What Do Engineers and Architects Need to Know about Finite Element Methods?

Course Outline

Finite Element Method is a powerful engineering analysis tool, and has been widely used in engineering since it was introduced in the 1950s. This course presents the basic theory and simple application of Finite Element Method (FEM) along with common FEM terminology. The emphasis of this course is on the fundamental concepts of finite element analysis. A list of major commercial software using FEM is also presented in the course, along with their features and capabilities. This course includes a multiple-choice quiz at the end, which is designed to enhance the understanding of course materials.

Learning Objectives

At the conclusion of this course, the student will:

1. Understand the concept of Finite Element Method (FEM);
2. Recognize the important role played by FEM in today's engineering world;
3. Understand the matrix operations used in FEM;
4. Get familiar with the different types of finite elements;
5. Get familiar with the terminology used in FEM; and
6. Be able to use solve simple structural problems using FEM.

Course Content

In this course, there are many hypertext links to related information. The reader is encouraged to visit the content within these links to get familiar with the linked subject for better understanding of the course content.

What Are Finite Element Method (FEM) and Finite Element Analysis (FEA)?

Many engineering phenomena can be expressed by "governing equations" and "boundary conditions". The governing equations are often in the form of partial differential equations (PDE) or ordinary differential equations (ODE). From mathematical standpoint, Finite Element Method (FEM) is a numerical method used for solving a set of related differential equations such as

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} + f_x = m\ddot{u}_x$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + f_y = m\ddot{u}_y$$

$$\frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + f_z = m\ddot{u}_z$$

From engineering standpoint, Finite Element Method (FEM) is a numerical method for solving a set of related equations by approximating continuous field variables as a set of field variables at discrete points (nodes). For structural problems, the related equations are equilibrium equations, and the field variables are nodal displacements and loads.

Finite element solutions are achieved by either eliminating the differential equation completely (steady state problems), or rendering the PDE into an equivalent ordinary differential equation, which is then solved using standard techniques such as finite differences. Use of Finite Element Method in engineering to analyze physical systems is commonly known as finite element analysis (FEA).

Why Use FEM?

Many engineering problems have complicated geometry and boundary conditions, which makes it impossible to come up with a closed-form solution. Therefore, numerical methods such as Finite Element Method, Finite Strip Method, Finite Difference Method, Finite Volume Method, Boundary Element Method and Hybrid BE-FE Method were introduced to provide approximate solutions to complicated engineering problems through the use of a computer. Among the above-mentioned numerical methods, Finite Element Method is the most powerful and most popular, and often forms the core of many commercially available engineering analysis programs.

Because FEM can be adapted to problems of great complexity and unusual geometry using grid or mesh, it is an extremely powerful tool for solving critical problems in heat transfer, fluid mechanics, electrostatics, and structural and mechanical systems. Furthermore, the availability of fast and inexpensive computers allows engineers and architects to solve daily engineering problems in a straightforward manner using Finite Element Method.

Common Finite Element Terminology

Domain - In mathematics, a domain is the set of independent variables for which a function is defined. In finite element analysis, a domain is a continuous system (region) over which the laws of physics govern. In structural engineering, a domain could be a beam or a complete building frame. In mechanical engineering, a domain could be a piece of machine parts or a thermal field.

Governing Equations - The governing equations for a system are the equations derived from the physics of the system. Many engineering systems can be described by governing equations, which determine the system's characteristics and behaviors.

Boundary Conditions – The boundary conditions of a function are values of the function at the edge of the range of some of its variables. Knowledge of some of the boundary conditions is needed to solve an engineering problem or to find an unknown function.

Element – An element is a portion of the problem domain, and is typically some simple shape like a triangle or quadrilateral in 2D, or tetrahedron or rectangular solid in 3D.

Node - A node is a point in the domain, and is often the vertex of several elements. A node is also called a nodal point.

Mesh (**Grid**) - The elements and nodes, together, form a mesh (grid), which is the central data structure in FEA.

Mesh Generation – Most FEA software automatically generates refined grid or mesh to achieve more accurate results. For large scale or complex finite element analysis, it is often imperative for computers to generate finite element mesh automatically. There are many different algorithms for automatic mesh generation. Click here to see some automatically generated mesh samples.

Linear Finite Element Analysis (FEA) – Linear Finite Element Analysis is based on the following assumptions: (1) Static; (2) Small displacements; (3) Material is linearly elastic.

Nonlinear Finite Element Analysis – Nonlinear Finite Element Analysis considers material nonlinearity and/or geometric nonlinearity of an engineering system. Geometric nonlinear analysis is also called large deformation analysis.

