for i=1:n
   for j=i+1:n+1
      lami=L(i,i);
      lamj=L(j,j);
      if lami>lamj
         qi=X(:,i);
         qj=X(:,j);
         L(i,i)=lamj;
         L(j,j)=lami;
         X(:,i)=qj;
         X(:,j)=qi;
      end
   end
end
L
for i=1:n+1
```

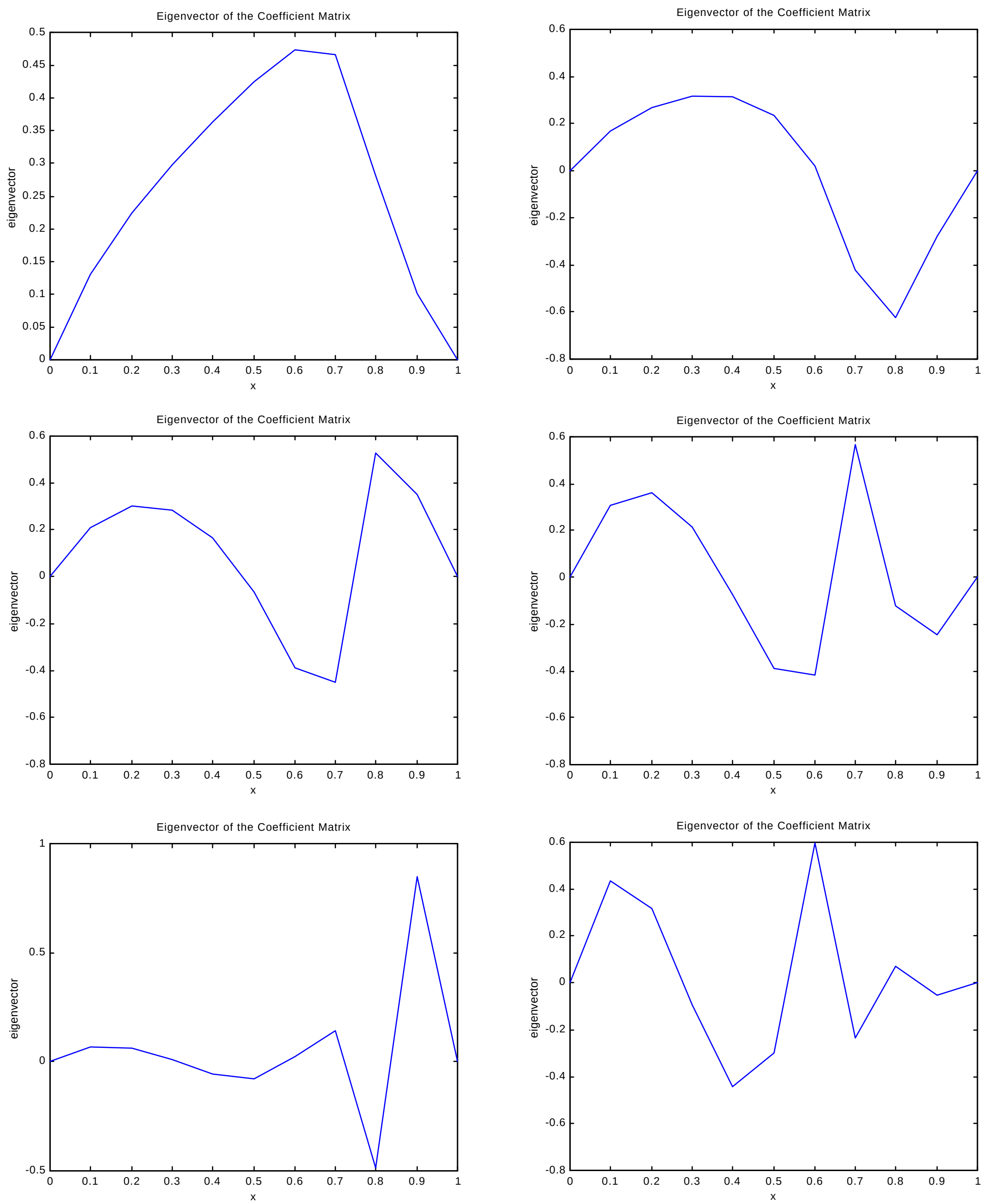
```
   L(i,i)
   plot(x,X(:,i))
   xlabel('x')
   ylabel('eigenvector')
   title('Eigenvector of the Coefficient Matrix')
   pause
end
% amount of domination by the eigenvectors
ua=zeros(n+1,1);
for i=3:n+1
   ui=u*X(:,i)
   ua=ua+ui*X(:,i);
   plot(x,ua,x,u)
   xlabel('x')
   ylabel('u & um')
   title('Stepwise Eigenvector Approximation')
   pause
end
```

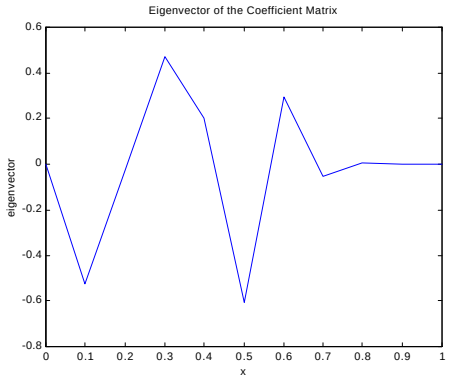
we shall solve the boundary value problem by the central finite difference approximation. For $n = 10$, the solution profile is obtained as follows:

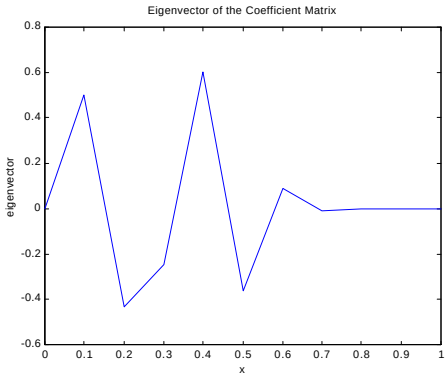


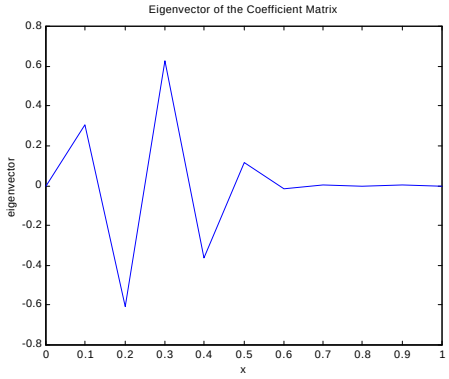
Now, using the nine eigenvectors of the coefficient matrix K except the ones related to

the boundary condition:

