Chapter 6

Finite Element Method

"Who is wise? He that learns from everyone.
Who is powerful? He that governs his passion.
Who is rich? He that is content.
Who is that? Nobody." Benjamin Franklin

6.1 Introduction

The finite element method (FEM) has its origin in the field of structural analysis. Although the earlier mathematical treatment of the method was provided by Courant [1] in 1943, the method was not applied to electromagnetic (EM) problems until 1968. Since then the method has been employed in diverse areas such as waveguide problems, electric machines, semiconductor devices, microstrips, and absorption of EM radiation by biological bodies.

Although the finite difference method (FDM) and the method of moments (MOM) are conceptually simpler and easier to program than the finite element method (FEM), FEM is a more powerful and versatile numerical technique for handling problems involving complex geometries and inhomogeneous media. The systematic generality of the method makes it possible to construct general-purpose computer programs for solving a wide range of problems. Consequently, programs developed for a particular discipline have been applied successfully to solve problems in a different field with little or no modification [2].

The finite element analysis of any problem involves basically four steps [3]:

- discretizing the solution region into a finite number of *subregions* or *elements,*
- deriving governing equations for a typical element,
- assembling of all elements in the solution region, and
- solving the system of equations obtained.

Discretization of the continuum involves dividing up the solution region into subdomains, called *finite elements*. Figure 6.1 shows some typical elements for one-, two-, and three-dimensional problems. The problem of discretization will be fully treated in Sections 6.5 and 6.6. The other three steps will be described in detail in the subsequent sections.

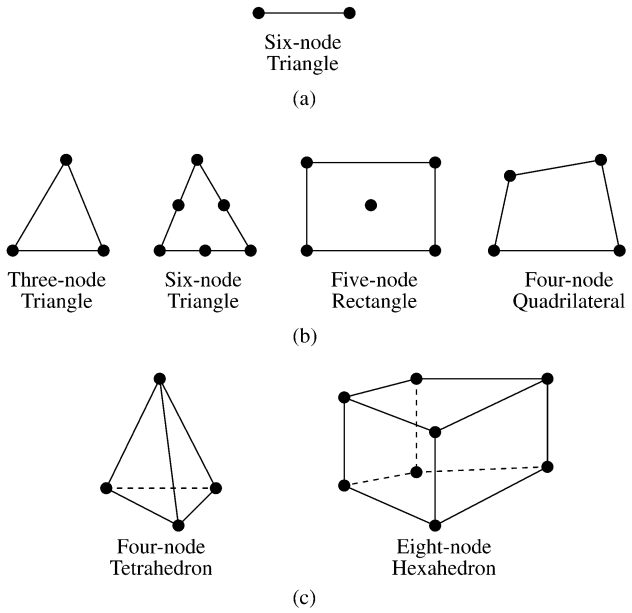


Figure 6.1

Typical finite elements: (a) One-dimensional, (b) two-dimensional, (c) three-dimensional.

6.2 Solution of Laplace's Equation

As an application of FEM to electrostatic problems, let us apply the four steps mentioned above to solve Laplace's equation, $\nabla^2 V = 0$. For the purpose of illustration, we will strictly follow the four steps mentioned above.

6.2.1 Finite Element Discretization

To find the potential distribution $V(x, y)$ for the two-dimensional solution region shown in Fig. 6.2(a), we divide the region into a number of finite elements as illustrated in Fig. 6.2(b). In Fig. 6.2(b), the solution region is subdivided into nine

nonoverlapping *finite elements*; elements 6, 8, and 9 are four-node quadrilaterals, while other elements are three-node triangles. In practical situations, however, it is preferred, for ease of computation, to have elements of the same type throughout the region. That is, in Fig. 6.2(b), we could have split each quadrilateral into two triangles so that we have 12 triangular elements altogether. The subdivision of the solution region into elements is usually done by hand, but in situations where a large number of elements is required, automatic schemes to be discussed in Sections 6.5 and 6.6 are used.

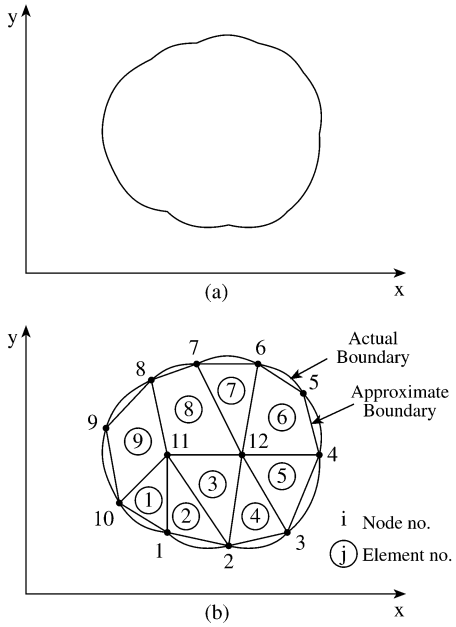


Figure 6.2
(a) The solution region; (b) its finite element discretization.

We seek an approximation for the potential V_e within an element e and then interrelate the potential distribution in various elements such that the potential is continuous across interelement boundaries. The approximate solution for the whole region is

$$V(x, y) \simeq \sum_{e=1}^{N} V_e(x, y), \tag{6.1}$$

where N is the number of triangular elements into which the solution region is divided. The most common form of approximation for V_e within an element is polynomial approximation, namely,

$$V_e(x, y) = a + bx + cy \tag{6.2}$$

for a triangular element and

$$V_e(x, y) = a + bx + cy + dxy \tag{6.3}$$

for a quadrilateral element. The constants a, b, c, and d are to be determined. The potential V_e in general is nonzero within element e but zero outside e. In view of the fact that quadrilateral elements do not conform to curved boundary as easily as triangular elements, we prefer to use triangular elements throughout our analysis in this chapter. Notice that our assumption of linear variation of potential within the triangular element as in Eq. (6.2) is the same as assuming that the electric field is uniform within the element, i.e.,

$$\mathbf{E}_e = -\nabla V_e = -\left(b\mathbf{a}_x + c\mathbf{a}_y\right) \tag{6.4}$$

6.2.2 Element Governing Equations

Consider a typical triangular element shown in Fig. 6.3. The potential V_{e1}, V_{e2}, and V_{e3} at nodes 1, 2, and 3, respectively, are obtained using Eq. (6.2), i.e.,

$$\begin{bmatrix} V_{e1} \\ V_{e2} \\ V_{e3} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{6.5}$$

The coefficients a, b and c are determined from Eq. (6.5) as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} V_{e1} \\ V_{e2} \\ V_{e3} \end{bmatrix} \tag{6.6}$$

Substituting this into Eq. (6.2) gives

$$V_e = \begin{bmatrix} 1 & x & y \end{bmatrix} \frac{1}{2A} \begin{bmatrix} (x_2y_3 - x_3y_2) & (x_3y_1 - x_1y_3) & (x_1y_2 - x_2y_1) \\ (y_2 - y_3) & (y_3 - y_1) & (y_1 - y_2) \\ (x_3 - x_2) & (x_1 - x_3) & (x_2 - x_1) \end{bmatrix} \begin{bmatrix} V_{e1} \\ V_{e2} \\ V_{e3} \end{bmatrix}$$

or

$$\boxed{V_e = \sum_{i=1}^{3} \alpha_i(x, y) V_{ei}} \tag{6.7}$$

where

$$\alpha_1 = \frac{1}{2A}\left[(x_2y_3 - x_3y_2) + (y_2 - y_3)\, x + (x_3 - x_2)\, y\right], \tag{6.8a}$$

$$\alpha_2 = \frac{1}{2A}\left[(x_3y_1 - x_1y_3) + (y_3 - y_1)\, x + (x_1 - x_3)\, y\right], \tag{6.8b}$$

$$\alpha_3 = \frac{1}{2A}\left[(x_1y_2 - x_2y_1) + (y_1 - y_2)\, x + (x_2 - x_1)\, y\right], \tag{6.8c}$$

and A is the area of the element e, i.e.,

$$2A = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$
$$= (x_1 y_2 - x_2 y_1) + (x_3 y_1 - x_1 y_3) + (x_2 y_3 - x_3 y_2)$$

or

$$A = \frac{1}{2}\left[(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)\right] \tag{6.9}$$

The value of A is positive if the nodes are numbered counterclockwise (starting from any node) as shown by the arrow in Fig. 6.3. Note that Eq. (6.7) gives the potential

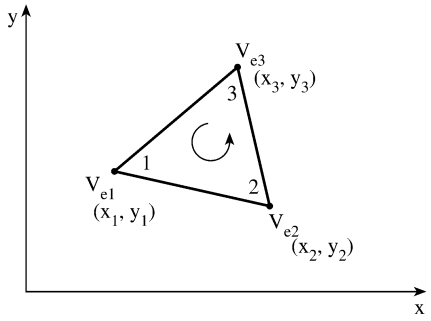


Figure 6.3
Typical triangular element; local node numbering 1-2-3 must proceed counterclockwise as indicated by the arrow.

at any point (x, y) within the element provided that the potentials at the vertices are known. This is unlike finite difference analysis, where the potential is known at the grid points only. Also note that α_i are linear interpolation functions. They are called the *element shape functions* and they have the following properties [4]:

$$\alpha_i = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{6.10a}$$

$$\sum_{i=1}^{3} \alpha_i(x, y) = 1 \tag{6.10b}$$

The shape functions α_1, α_2, and α_3 are illustrated in Fig. 6.4.