GUI – GUI stands for graphical user interface, which provides a visual tool to build a finite element model for a domain with complex geometry and boundary conditions.
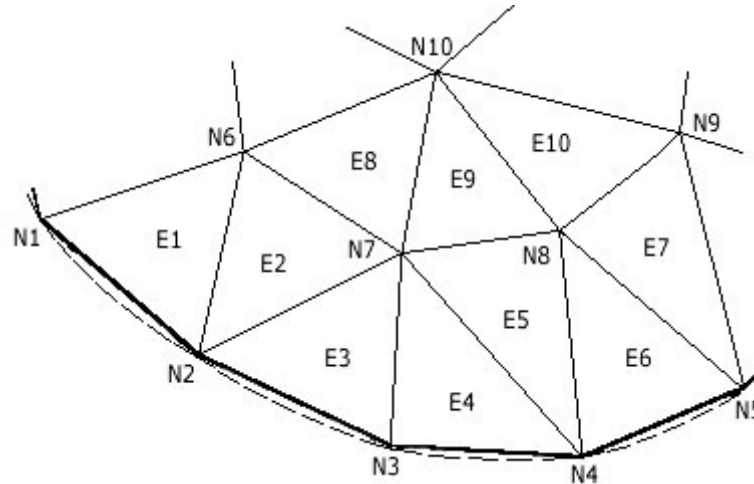

Figure 1 – Triangulation of a Surface Domain

Figure 1 shows an example of discretization of a surface domain using triangular elements. The actual boundary of the domain is shown in dashed lines. E1 and N1 represent Element 1 and Node 1, respectively.

What Are Matrices and Vectors?

To understand Finite Element Methods, one needs to understand the mathematical terms "matrix" and "vector". In mathematics, a matrix is a rectangular table of data. A matrix with m rows and n columns is said to be an m-by-n matrix. For example,

$$\begin{pmatrix} 4 & 3 & 2 & 5 \\ 3 & 2 & 1 & 6 \\ 2 & 1 & 1 & -1 \end{pmatrix}$$

is a 3-by-4 matrix. A matrix with m rows and m columns is called an m-by-m square matrix. The simplest matrix is called identity matrix, which is a square matrix with a value of 1 along its diagonal entries and a value of 0 for all other entries. The following matrix is a 3-by-3 identity matrix:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

In the context of Finite Element Methods, vectors are commonly denoted by matrices with dimensions $n \times 1$ (*column vector*) or $1 \times n$ (*row vector*). For example, the matrix below is also called a column vector:

$$\begin{Bmatrix} 8 \\ 4 \\ 2 \end{Bmatrix}$$

Matrices and vectors offer concise mathematical expression for a set of simultaneous algebraic equations. For example, the following set of simultaneous algebraic equations

$$\begin{aligned} 4x + 3y + 2z &= 8 \\ 3x + 2y + 1z &= 4 \\ 2x + 1y + 1z &= 2 \end{aligned} \tag{1}$$

can be written as a matrix equation:

$$\begin{pmatrix} 4 & 3 & 2 \\ 3 & 2 & 1 \\ 2 & 1 & 1 \end{pmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} 8 \\ 4 \\ 2 \end{Bmatrix}$$

Furthermore, if we use the following notations to represent each matrix and vector:

$$[K] = \begin{pmatrix} 4 & 3 & 2 \\ 3 & 2 & 1 \\ 2 & 1 & 1 \end{pmatrix} \qquad \{u\} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \qquad \{f\} = \begin{Bmatrix} 8 \\ 4 \\ 2 \end{Bmatrix}$$

the above matrix equation can be re-written as:

$[K]\{u\} = \{f\}$

or

Ku=**f** $\tag{2}$

The boldface letters **K** or **u** in the above matrix Equation (2) represent matrices or vectors in the context of FEM, and provide the most concise mathematical expression. The boldface letter **I** usually stands for the identity matrix.

Matrices in the mathematical sense are useful for keeping track of the coefficients of linear expressions such as linear transformations and systems of linear equations. The field of mathematics that studies matrices is called matrix theory, a branch of linear algebra. We can do addition, multiplication and many different operations on matrices. The matrix operation similar to number inversion is called matrix inverse.

How to Solve a Matrix Equation

In a typical engineering problem, matrix **K** and vector **f** are usually known while the vector **u** is an unknown. To solve for the unknown, one needs to perform the following operation:

u =K^{-1}f

in which **K**$^{-1}$ is the [matrix inverse](#) of **K**. The product of a square matrix and its inverse yields an [identity matrix](#) of the same dimensions. Math software such as [MathCad](#) can be used to easily find a matrix inverse. MathCad provides a complete, integrated environment for performing, documenting and communicating technical calculations. Within a MathCad worksheet one can perform "live" numeric or symbolic calculations, add graphics and animations, and annotate and format text. When one needs to change a variable, MathCad updates the results, formulae and graphs automatically and instantly. All the MathCad calculations contained herein are embedded in the MS Word document as an object before the document is converted to PDF format. For example, the above matrix Equation (2) can be solved by MathCad as follows:

$$\mathbf{K} := \begin{pmatrix} 4 & 3 & 2 \\ 3 & 2 & 1 \\ 2 & 1 & 1 \end{pmatrix} \qquad \mathbf{f} := \begin{pmatrix} 8 \\ 4 \\ 2 \end{pmatrix}$$

$$\mathbf{K}^{-1} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & 0 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

Check matrix inverse: $\qquad \mathbf{I} := \mathbf{K} \cdot \mathbf{K}^{-1} \qquad \mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad$ OK

Solve for unknowns: $\qquad \mathbf{u} := \mathbf{K}^{-1} \cdot \mathbf{f}$

$$\mathbf{u} = \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix} \qquad \begin{pmatrix} x \\ y \\ z \end{pmatrix} := \mathbf{u} \qquad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -2 \\ 4 \\ 2 \end{pmatrix}$$

The notation ":=" in the above MathCad calculation is called definition symbol, which assigns a value to a variable or values to a matrix.

