FINITE ELEMENTS AND ELASTICITY PROBLEMS

M AMEEN
Department of Civil Engineering
National Institute of Technology Calicut

1. APPROXIMATE METHODS

Exact closed-form solutions are seldom available to most of the engineering problems and resort to approximate numerical techniques is often warranted. Presence of irregularly shaped boundaries, odd-shaped holes and stiffeners, complex boundary conditions and nonlinearities introduced due to the material constitutive relations and/or strain-displacement relations ceasing to obey simple linear relationships are some of the difficulties faced by an engineering analyst. Analytical solutions to such problems rarely exist. However these are the types of problems the present day engineer is called upon to solve.

Of course, an analyst has some choice. Introducing a good number of simplifying assumptions and there by reducing the complexities of the problem to a manageable extent is one way out. However, such a solution may be far from the actual solution one is interested in. A more meaningful alternative is to make use of one of the approximate methods available and solve the problem at hand without additional simplifying assumptions.

Unfortunately, the terminologies used for the analytical and numerical methods are ambiguous and to some extent contradictory. The so-called "exact methods" are seldom exact! The "exact methods" refer to analytical methods that make use of the mathematical theories of differential and integral equations to solve problems which are simple in nature. That the "exact method" can be used to solve only trivial problems is a well-known fact. On the other hand, numerical methods that are suitable to obtain highly accurate solutions of practical engineering problems are classified under "approximate methods"!

Over the years several approximate numerical techniques have evolved. The *finite difference method* is one such method [1] which is in use even today in some of the branches of continuum mechanics such as fluid flow, heat conduction problems etc. The governing differential equations of the problem are approximated by their finite difference equivalents in a point-wise fashion, leading to a system of algebraic equations whose solution represents an approximate solution of the problem. Finite difference method has been used to solve problems of moderate complexity. But presence of irregular boundary geometry or unusual boundary conditions is difficult to handle by this method.

The *finite element method* is the most popular numerical tool available to the present day engineer [2]. It is quite versatile and is being used to solve almost any type of engineering problem. The entire domain of the problem is discretised into a large number of subregions of different sizes, shapes and orientations called *finite elements.* These finite elements are assumed to be connected to each other only at a finite number of points called *nodes*. The field variables are approximated by piece-wise continuous functions. The nodal values of the field variables are treated as the unknowns and these are determined by making use of some related variational principles such as the principle of minimum potential energy in elastostatics and the Hamilton's principle in structural dynamics. Methods based on *weighted residual technique* [3] are also available which are ideally suited for problems that do not have a variational formulation. The governing differential equation is directly inserted into the weighted residual statement, which is integrated by parts a few times. The resulting set of integral equations are solved using numerical integration schemes to obtain an approximate solution.

In addition to the above methods, which are classified under the so-called *domain methods*, emerged a new method of more recent origin called the *boundary element method* [4]. It is

based on the boundary integral equation formulation of the problem and, in contrast to the domain methods, needs only a boundary discretisation. As a result, improved numerical accuracy together with a lesser requirement on computational resources is possible using the boundary element method. The boundary element method is also being recognised as a truly *meshless method* and is increasingly used to model problems that may warrant a continuous mesh modification/refinement.

Consider, as an example, how the above three methods can be used to represent the geometrical shape of a machine component having an oval-shaped hole. For this problem, it is important to know the stress distribution, especially in the vicinity of the hole, due to some applied loading along the edges of the plate. A uniform finite difference mesh as shown in Fig. 1*a* can be used to cover the region of the plate reasonably well, but the curved boundary shape must be approximated by a series of horizontal and vertical lines. The finite element discretisation of the problem is shown in Fig. 1*b*. Simple triangular and quadrilateral elements are made use of. The finite elements are capable of representing the geometry very well and require lesser number of nodes. Moreover, it is possible to use finer discretisation over areas of steep stress variation, thereby improving the accuracy of the solution. Fig 1*c* shows the boundary element discretisation of the problem using linear boundary elements and results in a much lesser number of nodes that are situated only along the boundary of the region (both the external and internal boundary lines).

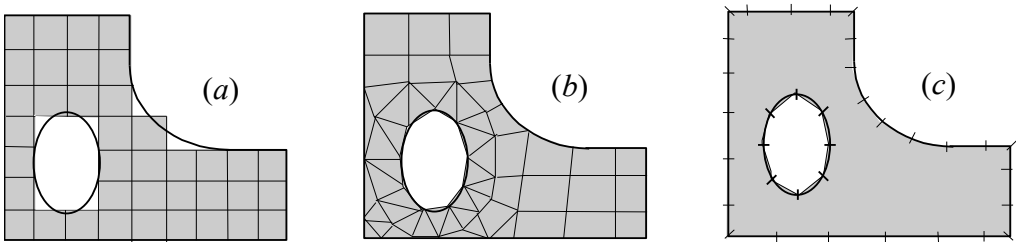


Figure 1 (*a*) Finite difference, (*b*) finite element and (*c*) boundary element discretisations of a machine component.

Finite element method is one of the most powerful and versatile computational mechanics tool available to the modern analyst. Its ever-expanding domain of application includes problems of stress analysis, heat transfer, fluid flow, lubrication, electric and magnetic fields, and many others. Though the method was first invented by a group of civil engineers as an extension of the matrix stiffness method, the finite element method by now has become an active area of research for both engineers and mathematicians alike.

The goal of this series of lectures is to provide an introduction to the finite element method to academicians and practising engineers. It should be borne in mind that the finite element method is one of the most popular computational mechanics tool and hence its study is not complete without taking at least a glimpse at the programming aspects and computer-implementation of the method. An introduction to the theoretical foundations of the method as well as an introduction to implement the method on a computer is intended by this series of lectures. Static problems of elasticity are chosen for demonstrating the formulation and implementation of finite element method. It is hoped that these lectures go as a nice and enjoyable educational experience for all the participants and for the speaker too.