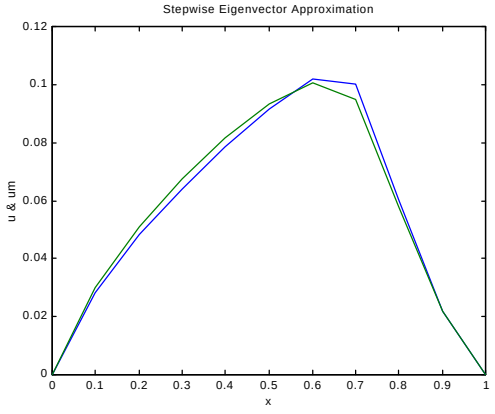


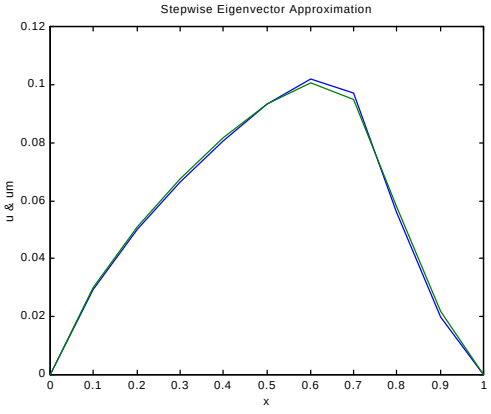


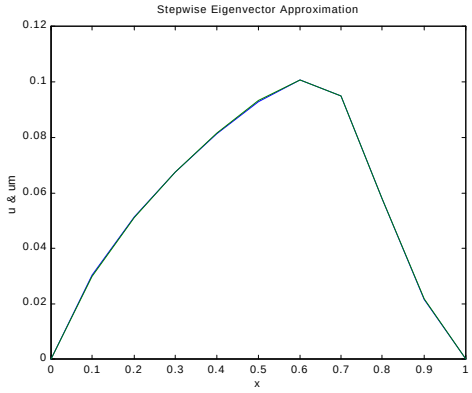


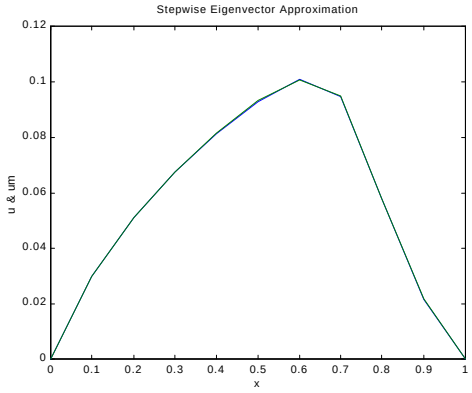


we shall approximate the solution by using the first *m* number of eigenvectors from small side:









It is clear that the first three eigenvectors are sufficient to approximate the solution. Indeed, if we compute the dot product of the solution and the eigenvectors, we have

Number of Eigenvectors	Dot Product $\mathbf{u} \bullet \mathbf{x}_i$
1	0.2152
2	0.0074
3	0.0041
4	6.5542×10^{-4}
5	3.5516×10^{-4}
6	-4.2776×10^{-4}
7	1.1575×10^{-4}
8	-1.7537×10^{-5}
9	-2.3127×10^{-6}

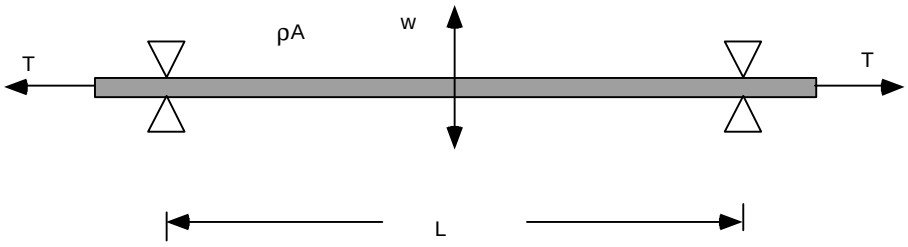
Even one eigenvector can represent the solution very well. It is noted that the right hand side **f** is given, and it is not related to the mechanical system of the elastic string spanned on the spring in the sense that it can be changed. The eigenvalues and eigenvectors are directly related to the mechanical system, not to the input data. In this sense, knowing the eigenvalues and eigenvectors is essential to grasp the characterization of the mechanical system.

Exercise 1 Compute the dot product of the right hand side f and the eigenvectors. From these numbers, you may observe that the first few eigenvectors are sufficient to represent the right hand side.

Example 2 : Vibration of an Elastic String The free vibration problem of an elastic string (or rope), that is tensioned by the force T, can be defined by the following initial-boundary value problem

$$\begin{aligned} &\rho A\frac{\partial^2 w}{\partial t^2} - T\frac{\partial^2 w}{\partial x^2} = 0 \qquad in \qquad (0, L) \\ &w(0,t) = w(L,t) = 0 \\ &w(x,0) = w_0 \qquad and \qquad \frac{\partial w}{\partial t}(x,0) = v_0 \end{aligned}$$

where ρ is the mass density of the string per unit volume, A is the cross sectional area, T is the tension force applied, w is the vertical deflection of the string, L is the length of the string, x is the coordinate along the string, and t is the time.



