Mechanical Vibrations

Chapter 10

Peter Avitabile
Mechanical Engineering Department
University of Massachusetts Lowell
Types of Models for Vibration Analysis

Models are developed to assist in the design and understanding of system dynamics

Analytical models (such as finite element models) are utilized in the design process

Experimental models are also used for many systems where modeling is not practical or models are too difficult to develop
Finite Element Model Considerations

Finite element models are commonly used

What are we trying to do when generating a model

CONTINUOUS SOLUTION

DISCRETIZED SOLUTION
Finite Element Model Considerations

Modeling Issues

- continuous solutions work well with structures that are well behaved and have no geometry that is difficult to handle
- most structures don't fit this simple requirement (except for frisbees and cymbals)

- real structures have significant geometry variations that are difficult to address for the applicable theory
- a discretized model is needed in order to approximate the actual geometry
- the degree of discretization is dependent on the waveform of the deformation in the structure
- finite element modeling meets this need
Finite Element Model Considerations

Finite element modeling involves the **discretization** of the structure into **elements** or **domains** that are defined by **nodes** which describe the **elements**.

A field quantity such as **displacement** is approximated using **polynomial interpolation** over each of the **domains**.

The **best** values of the field quantity at nodes results from a **minimization of the total energy**.

Since there are many nodes defining many elements, a **set of simultaneous equations** results.

Typically, this set of equations is very large and a computer is used to generate results.
Finite Element Model Considerations

Nodes represent geometric locations in the structure.

Elements boundary are defined by the nodes.

The type of displacement field that exists over the domain will determine the type of element used to characterize the domain.

Element characteristics are determined from

Theory of Elasticity
and
Strength of Materials.
Analytical Topics for Structural Dynamic Modeling

Structural element formulations use the same general assumptions about their respective behavior as their respective structural theories (such as truss, beam, plate, or shell).

Continuum element formulations (such as 2D and 3D solid elements) comes from theory of elasticity.

\[
[k] = \frac{EI}{L^3} \begin{bmatrix}
12 & 6L & -12 & 6L \\
6L & 4L^2 & -6L & 2L^2 \\
-12 & -6L & 12 & -6L \\
6L & 2L^2 & -6L & 4L
\end{bmatrix}
\]

\[
[m] = \rho AL \begin{bmatrix}
156 & -22L & 54 & 13L \\
-22L & 4L^2 & -13L & -3L^2 \\
54 & -13L & 156 & 22L \\
13L & -3L^2 & 22L & 4L^2
\end{bmatrix}
\]
Analytical Topics for Structural Dynamic Modeling

The basis of the finite element method is summarized below:

• subdivide the structure into small finite elements
• each element is defined by a finite number of node points
• assemble all elements to form the entire structure
• within each element, a simple solution to governing equations is formulated (the solution for each element becomes a function of unknown nodal values)
• general solution for all elements results in algebraic set of simultaneous equations
Finite Element Model Considerations

DEGREES OF FREEDOM

- maximum 6 dof can be described at a point in space
- finite element use a maximum of 6 dof
- most elements use less than 6 dof to describe the element
Finite Element Model Considerations

**Advantages**
- Models used for design development
- No prototypes are necessary

**Disadvantages**
- Modeling assumptions
- Joint design difficult to model
- Component interactions are difficult to predict
- Damping generally ignored
Finite Element Model Considerations

A typical finite element user may ask

- what kind of elements should be used?
- how many elements should I have?
- where can the mesh be coarse; where must it be fine?
- what simplifying assumptions can I make?
- should all of the physical structural detail be included?
- can I use the same static model for dynamic analysis?
- how can I determine if my answers are accurate?
- how do I know if the software is used properly?
**Finite Element Model Considerations**

ALL THESE QUESTIONS CAN BE ANSWERED, IF

- the general structural behavior is well understood
- the elements available are understood
- the software operation is understood
  (input procedures, algorithms, etc.)

BASICALLY - we need to know what we are doing !!!