The functional corresponding to Laplace's equation, $\nabla^2 V = 0$, is given by

$$W_e = \frac{1}{2}\int \epsilon\, |\mathbf{E}_e|^2\, dS = \frac{1}{2}\int \epsilon\, |\nabla V_e|^2\, dS \tag{6.11}$$

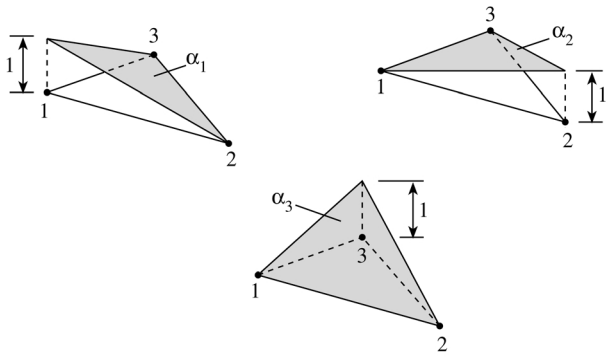


Figure 6.4
Shape functions α_1, α_2, and α_3 for a triangular element.

(Physically, the functional W_e is the energy per unit length associated with the element e.) From Eq. (6.7),

$$\nabla V_e = \sum_{i=1}^{3} V_{ei} \nabla \alpha_i \tag{6.12}$$

Substituting Eq. (6.12) into Eq. (6.11) gives

$$W_e = \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} \epsilon V_{ei} \left[\int \nabla \alpha_i \cdot \nabla \alpha_j \, dS \right] V_{ej} \tag{6.13}$$

If we define the term in brackets as

$$C_{ij}^{(e)} = \int \nabla \alpha_i \cdot \nabla \alpha_j \, dS \,, \tag{6.14}$$

we may write Eq. (6.13) in matrix form as

$$W_e = \frac{1}{2} \epsilon \left[V_e \right]^t \left[C^{(e)} \right] \left[V_e \right] \tag{6.15}$$

where the superscript t denotes the transpose of the matrix,

$$[V_e] = \begin{bmatrix} V_{e1} \\ V_{e2} \\ V_{e3} \end{bmatrix} \tag{6.16a}$$

and

$$\left[C^{(e)} \right] = \begin{bmatrix} C_{11}^{(e)} & C_{12}^{(e)} & C_{13}^{(e)} \\ C_{21}^{(e)} & C_{22}^{(e)} & C_{23}^{(e)} \\ C_{31}^{(e)} & C_{32}^{(e)} & C_{33}^{(e)} \end{bmatrix} \tag{6.16b}$$

The matrix $[C^{(e)}]$ is usually called the *element coefficient matrix* (or "stiffness matrix" in structural analysis). The element $C_{ij}^{(e)}$ of the coefficient matrix may be regarded as the coupling between nodes i and j; its value is obtained from Eqs. (6.8) and (6.14). For example,

$$\begin{aligned} C_{12}^{(e)} &= \int \nabla\alpha_1 \cdot \nabla\alpha_2 \, dS \\ &= \frac{1}{4A^2} \left[(y_2 - y_3)(y_3 - y_1) + (x_3 - x_2)(x_1 - x_3) \right] \int dS \\ &= \frac{1}{4A} \left[(y_2 - y_3)(y_3 - y_1) + (x_3 - x_2)(x_1 - x_3) \right] \end{aligned} \tag{6.17a}$$

Similarly,

$$C_{13}^{(e)} = \frac{1}{4A} \left[(y_2 - y_3)(y_1 - y_2) + (x_3 - x_2)(x_2 - x_1) \right] , \tag{6.17b}$$

$$C_{23}^{(e)} = \frac{1}{4A} \left[(y_3 - y_1)(y_1 - y_2) + (x_1 - x_3)(x_2 - x_1) \right] , \tag{6.17c}$$

$$C_{11}^{(e)} = \frac{1}{4A} \left[(y_2 - y_3)^2 + (x_3 - x_2)^2 \right] , \tag{6.17d}$$

$$C_{22}^{(e)} = \frac{1}{4A} \left[(y_3 - y_1)^2 + (x_1 - x_3)^2 \right] , \tag{6.17e}$$

$$C_{33}^{(e)} = \frac{1}{4A} \left[(y_1 - y_2)^2 + (x_2 - x_1)^2 \right] \tag{6.17f}$$

Also

$$C_{21}^{(e)} = C_{12}^{(e)}, \quad C_{31}^{(e)} = C_{13}^{(e)}, \quad C_{32}^{(e)} = C_{23}^{(e)} \tag{6.18}$$

6.2.3 Assembling of All Elements

Having considered a typical element, the next step is to assemble all such elements in the solution region. The energy associated with the assemblage of elements is

$$W = \sum_{e=1}^{N} W_e = \frac{1}{2} \epsilon [V]^t [C][V] \tag{6.19}$$

where

$$[V] = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \vdots \\ V_n \end{bmatrix} , \tag{6.20}$$

n is the number of nodes, N is the number of elements, and $[C]$ is called the overall or *global coefficient matrix,* which is the assemblage of individual element coefficient matrices. Notice that to obtain Eq. (6.19), we have assumed that the whole solution region is homogeneous so that ϵ is constant. For an inhomogeneous solution region such as shown in Fig. 6.5, for example, the region is discretized such that each finite element is homogeneous. In this case, Eq. (6.11) still holds, but Eq. (6.19) does not apply since $\epsilon(=\epsilon_r\epsilon_o)$ or simply ϵ_r varies from element to element. To apply Eq. (6.19), we may replace ϵ by ϵ_o and multiply the integrand in Eq. (6.14) by ϵ_r.

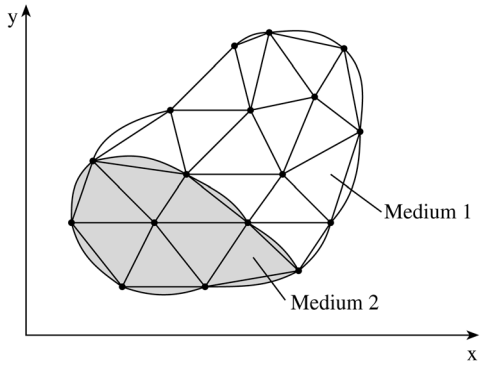


Figure 6.5
Discretization of an inhomogeneous solution region.

The process by which individual element coefficient matrices are assembled to obtain the global coefficient matrix is best illustrated with an example. Consider the finite element mesh consisting of three finite elements as shown in Fig. 6.6. Observe

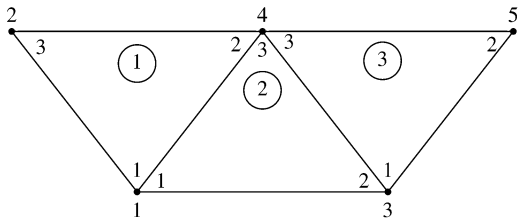


Figure 6.6
Assembly of three elements; i-j-k corresponds to local numbering (1-2-3) of the element in Fig. 6.3.

the numberings of the mesh. The numbering of nodes 1, 2, 3, 4, and 5 is called *global* numbering. The numbering i-j-k is called *local* numbering, and it corresponds with 1 - 2 - 3 of the element in Fig. 6.3. For example, for element 3 in Fig. 6.6, the global numbering 3 - 5 - 4 corresponds with local numbering 1 - 2 - 3 of the element in Fig. 6.3. (Note that the local numbering must be in counterclockwise sequence starting from any node of the element.) For element 3, we could choose 4 - 3 - 5

instead of 3 - 5 - 4 to correspond with 1 - 2 - 3 of the element in Fig. 6.3. Thus the numbering in Fig. 6.6 is not unique. But whichever numbering is used, the global coefficient matrix remains the same. Assuming the particular numbering in Fig. 6.6, the global coefficient matrix is expected to have the form

$$[C] = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} \end{bmatrix} \tag{6.21}$$

which is a 5×5 matrix since five nodes ($n = 5$) are involved. Again, C_{ij} is the coupling between nodes i and j. We obtain C_{ij} by using the fact that the potential distribution must be continuous across interelement boundaries. The contribution to the i, j position in $[C]$ comes from all elements containing nodes i and j. For example, in Fig. 6.6, elements 1 and 2 have node 1 in common; hence

