SO WHAT AM I TRYING TO DO WHEN MAKING A FEM ???

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 MODELING OVERVIEW

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 **many nodes** define **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 MODELING OVERVIEW

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?

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 OVERVIEW

**Nodes** are used to represent geometric locations in the structure.

**Element** boundary 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.*

Finite element method is a numerical method for solving a system of governing equations over the domain of a continuous physical system.

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 Diagram](image-url)
FINITE ELEMENT MODELING OVERVIEW

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 OVERVIEW

All structures are 3 dimensional in nature but many times simplifying assumptions can be assumed with no other loss in accuracy

Elements are typically categorized as

**Structural Elements**

**Continuum Elements**

- 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

A wide variety of different element types generally exists in most commercially available finite element software packages.

Typical structural elements are mass, truss, beam, membrane, plane stress/plane strain, thin plate, thin shell, thick plate, 3 dimensional solid with a variety of shape functions ranging from linear to higher order polynomial.
FINITE ELEMENT MODELING OVERVIEW

Element Definition

Each element is approximated by

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

where

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

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

A simple illustration of shape functions is shown below
FINITE ELEMENT MODELING OVERVIEW

**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

**Mass and Stiffness Formulation**

The mass and stiffness relationship is given by

\[
[M] = \iiint_V [N]^T \rho [N]^T \, \varepsilon \, dV
\]
\[
[K] = \iiint_V [B]^T [C] [B] \, \varepsilon \, dV
\]

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 OVERVIEW

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

\[ \{x_1\} = [T_{12}] \{x_2\} \]

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
FINITE ELEMENT MODELING OVERVIEW

**Boundary Conditions**

Elemental matrices are then assembled into the global master matrices using

\[
[K_n] [x_n] = [F_n]
\]

\[
\begin{bmatrix}
[K_{aa}] & [K_{ab}]
\end{bmatrix}
\begin{bmatrix}
x_a
\end{bmatrix}
= \begin{bmatrix}
F_a
\end{bmatrix}
\]

\[
\begin{bmatrix}
[K_{ba}] & [K_{bb}]
\end{bmatrix}
\begin{bmatrix}
x_b
\end{bmatrix}
= \begin{bmatrix}
F_b
\end{bmatrix}
\]

where 'a' identifies solution variable and 'b' identifies a bounded dof. The equation for solution is

\[
[K_{aa}] [x_a] + [K_{ab}] [x_b] = [F_a]
\]

\[
[K_{aa}] [x_a] = [F_a] - [K_{ab}] [x_b]
\]

where the equation for the reaction loads is

\[
[K_{ba}] [x_a] + [K_{bb}] [x_b] = [F_b]
\]

**Boundary Conditions - Method 1 - Decouple Equations**

Set off-diagonal terms to zero

\[
\begin{bmatrix}
[K_{aa}]
\end{bmatrix}
\begin{bmatrix}
x_a
\end{bmatrix}
= \begin{bmatrix}
{F_a} - [K_{ab}] [x_b]
\end{bmatrix}
\]

\[
[K_{bb}] [x_b] = \begin{bmatrix}
{F_b}
\end{bmatrix}
\]

**Boundary Conditions - Method 2 - Stiff Spring**

Apply stiff spring to bounded dofs (approx zero off-diagonal)

\[
\begin{bmatrix}
[K_{aa}] & [K_{ab}]
\end{bmatrix}
\begin{bmatrix}
x_a
\end{bmatrix}
= \begin{bmatrix}
{F_a}
\end{bmatrix}
\]

\[
\begin{bmatrix}
[K_{ba}] & [K_{bb} + K_{stiff}]
\end{bmatrix}
\begin{bmatrix}
x_b
\end{bmatrix}
= \begin{bmatrix}
[K_{stiff}] [F_b]
\end{bmatrix}
\]

**Boundary Conditions - Method 3 - Partition Equations**

Partition out bounded dofs

\[
[K_{aa}] [x_a] = [F_a]
\]
FINITE ELEMENT MODELING OVERVIEW

Types of Boundary Conditions

FREE - FREE
LEFT END - FREE
RIGHT END - FREE

SIMPLE SUPPORT
LEFT END - X=0, Y=0
RIGHT END - Y=0

CANTILEVER
LEFT END - X=0, Y=0, RZ=0
RIGHT END - FREE

BUILT IN BOTH ENDS
LEFT END - X=0, Y=0, RZ=0
RIGHT END - X=0, Y=0, RZ=0

BUILT IN BOTH ENDS - HALF MODEL
LEFT END - X=0, Y=0, RZ=0
RIGHT END - X=0, RZ=0
FINITE ELEMENT MODELING OVERVIEW

Solution Techniques

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 to extract a few modes from a large set of 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 time steps are required