2. FINITE ELEMENTS AND ELASTOSTATIC PROBLEMS

Theory of elasticity is a fascinating branch of mechanics. Elastostatic problems form a major branch of the mathematical theory of elasticity.

2.1 Governing Equations of Elasticity

Let us consider a linearly elastic, homogeneous, isotropic continuum subject to certain body forces and surface tractions. The following are the basic equations of elastostatics.

2.1.1 Equations of Equilibrium:

Considering the equilibrium of an infinitesimal rectangular parallelepiped element and summing up forces along the x and y coordinate directions we get the equations of equilibrium for a two-dimensional elastic continuum as:

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + b_x &= 0 \\ \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \sigma_y}{\partial y} + b_y &= 0 \end{aligned} \tag{1a}$$

σ_x, σ_y are the normal stresses and τ_{xy} is the shear stress component. The body forces are denoted by b_x and b_y. Eq (1a) could also be written as:

$$\begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \tag{1b}$$

Or
$$[L]^{\mathrm{T}}\{\sigma\} = \{b\} \tag{1c}$$

where $[L]$ is an operator matrix. It may be noted that there exists a third equation of equilibrium, which is obtained by summing up moments of various forces acting on the parallelepiped element. This equation in the absence of body couples, leads to the condition that the stress tensor is symmetric. That is, $\tau_{yx} = \tau_{xy}$.

In the case of a three-dimensional problem, the equations of equilibrium can be written as:

$$\sigma_{ji,j} + b_i = 0 \tag{1d}$$

In the above, the repeated index (dummy index) denotes summation and the comma followed by the subscript j denotes partial differentiation with respect to the independent variable x_j (Einstein's summation convention).

2.1.2 Stress-strain relations:

The algebraic relations which exist between the components of the stress and the strain tensors, can be conveniently represented using the matrix notation as:

$$\{\sigma\} = [D]\ \{\varepsilon\} \tag{2a}$$

where $\{\sigma\}^{\mathrm{T}} = [\sigma_x\ \sigma_y\ \tau_{xy}]$ and $\{\varepsilon\}^{\mathrm{T}} = [\varepsilon_x\ \varepsilon_y\ \tau_{xy}]$ are the stress and strain vectors; $[D]$ is called the *constitutive matrix* and is a symmetric matrix. For *plane stress* and *plane strain* problems in which the material is linearly elastic and isotropic, $[D]$ is given by,

$$[D] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \qquad [D] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \upsilon & 0 \\ \upsilon & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{2b}$$

(plane stress case) (plane strain case)

The sizes of the matrices in equation (2) will increase from three to six in the case of a three-dimensional problem. The stress-strain relations for an isotropic body can be written using the index notation as:

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} \tag{2c}$$

In the above, λ and μ are the so-called Lame's constants. They are related to the more commonly used elastic constants E and ν.

2.1.3 Strain-displacement relations:

The strain-displacement relations for a two-dimensional elasticity problem can be conveniently represented in matrix form as follows:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} \tag{3a}$$

or

$$\{\varepsilon\} = [L]\ \{u\} \tag{3b}$$

In the case of a three-dimensional problem, the strain-displacement relation takes the following form:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{3c}$$

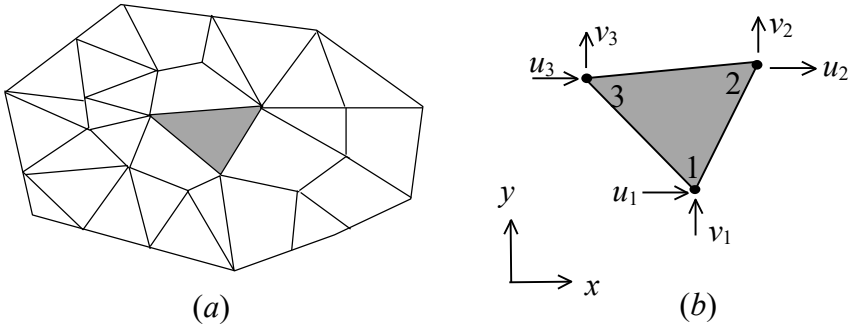
The above equation is equivalent to six different equations (i, j = 1 to 3). The shear strain component ε_{ij} ($i \neq j$) is equal to one half of the corresponding engineering shear strain component γ_{ij}. For example, $\varepsilon_{12} = 0.5\ \gamma_{xy}$.

2.2 Basic Steps Involved in Finite Element Analysis of Elastostatic Problems

Consider either a plane stress problem or a plane strain one. The following are the basic steps involved in finite element analysis of elastostatic problems (The steps are not much different even for three-dimensional problems, plate bending problems etc.).

1. Discretise the continuum into many subregions called *finite elements* of arbitrary size, shape and orientation (Fig 2a).
2. Each element is assumed to be connected to the neighbouring elements only at a finite number of discrete points called *nodes*.
3. The displacements at the nodes are assumed as the basic unknowns of the problem.
4. Let us imagine that, by some means, we have arrived at the nodal displacements at all the nodes. Consider any one element such as that shown in Fig. 2(b). For this element, thus, let us suppose that the element nodal displacement vector u^e is fully known, where u^e is given by:

$$\{u^e\} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}$$



(*a*) (*b*)

Figure 2 (*a*) Finite element discretisation of a two-dimensional continuum, and (*b*) one finite element and the nodal displacement degrees of freedom

Fig. 2(b) depicts a *three-noded triangular element*, which is one of the oldest of finite elements. Triangular elements are the simplest of all two-dimensional elements. However, for getting sufficiently accurate results one may have to use a fine mesh of triangles.