$$C_{11} = C_{11}^{(1)} + C_{11}^{(2)} \tag{6.22a}$$

Node 2 belongs to element 1 only; hence

$$C_{22} = C_{33}^{(1)} \tag{6.22b}$$

Node 4 belongs to elements 1, 2, and 3; consequently

$$C_{44} = C_{22}^{(1)} + C_{33}^{(2)} + C_{33}^{(3)} \tag{6.22c}$$

Nodes 1 and 4 belong simultaneously to elements 1 and 2; hence

$$C_{14} = C_{41} = C_{12}^{(1)} + C_{13}^{(2)} \tag{6.22d}$$

Since there is no coupling (or direct link) between nodes 2 and 3,

$$C_{23} = C_{32} = 0 \tag{6.22e}$$

Continuing in this manner, we obtain all the terms in the global coefficient matrix by inspection of Fig. 6.6 as

$$\begin{bmatrix} C_{11}^{(1)} + C_{11}^{(2)} & C_{13}^{(1)} & C_{12}^{(2)} & C_{12}^{(1)} + C_{13}^{(2)} & 0 \\ C_{31}^{(1)} & C_{33}^{(1)} & 0 & C_{32}^{(1)} & 0 \\ C_{21}^{(2)} & 0 & C_{22}^{(2)} + C_{11}^{(3)} & C_{23}^{(2)} + C_{13}^{(3)} & C_{12}^{(3)} \\ C_{21}^{(1)} + C_{31}^{(2)} & C_{23}^{(1)} & C_{32}^{(2)} + C_{31}^{(3)} & C_{22}^{(1)} + C_{33}^{(2)} + C_{33}^{(3)} & C_{32}^{(3)} \\ 0 & 0 & C_{21}^{(3)} & C_{23}^{(3)} & C_{22}^{(3)} \end{bmatrix} \tag{6.23}$$

Note that element coefficient matrices overlap at nodes shared by elements and that there are 27 terms (9 for each of the 3 elements) in the global coefficient matrix $[C]$. Also note the following properties of the matrix $[C]$:

(1) It is symmetric ($C_{ij} = C_{ji}$) just as the element coefficient matrix.

(2) Since $C_{ij} = 0$ if no coupling exists between nodes i and j, it is expected that for a large number of elements $[C]$ becomes sparse. Matrix $[C]$ is also banded if the nodes are carefully numbered. It can be shown using Eq. (6.17) that

$$\sum_{i=1}^{3} C_{ij}^{(e)} = 0 = \sum_{j=1}^{3} C_{ij}^{(e)}$$

(3) It is singular. Although this is not so obvious, it can be shown using the element coefficient matrix of Eq. (6.16b).

6.2.4 Solving the Resulting Equations

Using the concepts developed in Chapter 4, it can be shown that Laplace's equation is satisfied when the total energy in the solution region is minimum. Thus we require that the partial derivatives of W with respect to each nodal value of the potential be zero, i.e.,

$$\frac{\partial W}{\partial V_1} = \frac{\partial W}{\partial V_2} = \cdots = \frac{\partial W}{\partial V_n} = 0$$

or

$$\frac{\partial W}{\partial V_k} = 0, \qquad k = 1, 2, \ldots, n \tag{6.24}$$

For example, to get $\dfrac{\partial W}{\partial V_1} = 0$ for the finite element mesh of Fig. 6.6, we substitute Eq. (6.21) into Eq. (6.19) and take the partial derivative of W with respect to V_1. We obtain

$$\begin{aligned} 0 = \frac{\partial W}{\partial V_1} = {} & 2V_1C_{11} + V_2C_{12} + V_3C_{13} + V_4C_{14} + V_5C_{15} \\ & + V_2C_{21} + V_3C_{31} + V_4C_{41} + V_5C_{51} \end{aligned}$$

or

$$0 = V_1C_{11} + V_2C_{12} + V_3C_{13} + V_4C_{14} + V_5C_{15} \tag{6.25}$$

In general, $\dfrac{\partial W}{\partial V_k} = 0$ leads to

$$0 = \sum_{i=1}^{n} V_i C_{ik} \tag{6.26}$$

where n is the number of nodes in the mesh. By writing Eq. (6.26) for all nodes $k = 1, 2, \ldots, n$, we obtain a set of simultaneous equations from which the solution of $[V]^t = [V_1, V_2, \ldots, V_n]$ can be found. This can be done in two ways similar to those used in solving finite difference equations obtained from Laplace's equation in Section 3.5.

(1) Iteration Method: Suppose node 1 in Fig. 6.6, for example, is a free node. From Eq. (6.25),

$$V_1 = -\frac{1}{C_{11}} \sum_{i=2}^{5} V_i C_{1i} \tag{6.27}$$

Thus, in general, at node k in a mesh with n nodes

$$\boxed{V_k = -\frac{1}{C_{kk}} \sum_{i=1, i\neq k}^{n} V_i C_{ki}} \tag{6.28}$$

where node k is a free node. Since $C_{ki} = 0$ if node k is not directly connected to node i, only nodes that are directly linked to node k contribute to V_k in Eq. (6.28). Equation (6.28) can be applied iteratively to all the free nodes. The iteration process begins by setting the potentials of fixed nodes (where the potentials are prescribed or known) to their prescribed values and the potentials at the free nodes (where the potentials are unknown) equal to zero or to the average potential [5]

$$V_{\text{ave}} = \frac{1}{2}(V_{\min} + V_{\max}) \tag{6.29}$$

where $V_{\min}$ and $V_{\max}$ are the minimum and maximum values of V at the fixed nodes. With these initial values, the potentials at the free nodes are calculated using Eq. (6.28). At the end of the first iteration, when the new values have been calculated for all the free nodes, they become the old values for the second iteration. The procedure is repeated until the change between subsequent iterations is negligible enough.

(2) Band Matrix Method: If all free nodes are numbered first and the fixed nodes last, Eq. (6.19) can be written such that [4]

$$W = \frac{1}{2}\epsilon \begin{bmatrix} V_f & V_p \end{bmatrix} \begin{bmatrix} C_{ff} & C_{fp} \\ C_{pf} & C_{pp} \end{bmatrix} \begin{bmatrix} V_f \\ V_p \end{bmatrix} \tag{6.30}$$

where subscripts f and p, respectively, refer to nodes with free and fixed (or prescribed) potentials. Since V_p is constant (it consists of known, fixed values), we only differentiate with respect to V_f so that applying Eqs. (6.24) to (6.30) yields

$$\begin{bmatrix} C_{ff} & C_{fp} \end{bmatrix} \begin{bmatrix} V_f \\ V_p \end{bmatrix} = 0$$

or

$$\boxed{[C_{ff}][V_f] = -[C_{fp}][V_p]} \tag{6.31}$$

This equation can be written as

$$[A][V] = [B] \tag{6.32a}$$

or

$$[V] = [A]^{-1}[B] \tag{6.32b}$$

where $[V] = [V_f], [A] = [C_{ff}], [B] = -[C_{fp}][V_p]$. Since $[A]$ is, in general, nonsingular, the potential at the free nodes can be found using Eq. (6.32). We can solve for $[V]$ in Eq. (6.32a) using Gaussian elimination technique. We can also solve for $[V]$ in Eq. (6.32b) using matrix inversion if the size of the matrix to be inverted is not large.

It is sometimes necessary to impose Neumann condition ($\frac{\partial V}{\partial n} = 0$) as a boundary condition or at the line of symmetry when we take advantage of the symmetry of the problem. Suppose, for concreteness, that a solution region is symmetric along the y-axis as in Fig. 6.7. We impose condition ($\frac{\partial V}{\partial x} = 0$) along the y-axis by making

$$V_1 = V_2, \quad V_4 = V_5, \quad V_7 = V_8 \tag{6.33}$$

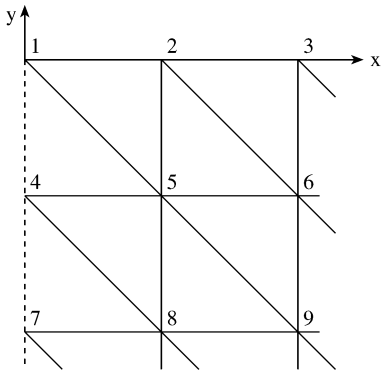


Figure 6.7
A solution region that is symmetric along the y-axis.

Notice that as from Eq. (6.11) onward, the solution has been restricted to a two-dimensional problem involving Laplace's equation, $\nabla^2 V = 0$. The basic concepts developed in this section will be extended to finite element analysis of problems involving Poisson's equation ($\nabla^2 V = -\rho_v/\epsilon$, $\nabla^2 \mathbf{A} = -\mu \mathbf{J}$) or wave equation ($\nabla^2 \Phi - \gamma^2 \Phi = 0$) in the next sections.

The following two examples were solved in [3] using the band matrix method; here they are solved using the iterative method.