FINITE ELEMENT MODELING OVERVIEW - THE ELEMENTS

STRUCTURAL ELEMENTS
- TRUSS
- 3D BEAM
- TORSIONAL ROD

CONTINUUM ELEMENTS
- PLATE
- CUBE

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 characteristics
FINITE ELEMENT MODELING OVERVIEW - THE ELEMENTS

**TRUSS**
slender element (length>>area) which supports only tension or compression along its length; essentially a 1D spring

**BEAM**
slender element whose length is much greater than its transverse dimension which supports lateral loads which cause flexural bending

**TORSION**
same as truss but supports torsion

**2D SOLID**
element whose geometry definition lies in a plane and applied loads also lie in the same plane
- plane stress occurs for structures with small thickness compared with its in plane dimension - stress components associated with the out of plane coordinate are zero
- plane strain occurs for structures where the thickness becomes large compared to its in plane dimension - strain component associated with the out of plane coordinate are zero

**PLATES**
element whose geometry lies in the plane with loads acting out of the plane which cause flexural bending and with both in plane dimensions large in comparison to its thickness - two dimensional state of stress exists similar to plane stress except that there is a variation of tension to compression through the thickness

**SHELLS**
element similar in character to a plate but typically used on curved surface and supports both in plane and out of plane loads - numerous formulations exist

**3D SOLID**
element classification that covers all elements - element obeys the strain displacement and stress strain relationships
ELEMENT TYPES

TRUSS

slender element (length>>area) which supports only tension or compression along its length; essentially a 1D spring

\[ \varepsilon = \frac{du}{dx} \]

The truss strain is defined as \( \varepsilon = \frac{du}{dx} \)

The truss stiffness and lumped/consistent mass matrices are

\[
[k] = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} [m_1] = \rho AL \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} [m_c] = \rho AL \begin{bmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{bmatrix}
\]

TORSION

similar to truss but supports torsion

\[
[k_t] = \frac{JG}{L \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}}
\]
ELEMENT TYPES

**BEAM** slender element whose length is much greater that its transverse dimension which supports lateral loads which cause flexural bending

Beam assumptions are
- constant cross section
- cross section small compared to length
- stress and strain vary linearly across section depth

The beam elastic curvature due to lateral loading is satisfied by \( \frac{E I d^4 \nu}{d x^4} = q \)

The longitudinal strain is proportional to the distance from the neutral axis and second derivative of the elastic curvature given as \( \varepsilon = \frac{y d^2 \nu}{dx^2} \)

The stiffness and consistent mass matrices are

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

\[
[m] = \frac{\rho AL}{420} \begin{bmatrix}
156 & -22L & 54 & 13L \\
-22L & 4L^2 & -13L & -3L^2 \\
54 & -13L & 156 & 22L \\
13L & -3L^2 & 22L & 4L^2
\end{bmatrix}
\]
ELEMENT TYPES

BEAM

The full beam stiffness matrix can be assembled using the truss, torsion and two planar beam elements (one on plane and one out of plane)
FINITE ELEMENT MODELING OVERVIEW

A simple spring example is useful to illustrate the finite element process

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 OVERVIEW

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 OVERVIEW

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 OVERVIEW

Three equations can now be written as
\[
k_1 u_1 - k_1 u_2 = f_1 \\
-k_1 u_1 + k_1 u_2 + k_2 u_2 - k_2 u_3 = f_2 \\
-k_2 u_2 + k_2 u_3 = f_3
\]
or in matrix form
\[
\begin{bmatrix}
k_1 & -k_1 \\
-k_1 & k_1 + k_2 & -k_2 \\
-k_2 & k_2 \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
\end{bmatrix}
=
\begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
\end{bmatrix}
\]

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
\[
\begin{bmatrix}
k_1 + k_2 & -k_2 \\
-k_2 & k_2 \\
\end{bmatrix}
\begin{bmatrix}
u_2 \\
u_3 \\
\end{bmatrix}
=
\begin{bmatrix}
0 \\
f_3 \\
\end{bmatrix}
\]
FINITE ELEMENT MODELING CONSIDERATIONS

TRUSS ELEMENTS ASSEMBLED TOGETHER

ELEMENT ASSEMBLY

- the elements can be assembled into one matrix
FINITE ELEMENT MODELING CONSIDERATIONS