This initial-boundary value problem can be approximated by the finite element method, and can be replaced by the system of ordinary differential equations

$$\begin{aligned} &\mathbf{M}\frac{d^2\mathbf{u}}{dt^2} + \mathbf{K}\mathbf{u} = \mathbf{0} \\ &\mathbf{u}(0) = \mathbf{u}_0 \qquad and \qquad \frac{d\mathbf{u}}{dt}(0) = \mathbf{v}_0 \end{aligned}$$

where $\boldsymbol{M}$ is the global mass matrix, $\boldsymbol{K}$ is the global stiffness matrix, $\boldsymbol{u}$ is the generalized displacement defined by the nodal deflection of the string, $\boldsymbol{u}_0$ is the discretized initial deflection specified, and $\boldsymbol{v}_0$ is the initial nodal velocity specified. If the equal size two node linear elements are applied, the mass and stiffness matrices are computed by

$$\mathbf{M} = \frac{\rho A L}{6 N_E}\begin{bmatrix} 2 & 1 & 0 & 0 & 0 & ..... & \\ 1 & 4 & 1 & 0 & 0 & ..... & 0 \\ 0 & 1 & 4 & 1 & 0 & ..... & 0 \\ 0 & 0 & 1 & 4 & 1 & ..... & 0 \\ 0 & 0 & 0 & 1 & 4 & ..... & 0 \\ : & : & : & : & : & & : \\ 0 & 0 & 0 & 0 & 0 & ..... & 2 \end{bmatrix}$$

(if the consistent mass matrix is computed)

$$\hat{\mathbf{M}} = \frac{\rho A L}{2 N_E}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & ..... & \\ 0 & 2 & 0 & 0 & 0 & ..... & 0 \\ 0 & 0 & 2 & 0 & 0 & ..... & 0 \\ 0 & 0 & 0 & 2 & 0 & ..... & 0 \\ 0 & 0 & 0 & 0 & 2 & ..... & 0 \\ : & : & : & : & : & & : \\ 0 & 0 & 0 & 0 & 0 & ..... & 1 \end{bmatrix}$$

(if the lumped mass matrix is computed)

$$\mathbf{K} = \frac{T N_E}{L}\begin{bmatrix} 1+k_0 & -1 & 0 & 0 & 0 & ..... & \\ -1 & 2 & -1 & 0 & 0 & ..... & 0 \\ 0 & -1 & 2 & -1 & 0 & ..... & 0 \\ 0 & 0 & -1 & 2 & -1 & ..... & 0 \\ 0 & 0 & 0 & -1 & 2 & ..... & 0 \\ : & : & : & : & : & & : \\ 0 & 0 & 0 & 0 & 0 & ..... & 1+k_0 \end{bmatrix}$$

where N_E is the total number of elements, and $TN_E k0/L$ is the spring constant that represents the fixed support. That is, the boundary condition $w = 0$ at $x = 0$ and L, is approximated by the penalty method, i.e., the support is replaced by the very stiff spring. It should be very large number. Using the lumped mass matrix, the above problem can be represented by

$$\frac{d^2\mathbf{u}}{dt^2} + \hat{\mathbf{M}}^{-1}\mathbf{K}\mathbf{u} = \mathbf{0}$$

$$\mathbf{u}(0) = \mathbf{u}_0 \qquad \textit{and} \qquad \frac{d\mathbf{u}}{dt}(0) = \mathbf{v}_0$$

For the following example, we shall assume $N_E = 10$, $L = 1$, $T/\rho A = 1$, and k_0=10^5.