Example 6.1

Consider the two-element mesh shown in Fig. 6.8(a). Using the finite element method, determine the potentials within the mesh. □

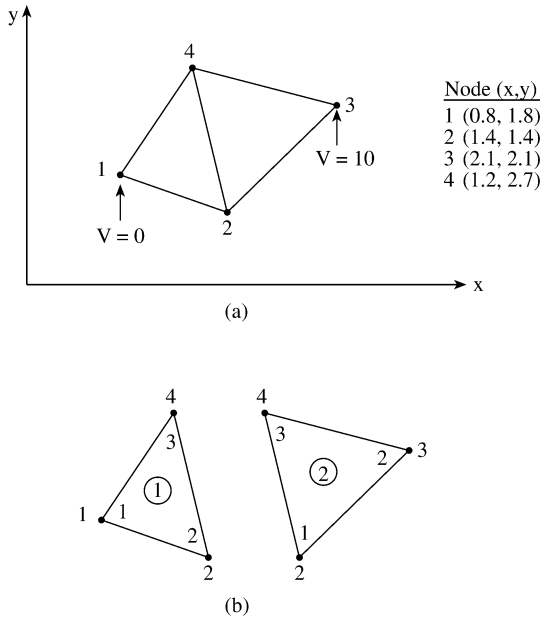


Figure 6.8

For Example 6.1: (a) Two-element mesh, (b) local and global numbering at the elements.

Solution

The element coefficient matrices can be calculated using Eqs. (6.17) and (6.18). However, our calculations will be easier if we define

$$P_1 = (y_2 - y_3)\,, \quad P_2 = (y_3 - y_1)\,, \quad P_3 = (y_1 - y_2)\,, \tag{6.34}$$
$$Q_1 = (x_3 - x_2)\,, \quad Q_2 = (x_1 - x_3)\,, \quad Q_3 = (x_2 - x_1)$$

With P_i and Q_i ($i = 1, 2, 3$ are the local node numbers), each term in the element coefficient matrix is found as

$$C_{ij}^{(e)} = \frac{1}{4A}\left(P_i P_j + Q_i Q_j\right) \tag{6.35}$$

where $A = \frac{1}{2}(P_2 Q_3 - P_3 Q_2)$. It is evident that Eq. (6.35) is more convenient to use than Eqs. (6.17) and (6.18). For element 1 consisting of nodes 1 - 2 - 4 corresponding to the local numbering 1 - 2 - 3 as in Fig. 6.8(b),

$$
\begin{aligned}
P_1 &= -1.3, \quad P_2 = 0.9, \quad P_3 = 0.4\,, \\
Q_1 &= -0.2, \quad Q_2 = -0.4, \quad Q_3 = 0.6\,, \\
A &= \frac{1}{2}(0.54 + 0.16) = 0.35
\end{aligned}
$$

Substituting all of these into Eq. (6.35) gives

$$
\left[C^{(1)}\right] = \begin{bmatrix} 1.2357 & -0.7786 & -0.4571 \\ -0.7786 & 0.6929 & 0.0857 \\ -0.4571 & 0.0857 & 0.3714 \end{bmatrix} \tag{6.36}
$$

Similarly, for element 2 consisting of nodes 2 - 3 - 4 corresponding to local numbering 1 - 2 - 3 as in Fig. 6.8(b),

$$
\begin{aligned}
P_1 &= -0.6, \quad P_2 = 1.3, \quad P_3 = -0.7\,, \\
Q_1 &= -0.9, \quad Q_2 = 0.2, \quad Q_3 = 0.7\,, \\
A &= \frac{1}{2}(0.91 + 0.14) = 0.525
\end{aligned}
$$

Hence

$$
\left[C^{(2)}\right] = \begin{bmatrix} 0.5571 & -0.4571 & -0.1 \\ -0.4571 & 0.8238 & -0.3667 \\ -0.1 & -0.3667 & 0.4667 \end{bmatrix} \tag{6.37}
$$

The terms of the global coefficient matrix are obtained as follows:

$$
\begin{aligned}
C_{22} &= C_{22}^{(1)} + C_{11}^{(2)} = 0.6929 + 0.5571 = 1.25 \\
C_{24} &= C_{23}^{(1)} + C_{13}^{(2)} = 0.0857 - 0.1 = -0.0143 \\
C_{44} &= C_{33}^{(1)} + C_{33}^{(2)} = 0.3714 + 0.4667 = 0.8381 \\
C_{21} &= C_{21}^{(1)} = -0.7786 \\
C_{23} &= C_{12}^{(2)} = -0.4571 \\
C_{41} &= C_{31}^{(1)} = -0.4571 \\
C_{43} &= C_{32}^{(2)} = -0.3667
\end{aligned}
$$

Note that we follow local numbering for the element coefficient matrix and global numbering for the global coefficient matrix. Thus

$$[C] = \begin{bmatrix} C_{11}^{(1)} & C_{12}^{(1)} & 0 & C_{13}^{(1)} \\ C_{21}^{(1)} & C_{22}^{(1)} + C_{11}^{(2)} & C_{12}^{(2)} & C_{23}^{(1)} + C_{12}^{(2)} \\ 0 & C_{21}^{(2)} & C_{22}^{(2)} & C_{23}^{(2)} \\ C_{31}^{(1)} & C_{32}^{(1)} + C_{31}^{(2)} & C_{32}^{(2)} & C_{33}^{(1)} + C_{33}^{(2)} \end{bmatrix}$$

$$= \begin{bmatrix} 1.2357 & -0.7786 & 0 & -0.4571 \\ -0.7786 & 1.25 & -0.4571 & -0.0143 \\ 0 & -0.4571 & 0.8238 & -0.3667 \\ -0.4571 & -0.0143 & -0.3667 & 0.8381 \end{bmatrix} \tag{6.38}$$

Note that $\sum_{i=1}^{4} C_{ij} = 0 = \sum_{j=1}^{4} C_{ij}$. This may be used to check if C is properly obtained. We now apply Eq. (6.28) to the free nodes 2 and 4, i.e.,

$$V_2 = -\frac{1}{C_{22}} (V_1 C_{12} + V_3 C_{32} + V_4 C_{42})$$

$$V_4 = -\frac{1}{C_{44}} (V_1 C_{14} + V_2 C_{24} + V_3 C_{34})$$

or

$$V_2 = -\frac{1}{1.25} (-4.571 - 0.0143 V_4) \tag{6.39a}$$

$$V_4 = -\frac{1}{0.8381} (-0.143 V_2 - 3.667) \tag{6.39b}$$

By initially setting $V_2 = 0 = V_4$, we apply Eqs. (6.39a), (6.39b) iteratively. The first iteration gives $V_2 = 3.6568$, $V_4 = 4.4378$ and at the second iteration $V_2 = 3.7075$, $V_4 = 4.4386$. Just after two iterations, we obtain the same results as those from the band matrix method [3]. Thus the iterative technique is faster and is usually preferred for a large number of nodes. Once the values of the potentials at the nodes are known, the potential at any point within the mesh can be determined using Eq. (6.7). ∎

Example 6.2

Write a FORTRAN program to solve Laplace's equation using the finite element method. Apply the program to the two-dimensional problem shown in Fig. 6.9(a). □

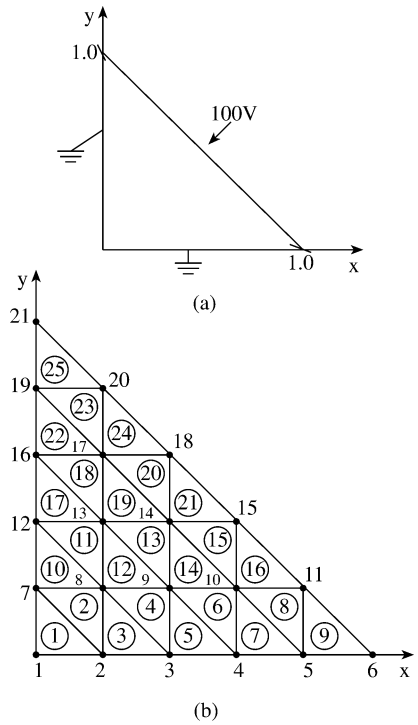


Figure 6.9

For Example 6.2: (a) Two-dimensional electrostatic problem, (b) solution region divided into 25 triangular elements.

Solution

The solution region is divided into 25 three-node triangular elements with total number of nodes being 21 as shown in Fig. 6.9(b). This is a necessary step in order to have input data defining the geometry of the problem. Based on the discussions in Section 6.2, a general FORTRAN program for solving problems involving Laplace's equation using three-node triangular elements is developed as shown in Fig. 6.10. The development of the program basically involves four steps indicated in the program and explained as follows.