SOME COMMON MATERIAL TERMS/DEFINITIONS

- Elastic or Young's Modulus (E) gives a direct indication of stiffness and is the ratio of stress to strain
- Shear Modulus (G) or Modulus of Rigidity is the ration of shear stress to shear strain
- Mass density (r) is the weight density divided by the acceleration due to gravity
- Poisson's Ratio (n) is the ration of lateral strain to extensional strain
- Linear Isotropic material has material constants of elastic modulus, shear modulus, Poisson's ratio and thermal expansion which are all constant properties which are independent of the coordinate system of the element
- Linear Anisotropic material has material constants defined by a 6x6 symmetrical matrix and 6 terms for thermal expansion which are dependent on directional orientation in the material
- Linear Orthotropic material is a special case of Anisotropic material which contains 4 independent constants

FINITE ELEMENT MODELING APPROXIMATIONS

- Approximation of the boundary condition is applied in the finite element model at the node points and not along the surface of the element
- Distributed forces are applied in an approximate sense at the nodes of the model and not actually distributed as in the real world sense
FINITE ELEMENT MODELING CONSIDERATIONS

COMMON MODELING BLUNDEES

• inconsistent set of units (ie, material in psi - model in feet)
• weight density used instead of mass density
• polar moment of inertia (J) used instead of torsional constant (J)
• beam orientation 2-2 and 3-3 switched
• aspect ratio incorrect
• symmetry boundary conditions incorrectly specified
• never use simple model first to assure closed form solution can be obtained or understand the usage of the modeling technique
• parts of the model not hooked together
• misinterpretation of local/global coordinate systems
• a finer mesh never used to assure convergence of the model
• reluctance to read user & theoretical manuals
• assume software should behave a certain way because of familiarity of how a different software package behaves
• ignorance of warning and error messages since they appear to be written in a foreign language
FINITE ELEMENT MODELING – MATLAB SCRIPT FILE

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

This MATLAB file is used to develop the frequencies and
mode shapes for a cantilever beam used for ME22.403 Final Project

The model is defined with 10 beam elements with 2 dof/node (shear/rotary)
The parameters are defined below

| 1 2 3 4 5 6 7 8 9 10 | (node numbers)
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-----------------------</td>
</tr>
</tbody>
</table>
| O (mass at tip of beam)
|

incrementers and counters

% number of beam elements
nel = 10;

% total number of nodes
nodes = nel + 1;

% number of DOF per node
ndfpn = 2;

% total number of DOF in model before BC added
nf = nodes*ndfpn;

% total number of DOF after built-in BC added
n = nf - ndfpn;

% increment for beam element assembly in mass and stiffness matrices
ninc = 2;

physical parameters

% Young's Modulus (psi)
E = 10e6;

% beam dimension (inch)
b = 0.998;

% beam dimension (inch)
h = 0.252;

% area moment of inertia (inch**4)
I = 1/12*b*h^3;

% total length of beam from constraint (inch)
length = 11.75;

% length of individual beam element (inch)
len = length/nel;

% mass density (not weight density)
rho = 0.1/386.4;

% cross sectional area (inch**2)
A = b*h;

% assume accelerometer weights 0.01 lb
m_acc = .01/386.4;
FINITE ELEMENT MODELING – MATLAB SCRIPT FILE (CONTINUED)

% Setup and Assemble Mass and Stiffness Matrices

Knf = zeros(nf,nf); % setup initial matrix space for stiffness
Mnf = zeros(nf,nf); % setup initial matrix space for stiffness

% individual element characteristics

kelement = kbeam(E,I,len); % ==>> EXTERNAL SCRIPT FILE NEEDED !!!
melement = mcbeam(rho,A,len); % ==>> EXTERNAL SCRIPT FILE NEEDED !!!

% assemble individual elements into matrices

[Knf] = assemble(Knf,kelement,[1,2,3,4],nel,ninc); %==>>SCRIPT FILE NEEDED !!!
[Mnf] = assemble(Mnf,melement,[1,2,3,4],nel,ninc); %==>>SCRIPT FILE NEEDED !!!
[Mnf] = assemble(Mnf,m_acc,21,1,1); % add accel mass at dof=21 at tip dof

% constrain system by removing equations associated with boundary

Kn = Knf(3:nf,3:nf); % remove first two equations from stiffness matrix
Mn = Mnf(3:nf,3:nf); % remove first two equations from mass matrix

% perform eigensolution to obtain frequencies and mode shapes

[shapes,freq] = eigen(Kn,Mn); % ==> ==>> EXTERNAL SCRIPT FILE NEEDED !!!
figure(1)
plot(shapes((1:2:(n-1)),(1:1:3))); % plot all three modes
title('Mode Shape - first three modes - 10 elements')
freq(1:1:3)