There are several algorithms to solve a system of linear equations numerically on a computer. One of the algorithms is called the [Gauss Elimination Method](#), which consists of two steps in its solution process:

1. Forward Elimination of Unknowns: In this step, the unknown is eliminated in each equation starting with the first equation. This way, the equations are "reduced" to one equation and one unknown in the last equation.

2. Back Substitution: By eliminating all but one unknown in the last equation, one can solve for the unknown variables from last equation up. In this step, starting from the last equation, each of the unknowns is found by back-substitution.

Fundamental Concepts of FEM

From the mathematics standpoint, many engineering phenomena can be expressed by "governing equations" and "boundary conditions" such as

$$G(\phi) + f = 0 \tag{3}$$
$$B(\phi) + h = 0 \tag{4}$$

The governing equation (3) is often in the form of a differential equation, which can be converted into a matrix equation using FEM:

$$\mathbf{Ku}=\mathbf{f} \tag{5}$$

The solution to the above matrix equation is

$$\mathbf{u} =\mathbf{K}^{-1}\mathbf{f} \tag{6}$$

in which **K** represents the property, **u** the behavior, and **f** the action. The following table summarizes the physical significance of these variables for different types of engineering problems:

Table 1 – Summary of Property, Behavior and Action in FEM

Problem Type	Property, K	Behavior, u	Action, f	Applicable Field
Elastic Problem	Stiffness	Displacement	Force	Structural Engineering
Thermal Problem	Conductivity	Temperature	Heat Source	Mechanical Engineering
Electrostatic Problem	Dialectric Permittivity	Electric Potential	Charge	Electrical Engineering
Fluid Problem	Viscosity	Velocity	Body Force	Civil & Mechanical Engineering

In order to use FEM to analyze an engineering problem, it is necessary to divide the entire domain (structure) into a number of small, simple elements. A field variable is interpolated by a polynomial over an element. The adjacent elements have to share the same degrees of freedom at connecting nodes to achieve compatibility. By connecting elements together, the field variable becomes interpolated over the entire domain (structure) in piecewise fashion, which results in a set of simultaneous algebraic equations at nodes. By solving the algebraic equations, one can obtain primary unknown field variables at nodes.

One may visualize FEM as follows:

1. First break a structure (domain) into several smaller pieces (elements).
2. Derive the governing equations in matrix form for each element based on the physics of the problem.
3. Stitch elements together at nodes (nodal points).
4. Assemble a set of simultaneous algebraic equations in matrix form for the entire structure (a global equation).
5. Solve for primary unknown variables at nodes by using the known boundary conditions.
6. Solve for other unknown variables at the element level.

What Types of Engineering Problems Can Be Solved by FEM?

Finite element methods are employed in a wide variety of engineering disciplines, and have been widely used in solving problems related to:

1. Stress analysis
2. Electrostatics
3. Magnetic fields
4. Fluid flow
5. Dynamics
6. Heat-transfer

What Is the Procedure of FEA?

Various methods such as energy methods or virtual work principles can be used to derive Finite Element equations. The typical procedure for Finite Element Analysis (FEA) is as follows:

1. Preprocessing: user builds a Finite Element (FE) model and defines the boundary conditions and loads for an engineering problem.
2. Processing: computer conducts numerical analysis and prints out results for the model.
3. Postprocessing: user analyzes and interprets the results of FEA.

The followings are the typical steps in performing Finite Element Analysis:

Step 1 – Select Element Type: Element type affects the ease of modeling and the accuracy of solution.

Step 2 – Discretize: The problem domain (structure) is divided into a collection of simple shapes, or elements.

Step 3 – Derive Governing Equations for Each Element: Element matrix equation can be developed based on the physics of the problem using methods such as energy method or virtual work principle.

Step 4 – Assemble Global Governing Equations: The element matrices are assembled into a global matrix equation that models the properties of the entire domain (structure).

Step 5 – Apply Boundary Conditions: Boundary conditions reflect the known values for certain primary unknowns. Imposing the boundary conditions modifies the global matrix equations.

Step 6 – Solve for Primary Unknowns: The modified global matrix equations are solved for the primary unknowns at the nodes.

Step 7 – Calculate Derived Variables: Other field variables can be calculated using the nodal values of the primary variables.

Step 8 – Check and Interpret the Results: This step is probably the most critical. While checking and interpreting the results, one may find results unreasonable, thus prompting revisions and re-analysis.

To build a realistic FE model for a specific problem, one needs to select appropriate type of finite elements to start with.
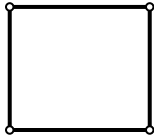
What Types of Finite Elements Are Available?