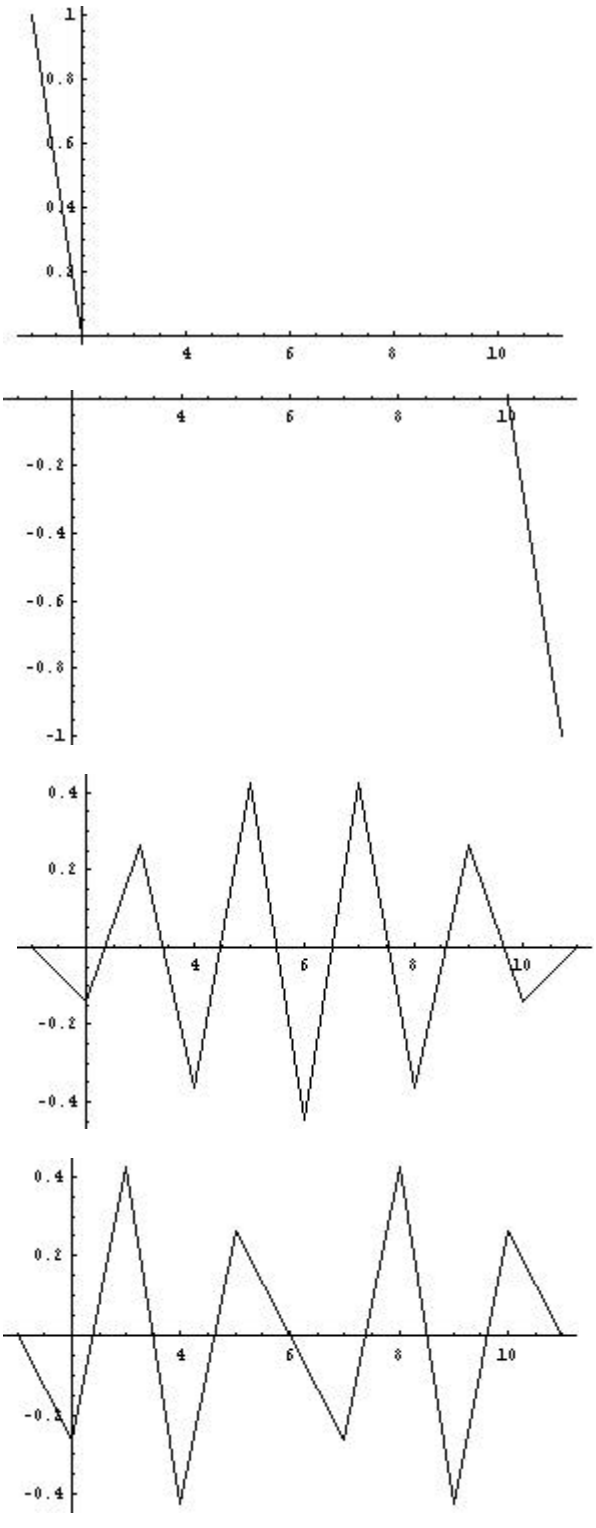
MATHEMATICA Script Program for Transient Analysis

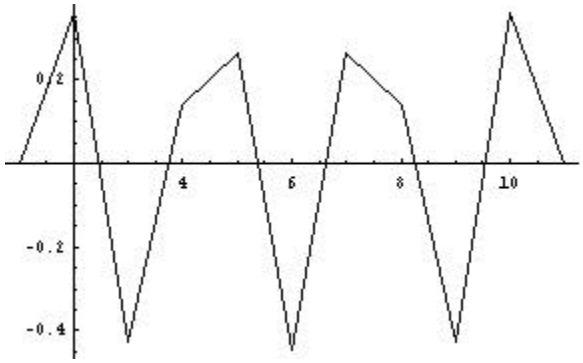
```
(* L       = length of the string                    *)
(* ne      = number of finite elements               *)
(* he      = element size                            *)
(* spring = spring constant to replace the support *)
L=1;
ne=10;
nx=ne+1;
he=L/ne;
spring=10^5/he^2;
(* A = Inverse[M].K          *)
(* M = lumped mass matrix *)
(* K = stiffness matrix     *)
A=Table[0,{i,1,nx},{j,1,nx}];
Block[{i},
  Do[If[i>1,A[[i-1,i]]=-1/he^2,];
     If[i<nx,A[[i+1,i]]=-1/he^2,];
     A[[i,i]]=2/he^2,
     {i,1,nx}]
  ]
A[[1,1]]=(1/he^2+spring)*2;
A[[nx,nx]]=(1/he^2+spring)*2;
(* R = eigenvalues                    *)
(* X = eigenvectors {x1,x2,.....} *)
R=Eigenvalues[N[A]]
X=Eigenvectors[N[A]];
(* Plot the Eigenvectors *)
Block[{i},
  Do[xi=Table[X[[i,j]],{j,1,nx}];
     ListPlot[xi,PlotJoined->True],
     {i,1,nx}]
  ]
(* Define the initial displacement x0 *)
```

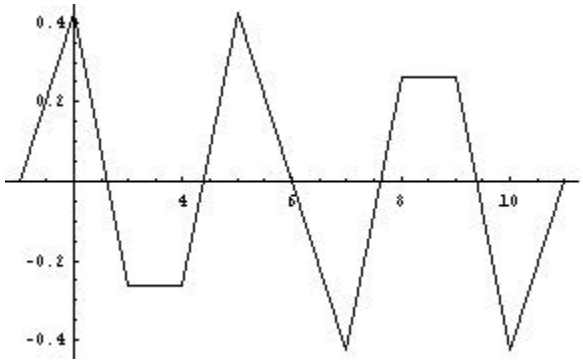
```
x0=N[Table[(he*(i-1))^2 Sin[Pi he*(i-1)],{i,1,nx}]]
(* Define the initial velocity y0        *)
y0=N[Table[0,{i,1,nx}]]
ListPlot[x0,PlotJoined->True]
ListPlot[y0,PlotJoined->True]