Step 1: This involves inputting the necessary data defining the problem. This is the only step that depends on the geometry of the problem at hand. Through a data file, we input the number of elements, the number of nodes, the number of fixed nodes, the prescribed values of the potentials at the free nodes, the x and y coordinates of all nodes, and a list identifying the nodes belonging to each element in the order of the local numbering 1 - 2 - 3. For the problem in Fig. 6.9, the three sets of data for coordinates, element-node relationship, and prescribed potentials at fixed nodes are shown in Tables 6.1, 6.2, and 6.3, respectively.

```
0001    C       FINITE ELEMENT SOLUTION OF LAPLACE'S EQUATION FOR
0002    C       TWO-DIMENSIONAL PROBLEMS
0003    C       TRIANGULAR ELEMENTS ARE USED
0004    C
0005    C       THE UNKNOWN POTENTIALS ARE OBTAINED USING
0006    C       ITERATION METHOD
0007    C
0008    C       ND = NO. OF NODES
0009    C       NE = NO. OF ELEMENTS
0010    C       NP = NO. OF FIXED NODES (WHERE POTENTIAL IS PRESCRIBED)
0011    C       NDP(I) = NODE NO. OF PRESCRIBED POTENTIAL, I = 1,2,...NP
0012    C       VAL(I)) = VALUE OF PRESCRIBED POTENTIAL AT NODE NDP(I)
0013    C       NL(I,J) = LIST OF NODES FOR EACH ELEMENT I, WHERE
0014    c       LF(I) = LIST OF FREE NODES I = 1,2,...,NF=ND-NP
0015    C       J = 1, 2, 3  IS THE LOCAL NODE NUMBER
0016    C       CE(I,J) = ELEMENT COEFFICIENT MATRIX
0017    C       ER(I) = VALUE OF THE RELATIVE PERMITTIVITY FOR ELEMENT I
0018    C       C(I,J)  = GLOBAL COEFFICIENT MATRIX
0019    C       X(I), Y(I) = GLOBAL COORDINATES OF NODE I
0020    C       XL(J), YL(J) = LOCAL COORDINATES OF NODE J = 1,2,3
0021    C       V(I) = POTENTIAL AT NODE I
0022    C       MATRICES P(I) AND Q(I) ARE DEFINED IN EQ.(6.1.1)
0023
0024            DIMENSION  X(100), Y(100), C(100,100), CE(100,100)
0025            DIMENSION  NL(100,3), NDP(100), VAL(100),LF(100)
0026            DIMENSION  V(100),P(3),Q(3),XL(3),YL(3),ER(100)
0027
0028    C       ************************************************
0029    C       FIRST STEP - INPUT DATA DEFINING GEOMETRY AND
0030    C                    BOUNDARY CONDITIONS
0031    C       ************************************************
0032
0033            NI = 50  ! NO. OF ITERATIONS
0034            READ(5,*) NE,ND, NP
0035            READ(5,*)( I, ( NL(I,J), J=1,3),I=1,NE)
0036            READ(5,*) (I, X(I), Y(I), I=1,ND)
0037            READ(5,*) ( NDP(I), VAL(I), I=1,NP)
0038            PIE = 4.0*ATAN(1.0)
0039            E0 = 1.0E-9/(36.0*PIE)
0040            DO 10 I=1,NE
0041            ER(I) = 1.0
0042    10      CONTINUE
0043    C       ************************************************
0044    C       SECOND STEP - EVALUATE COEFFICIENT MATRIX FOR EACH
0045    C                     ELEMENT AND ASSEMBLE GLOBALLY
0046    C       ************************************************
0047            DO 20 M =1, ND
0048            DO 20 N=1,ND
0049            C(M,N) = 0.0
0050    20      CONTINUE
0051            DO 70 I = 1, NE
0052    C       FIND LOCAL COORDINATES XL(J), YL(J) FOR ELEMENT I
0053            DO 30 J=1,3
0054            K=NL(I,J)
0055            XL(J) = X(K)
0056            YL(J) = Y(K)
0057    30      CONTINUE
0058            P(1) = YL(2) - YL(3)
0059            P(2) = YL(3) - YL(1)
0060            P(3) = YL(1) - YL(2)
0061            Q(1) = XL(3) - XL(2)
0062            Q(2) = XL(1) - XL(3)
```

Figure 6.10

Computer program for Example 6.2 ***(Continued).***

```
0063          Q(3) = XL(2) - XL(1)
0064          AREA = 0.5*ABS( P(2)*Q(3) - Q(2)*P(3) )
0065    C     DETERMINE COEFFICIENT MATRIX FOR ELEMENT I
0066          DO 40 M=1,3
0067          DO 40 N=1,3
0068          CE(M,N) = ER(I)*( P(M)*P(N) + Q(M)*Q(N) )/(4.0*AREA)
0069    40    CONTINUE
0070    C     ASSEMBLE GLOBALLY - FIND C(I,J)
0071          DO 60 J=1,3
0072          IR = NL(I,J)
0073          DO 50 L=1,3
0074          IC = NL(I,L)
0075          C(IR,IC) = C(IR,IC) + CE(J,L)
0076    50    CONTINUE
0077    60    CONTINUE
0078    70    CONTINUE
0079    C     ************************************************
0080    C     THIRD STEP - SOLVE THE RESULTING SYSTEM
0081    C                  ITERATIVELY
0082    C     ************************************************
0083    C
0084    C     INITIALIZE AND DETERMINE LF(I) - LIST OF FREE NODES I
0085    C
0086          NF = 0
0087          DO 120 I=1,ND
0088          V(I) = 0.0
0089          DO 110 K=1,NP ! CHECK IF NODE I IS A PRESCRIBED NODE
0090          IF(I.EQ.NDP(K)) THEN
0091               V(I) = VAL(K)
0092          print *, i, v(i)
0093               GO TO 120
0094          ENDIF
0095    110   CONTINUE
0096          NF = NF + 1
0097          LF(NF) = I ! IF I IS NOT A PRESCRIBED NODE, IT IS FREE
0098    120   CONTINUE
0099          PRINT *,NF,ND-NP,'CHECK IF THESE ARE EQUAL'
0100    C
0101    C     NOW, APPLY ITERATIVE METHOD
0102    C
0103           DO 150  N = 1,NI
0104           DO 140 I = 1,NF
0105           SUM = 0.0
0106           K = LF(I)
0107           DO 130 J=1,ND
0108            IF(J.EQ.K) GO TO 130
0109           SUM = SUM + V(J)*C(J,K)
0110    130    CONTINUE
0111           V(K) = - SUM/C(K,K) ! APPLIES ONLY TO FREE NODES
0112    140    CONTINUE
0113    150    CONTINUE
0114    C     ************************************************
0115    C     FOURTH STEP - FINALLY OUTPUT THE RESULTS
0116    C     ************************************************
0117          WRITE(6,170) ND,NE,NP
0118          DO 160 I=1,ND
0119          WRITE(6,*)I, X(I),Y(I),V(I)
0120    160   CONTINUE
0121    170   FORMAT(2X,'NO. OF NODES = ',I3,2X,'NO. OF ELEMENTS =',
0122         1   I3,2X,'NO. OF FIXED NODES = ',I3,/)
0123          STOP
0124          END
```

Figure 6.10
(Cont.) **Computer program for Example 6.2.**

Table 6.1 Nodal Coordinates of the Finite Element Mesh in Fig. 6.9

Node	x	y	Node	x	y
1	0.0	0.0	12	0.0	0.4
2	0.2	0.0	13	0.2	0.4
3	0.4	0.0	14	0.4	0.4
4	0.6	0.0	15	0.6	0.4
5	0.8	0.0	16	0.0	0.6
6	1.0	0.0	17	0.2	0.6
7	0.0	0.2	18	0.4	0.6
8	0.2	0.2	19	0.0	0.8
9	0.4	0.2	20	0.2	0.8
10	0.6	0.2	21	0.0	1.0
11	0.8	0.2			

Table 6.2 Element-Node Identification

	Local node no.				Local node no.		
Element	1	2	3	Element	1	2	3
1	1	2	7	14	9	10	14
2	2	8	7	15	10	15	14
3	2	3	8	16	10	11	15
4	3	9	8	17	12	13	16
5	3	4	9	18	13	17	16
6	4	10	9	19	13	14	17
7	4	5	10	20	14	18	17
8	5	11	10	21	14	15	18
9	5	6	11	22	16	17	19
10	7	8	12	23	17	20	19
11	8	13	12	24	17	18	20
12	8	9	13	25	19	20	21
13	9	14	13				

Step 2: This step entails finding the element coefficient matrix $[C^{(e)}]$ for each element and using the terms to form the global matrix $[C]$.

Step 3: At this stage, we first find the list of free nodes using the given list of prescribed nodes. We now apply Eq. (6.28) iteratively to all the free nodes. The solution converges at 50 iterations or less since only 6 nodes are involved in this case. The solution obtained is exactly the same as those obtained using the band matrix method [3].