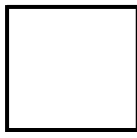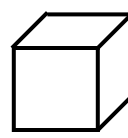Common Finite Elements for structural problems are

Truss Element   Beam Element

Shell Element   Plate Element   Solid Element

Within the above elements, one can specify linear or quadratic type of elements depending on the desired accuracy of numerical approximation through linear polynomials or quadratic polynomials. Linear elements are used in most engineering applications for simplicity. Commercial engineering analysis software may contain hundreds of different finite element types for various engineering applications.

The difference between a truss element and a beam element is in the degrees-of-freedom at the nodes. For a 3-D truss element, there are only three translational degrees-of-freedom at each node. A 3-D beam element contains three rotational degrees-of-freedom in addition to three translational degrees-of-freedom at each node.

Table 2 – Degrees-of-freedom for Some Common Structural Element Types

Element	Degrees of Freedom
3-D Truss Element	Translation in X, Y, Z
3-D Beam Element	Translation in X, Y, Z; Rotation in X, Y, Z
2-D Truss Element	Translation in X, Y
2-D Beam Element	Translation in X, Y; Rotation in Z

What Is the Simplest Finite Element Model?

A 1-D spring problem can be transformed into the simplest Finite Element model, which is very helpful in understanding the basic concept of FEM.

Ku=f

K – Spring Stiffness
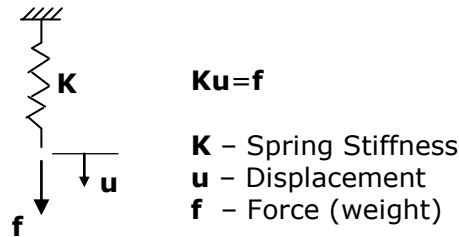**u** – Displacement
f – Force (weight)

Figure 2 – Typical Spring Problem

If the spring stiffness is equal to 100 pounds/inch and the weight is equal to 200 pounds, we can easily solve this one-element problem by hand as follows:

K = 100   **f** = 200

100**u** = 200   ⟹   **u** = 200/100 = 2 inches

In the above simple spring problem, there is only one entry in the stiffness matrix **K**. In the real world, the stiffness matrix could contain thousands of entries for a multistory building model.

A Real World Problem

Let us take a look at the following 2-D beam problem. This beam is subject to a concentrated load P_2 at the midspan. The boundary conditions are: the left end of the beam is fixed, and the right end has a roller-type support.
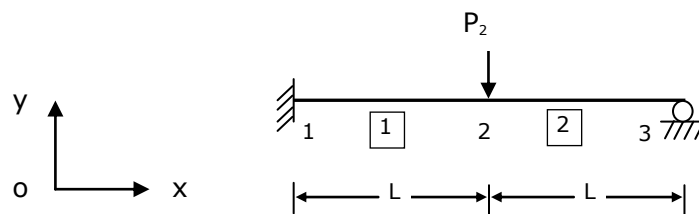
Figure 3 – Finite Element Model of a 2-D Beam

If we want to find out the vertical displacement at the location of the concentrated load, one way is to introduce a node at the midspan of the beam. So for this simple beam problem, we have a finite element model with 3 nodes and 2 beam elements (element numbers are shown in a box).

There are three degrees-of-freedom at each node of a 2-D beam element (see Table 2 above and Figure 4 below).
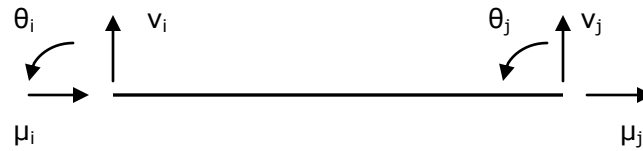
Figure 4 – Degrees-of-Freedom of a Typical 2-D Beam Element

The typical stiffness matrix, displacement vector and force vector of a 2-D beam element are as follows (click here for details of beam stiffness matrix development):

$$\mathbf{K} := \begin{pmatrix} \dfrac{EA}{L^1} & 0 & 0 & \dfrac{-EA}{L^1} & 0 & 0 \\[2mm] 0 & \dfrac{12EI}{L^3} & \dfrac{6EI}{L^2} & 0 & \dfrac{-12EI}{L^3} & \dfrac{6EI}{L^2} \\[2mm] 0 & \dfrac{6EI}{L^2} & \dfrac{4EI}{L^1} & 0 & \dfrac{-6EI}{L^2} & \dfrac{2EI}{L^1} \\[2mm] \dfrac{-EA}{L^1} & 0 & 0 & \dfrac{EA}{L^1} & 0 & 0 \\[2mm] 0 & \dfrac{-12EI}{L^3} & \dfrac{-6EI}{L^2} & 0 & \dfrac{12EI}{L^3} & \dfrac{-6EI}{L^2} \\[2mm] 0 & \dfrac{6EI}{L^2} & \dfrac{2EI}{L^1} & 0 & \dfrac{-6EI}{L^2} & \dfrac{4EI}{L^1} \end{pmatrix} \qquad \mathbf{u} := \begin{pmatrix} \mu_i \\ v_i \\ \theta_i \\ \mu_j \\ v_j \\ \theta_j \end{pmatrix} \qquad \mathbf{f} := \begin{pmatrix} a_i \\ p_i \\ m_i \\ a_j \\ p_j \\ m_j \end{pmatrix}$$