x0i=N[X.x0]
y0i=N[X.y0]
xt=Table[0,{i,1,nx},{j,1,nx}];
(* Plot the Dynamical Response                          *)
(* set up the time increment and the final time *)
dt=0.01;
Tmax=0.3;
imax=Round[Tmax/dt]+1;
kmax=5;
Block[{i,j,k},
  Do[
  Do[xt[[j,k]]=Sum[(Cos[R[[i]] t] x0i[[i]] +
                Sin[R[[i]] t] y0i[[i]]/R[[i]]) X[[i,j]],
                    {i,nx-k+1,nx}],
      {j,1,nx}];
  xtp=Table[xt[[j,k]]/.{t->dt*(i-1)},
                {j,1,nx},{i,1,imax}];
  ListPlot3D[xtp,PlotRange->All],
  {k,1,kmax}]
 ]
```

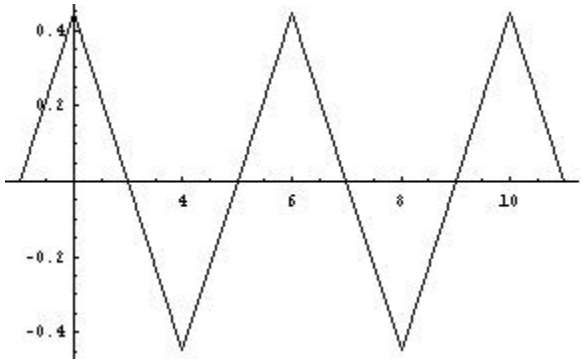
OUTPUT from the above MATHEMATICA program

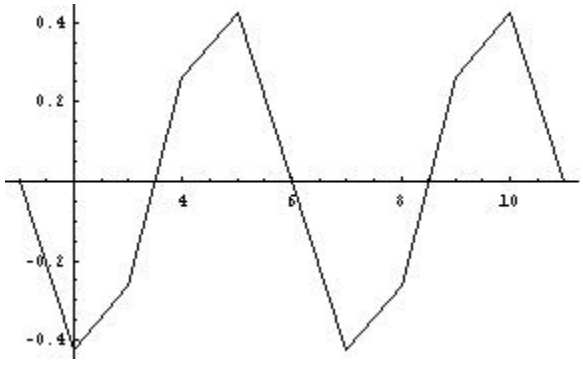
```
Computed Eigenvalues ( Reverse Ordering )
                7                7
{2.00002 10 , 2.00002 10 , 390.211, 361.803, 317.557,

  261.803, 200., 138.196, 82.4428, 38.1965, 9.78868}

Associated Eigenvectors ( Free Vibration Modes )
```

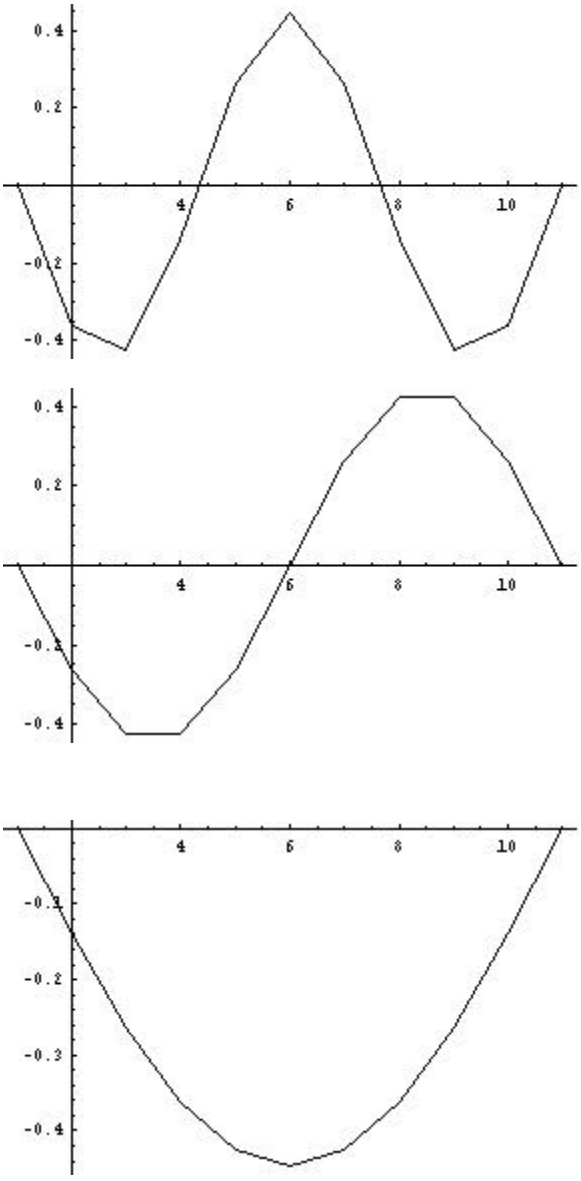

1
0.8
0.6
0.4
0.2
4
6
8
10
4
6
8
10
-0.2
-0.4
-0.6
-0.8
-1
0.4
0.2
4
6
8
10
-0.2
-0.4
0.4
0.2
4
6
8
10
-0.4


0.2
4
6
8
10
-0.2
-0.4


0.4
0.2
4
6
8
10
-0.2
-0.4


0.4
0.2
4
6
8
10
-0.2
-0.4


0.4
0.2
4
6
8
10
-0.2
-0.4



Initial nodal displacement (specified)