Step 4: This involves outputting the result of the computation. The output data for the problem in Fig. 6.9 is presented in Table 6.4. The validity of the result in Table 6.4 is checked using the finite difference method. From the finite difference analysis, the

Table 6.3 Prescribed Potentials at Fixed Nodes

Node	Prescribed potential	Node	Prescribed potential
1	0.0	18	100.0
2	0.0	20	100.0
3	0.0	21	50.0
4	0.0	19	0.0
5	0.0	16	0.0
6	50.0	12	0.0
11	100.0	7	0.0
15	100.0		

Table 6.4 Output Data of the Program in Fig. 6.10. No. of Nodes = 21, No. of Elements = 25, No. of Fixed Nodes = 15

Node	X	Y	Potential
1	0.00	0.00	0.000
2	0.20	0.00	0.000
3	0.40	0.00	0.000
4	0.60	0.00	0.000
5	0.80	0.00	0.000
6	1.00	0.00	50.000
7	0.00	0.20	0.000
8	0.20	0.20	18.182
9	0.40	0.20	36.364
10	0.60	0.20	59.091
11	0.80	0.20	100.000
12	0.00	0.40	0.000
13	0.20	0.40	36.364
14	0.40	0.40	68.182
15	0.60	0.40	100.000
16	0.00	0.60	0.000
17	0.20	0.60	59.091
18	0.40	0.60	100.000
19	0.00	0.80	0.000
20	0.20	0.80	100.000
21	0.00	1.00	50.00

potentials at the free nodes are obtained as:

$$V_8 = 15.41, \quad V_9 = 26.74, \quad V_{10} = 56.69\ ,$$
$$V_{13} = 34.88, \quad V_{14} = 65.41, \quad V_{17} = 58.72V$$

Although the result obtained using finite difference is considered more accurate in this problem, increased accuracy of finite element analysis can be obtained by dividing the solution region into a greater number of triangular elements, or using higher-order elements to be discussed in Section 6.8. As alluded to earlier, the finite element method has two major advantages over the finite difference method. Field quantities are obtained only at discrete positions in the solution region using FDM; they can be obtained at any point in the solution region in FEM. Also, it is easier to handle complex geometries using FEM than using FDM. ∎

6.3 Solution of Poisson's Equation

To solve the two-dimensional Poisson's equation,

$$\nabla^2 V = -\frac{\rho_s}{\epsilon} \tag{6.40}$$

using FEM, we take the same steps as in Section 6.2. Since the steps are essentially the same as in Section 6.2 except that we must include the source term, only the major differences will be highlighted here.

6.3.1 Deriving Element-governing Equations

After the solution region is divided into triangular elements, we approximate the potential distribution $V_e(x, y)$ and the source term ρ_{se} (for two-dimensional problems) over each triangular element by linear combinations of the local interpolation polynomial α_i, i.e.,

$$\boxed{V_e = \sum_{i=1}^{3} V_{ei}\alpha_i(x, y)} \tag{6.41}$$

$$\boxed{\rho_{se} = \sum_{i=1}^{3} \rho_{ei}\alpha_i(x, y)} \tag{6.42}$$

The coefficients V_{ei} and ρ_{ei}, respectively, represent the values of V and ρ_s at vertex i of element e as in Fig. 6.3. The values of ρ_{ei} are known since $\rho_s(x, y)$ is prescribed, while the values of V_{ei} are to be determined.

From Table 4.1, an energy functional whose associated Euler equation is Eq. (6.40) is

$$F(V_e) = \frac{1}{2}\int_S \left[\epsilon\,|\nabla V_e|^2 - 2\rho_{se}V_e\right] dS \tag{6.43}$$

$F(V_e)$ represents the total energy per length within element e. The first term under the integral sign, $\frac{1}{2}\mathbf{D}\cdot\mathbf{E} = \frac{1}{2}\epsilon|\nabla V_e|^2$, is the energy density in the electrostatic system, while the second term, $\rho_{se}V_e dS$, is the work done in moving the charge $\rho_{se}dS$ to its location at potential V_e. Substitution of Eqs. (6.41) and (6.42) into Eq. (6.43) yields

$$F(V_e) = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\epsilon V_{ei}\left[\int \nabla\alpha_i\cdot\nabla\alpha_j\,dS\right]V_{ej} - \sum_{i=1}^{3}\sum_{j=1}^{3}V_{ei}\left[\int \alpha_i\alpha_j\,dS\right]\rho_{ej}$$

This can be written in matrix form as

$$F(V_e) = \frac{1}{2}\epsilon\left[V_e\right]^t\left[C^{(e)}\right]\left[V_e\right] - \left[V_e\right]^t\left[T^{(e)}\right]\left[\rho_e\right] \tag{6.44}$$

where

$$C_{ij}^{(e)} = \int \nabla\alpha_i\cdot\nabla\alpha_j\,dS \tag{6.45}$$

which is already defined in Eq. (6.17) and

$$T_{ij}^{(e)} = \int \alpha_i\alpha_j\,dS \tag{6.46}$$

It will be shown in Section 6.8 that

$$\boxed{T_{ij}^{(e)} = \begin{cases} A/12, & i \neq j \\ A/6 & i = j \end{cases}} \tag{6.47}$$

where A is the area of the triangular element.

Equation (6.44) can be applied to every element in the solution region. We obtain the discretized functional for the whole solution region (with N elements and n nodes) as the sum of the functionals for the individual elements, i.e., from Eq. (6.44),

$$F(V) = \sum_{e=1}^{N} F(V_e) = \frac{1}{2}\epsilon[V]^t[C][V] - [V]^t[T][\rho] \tag{6.48}$$

where t denotes transposition. In Eq. (6.48), the column matrix $[V]$ consists of the values of V_{ei}, while the column matrix $[\rho]$ contains n values of the source function ρ_s at the nodes. The functional in Eq. (6.48) is now minimized by differentiating with respect to V_{ei} and setting the result equal to zero.

6.3.2 Solving the Resulting Equations

The resulting equations can be solved by either the iteration method or the band matrix method as discussed in Section 6.2.4.

Iteration Method: Consider a solution region in Fig. 6.6 having five nodes so that $n = 5$. From Eq. (6.48),

$$F = \frac{1}{2}\epsilon \, [V_1 \; V_2 \; \cdots \; V_5] \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{15} \\ C_{21} & C_{22} & \cdots & C_{25} \\ \vdots & & & \vdots \\ C_{51} & C_{52} & \cdots & C_{55} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_5 \end{bmatrix}$$
$$- \, [V_1 \; V_2 \; \cdots \; V_5] \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{15} \\ T_{21} & T_{22} & \cdots & T_{25} \\ \vdots & & & \vdots \\ T_{51} & T_{52} & \cdots & T_{55} \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_5 \end{bmatrix} \tag{6.49}$$

We minimize the energy by applying

$$\frac{\partial F}{\partial V_k} = 0, \quad k = 1, 2, \ldots, n \tag{6.50}$$

From Eq. (6.49), we get $\dfrac{\partial F}{\partial V_1} = 0$, for example, as

$$\frac{\partial F}{\partial V_1} = \epsilon \, [V_1 C_{11} + V_2 C_{21} + \cdots + V_5 C_{51}] - [T_{11}\rho_1 + T_{21}\rho_2 + \cdots + T_{51}\rho_5] = 0$$

or

$$V_1 = -\frac{1}{C_{11}} \sum_{i=2}^{5} V_i C_{i1} + \frac{1}{\epsilon C_{11}} \sum_{i=1}^{5} T_{i1}\rho_i \tag{6.51}$$

Thus, in general, for a mesh with n nodes

$$\boxed{V_k = -\frac{1}{C_{kk}} \sum_{i=1, i\neq k}^{n} V_i C_{ki} + \frac{1}{\epsilon C_{kk}} \sum_{i=1}^{n} T_{ki}\rho_i} \tag{6.52}$$

where node k is assumed to be a free node.

By fixing the potential at the prescribed nodes and setting the potential at the free nodes initially equal to zero, we apply Eq. (6.52) iteratively to all free nodes until convergence is reached.

Band Matrix Method: If we choose to solve the problem using the band matrix method, we let the free nodes be numbered first and the prescribed nodes last. By

doing so, Eq. (6.48) can be written as

$$F(V) = \frac{1}{2}\epsilon \begin{bmatrix} V_f & V_p \end{bmatrix} \begin{bmatrix} C_{ff} & C_{fp} \\ C_{pf} & C_{pp} \end{bmatrix} \begin{bmatrix} V_f \\ V_p \end{bmatrix} - \begin{bmatrix} V_f & V_p \end{bmatrix} \begin{bmatrix} T_{ff} & T_{fp} \\ T_{pf} & T_{pp} \end{bmatrix} \begin{bmatrix} \rho_f \\ \rho_p \end{bmatrix} \tag{6.53}$$

Minimizing $F(V)$ with respect to V_f, i.e.,

$$\frac{\partial F}{\partial V_f} = 0$$

gives

$$0 = \epsilon \left(C_{ff} V_f + C_{pf} V_p \right) - \left(T_{ff} \rho_f + T_{fp} \rho_p \right)$$

or

$$\boxed{\begin{bmatrix} C_{ff} \end{bmatrix} \begin{bmatrix} V_f \end{bmatrix} = - \begin{bmatrix} C_{fp} \end{bmatrix} \begin{bmatrix} V_p \end{bmatrix} + \frac{1}{\epsilon} \begin{bmatrix} T_{ff} \end{bmatrix} \begin{bmatrix} \rho_f \end{bmatrix} + \frac{1}{\epsilon} \begin{bmatrix} T_{fp} \end{bmatrix} \begin{bmatrix} \rho_p \end{bmatrix}} \tag{6.54}$$

This can be written as

$$[A][V] = [B] \tag{6.55}$$

where $[A] = [C_{ff}]$, $[V] = [V_f]$ and $[B]$ is the right-hand side of Eq. (6.54). Equation (6.55) can be solved to determine $[V]$ either by matrix inversion or Gaussian elimination technique discussed in Appendix D. There is little point in giving examples on applying FEM to Poisson's problems, especially when it is noted that the difference between Eqs. (6.28) and (6.52) or Eqs. (6.54) and (6.31) is slight. See [19] for an example.