IF A ROUGH BACK OF THE ENVELOP ANALYSIS CAN NOT BE FORMULATED, THEN MOST LIKELY THE ANALYST DOES NOT KNOW ENOUGH ABOUT THE PROBLEM AT HAND TO FORMULATE A FINITE ELEMENT MODEL
Finite Element Modeling

Using standard finite element modeling techniques, the following steps are usually followed in the generation of an analytical model:

- node generation
- element generation
- coordinate transformations
- assembly process
- application of boundary conditions
- model condensation
- solution of equations
- recovery process
- expansion of reduced model results
Finite Element Modeling

Element Definition

Each element is approximated by

\[ \{\delta\} = [N]\{x\} \]

where

- \(\{\delta\}\) - vector of displacements in element
- \([N]\) - shape function for selected element
- \(\{x\}\) - nodal variable

Element shape functions can range from linear interpolation functions to higher order polynomial functions.
**Finite Element Modeling**

**Strain Displacement Relationship**

The strain displacement relationship is given by

\[
\{\varepsilon\} = [B]\{x\}
\]

where

\{\varepsilon\} - vector of strain within element

\[B\] - strain displacement matrix

(proportional to derivatives of \[N\])

\{x\} - nodal variable
**Finite Element Modeling**

**Mass and Stiffness Formulation**

The mass and stiffness relationship is given by

\[
[M] = \iiint_V [N] \rho [N]^T \partial V
\]

\[
[K] = \iiint_V [B]^T [C][B] \partial V
\]

where

- \([M]\) - element mass matrix
- \([K]\) - element stiffness matrix
- \([N]\) - shape function for element
- \(\{\rho}\) - density
- \([B]\) - strain displacement matrix
- \([C]\) - stress-strain (elasticity) matrix
Finite Element Modeling

Coordinate Transformation

Generally, elements are formed in a local coordinate system which is convenient for generation of the element. Elemental matrices are transformed from the local elemental coordinate system to the global coordinate system using

$$\{x_1\} = [T_{12}]\{x_2\}$$
Finite Element Modeling

Assembly Process

Elemental matrices are then assembled into the global master matrices using

\[ \{x_k\} = [c_k]\{x_g\} \]

where

- \(\{x_k\}\) - element degrees of freedom
- \([c_k]\) - connectivity matrix
- \(\{x_g\}\) - global degrees of freedom

The global mass and stiffness matrices are assembled and boundary conditions applied for the structure.
Finite Element Modeling

Static Solutions
- typically involve decomposition of a large matrix
- matrix is usually sparsely populated
- majority of terms concentrated about the diagonal

Eigenvalue Solutions
- use either direct or iterative methods
- direct techniques used for small matrices
- iterative techniques used for a few modes from large matrices

Propagation Solutions
- most common solution uses derivative methods
- stability of the numerical process is of concern
- at a given time step, the equations are reduced to an equivalent static form for solution
- typically many times steps are required
Finite Element Modeling - Simple Example

Consider the 2 spring system shown below

- each spring element is denoted by a box with a number
- each element is defined by 2 nodes denoted by the circle with a number assigned to it
- the springs have a node at each end and have a common node point
- the displacement of each node is denoted by u with a subscript to identify which node it corresponds to
- there is an applied force at node 3
**Finite Element Modeling - Simple Example**

The first step is to formulate the spring element in a general sense

- the element label is p
- the element is bounded by node i and j
- assume positive displacement conditions at both nodes
- define the force at node i and node j for the p element

Application of simple equilibrium gives

\[ f_{ip} = k_p (u_i - u_j) = +k_p u_i - k_p u_j \]
\[ f_{jp} = k_p (u_j - u_i) = -k_p u_i + k_p u_j \]
Finite Element Modeling - Simple Example

This can be written in matrix form to give

\[
\begin{bmatrix}
  k_p & -k_p \\
  -k_p & k_p \\
\end{bmatrix}
\begin{bmatrix}
  u_i \\
  u_j \\
\end{bmatrix}
= \begin{bmatrix}
  f_{ip} \\
  f_{jp} \\
\end{bmatrix}
\]

Now for element #1

\[
\begin{bmatrix}
  k_1 & -k_1 \\
  -k_1 & k_1 \\
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_2 \\
\end{bmatrix}
= \begin{bmatrix}
  f_{11} \\
  f_{21} \\
\end{bmatrix}
\]