5. Given the element nodal displacement vector u^e, the interior displacement field within the element is obtained by interpolation. Thus,

$$\begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}$$

or

$$\{u\} = [N]\ \{u^e\} \tag{4}$$

N_1, N_2, N_3, are called the interpolation functions. $[N]$ is called the interpolation matrix or the shape function matrix. Polynomial interpolation is a common feature of most of the present day finite element formulations.

6. Once the displacement field within the element is known, the strain field can be obtained by making use of the strain-displacement relations, viz., Eq (3). Thus,

$$\begin{aligned} \{\varepsilon\} &= [L]\ \{u\} = [L]\ [N]\ \{u^e\} \\ &= [B]\ \{u^e\} \end{aligned} \tag{5}$$

7. By making use of the stress-strain relationss, the stress field in the element is obtained as

$$\begin{aligned} \{\sigma\} &= [D]\ \{\varepsilon\} \\ &= [D]\ [B]\ \{u^e\} \end{aligned} \tag{6}$$

8. The principle of virtual work is now used to arrive at the basic finite element equation as follows:

Impose a virtual nodal displacement field $\{\delta u^e\}$ within the element. The virtual displacement field within the element is given by:

$$\{\delta u\} = [N]\ \{\delta u^e\} \tag{7}$$

The virtual strain field corresponding to the above nodal virtual displacement vector $\{\delta u^e\}$ is given by:

$$\{\delta \varepsilon\} = [B]\ \{\delta u^e\} \tag{8}$$

Now, equating the internal virtual work done to the external virtual work, we get:

$$\int_{v_e} \{\delta\varepsilon\}^T \{\sigma\} dV_e = \int_{v_e} \{\delta u\}^T \{b\} dV_e + \int_{s_e} \{\delta u\}^T \{p\} dS_e \tag{9}$$

where V_e and S_e are the element volume and surface area respectively; $\{b\}$ is the body force vector and $\{p\}$ denotes the surface force acting over the periphery of the element. Making use of Eqs. (6), (7) and (8) in Eq (9) we get:

$$\{\delta u^e\}^T [\int_{V_e} B^T DB dV_e] \{u^e\} = \{\delta u^e\}^T [\int_{V_e} N^T \{b\} dV_e + \int_{S_e} N^T \{p\} dS_e] \tag{10}$$

Since $\{\delta\ u^e\}$ represents an arbitrary nodal virtual displacement vector, Eq (10) can be written as:

$$[k^e]\{u^e\} = \{r^e\} \tag{11}$$

where

$$[k^e] = \int_{V_e} B^T DB\ dV_e \tag{12}$$

and

$$\{r^e\} = \int_{V_e} N^T \{b\} dV_e + \int_{S_e} N^T \{p\} dS_e \tag{13}$$

In the above, $[k^e]$ is the *element stiffness matrix* and $\{r^e\}$ is the *element load vector*.

9. The element stiffness matrices and the element load vectors for various elements are added together to arrive at a system of linear algebraic equations as given by

$$[K]\ \{U\} = \{R\} \tag{14}$$

where $[K]$ is called the *global stiffness matrix* and $\{R\}$ is called the *global load vector*. These matrices are obtained by a process called "*assembly"*, that can be mathematically represented as,

$$[K] = \sum_{e=1}^{nElems} [k^e] \qquad \text{and} \qquad \{R\} = \sum_{e=1}^{nElems} \{r^e\} \tag{15}$$

where *nElems* represents the number of elements. The above summations (assembly) imply a special type of matrix addition that is explained later.

10. Eq (14) is solved to determine the unknown nodal displacement vector $\{U\}$. Now, for each element, the element nodal displacement vector $\{u^e\}$ is extracted from $\{U\}$.
11. Once $\{u^e\}$ is known for an element *"e"*, the displacement field within the element is found using Eq (4).
12. Eq (5) gives the components of the strain within the element
13. Eq (6) provides the stress components within the element

EXAMPLE:(A) AXIAL BAR ELEMENT

Consider a two noded axial bar element which is called a *truss element* or a *bar element*. It has two degrees of freedom, one at each node as shown in Fig. 3(a). Given the nodal dis-

placements u_1 and u_2, the displacement at any intermediate point is obtained by a simple linear interpolation as,

$$u(x) = [N_1 \quad N_2] \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = [N] \ \{u^e\} \tag{a}$$

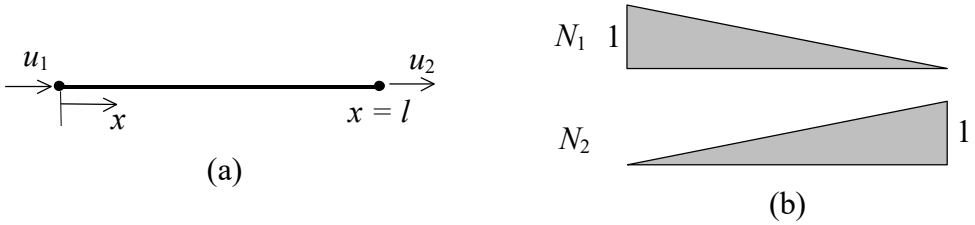


Figure 3 (*a*) Two-noded truss element, and (*b*) the shape functions N_1 and N_2

where $N_1 = 1 - x/l$ and $N_2 = x/l$. The plots of N_1 and N_2 are shown in Fig. 2(b). N_1, N_2 are known as shape functions or interpolation functions.