6.4 Solution of the Wave Equation

A typical wave equation is the inhomogeneous scalar Helmholtz's equation

$$\nabla^2 \Phi + k^2 \Phi = g \tag{6.56}$$

where Φ is the field quantity (for waveguide problem, $\Phi = H_z$ for TE mode or E_z for TM mode) to be determined, g is the source function, and $k = \omega\sqrt{\mu\epsilon}$ is the wave number of the medium. The following three distinct special cases of Eq. (6.56) should be noted:

(i) $k = 0 = g$: Laplace's equation;

(ii) $k = 0$: Poisson's equation; and

(iii) k is an unknown, $g = 0$: homogeneous, scalar Helmholtz's equation.

We know from Chapter 4 that the variational solution to the operator equation

$$L\Phi = g \tag{6.57}$$

is obtained by extremizing the functional

$$I(\Phi) =< L, \Phi > -2 < \Phi, g > \tag{6.58}$$

Hence the solution of Eq. (6.56) is equivalent to satisfying the boundary conditions and minimizing the functional

$$I(\Phi) = \frac{1}{2}\iint \left[|\nabla\Phi|^2 - k^2\Phi^2 + 2\Phi g\right] dS \tag{6.59}$$

If other than the natural boundary conditions (i.e., Dirichlet or homogeneous Neumann conditions) must be satisfied, appropriate terms must be added to the functional as discussed in Chapter 4.

We now express potential Φ and source function g in terms of the shape functions α_i over a triangular element as

$$\boxed{\Phi_e(x, y) = \sum_{i=1}^{3} \alpha_i \Phi_{ei}} \tag{6.60}$$

$$\boxed{g_e(x, y) = \sum_{i=1}^{3} \alpha_i g_{ei}} \tag{6.61}$$

where Φ_{ei} and g_{ei} are, respectively, the values of Φ and g at nodal point i of element e.

Substituting Eqs. (6.60) and (6.61) into Eq. (6.59) gives

$$\begin{aligned} I(\Phi_e) = {} & \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} \Phi_{ei}\Phi_{ej} \iint \nabla\alpha_i \cdot \nabla\alpha_j \, dS \\ & - \frac{k^2}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} \Phi_{ei}\Phi_{ej} \iint \alpha_i\alpha_j \, dS \\ & + \sum_{i=1}^{3}\sum_{j=1}^{3} \Phi_{ei} g_{ej} \iint \alpha_i\alpha_j \, dS \\ = {} & \frac{1}{2}\left[\Phi_e\right]^t\left[C^{(e)}\right]\left[\Phi_e\right] \\ & - \frac{k^2}{2}\left[\Phi_e\right]^t\left[T^{(e)}\right]\left[\Phi_e\right] + \left[\Phi_e\right]^t\left[T^{(e)}\right]\left[G_e\right] \end{aligned} \tag{6.62}$$

where $[\Phi_e] = [\Phi_{e1}, \Phi_{e2}, \Phi_{e3}]^t$, $[G_e] = [g_{e1}, g_{e2}, g_{e3}]^t$, and $[C^{(e)}]$ and $[T^{(e)}]$ are defined in Eqs. (6.17) and (6.47), respectively.

Equation (6.62), derived for a single element, can be applied for all N elements in the solution region. Thus,

$$I(\Phi) = \sum_{e=1}^{N} I(\Phi_e) \tag{6.63}$$

From Eqs. (6.62) and (6.63), $I(\Phi)$ can be expressed in matrix form as

$$I(\Phi) = \frac{1}{2}[\Phi]^t[C][\Phi] - \frac{k^2}{2}[\Phi]^t[T][\Phi] + [\Phi]^t[T][G] \tag{6.64}$$

where

$$[\Phi] = [\Phi_1, \Phi_2, \ldots, \Phi_N]^t \ , \tag{6.65a}$$

$$[G] = [g_1, g_2, \ldots, g_N]^t \ , \tag{6.65b}$$

$[C]$, and $[T]$ are global matrices consisting of local matrices $[C^{(e)}]$ and $[T^{(e)}]$, respectively.

Consider the special case in which the source function $g = 0$. Again, if free nodes are numbered first and the prescribed nodes last, we may write Eq. (6.64) as

$$\begin{aligned} I = \frac{1}{2}\begin{bmatrix}\Phi_f & \Phi_p\end{bmatrix}\begin{bmatrix}C_{ff} & C_{fp} \\ C_{pf} & C_{pp}\end{bmatrix}\begin{bmatrix}\Phi_f \\ \Phi_p\end{bmatrix} \\ - \frac{k^2}{2}\begin{bmatrix}\Phi_f & \Phi_p\end{bmatrix}\begin{bmatrix}T_{ff} & T_{fp} \\ T_{pf} & T_{pp}\end{bmatrix}\begin{bmatrix}\Phi_f \\ \Phi_p\end{bmatrix} \end{aligned} \tag{6.66}$$

Setting $\dfrac{\partial I}{\partial \Phi_f}$ equal to zero gives

$$\begin{bmatrix}C_{ff} & C_{fp}\end{bmatrix}\begin{bmatrix}\Phi_f \\ \Phi_p\end{bmatrix} - k^2\begin{bmatrix}T_{ff} & T_{fp}\end{bmatrix}\begin{bmatrix}\Phi_f \\ \Phi_p\end{bmatrix} = 0 \tag{6.67}$$

For TM modes, $\Phi_p = 0$ and hence

$$\left[C_{ff} - k^2 T_{ff}\right]\Phi_f = 0 \tag{6.68}$$

Premultiplying by T_{ff}^{-1} gives

$$\boxed{\left[T_{ff}^{-1}C_{ff} - k^2 I\right]\Phi_f = 0} \tag{6.69}$$

Letting

$$A = T_{ff}^{-1}C_{ff}, \quad k^2 = \lambda, \quad X = \Phi_f \tag{6.70a}$$

we obtain the standard eigenproblem

$$(A - \lambda I)X = 0 \tag{6.70b}$$

where I is a unit matrix. Any standard procedure [7] (or see Appendix D) may be used to obtain some or all of the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{n_f}$ and eigenvectors $X_1, X_2, \ldots, X_{n_f}$, where n_f is the number of free nodes. The eigenvalues are always real since C and T are symmetric.

Solution of the algebraic eigenvalue problems in Eq. (6.70) furnishes eigenvalues and eigenvectors, which form good approximations to the eigenvalues and eigenfunctions of the Helmholtz problem, i.e., the cuttoff wavelengths and field distribution patterns of the various modes possible in a given waveguide.

The solution of the problem presented in this section, as summarized in Eq. (6.69), can be viewed as the finite element solution of homogeneous waveguides. The idea can be extended to handle inhomogeneous waveguide problems [8]–[11]. However, in applying FEM to inhomogeneous problems, a serious difficulty is the appearance of spurious, nonphysical solutions. Several techniques have been proposed to overcome the difficulty [12]–[18].

Example 6.3

To apply the ideas presented in this section, we use the finite element analysis to determine the lowest (or dominant) cutoff wavenumber k_c of the TM_{11} mode in waveguides with square $(a \times a)$ and rectangular $(a \times b)$ cross sections for which the exact results are already known as

$$k_c = \sqrt{(m\pi/a)^2 + (n\pi/b)^2}$$

where $m = n = 1$.

It may be instructive to try with hand calculation the case of a square waveguide with 2 divisions in the x and y directions. In this case, there are 9 nodes, 8 triangular elements, and 1 free node ($n_f = 1$). Equation (6.68) becomes

$$C_{11} - k^2 T_{11} = 0$$

where C_{11} and T_{11} are obtained from Eqs. (6.34), (6.35), and (6.47) as

$$C_{11} = \frac{a^2}{2A}, \qquad T_{11} = A, \qquad A = \frac{a^2}{8}$$

Hence

$$k^2 = \frac{a^2}{2A^2} = \frac{32}{a^2}$$

or

$$ka = 5.656$$

which is about 27% off the exact solution. To improve the accuracy, we must use more elements.