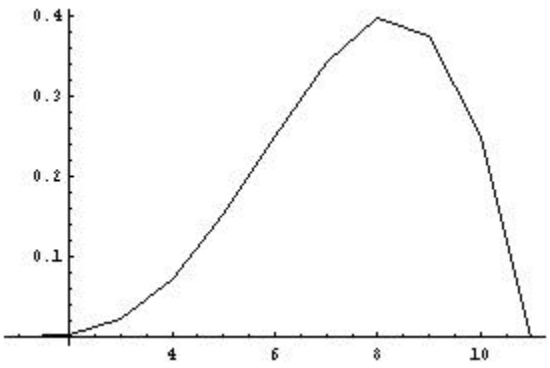
{0, 0.00309017, 0.0235114, 0.0728115, 0.152169, 0.25,

0.34238, 0.396418, 0.376183, 0.250304, 0}

Initial nodal velocity (specified)

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Profile of the initial displacement



Influence coefficients corresponding to the eigenvalues

-8 -6

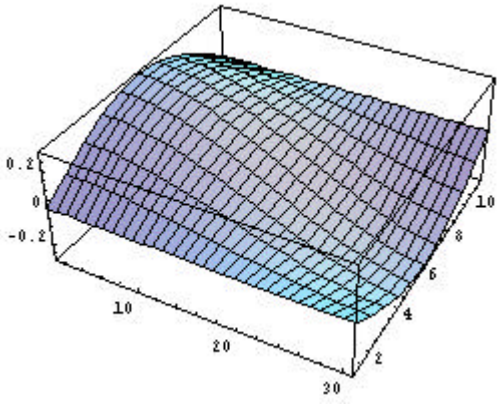
{1.54503 10 , 1.25151 10 , 0.00118029, -0.00262202,

0.00472119, -0.0082775, 0.0152783, 0.0318841, 0.0847209,

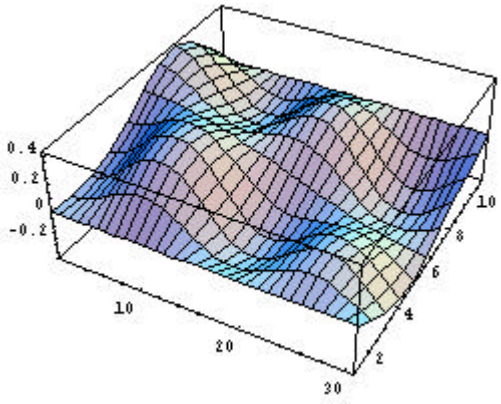
-0.402622, 0.632001}

{0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}

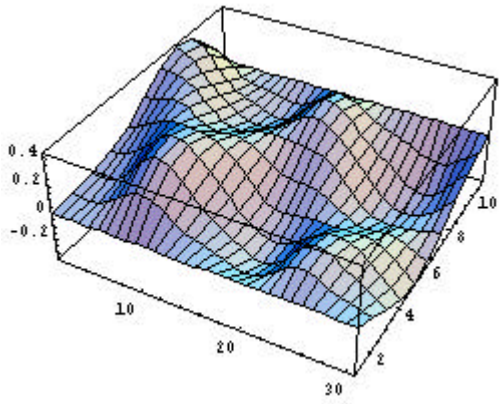
Profiles of the transient response (displacement)



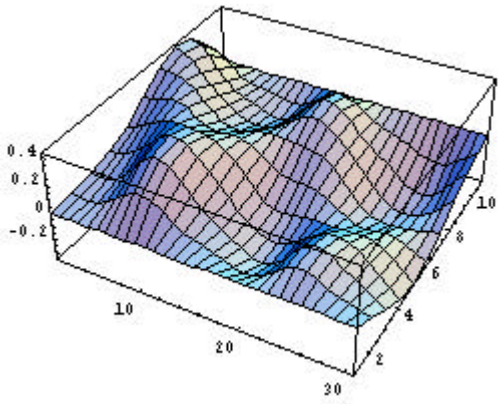
Only by the 11th Mode



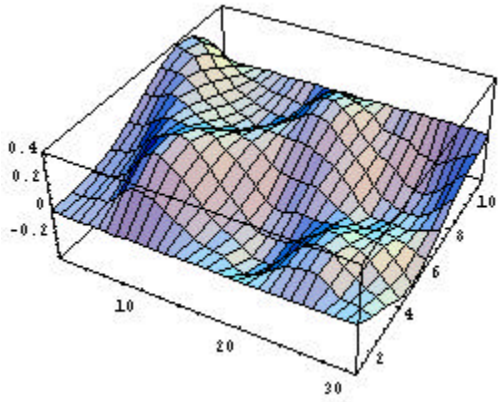
11th + 12th Modes



11th + 10th + 9th Modes



11th + 10th + 9th + 8th Modes



All Modes

It is clear that the first few modes can provide almost exact transient response. In other words, we need not compute all of the eigenvalues and eigenvectors to see transient response of a given dynamic problem. It is also noted that accuracy of the finite element approximation must be taken into account when eigenvalues and eigenvectors are