The strain in the bar at x is given by,

$$\begin{aligned} \varepsilon_x(x) = \frac{du}{dx} &= \left[\frac{dN_1}{dx} \frac{dN_2}{dx} \right] \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \\ &= \left[-\frac{1}{l} \quad \frac{1}{l} \right] \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = [B] \{u^e\} \end{aligned} \tag{b}$$

Thus, $[B] = [-1/l \quad 1/l]$. The stress σ_x at x is given by,

$$\sigma_x = E \ \varepsilon_x = E[B] \ \{u^e\} \tag{c}$$

i.e., in the case of the bar element, the constitutive matrix reduces to a 1×1 matrix or scalar, E. Hence we can get the stiffness matrix k^e as:

$$\begin{aligned} [k^e] &= \int_{v_e} B^T DB dV_e \\ &= \int_0^l \begin{bmatrix} -1/l \\ 1/l \end{bmatrix} E \left[-\frac{1}{l} \quad \frac{1}{l} \right] A dx = \frac{AE}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \end{aligned} \tag{d}$$

The above procedure can be followed for any other element too to arrive at the element stiffness matrix.

EXAMPLE: (B) CONSTANT STRAIN TRIANGLE ELEMENT

Consider the three-noded triangular element with a total of 6 degrees of freedom as shown in Fig 4. The displacement field within the element is given by,

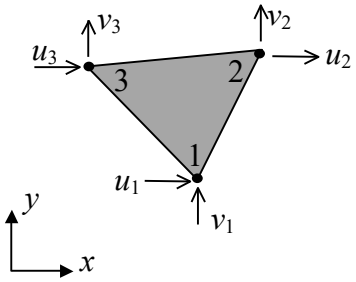


Figure 4 Three-noded triangle element

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} = [N]\{u^e\} \tag{a}$$

where N_1, N_2 and N_3 are linear interpolation polynomials in x and y, which are given by:

$$N_i = \frac{1}{2\Delta}(a_i + b_i x + c_i y),\ i = 1, 2, 3 \tag{b}$$

where $a_i = x_j\, y_k$ - $x_k\, y_j$
$b_i = y_j$ - y_k
$c_i = x_k$ - x_j

i,j and k are to be taken in a cyclic order. For example, when i = 1, j and k should be 2 and 3 respectively. And,

$$\Delta = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 2 \times \text{Area of the triangle}$$

The strain displacement matrix is given by,

$$[B] = \frac{1}{2\Delta} \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}$$

$$= \frac{1}{2\Delta} \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix} \tag{c}$$

It can be noticed that [B] is a matrix of constant elements. Hence a constant strain field is obtained (hence, the name-*constant strain triangle*).

The element stress field is given by:

$$\{\sigma\} = [D]\ [B]\ \{u^e\} \tag{d}$$

where the elements of the 3 × 3 constitutive matrix [D] are obtained from Eq (2) depending on whether the problem is a plane stress or plane strain one. The element stiffness matrix is obtained as:

$$[k^e] = \int_{v_e} B^T DB\, dV_e$$

$$= B^T DB \times \text{area of triangle} \times \text{thickness of element} \tag{e}$$

Explicit expressions for k^e are available in standard textbooks. In the above, for plane stress problems the thickness of the plate is to be used in the place of thickness of the element. On the other hand, for plane strain problems a unit thickness is usually adopted.

2.2.1 Assembly of Equations

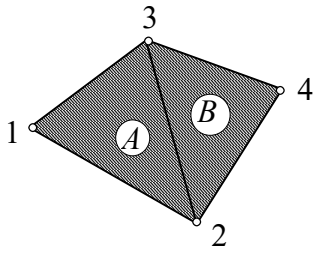


Figure 5 Assembly of element matrices

Consider two imaginary triangular elements with just one degree of freedom per node. Let the element load displacement relationships be given by,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix}, \text{ for element } A$$

and

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{Bmatrix} u_2 \\ u_4 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix}, \text{ for element } B$$

Adding the above two sets of equations, (noting that the displacement at common nodes such as nodes 2 and 3 are the same for all the elements meeting at those nodes), we get,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22}+b_{11} & a_{23}+b_{13} & b_{12} \\ a_{31} & a_{32}+b_{31} & a_{33}+b_{33} & b_{32} \\ 0 & b_{21} & b_{23} & b_{22} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{Bmatrix} p_1 \\ p_2+q_1 \\ p_3+q_3 \\ q_2 \end{Bmatrix}$$

The assembly operation basically requires the identification of the addresses in the global stiffness matrix and global load vector to which the elements of the element stiffness matrix and element load vector should be posted.

2.2.2 Solution of Equations

Eq (14) can be written in partitioned form, to show the restrained and free (subscripts *r* and *f*) degrees of freedom explicitly as follows:

$$\begin{bmatrix} K_{ff} & K_{fr} \\ K_{rf} & K_{rr} \end{bmatrix} \begin{Bmatrix} U_f \\ U_r \end{Bmatrix} = \begin{Bmatrix} R_f \\ R_r \end{Bmatrix} \tag{16}$$

In general, $\{U_r\} = \{0\}$, the null vector. Hence, we get:

$$[K_{ff}]\{U_f\} = \{R_f\} \tag{17}$$

Thus, only the K_{ff} part of the stiffness matrix (which is called the *reduced stiffness matrix*) need be assembled and stored. R_f represents the applied loads and R_r the support reactions. The support reactions are obtained after solving Eq (17) for the unknown free degrees of freedom $\{U_f\}$. The element nodal displacements are extracted from $\{U\}$ and

$$[k^e]\ \{u^e\} = \{r^e\} \tag{18}$$

is used to calculate $\{r^e\}$, now that $\{u^e\}$ is known.

2.3 Some of the Programming Aspects

Learning finite element method is incomplete unless one takes a look at the computer implementation of the method. Let us try to get into at least some of the basic details.