And for element #2

\[
\begin{bmatrix}
  k_2 & -k_2 \\
  -k_2 & k_2 \\
\end{bmatrix}
\begin{bmatrix}
  u_2 \\
  u_3 \\
\end{bmatrix}
= \begin{bmatrix}
  f_{22} \\
  f_{32} \\
\end{bmatrix}
\]

The equilibrium requires that the sum of the internal forces equals the applied force acting on each node.
Finite Element Modeling - Simple Example

The three equations can be written as

\[ k_1u_1 - k_1u_2 = f_1 \]
\[ -k_1u_1 + k_1u_2 + k_2u_2 - k_2u_3 = f_2 \]
\[ -k_2u_2 + k_2u_3 = f_3 \]

or in matrix form

\[
\begin{bmatrix}
  k_1 & -k_1 \\
  -k_1 & k_1 + k_2 & -k_2 \\
  -k_2 & k_2
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3
\end{bmatrix}
=
\begin{bmatrix}
  f_1 \\
  f_2 \\
  f_3
\end{bmatrix}
\]
Finite Element Modeling - Simple Example

Now applying a boundary condition of zero displacement at node 1 has the effect of zeroing the first column of the K matrix which gives three equations with 2 unknowns. Solving for the second and third equation gives

\[
\begin{bmatrix}
k_1 & -k_1 \\
-k_1 & k_1 + k_2 & -k_2 \\
& -k_2 & k_2
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3
\end{bmatrix}
=
\begin{bmatrix}
f_1 \\
f_2 \\
f_3
\end{bmatrix}
\]

\[
\begin{bmatrix}
k_1 + k_2 & -k_2 \\
-k_2 & k_2
\end{bmatrix}
\begin{bmatrix}
u_2 \\
u_3
\end{bmatrix}
=
\begin{bmatrix}
0 \\
f_3
\end{bmatrix}
\]
Finite Element Modeling - Simple Example

Assembly of the stiffness matrix with more elements

\[
\begin{bmatrix}
k_1 & -k_1 \\
-k_1 & k_1 + k_2 + k_5 & -k_2 & -k_5 \\
-k_2 & k_2 + k_3 & -k_3 \\
-k_3 & k_3 + k_4 & -k_4 \\
-k_5 & -k_4 & k_4 + k_5
\end{bmatrix}
\]

Notice that the banded nature of the matrix is not preserved when elements are arbitrarily added to the assembly.
Finite Element Modeling - Beam Elements

The force-stiffness equation for a beam is given by the following equation

\[
\begin{bmatrix}
F_1 \\
M_1 \\
F_2 \\
M_2
\end{bmatrix}
= \frac{EI}{l^3}
\begin{bmatrix}
12 & 6l & -12 & 6l \\
6l & 4l^2 & -6l & 2l^2 \\
-12 & -6l & 12 & -6l \\
6l & 2l^2 & -6l & 4l^2
\end{bmatrix}
\begin{bmatrix}
u_1 \\
\theta_1 \\
u_2 \\
\theta_2
\end{bmatrix}
\]
Finite Element Modeling - Beam Elements

The beam element stiffness matrix is

\[
[K] = \frac{EI}{l^3} \begin{bmatrix}
12 & 6l & -12 & 6l \\
6l & 4l^2 & -6l & 2l^2 \\
-12 & -6l & 12 & -6l \\
6l & 2l^2 & -6l & 4l^2
\end{bmatrix}
\]

The beam element mass matrix is

\[
[M] = \frac{\rho Al}{420} \begin{bmatrix}
156 & 22l & 54 & -13l \\
22l & 4l^2 & 13l & -3l^2 \\
54 & 13l & 156 & -22l \\
-13l & -3l^2 & -22l & 4l^2
\end{bmatrix}
\]
A three beam model with spring support

\[
[K] = \begin{bmatrix}
12 & 6l & -12 & 6l \\
6l & 4l^2 & -6l & 2l^2 \\
-12 & -6l & 12 & -6l \\
6l & 2l^2 & -6l & 4l^2
\end{bmatrix}
\]