in which E=Elastic modulus, A=Area, I=Moment of Inertia, L=Length of the beam, μ=Longitudinal displacement, υ=Transverse displacement, θ=Rotation, a=Axial force, p=Shear force, m=Internal moment, **u**=Displacement vector, and **f**=Force vector. The field variables with subscripts "i" and "j" represent the values at the left and right end of the element, respectively. If the axial deformations are ignored, the typical stiffness matrix, displacement vector and force vector of the above beam element can be simplified as follows:

$$\mathbf{K} := \begin{pmatrix} \dfrac{12EI}{L^3} & \dfrac{6EI}{L^2} & \dfrac{-12EI}{L^3} & \dfrac{6EI}{L^2} \\[2mm] \dfrac{6EI}{L^2} & \dfrac{4EI}{L^1} & \dfrac{-6EI}{L^2} & \dfrac{2EI}{L^1} \\[2mm] \dfrac{-12EI}{L^3} & \dfrac{-6EI}{L^2} & \dfrac{12EI}{L^3} & \dfrac{-6EI}{L^2} \\[2mm] \dfrac{6EI}{L^2} & \dfrac{2EI}{L^1} & \dfrac{-6EI}{L^2} & \dfrac{4EI}{L^1} \end{pmatrix} \qquad \mathbf{u} := \begin{pmatrix} v_i \\ \theta_i \\ v_j \\ \theta_j \end{pmatrix} \qquad \mathbf{f} := \begin{pmatrix} p_i \\ m_i \\ p_j \\ m_j \end{pmatrix}$$

Considering the fixed boundary conditions $\upsilon_1=0$ (no vertical displacement at Node 1) and $\theta_1=0$ (no rotational displacement at Node 1), we can further simplify the stiffness matrix, displacement vector and force vector for beam Element 1 as follows:

$$\mathbf{K}_1 := \begin{pmatrix} \dfrac{12EI}{L^3} & \dfrac{-6EI}{L^2} \\ \dfrac{-6EI}{L^2} & \dfrac{4EI}{L^1} \end{pmatrix} \qquad \mathbf{u}_1 := \begin{pmatrix} \nu_2 \\ \theta_2 \end{pmatrix} \qquad \mathbf{f}_1 := \begin{pmatrix} p_{12} \\ m_{12} \end{pmatrix}$$

In the above expression, p_{12} and m_{12} represent shear force and internal moment at the second node (j node, right end) of Element 1, respectively. Similarly for beam Element 2, the stiffness matrix, displacement vector and force vector can be written as follows:

$$\mathbf{K}_2 := \begin{pmatrix} \dfrac{12EI}{L^3} & \dfrac{6EI}{L^2} & \dfrac{6EI}{L^2} \\ \dfrac{6EI}{L^2} & \dfrac{4EI}{L^1} & \dfrac{2EI}{L^1} \\ \dfrac{6EI}{L^2} & \dfrac{2EI}{L^1} & \dfrac{4EI}{L^1} \end{pmatrix} \qquad \mathbf{u}_2 := \begin{pmatrix} \nu_2 \\ \theta_2 \\ \theta_3 \end{pmatrix} \qquad \mathbf{f}_2 := \begin{pmatrix} p_{21} \\ m_{21} \\ m_{22} \end{pmatrix}$$

in which p_{21} and m_{21} represent shear force and internal moment at the first node (i node, left end) of Element 2, respectively.

Now that we have the stiffness matrix, displacement vector and force vector for each element, we need to join the elements together using the corresponding degree-of-freedom at the adjacent node. This step is called stiffness matrix assembly for structural type of problems. In this example, we use the method of direct formulation to form the global matrix equation. To assemble the element matrix equations into a global matrix equation, we first need to expand the element stiffness matrix, displacement vector and force vector so that their sizes correspond to the size of the global stiffness matrix, displacement vector and force vector, and then add the two element matrix equations together. Here are the expanded element stiffness matrix, displacement vector and force vector for Element 1:

$$\mathbf{K}_1 := \begin{pmatrix} \dfrac{12EI}{L^3} & \dfrac{-6EI}{L^2} & 0 \\ \dfrac{-6EI}{L^2} & \dfrac{4EI}{L^1} & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{u}_1 := \begin{pmatrix} \nu_2 \\ \theta_2 \\ \theta_3 \end{pmatrix} \qquad \mathbf{f}_1 := \begin{pmatrix} p_{12} \\ m_{12} \\ 0 \end{pmatrix}$$

The element stiffness matrix, displacement vector and force vector for Element 2 happen to be the same size of the global stiffness matrix, displacement vector and force vector,

respectively. To assemble the two element matrix equations into the global matrix equation, we just perform the following matrix additions:

K = **K**$_1$ + **K**$_2$

f= **f**$_1$ + **f**$_2$

Hence, the global stiffness matrix, displacement vector and force vector for this 2-D beam problem can be written as:

$$
\mathbf{K} := \begin{pmatrix} \dfrac{2(\,EI}{L^3} & 0 & \dfrac{6EI}{L^2} \\[2mm] 0 & \dfrac{8EI}{L^1} & \dfrac{2EI}{L^1} \\[2mm] \dfrac{6EI}{L^2} & \dfrac{2EI}{L^1} & \dfrac{4EI}{L^1} \end{pmatrix}
\qquad
\mathbf{u} := \begin{pmatrix} v_2 \\ \theta_2 \\ \theta_3 \end{pmatrix}
\qquad
\mathbf{f} := \begin{pmatrix} P_2 \\ M_2 \\ M_3 \end{pmatrix}
$$

in which $P_2 = p_{12} + p_{21}$, $M_2 = m_{12} + m_{21}$, and $M_3 = m_{22}$. These represent the equilibrium relationship between the external and internal forces at the nodes. P_2 is the external force at Node 2. M_2 is the external moment at Node 2. M_3 is the external moment at Node 3.

In concise matrix form, the finite element formulation for the above structural problem can be written as:

Ku=**f**   (7)

Assuming one vertical downward concentrated load (P_2=-10000 pounds), zero moment at Node 2 (M_2=0), zero moment at Node 3 (M_3=0), L=100 inches, E=30,000,000 psi, and I=100 in^4, we can solve the above Equation (7) using MathCad as follows:

$$L := 100 \qquad E := 30000000 \qquad I := 100 \qquad P_2 := -10000 \qquad EI := E \cdot I$$

Assembled Stiffness Matrix:

Known Boundary Conditions:

$$\mathbf{K} := \begin{pmatrix} \dfrac{24 \cdot EI}{L^3} & 0 & \dfrac{6 \cdot EI}{L^2} \\[2ex] 0 & \dfrac{8 \cdot EI}{L^1} & \dfrac{2 \cdot EI}{L^1} \\[2ex] \dfrac{6 \cdot EI}{L^2} & \dfrac{2 \cdot EI}{L^1} & \dfrac{4 \cdot EI}{L^1} \end{pmatrix}$$

$$\begin{pmatrix} P_2 \\ M_2 \\ M_3 \end{pmatrix} := \begin{pmatrix} -10000 \\ 0 \\ 0 \end{pmatrix} \qquad \begin{pmatrix} v_1 \\ \theta_1 \\ v_3 \end{pmatrix} := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{K} = \begin{pmatrix} 7.2 \times 10^4 & 0 & 1.8 \times 10^6 \\ 0 & 2.4 \times 10^8 & 6 \times 10^7 \\ 1.8 \times 10^6 & 6 \times 10^7 & 1.2 \times 10^8 \end{pmatrix}$$

$$\mathbf{f} := \begin{pmatrix} P_2 \\ M_2 \\ M_3 \end{pmatrix} \qquad \mathbf{f} = \begin{pmatrix} -1 \times 10^4 \\ 0 \\ 0 \end{pmatrix}$$

Stiffness Matrix Inverse:

$$\mathbf{K}^{-1} = \begin{pmatrix} 2.431 \times 10^{-5} & 1.042 \times 10^{-7} & -4.167 \times 10^{-7} \\ 1.042 \times 10^{-7} & 5.208 \times 10^{-9} & -4.167 \times 10^{-9} \\ -4.167 \times 10^{-7} & -4.167 \times 10^{-9} & 1.667 \times 10^{-8} \end{pmatrix}$$

Solve for unknown displacement vector:

$$\mathbf{u} := \mathbf{K}^{-1} \mathbf{f}$$

$$\mathbf{u} = \begin{pmatrix} -0.243 \\ -1.042 \times 10^{-3} \\ 4.167 \times 10^{-3} \end{pmatrix} \qquad \begin{pmatrix} v_2 \\ \theta_2 \\ \theta_3 \end{pmatrix} := \mathbf{u} \quad \text{or} \quad \begin{pmatrix} v_2 \\ \theta_2 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} -0.243 \\ -1.042 \times 10^{-3} \\ 4.167 \times 10^{-3} \end{pmatrix}$$

Therefore, the vertical displacement at Node 2 is equal to 0.243 inches (downward) under a 10000-pound concentrated load, and the rotation at Node 3 is equal to 0.004167 radians. Notes: (a) Positive rotation means counterclockwise rotation using the standard sign convention (right-hand rule). (b) The unit of angular rotation in FEA is normally in radians, not degrees.

Based on the above displacements, we can also solve for the support reactions using the element matrix equation based on the complete element stiffness matrix, displacement vector and force vector (see page 10). The detailed MathCad calculations for the support reactions are shown on the next page.