2.3.1 Input Data

The various input data required by a finite element analysis program (for example, say a two-dimensional elastostatic problem) are:

- number of nodes (*nNodes*), number of elements(*nElems*)
- modulus of elasticity (*E*)
- nodal coordinate data

```
// input nodal coordinates of all nodes in any order
for i = 1, nNodes
input k, x(k), y(k)
```

- element connectivity data

```
// input the three node numbers and thickness of the element (follow an
// anti-clockwise order) for each element.
for i = 1, nElems
input k, nod(1, k), nod(2, k), nod(3, k), thick(k), E(k)
```

- displacement boundary conditions

```
// Initialise destination array
for i = 1, nNodes
for j = 1, 2
destn(i,j) = 0

input nDispRestrains        // nDispRestrains = number of nodes with
                            // displacement boundary conditions prescribed
for i = 1, nDispRestrains
read k, destn(1,k), destn(2,k)      // input 1 if a d.o.f. is restrained;
                                    //input 0 if a d.o.f. is free
```

The destination array, viz., destn array, is used to store addresses to which the elements of the matrices $[k^e]$ and $\{r^e\}$ must be posted. The destination array is a rectangular matrix which has as many columns as the number of nodes and as many rows as the number of degrees of freedom per node. As shown above in the pseudocode, the destination array is first initialised with zeros. The displacement boundary conditions are then fed into this array as 1's and 0's. Input a value of unity for a fixed degree of freedom, and a value of zero for a free degree of freedom for all the nodes at which the displacements are restrained. Consider a small example problem as shown in Fig. 6(a). It corresponds to a plate under plane stress conditions (i.e., the thickness of the plate is small when compared to the other two dimensions). It is subjected to

a uniform loading along one edge and is supported in a stable manner. The first step in solving the problem by the finite element method consists in discretising the domain into many elements. Let us use just four elements as a demonstration. The elements and the nodes are to be numbered next as shown. It is a common practice to encircle the node numbers while enclosing the element numbers inside squares in order that the node numbers are distinct from the element numbers. The degree of freedom numbers are generated next. The rule is to follow the node numbers starting from the first node. If a degree of freedom is restrained, mark it as a zero; if a degree of freedom is free, number it in an ascending order as 1, 2, 3, etc. The horizontal degree of freedom is numbered prior to the vertical degree of freedom at any node. The degree of freedom numbering for the above discretisation is shown in Fig. (b).

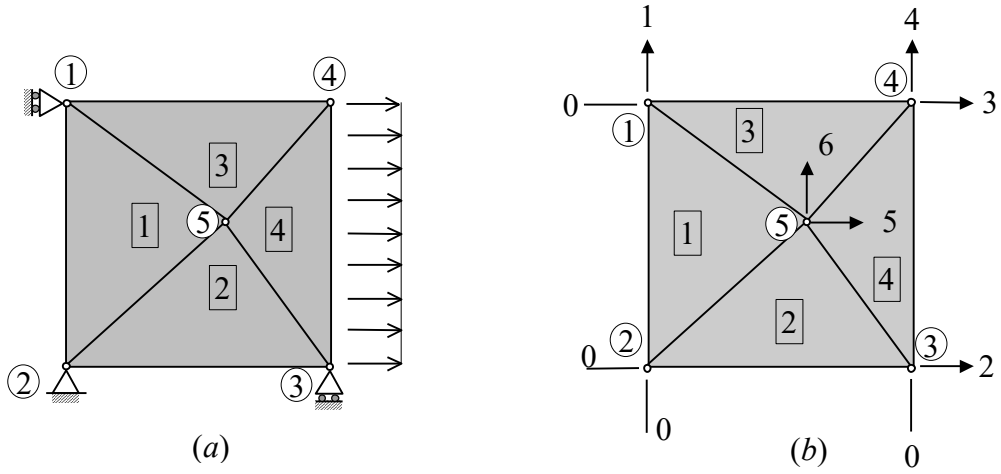


Figure 6 (a) The discretisation and **(b)** the degree of freedom numbering of square plate problem

These degrees of freedom numbers are generated by the computer quite easily. The destination array, for this example, after reading in the boundary restrain list (i.e., displacement boundary conditions) will look as follows:

$$[destn] = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Now, the *destn* array is scanned column wise, and each column from top to bottom. Any '1' is converted into a '0'; and a '0' is replaced by a counter. Thus, the above *destn* array gets modified into,

$$[destn] = \begin{bmatrix} 0 & 0 & 2 & 3 & 5 \\ 1 & 0 & 0 & 4 & 6 \end{bmatrix}$$

In the above, the elements of the [destn] array represent the d.o.f. numbers at each node (the column number represents the node number). Also, the last number in the *destn* array represents the number of equations. The corresponding algorithm runs as follows:

```
// Update destination array
neq = 0 // neq is initialised to zero
for i = 1, nNodes
for j = 1, 2
  if (destn(i, j) = 0)
     neq = neq + 1
     destn(i, j) = neq
  else
     destn(i, j) = 0;
     destn(i,j) = 0
  endif
```

The *nod* array and the *destn* array are used to assemble the element stiffness matrices and the load vectors to get the global stiffness matrix and the global load vector respectively. Before assembly, these global arrays are initialised. If any concentrated load acts at any joint along any d.o.f. direction, it is read directly into the global load vector as follows:

```
for i = 1, neq          //neq = number  of equations
R(i) = 0
for j = 1, neq
K(i,j) = 0
input nNconc                //nNconc = number of nodes at which concen-
trated                //          loads are applied
for i = 1, nNconc
    input k, forcx, forcy
    dof1 = destn(1,k)
    dof2 = destn(2,k)
   R(dof1) = forcx
R(dof2) = forcy
```