\[
[M] = \begin{bmatrix}
156 & 22l & 54 & -13l \\
22l & 4l^2 & 13l & -3l^2 \\
-13l & -3l^2 & -22l & 4l^2
\end{bmatrix}
\]

\[
[K] = \frac{EI}{l^3}
\]

\[
[M] = \frac{\rho AI}{420}
\]
The individual stiffness elements assemble as

\[
\begin{bmatrix}
(K + 12) & 6l & -12 & 6l \\
6l & 4l^2 & -6l & 2l^2 \\
-12 & -6l & (12 + 12) & (-6l + 6l) \\
6l & 2l^2 & (-6l + 6l) & (4l^2 + 4l^2)
\end{bmatrix}
\]

\[
\begin{bmatrix}
-12 & 6l \\
-6l & 2l^2 \\
-12 & -6l & (12 + 12) & (-6l + 6l) \\
6l & 2l^2 & (-6l + 6l) & (4l^2 + 4l^2)
\end{bmatrix}
\]

\[
\begin{bmatrix}
-12 & 6l \\
-6l & 2l^2 \\
-12 & -6l & (K + 12) & -6l \\
6l & 2l^2 & -6l & 4l^2
\end{bmatrix}
\]
MATLAB / MATSAP Script File

Peter Avitabile - Modal Analysis & Controls Laboratory
University of Massachusetts Lowell

This MATLAB file is used to illustrate the use of MATLAB and four scripts developed by John O'Callahan to generate a beam element mass and stiffness KBEAM, MBEAM, ASSEMBLE, EIGEN

A simple beam model has four different beam elements
It is supported by two linear springs (translation)
There is also a lumped mass at the end of the beam
The beam elements have 2 dof/node (shear/rotary)

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>(3,4)</td>
<td>(5,6)</td>
<td>(7,8)</td>
<td>(9,10)</td>
</tr>
<tr>
<td>(1)</td>
<td>(2)</td>
<td>(3)</td>
<td>(4)</td>
<td>(M)</td>
</tr>
</tbody>
</table>

x-----------x-----------x-----------x
/            /            /            |
\            \            \            |
\            /            KGROUND      |
\            \            |
\            \            |
---          ---          

(node numbers)  (DOF numbers)  (element numbers)  (lumped mass at tip of beam)  (stiffness to ground)
MATLAB / MATSAP Script File

% Setup and Assemble Mass and Stiffness Matrices
% total dof = 10; % total number of degrees of freedom for model
GLOBAL_K = zeros(total_dof,total_dof); % setup initial matrix space for stiffness
GLOBAL_M = zeros(total_dof,total_dof); % setup initial matrix space for stiffness

% Physical parameters - assume aluminum is material
E = 10e6; % Young's Modulus (psi)
rho = 0.1/386.4; % mass density (not weight density)

% Support spring stiffness

K_ground_left = 200000; % assume left spring stiffness is 200,000 lb/in
K_ground_right = 50000; % assume left spring stiffness is 50,000 lb/in

% Mass of weight at end of beam - 5 lbs
Mass = 5/386.4; % assume weight 5 lb
MATLAB / MATSAP Script File

% Beam #1 - Aluminum, I=2.0 in^4, A=1.5 in^2, L=12 in
A_1 = 1.5; I_1 = 2; L_1 = 12; % beam 1 properties
k_beam_1 = kbeam(E,I_1,L_1); % ==> ==> EXTERNAL SCRIPT FILE NEEDED !!!
m_beam_1 = mcbeam(rho,A_1,L_1); % ==> ==> EXTERNAL SCRIPT FILE NEEDED !!!
%
% Beam #2 - Aluminum, I=0.5 in^4, A=0.02 in^2, L=18 in
A_2 = 0.02; I_2 = 0.5; L_2 = 18; % beam 2 properties
k_beam_2 = kbeam(E,I_2,L_2); % ==> ==> EXTERNAL SCRIPT FILE NEEDED !!!
m_beam_2 = mcbeam(rho,A_2,L_2); % ==> ==> EXTERNAL SCRIPT FILE NEEDED !!!
%
% Beam #3 - Aluminum, I=0.25 in^4, A=0.005 in^2, L=24 in
A_3 = 0.005; I_3 = 0.25; L_3 = 24; % beam 3 properties
k_beam_3 = kbeam(E,I_3,L_3); % ==> ==> EXTERNAL SCRIPT FILE NEEDED !!!
m_beam_3 = mcbeam(rho,A_3,L_3); % ==> ==> EXTERNAL SCRIPT FILE NEEDED !!!
%
% Beam #4 - Aluminum, I=2.5 in^4, A=1.5 in^2, L=12 in
A_4 = 1.5; I_4 = 2.5; L_4 = 12; % beam 4 properties
k_beam_4 = kbeam(E,I_4,L_4); % ==> ==> EXTERNAL SCRIPT FILE NEEDED !!!
m_beam_4 = mcbeam(rho,A_4,L_4); % ==> ==> EXTERNAL SCRIPT FILE NEEDED !!!
assemble individual elements into matrices