Element Stiffness Matrix:

$$\mathbf{K} := \begin{pmatrix} \dfrac{12 \cdot EI}{L^3} & \dfrac{6 \cdot EI}{L^2} & \dfrac{-12 \cdot EI}{L^3} & \dfrac{6 \cdot EI}{L^2} \\[2mm] \dfrac{6 \cdot EI}{L^2} & \dfrac{4 \cdot EI}{L^1} & \dfrac{-6 \cdot EI}{L^2} & \dfrac{2 \cdot EI}{L^1} \\[2mm] \dfrac{-12 \cdot EI}{L^3} & \dfrac{-6 \cdot EI}{L^2} & \dfrac{12 \cdot EI}{L^3} & \dfrac{-6 \cdot EI}{L^2} \\[2mm] \dfrac{6 \cdot EI}{L^2} & \dfrac{2 \cdot EI}{L^1} & \dfrac{-6 \cdot EI}{L^2} & \dfrac{4 \cdot EI}{L^1} \end{pmatrix}$$

$$\mathbf{K} = \begin{pmatrix} 3.6 \times 10^4 & 1.8 \times 10^6 & -3.6 \times 10^4 & 1.8 \times 10^6 \\ 1.8 \times 10^6 & 1.2 \times 10^8 & -1.8 \times 10^6 & 6 \times 10^7 \\ -3.6 \times 10^4 & -1.8 \times 10^6 & 3.6 \times 10^4 & -1.8 \times 10^6 \\ 1.8 \times 10^6 & 6 \times 10^7 & -1.8 \times 10^6 & 1.2 \times 10^8 \end{pmatrix}$$

Displacement vector for Element 1:

$$\begin{pmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -0.243 \\ -1.042 \times 10^{-3} \end{pmatrix} \qquad \mathbf{u}_1 := \begin{pmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{pmatrix} \qquad \mathbf{f}_1 := \mathbf{K} \cdot \mathbf{u}_1$$

Force vector for Element 1:

$$\mathbf{f}_1 = \begin{pmatrix} 6.875 \times 10^3 \\ 3.75 \times 10^5 \\ -6.875 \times 10^3 \\ 3.125 \times 10^5 \end{pmatrix} \qquad \begin{pmatrix} p_{11} \\ m_{11} \\ p_{12} \\ m_{12} \end{pmatrix} := \mathbf{f}_1 \qquad \begin{pmatrix} p_{11} \\ m_{11} \\ p_{12} \\ m_{12} \end{pmatrix} = \begin{pmatrix} 6.875 \times 10^3 \\ 3.75 \times 10^5 \\ -6.875 \times 10^3 \\ 3.125 \times 10^5 \end{pmatrix}$$

Displacement vector for Element 2:

$$\begin{pmatrix} v_2 \\ \theta_2 \\ v_3 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} -0.243 \\ -1.042 \times 10^{-3} \\ 0 \\ 4.167 \times 10^{-3} \end{pmatrix} \qquad \mathbf{u}_2 := \begin{pmatrix} v_2 \\ \theta_2 \\ v_3 \\ \theta_3 \end{pmatrix} \qquad \mathbf{f}_2 := \mathbf{K} \cdot \mathbf{u}_2$$

Force vector for Element 2:

$$\mathbf{f}_2 = \begin{pmatrix} -3.125 \times 10^3 \\ -3.125 \times 10^5 \\ 3.125 \times 10^3 \\ -5.116 \times 10^{-13} \end{pmatrix} \qquad \begin{pmatrix} p_{21} \\ m_{21} \\ p_{22} \\ m_{22} \end{pmatrix} := \mathbf{f}_2 \qquad \begin{pmatrix} p_{21} \\ m_{21} \\ p_{22} \\ m_{22} \end{pmatrix} = \begin{pmatrix} -3.125 \times 10^3 \\ -3.125 \times 10^5 \\ 3.125 \times 10^3 \\ -5.116 \times 10^{-13} \end{pmatrix}$$

Based on the above calculations, the support reactions at Nodes 1 and 3 are as follows:
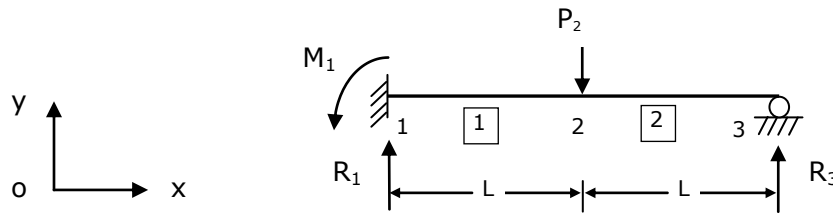

Figure 5 – 2-D Beam Load Diagram

$R_1 = p_{11} = 6875$ pounds $M_1 = m_{11} = 375000$ in-pounds
$R_3 = p_{22} = 3125$ pounds $M_3 = m_{22} = -5.116 \times 10^{-13}$ in-pounds

To verify the correctness of the above calculations, one may add up all the vertical reactions to see if the sum is equal to the applied load. In this case, sum of the vertical reactions = $R_1 + R_3$ = 6875 + 3125 = 10000 pounds, which is the same as the applied concentrated load P_2.

Based on the above calculations, M_3 (the external moment at Node 3) has a very small value, but should be equal to zero. This is caused by the numerical error due to the limited number of digits used in the MathCad calculations.