2.3.2 Assembly of global matrices

```
for e = 1, nElems
   call element(e)          //invoke element subprogram to get element
                            //stiffness matrix and load vectors
   nod1 = nod(1,n)
   nod2 = nod(2,n)
   nod3 = nod(3,n)
   kk(1) = destn(1, nod1)
   kk(2) = destn(2, nod1)
   kk(3) = destn(1, nod2)
   kk(4) = destn(2, nod2)
   kk(5) = destn(1, nod3)
   kk(6) = destn(2, nod3)
   for i = 1, 6
      if kk(i) > (0) then
         krow = kk(i)
         R(krow) = R(krow) + r(i)
      for j = 1,6
         if kk(j) > (0) then
            kcol = kk(j)
            K(krow,kcol) = K(krow,kcol) + ke(i,j)
         end if
       endif
```

2.3.3 Solution of Equations

Once the global stiffness matrix and the global load vector have been assembled, we arrive at a system of linear algebraic equations (Eq. 17) whose solution gives the unknown free degrees of freedom $\{U_f\}$ of the problem. Several variants of Gauss elimination procedures are in use. The global stiffness matrix, for structural mechanics problems is symmetric, banded and sparse. A number of very efficient (both in terms of memory requirement and computational time) algorithms are available which make use of the symmetry, bandedness and sparsity of the stiffness matrix.

2.3.4 Computation of Stresses, Support Reactions etc.

Once $\{U_f\}$ is determined, the complete nodal displacement vector $\{U\}$ is known (remember $\{U_r\}$ contains prescribed displacement degrees of freedom, and hence is fully known). Now, the element subprogram is called once again. The element nodal displacements are extracted from the global displacement vector. Then the stresses within any element is given by Eq (6)

Support reactions are obtained using Eq (18), by multiply the element stiffness matrix by the element nodal displacement vector.

3. STRUCTURAL DYNAMICS

If the frequency of excitation applied to a structure is less than roughly one-third the structure's lowest natural frequency, the effects of inertia can be neglected and the problem can be treated as *quasistatic*. That is the equations $[K]\ \{U\} = \{R\}$ are sufficiently accurate even though loads $\{R\}$, and hence displacements $\{U\}$ vary (slowly) with time.

The equations of motion can be developed by using the virtual work principle itself in which the inertia force is treated just like another body force. Thus, Eq (9) gets replaced by:

$$\int_{v_e} \{\delta\varepsilon\}^T \{\sigma\} dV_e = \int_{v_e} \{\delta u\}^T \{b\} dV_e + \int_{s_e} \{\delta u\}^T \{p\} ds_e - \int_{v_e} \{\delta u\}^T \rho \{\ddot{u}\} dV_e \tag{19}$$

which leads to:

$$[M]\ \{\ddot{U}\} + [K]\ \{U\} = \{R\} \tag{20}$$

where, $[M] = \sum_{e=1}^{nelem} [m^e]$, where $[m^e]$ is the element mass matrix which is given by,

$$[m^e] = \int_{V_e} \rho\ N^T N\, dV_e \tag{21}$$

The mass matrix can be assembled in exactly the same way as the stiffness matrix. Depending on the nature of $\{R\}$, there are different types of structural dynamic problems one may come across.

3.1 Free Vibration Problems (Eigenvalue Problem)

When {R} is a null vector, we come across the free vibration problem in which one is interested in knowing the response of the system due to some prescribed initial conditions in the form of initial displacements and initial velocities. A more important free vibration problem deals with the determination of the natural frequencies of the system.

Eq (20) in the absence of any external excitation reduces to

$$M\ddot{U} + KU = 0 \tag{22}$$

We seek a kind of synchronous solution wherein all the degrees of freedom oscillate at the same frequency, but with different amplitudes. Substituting $\{U\} = \{U_o\}\ Sin\ \omega t$ in the above equation, we end up with an algebraic eigenvalue problem.

$$(-\omega^2 M U_o + K U_o) \sin \omega t = 0 \tag{23}$$

Since $Sin\ \omega t$ cannot be zero for different t values, we must have,

$$[K]\ \{U_o\} = \omega^2 [M]\ \{U_o\} \tag{24}$$

This is a generalised eigenvalue problem of the form $[A]\ \{X\} = \lambda\ [B]\ \{X\}$. Some of the popular methods of solving Eq. 24 in order to obtain the eigenvalues (squares of natural frequencies of vibration) and the eigenvectors (vibration modes) are:

- ***Jacobi Method***

An iterative method by which both K and M matrices are brought to diagonal form using a number of orthogonal transformations.

- ***Householder Method***

K and M matrices are first brought down to tri-diagonal forms. The eigenvalues and eigenvectors of the tri diagonal system is extracted using methods such as QR algorithm.

- ***Forward and Inverse Iteration Methods***

Methods which are very easy to implement on a computer. First the smallest eigenvalue and the corresponding eigenvector is obtained; then the second smallest and so on.

3.2 Forced Vibration — Mode Superposition Method

The equations of motion as given by Eq (20) are transformed by making use of the modal matrix. Thus, the transformation,

$$\{U\} = [P]\ \{X\} \tag{25}$$

where $[P]$ is called the modal matrix and is obtained by stacking the eigenvectors column-wise. It is usually found by means of an eigenvalue extraction routine. The modal matrix has a special property that it can diagonalize $[M]$ and $[K]$ matrices simultaneously. Using Eq (25) in Eq (20), and pre-multiplying each term by $[P]^{\mathrm{T}}$, we get,

$$[P]^{\mathrm{T}}\ [M]\ [P]\ \{\ddot{X}\} + [P]^{\mathrm{T}}\ [K]\ [P]\ \{X\} = [P]^{\mathrm{T}}\ \{R\}$$