involved. Indeed, in the above example, only 10 finite elements are used to make a finite element model of an elastic string. In this case, if we compare the FEA result with the exact (analytical) solution, we can find the error (%)

$$\{1.67465\times10^{6},\ 2.02634\times10^{6},\ -51.1893,\ -42.7213,\ -34.3362,$$

$$-26.3161,\ -18.9431,\ -12.4861,\ -7.18662,\ -3.24705,\ -0.819959\}$$

That is, the first eigenvalue is computed only with 0.82% error, while the error is increased for the higher eigenvalues. The second one has 3.2% error, the third one has

7.2% error, and so on. If the number of finite elements is increased, then the error should be reduced. Only two eigenvalues are estimated within 5% error. If 20 finite elements are applied, then the error becomes

$$\{1.83794 \times 10^{6}, 2.02634 \times 10^{6}, -55.3695, -51.1893, -46.9622,$$

$$-42.7213, -38.501, -34.3362, -30.2627, -26.3161,$$

$$-22.5314, -18.9431, -15.5842, -12.486, -9.6779,$$

$$-7.18655, -5.03596, -3.24697, -1.837, -0.819863, \quad -0.205547\}$$

In this case, four eigenvalues are estimated within 5% error. It is important to note that in order to obtain accurate eigenvalues and eigenvectors, we must develop very refined models.