----------------------assemble beam 1 mass and stiffness-------------------------
[GLOBAL_K] = assemble(GLOBAL_K,k_beam_1,[1,2,3,4]); %==>>SCRIPT FILE NEEDED !!!
[GLOBAL_M] = assemble(GLOBAL_M,m_beam_1,[1,2,3,4]); %==>>SCRIPT FILE NEEDED !!!
----------------------assemble beam 2 mass and stiffness-------------------------
[GLOBAL_K] = assemble(GLOBAL_K,k_beam_2,[3,4,5,6]); %==>>SCRIPT FILE NEEDED !!!
[GLOBAL_M] = assemble(GLOBAL_M,m_beam_2,[3,4,5,6]); %==>>SCRIPT FILE NEEDED !!!
----------------------assemble beam 3 mass and stiffness-------------------------
[GLOBAL_K] = assemble(GLOBAL_K,k_beam_3,[5,6,7,8]); %==>>SCRIPT FILE NEEDED !!!
[GLOBAL_M] = assemble(GLOBAL_M,m_beam_3,[5,6,7,8]); %==>>SCRIPT FILE NEEDED !!!
----------------------assemble beam 4 mass and stiffness-------------------------
[GLOBAL_K] = assemble(GLOBAL_K,k_beam_4,[7,8,9,10]); %==>>SCRIPT FILE NEEDED !!!
[GLOBAL_M] = assemble(GLOBAL_M,m_beam_4,[7,8,9,10]); %==>>SCRIPT FILE NEEDED !!!
----------------------assemble support beam stiffness--------------------------
[GLOBAL_K] = assemble(GLOBAL_K,k_ground_left,[3]); %==>>SCRIPT FILE NEEDED !!!
[GLOBAL_K] = assemble(GLOBAL_K,k_ground_right,[7]); %==>>SCRIPT FILE NEEDED !!!
----------------------assemble lumped mass at end of beam----------------------
[GLOBAL_M] = assemble(GLOBAL_M,Mass,9); % add lumped mass at dof=9 (tip dof)
MATLAB / MATSAP Script File

% perform eigen solution to obtain frequencies and mode shapes
%---------------------------------------------------------------
[shapes, freq] = eigen(GLOBAL_K, GLOBAL_M); %==>> EXTERNAL SCRIPT FILE NEEDED !!!
freq(1:1:2); % list frequencies in Hz
length=[0 (L_1) (L_1+L_2) (L_1+L_2+L_3) (L_1+L_2+L_3+L_4)]; % create total length
figure(1)
plot(length, shapes((1:2:(total_dof-1)),1)); % plot mode 1
title('Mode Shape - first mode')
figure(2)
plot(length, shapes((1:2:(total_dof-1)),2)); % plot mode 2
title('Mode Shape - second mode')
Simplistic Model of Mixing Device

Natural frequency determination and support location are required for a mixing device.

PROPOSED SUPPORT LOCATIONS

MOTOR = 1000 LB
Simplistic Model of Mixing Device

The device can be broken down into simplistic pieces of beam elements, mass elements and support characteristics.
Simplistic Model of Mixing Device

The characteristics of each individual component are needed in order to make a rough model of the system for preliminary evaluations. The model is developed to determine the system characteristics to assess whether a more detailed model is needed.