Due to the special property of the modal matrix, $P^T\ MP$ and $P^T\ KP$ are both diagonal matrices. Thus, we have,

$$[M^*]\ \{\ddot{X}\} + [K^*]\ \{X\} = \{R^*\} \tag{26}$$

where both $[M^*]$ and $[K^*]$ matrices are diagonal. Hence Eq (26) gets decoupled into several single degree of freedom system equations. The i^{th} equation will look like,

$$m_i\ \ddot{x}_i + k_i\, x_i = r_i \tag{27}$$

which can be easily solved. For example, the solutions to Eq (27) can be written using Duhamel integral as,

$$x_i(t) = \frac{1}{\omega_i} \int_0^t r_i(\tau)\, Sin\, \omega_i (t-\tau) d\tau + A_i\, Sin\, \omega_i t + B_i\, Cos\, \omega_i t \tag{28}$$

where $\omega_i = \sqrt{k_i / m_i}$ is the corresponding natural frequency; A_i and B_i are determined from the initial conditions. We can get the required solution $\{U\}$ once $\{X\}$ is known, using Eq (25).

3.3 Forced Vibration — Direct Integration Method

In direct integration equations in (20) are integrated using a numerical step-by-step procedure. Some of the common direct integration methods are discussed below.

(a) ***The Central Difference Method*****:**

The derivatives appearing in the equations of motion are replaced by their finite difference equivalents. Let us consider a damped multi-degree of freedom system as given by,

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \tag{29}$$

where **C** is the damping matrix. C is obtained either by a finite element procedure or more commonly, **C** is taken to be proportional to **K** and **M** matrices. Thus,

$$\mathbf{C} = \alpha\,\mathbf{K} + \beta\,\mathbf{M}$$

where α and β are appropriately chosen. The damping matrix thus obtained is called *proportional damping*.

In the Eq.(29), $\dot{U}$ and $\ddot{U}$ are replaced by the finite difference approximations (central differences),

$$\dot{U}(t) = \frac{1}{2\Delta t}[U(t+\Delta t) - U(t-\Delta t)] \tag{30}$$

$$\ddot{U}(t) = \frac{1}{\Delta t^2}[U(t+\Delta t) - 2U(t) + U(t-\Delta t)] \tag{31}$$

Considering Eq (29) at time *t* and substituting Eqns. 30 and 31, we get,

$$\left(\frac{1}{\Delta t^2}M + \frac{1}{2\Delta t}C\right)U(t+\Delta t) = R(t) - \left(K - \frac{2}{\Delta t^2}M\right)U(t) - \left(\frac{1}{\Delta t^2}M + \frac{1}{2\Delta t}C\right)U(t-\Delta t)] \tag{32}$$

from which we can solve for $U(t + \Delta t)$. However, a special starting procedure is called for to get $\ddot{U}(0)$ and $U(-\Delta t)$. The complete algorithm is given below:

Table 1 : Step-by-step solution using central difference method

A. *Initial Calculations*:

1. Form stiffness matrix **K**, mass matrix **M**, and damping matrix **C**.
2. Initialise $\mathbf{U}_o$, $\dot{\mathbf{U}}_o$ and $\ddot{\mathbf{U}}_o$
3. Select time step Δt, $\Delta t < \Delta t_{cr}$, and calculate integration constants:

$$a_o = \frac{1}{\Delta t^2}; a_1 = \frac{1}{2\Delta t}; a_2 = 2a_o; a_3 = \frac{1}{a_2}$$

4. Calculate $\boldsymbol{U}_{-\Delta t} = \boldsymbol{U}_o - \Delta t\,\dot{\boldsymbol{U}}_0 + a_3\,\ddot{\boldsymbol{U}}_0$
5. Form effective mass matrix $\hat{\mathbf{M}} = a_0\mathbf{M} + a_1\mathbf{C}$
6. Triangularize $\hat{\mathbf{M}}$: $\hat{\mathbf{M}} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathbf{T}}$

B. *For Each Time Step*:

1. Calculate effective loads at time *t*:

$$\hat{\mathbf{R}}_t = \mathbf{R}_t - (\mathbf{K} - a_2\mathbf{M})\mathbf{U}_t - (a_0\mathbf{M} - a_1\mathbf{C})\mathbf{U}_{t-\Delta t}$$

2. Solve for displacements at time $t + \Delta t$:

$$\mathbf{L}\mathbf{D}\mathbf{L}^T\mathbf{U}_{t+\Delta t} = \hat{\mathbf{R}}_t$$

3. If required, evaluate accelerations and velocities at time *t*:

$$\ddot{\mathbf{U}}_t = a_0(\mathbf{U}_{t-\Delta t} - 2\mathbf{U}_t + \mathbf{U}_{t+\Delta t})$$
$$\dot{\mathbf{U}}_t = a_1(-\mathbf{U}_{t-\Delta t} + \mathbf{U}_{t+\Delta t})$$

The central difference method has some major drawbacks (such as only conditional stability) and hence is not very popular. The method has been described here as this is one of the

easiest to understand amongst the direct integration methods. Choice of an appropriate time step is very important in this method.

(b) *Houbolt Method*:

In the Houbolt method, the following finite difference approximations are used:

$$\dot{U}(t+\Delta t) = \frac{1}{6\Delta t}\{11U(t+\Delta t) - 18U(t) + 9U(t-\Delta t) - 2U(t-2\Delta t)\} \tag{33}$$

$$\ddot{U}(t+\Delta t) = \frac{1}{\Delta t^2}\{2U(t+\Delta t) - 5U(t) + 4U(t-\Delta t) - U(t-2\Delta t)] \tag{34}$$

The above are special backward difference formulae with error of order $(\Delta t)^2$. The equations of motion (Eq. 29) are written at time $t + \Delta t$ (and not at t, as in the central difference method). Making use of Eqs. 33 and 34 in the resulting equation gives,

$$\left(\frac{2}{\Delta t^2}M + \frac{11}{6\Delta t}C + K\right)U(t+\Delta t) = R(t+\Delta t) + \left(\frac{5}{\Delta t^2}M + \frac{3}{\Delta t}C\right)U(t)$$
$$-\left(\frac{4}{\Delta t^2}M + \frac{3}{2\Delta t}C\right)U(t-\Delta t) + \left(\frac{1}{\Delta t^2}M + \frac{1}{3\Delta t}C\right)U(t-2\Delta t) \tag{35}$$

The stepwise algorithm is given below.