Boundary conditions on a structure may occur as either applied displacements or external forces. In the above example, there are three known displacements (if the axial deformation is ignored) and three known external forces, including two zero moments at Nodes 2 and 3.

FEM for Dynamic Problems

The typical Finite Element equation **Ku**=**f** is applicable to static problems only. For dynamic problems, the finite element equation of motion takes the form of

$$\mathbf{M\ddot{u} + C\dot{u} + Ku = f} \tag{8}$$

in which **M** is the mass matrix, **C** the damping matrix, **K** the stiffness matrix, **ü** the acceleration vector, **u̇** the velocity vector, **u** the displacement vector, and **f** the force vector. Equation (8) basically represents an equilibrium equation for any physical system. For a static problem, Equation (8) becomes Equation (2) (see page 3) since there is no acceleration (**ü**=0) and no velocity (**u̇**=0).

Dynamic problems or forced vibration problems are time-dependent problems, which means that we also need to discretize the time domain to achieve Finite Element solutions. If there is no damping and no external force, Equation (8) becomes

$$\mathbf{M\ddot{u} + Ku = 0} \tag{9}$$

Equation (9) represents a free dynamic vibration problem, which is an eigenvalue problem from mathematical standpoint. The solution to the eigenvalue problem is the natural frequency of the structure.

What Are the Advantages of FEM?

1. FEM can readily handle very complex geometry and boundary conditions (restraints).
2. FEM can handle complex loading, from point loads to uniform loads to dynamic loads.
3. FEM can solve various engineering problems, from solid mechanics to dynamics to heat transfer, and to electrostatic fields.

With the widespread use of PCs in engineering offices, FEM can provide an engineering solution at a few clicks of a mouse (for simple problems, of course).

What Are the Disadvantages of FEM?

1. The FE solutions are often approximate. The more refined the grid (mesh), the more accurate the FE solution.
2. The FE solution may contain "inherent" computational errors as a result of error accumulation during the numerical computation.
3. The FE solution may contain "fatal" errors as a result of incorrect modeling of structures, loads or boundary conditions.

To minimize the "inherent" computational errors and to eliminate "fatal" errors as a result of incorrect modeling, engineers and architects need to understand the fundamentals of Finite Element Methods, and to be able to model a system and its boundary conditions correctly.

A Sampling of Commercial FE Software

Hundreds of commercial FEM programs are available in the world. The following is just a sampling of commonly used FEM software packages:

Software Name	Features
ANSYS	The **ANSYS** finite element analysis software package is a flexible, robust design analysis and optimization package. ANSYS features file compatibility throughout its family of products and across all platforms. The multiphysics nature of ANSYS allows the same model to be used for a variety of coupled-field applications, such as thermal-structural, magneto-structural, and electrical-magnetic-flow-thermal. In addition to solution generation tools, comprehensive analysis and graphics tools are also included, which allow the user to effectively visually model various types of systems.
GT STRUDL	**GT STRUDL** is a fully integrated and database-driven software system for general finite element analysis and comprehensive structural engineering design. GT STRUDL's nine functional areas that operate seamlessly with one another combined with its amazingly fast computational speed provide virtually unlimited power and flexibility for projects of any size or complexity.
RISA	**RISA** is a general-purpose structural engineering software for Windows. It consists of several components: RISAFloor - Structural Engineering Software for Buildings to manage loads, design members, generate plans, and serve RISA-3D lateral system design information; RISA-3D - Easy 3D Structural Engineering Analysis & Design Software for general frame, truss, and plate/shell structures; and RISA-2D - Simple 2D Structural Engineering Software for Analysis & Design of continuous beams, 2D trusses, and 2D frames.

SAP2000	**SAP2000** is an integrated, productive and practical general purpose structural program on the market today. It features a sophisticated, intuitive and versatile user interface for engineers working on transportation, industrial, public works, sports, and other facilities. Structures can be analyzed for the effects of staged construction, including the adding and removing of temporary shoring. Nonlinear effects can be considered such as large deflections, yielding, and gap opening and closing. Arbitrary loading sequences may be applied. Multiple construction sequences may be analyzed in the same model and compared or enveloped. Dynamic, buckling, and other types of analyses can be performed at the end of any construction sequence, so that the behavior of a structure can be examined before and after a retrofit.
STAAD PRO	**STAAD.*Pro*,** a structural engineering software product for 3D model generation, analysis and multi-material design, is the flagship product of former Research Engineers International, now a part of Bentley Systems, Incorporated. It has an intuitive, user-friendly GUI, visualization tools, powerful analysis and design facilities and seamless integration to several other modeling and design software products. STAAD.*Pro* can be used for static or dynamic analysis of bridges, containment structures, embedded structures (tunnels and culverts), pipe racks, steel, concrete, aluminum or timber buildings, transmission towers, stadiums or any other simple or complex structure.

Summary

Finite Element Method is a very powerful computational tool for engineers and architects to analyze complex electrostatic, structural and mechanical systems. A basic understanding of FEM helps engineers and architects solve the problems that are intractable using analytic or mechanical methods.