Table 2 : Step-by-step solution using Houbolt method

A. *Initial Calculations*:

1. From stiffness matrix **K**, mass matrix **M**, and damping matrix **C**.
2. Initialise $\mathbf{U}_o$, $\dot{\mathbf{U}}_o$ and $\ddot{\mathbf{U}}_o$.
3. Select time step Δt, and calculate integration constants:

$$a_o = \frac{1}{\Delta t^2}; a_1 = \frac{11}{6\Delta t}; a_2 = \frac{5}{\Delta t^2}; a_3 = \frac{3}{\Delta t}; a_4 = -2a_0; a_5 = \frac{-a_3}{2}; a_6 = \frac{a_0}{2}; a_7 = \frac{a_3}{9}$$

4. Use special starting procedure to calculate $\mathbf{U}_{\Delta t}$ and $\mathbf{U}_{\Delta t}$
5. Calculate effective stiffness matrix $\hat{\mathbf{K}}$: $\mathbf{K} + a_o\mathbf{M} + a_1\mathbf{C}$
6. Triangularize $\hat{\mathbf{K}}$: $\mathbf{K} = \mathbf{L}\mathbf{D}\mathbf{L}^T$

B. *For Each Time Step*:

1. Calculate effective loads at time t:

$$\hat{\mathbf{R}}_{t+\Delta t} = \mathbf{R}_{t+\Delta t}\mathbf{M}(a_2\mathbf{U}_t + a_4\mathbf{U}_{t-\Delta t} + a_6\mathbf{U}_{t-2\Delta t}) + \mathbf{C}(a_3\mathbf{U}_t + a_5\mathbf{U}_{t-\Delta t} + a_7\mathbf{U}_{t-2\Delta t})$$

2. Solve for displacements at time $t + \Delta t$:

$$LDL^T\mathbf{U}_{t+\Delta t} = \hat{\mathbf{R}}_{t+\Delta t}$$

3. If required, evaluate accelerations and velocities at time $t + \Delta t$:

$$\ddot{\mathbf{U}}_{t+\Delta t} = a_0\mathbf{U}_{t+\Delta t} - a_2\mathbf{U}_t - a_4\mathbf{U}_{t-\Delta t} - a_6\mathbf{U}_{t-2\Delta t}$$

$$\dot{\mathbf{U}}_{t+\Delta t} = a_1\mathbf{U}_{t+\Delta t} - a_3\mathbf{U}_t - a_5\mathbf{U}_{t-\Delta t} - a_7\mathbf{U}_{t-2\Delta t}$$

Houbolt method is an unconditionally stable method and works well with even much larger a Δt than that could be used with the central difference method.
There are other important and powerful methods such as the Wilson θ method and the Newmark β method. Interested reader is referred to the books by Bathe given in bibliography.

Reference

1. Forsythe, G.E., and Wasow, W.R., *Finite Difference Methods for Partial Differential Equations*, John Wiley & Sons, New York, 1960.
2. Zienkiewicz, O.C., and Taylor, R.L., *The Finite Element Method*, Vols I & II, McGraw-Hill, London, 1989.
3. Finlayson, B.A., *The Method of Weighted Residuals and Variational Principles*, Academic Press, New York, 1972.
4. M. Ameen, *Boundary Element Analysis – Theory and Programming*, Narosa Publishing House, New Delhi, 2001. (Also published by Alpha Science International, UK)
5. Greenberg, M.D., *Application of Green's Functions in Science and Engineering*, Prentice-Hall, Inc., Engle Wood, New Jersey, 1971.
6. Ameen, M., *Boundary Element Analysis-Theory and Programming*, Narosa Publishing House, New Delhi, 2001.

Bibliography

Some of the useful books on FEM are given below with a very brief description.

1. R.D. Cook, D.S. Malkus and M.E. Plesha, *Concept and Application of Finite Element Analysis*, John Wiley, New York, 1989.
   (An excellent book to both beginners and users of FEM, very well written. Worth owing)
2. J.K. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice Hall of India, New Delhi, 1992.

(A very good reference book especially for problems related to structural dynamics. An exhaustive book, which is a bit hard to read at places where even the notations are complicated)

3. J.K. Bathe and E.L. Wilson, *Numerical Methods in Finite Element Analysis*, Prentice Hall India, New Delhi, 1987.
   (A much smaller book in comparison to the above; contains almost all materials related to dynamics as in the above)
4. O.C. Zienkiewicz, and R.L. Taylor, *The Finite Element Method*, Volume I & II McGraw-Hill, London, 1989.
   (An excellent reference book. Vol II covers advanced topics such as nonlinear problems)
5. O.C. Zienkiewicz, and K. Morgan, *Finite Elements and Approximation*, John Wiley, New York, 1983.
   (A good book on weighted residual approach)
6. B. Szabo and I. Babuska, *Finite Element Analysis*, John Wiley, New York, 1991.
   (For a good taste in more mathematical treatment of FEM)
7. S.S. Rao, *The Finite Element Method in Engineering*, Pergamon Press, New York, 1982.
8. I. Shames and C.L. Dym, *Energy and Finite Element Methods in Structural Mechanics*, New Age Intrnl-Wiley Eastern, New Delhi, 1991.