The computer program in Fig. 6.11 applies the ideas in this section to find k_c. The main program calls subroutine GRID (to be discussed in Section 6.5) to generate the necessary input data from a given geometry. If n_x and n_y are the number of divisions in the x and y directions, the total number of elements $n_e = 2n_x n_y$. By simply specifying the values of a, b, n_x, and n_y, the program determines k_c using subroutines GRID, INVERSE, and POWER or EIGEN. Subroutine INVERSE available in Appendix D finds T_{ff}^{-1} required in Eq. (6.70a). Either subroutine POWER or EIGEN calculates the eigenvalues. EIGEN finds all the eigenvalues, while POWER only determines the lowest eigenvalue; both subroutines are available in Appendix D. The results for the square ($a = b$) and rectangular ($b = 2a$) waveguides are presented in Tables 6.5a and 6.5b, respectively. □

Table 6.5 (a) Lowest Wavenumber for a Square Waveguide ($b = a$)

n_x	n_e	$k_c a$	% error
2	8	5.656	27.3
3	18	5.030	13.2
5	50	4.657	4.82
7	98	4.553	2.47
10	200	4.497	1.22
Exact: $k_c a = 4.4429$, $n_y = n_x$			

Table 6.5 (b) Lowest Wavenumber for a Rectangular Waveguide ($b = 2a$)

n_x	n_e	$k_c a$	% error
2	16	4.092	16.5
4	64	3.659	4.17
6	144	3.578	1.87
8	256	3.549	1.04
Exact: $k_c a = 3.5124$, $n_y = 2n_x$			

```
0001    C*****************************************************
0002    C      FINITE ELEMENT SOLUTION OF THE WAVE EQUATION
0003    C      TRIANGULAR ELEMENTS ARE USED
0004    C
0005    C      ND = NO. OF NODES
0006    C      NE = NO. OF ELEMENTS
0007    C      NL(I,J) = LIST OF NODES FOR EACH ELEMENT I, WHERE
0008    C      CE(I,J) = ELEMENT COEFFICIENT MATRIX
0009    C      C(I,J)  = GLOBAL COEFFICIENT MATRIX
0010    C      X(I), Y(I) = GLOBAL COORDINATES OF NODE I
0011    C      XL(J), YL(J) = LOCAL COORDINATES OF NODE J = 1,2,3
0012    C      MATRICES P(I) AND Q(I) ARE DEFINED IN EQ.(9)
0013    C      LF(I) = LIST OF FREE NODES
0014    C      ALAM(I) = CONTAINS EIGENVALUES
0015
0016           DIMENSION  C(400,400), CE(400,400),LF(400)
0017           DIMENSION  V(400), P(3), Q(3), XL(3), YL(3)
0018           DIMENSION  T(400,400), A(400,400), ALAM(400)
0019           COMMON X(400),Y(400),DX(50),DY(50),NL(400,3),NDP(400)
0020
0021    C     ***************************************************
0022    C     FIRST STEP - INPUT DATA DEFINING GEOMETRY AND
0023    C                  BOUNDARY CONDITIONS (USE SUBROUTINE GRID)
0024    C     ***************************************************
0025           PRINT *, 'INPUT NX'
0026           READ(5,*)  NX
0027           NY = 2.0*NX
0028           AA = 1.0
0029           BB = 2.0
0030           DELTAX = AA/FLOAT(NX)
0031           DELTAY = BB/FLOAT(NY)
0032           DO 10 I=1,NX
0033           DX(I)=DELTAX
0034    10     CONTINUE
0035           DO 20 I=1,NY
0036           DY(I)=DELTAY
0037    20     CONTINUE
0038           CALL GRID(NX,NY,ND,NE,NP)
0039    C     ***************************************************
0040    C     SECOND STEP - EVALUATE COEFFICIENT MATRIX FOR EACH
0041    C                   ELEMENT AND ASSEMBLE GLOBALLY
0042    C     ***************************************************
0043          DO 30 M =1, ND
0044          DO 30 N=1,ND
0045          C(M,N) = 0.0
0046    30    CONTINUE
0047          DO 80 I = 1, NE
0048          DO 40 J=1,3
0049          K=NL(I,J)
0050          XL(J) = X(K)
0051          YL(J) = Y(K)
0052    40    CONTINUE
0053          P(1) = YL(2) - YL(3)
0054          P(2) = YL(3) - YL(1)
0055          P(3) = YL(1) - YL(2)
0056          Q(1) = XL(3) - XL(2)
0057          Q(2) = XL(1) - XL(3)
0058          Q(3) = XL(2) - XL(1)
0059          AREA = 0.5*ABS( P(2)*Q(3) - Q(2)*P(3) )
0060    C     DETERMINE COEFFICIENT MATRIX FOR ELEMENT I
0061          DO 50 M=1,3
```

Figure 6.11
Computer program for Example 6.3 ***(Continued).***

```
0062          DO 50 M=1,3
0063          CE(M,N) = ( P(M)*P(N) + Q(M)*Q(N) )/(4.0*AREA)
0064     50   CONTINUE
0065     C    ASSEMBLE GLOBALLY - FIND C(I,J) AND T(I,J)
0066          DO 70 J=1,3
0067          IR = NL(I,J)
0068          DO 60 L=1,3
0069          IC = NL(I,L)
0070          C(IR,IC) = C(IR,IC) + CE(J,L)
0071          IF(J.EQ.L) THEN
0072          T(IR,IC) = T(IR,IC) + AREA/6.0
0073          GO TO 60
0074          ELSE
0075             T(IR,IC) = T(IR,IC) + AREA/12.0
0076          ENDIF
0077     60   CONTINUE
0078     70   CONTINUE
0079     80   CONTINUE
0080          PRINT *, 'C AND T HAVE BEEN CALCULATED'
0081     C    ************************************************
0082     C    THIRD STEP - SOLVE THE RESULTING SYSTEM
0083     C    ************************************************
0084     C   DETERMINE LF(I) - LIST OF FREE NODES
0085          NF = 0
0086          DO 100 I=1,ND
0087          DO 90 K=1,NP ! CHECK IF NODE I IS PRESCRIBED
0088          IF(I.EQ.NDP(K)) GO TO 100
0089     90   CONTINUE
0090          NF = NF + 1
0091          LF(NF) = I   ! NODE I IS FREE
0092     100  CONTINUE
0093          PRINT *,NF,ND-NP,'  CHECK IF THESE ARE EQUAL'
0094     C
0095     C   FROM GLOBAL C AND T, FIND C_ff AND T_ff
0096     C
0097          DO 110 I=1,NF
0098          DO 110 J=1,NF
0099          C(I,J) = C(LF(I),LF(J))
0100          T(I,J) = T(LF(I),LF(J))
0101     110  CONTINUE
0102          NMAX = 400
0103          CALL INVERSE(T,NF,NMAX)
0104          DO 120 I = 1,NF
0105          DO 120 J = 1,NF
0106          DO 120 K=1,NF
0107          A(I,J) = A(I,J) + T(I,K)*C(K,J)
0108     120  CONTINUE
0109     C    CALL INVERSE(A,NF,NMAX)
0110     C    CALL POWER(A,ALAMBDA,X,NMAX,NF,IT)
0111          CALL  EIGEN(A,X,NMAX,NF,ALAM)
0112     C    ************************************************
0113     C    FOURTH STEP - OUTPUT THE RESULTS
0114     C    ************************************************
0115          WRITE(6,130) ND,NE,NP
0116     130  FORMAT(2X,'NO. OF NODES = ',I3,2X,'NO. OF ELEMENTS =',
0117        1    I3,2X,'NO. OF PRESCRIBED NODES',I3,/)
0118     C    AK = 1.0/SQRT(ALAMBDA)
0119     C    WRITE (6,*)NX,NY,AK,IT
```

Figure 6.11

(Cont.) **Computer program for Example 6.3** ***(Continued).***

```
0120        DO 140 I=1,NF
0121        ALAM(I) = SQRT( ALAM(I) )
0122        PRINT *,I,ALAM(I)
0123        WRITE(6,*) I,ALAM(I)
0124    140 CONTINUE
0125        STOP
0126        END
```

Figure 6.11
(Cont.) **Computer program for Example 6.3.**

6.5 Automatic Mesh Generation I — Rectangular Domains

One of the major difficulties encountered in the finite element analysis of continuum problems is the tedious and time-consuming effort required in data preparation. Efficient finite element programs must have node and element generating schemes, referred to collectively as *mesh generators*. Automatic mesh generation minimizes the input data required to specify a problem. It not only reduces the time involved in data preparation, it eliminates human errors introduced when data preparation is performed manually. Combining the automatic mesh generation program with computer graphics is particularly valuable since the output can be monitored visually. Since some applications of the FEM to EM problems involve simple rectangular domains, we consider the generation of simple meshes [19] here; automatic mesh generator for arbitrary domains will be discussed in Section 6.6.

Consider a rectangular solution region of size $a \times b$ as in Fig. 6.12. Our goal is to divide the region into rectangular elements, each of which is later divided into two triangular elements. Suppose n_x and n_y are the number of divisions in x and y directions, the total number of elements and nodes are, respectively, given by

$$\begin{aligned} n_e &= 2\, n_x n_y \\ n_d &= (n_x + 1)\left(n_y + 1\right) \end{aligned} \tag{6.71}$$

Thus it is easy to figure out from Fig. 6.12 a systematic way of numbering the elements and nodes. To obtain the global coordinates (x, y) for each node, we need an array containing Δx_i, $i = 1, 2, \ldots, n_x$ and Δy_j, $j = 1, 2, \ldots, n_y$, which are, respectively, the distances between nodes in the x and y directions. If the order of node numbering is from left to right along horizontal rows and from bottom to top along the vertical rows, then the first node is the origin (0,0). The next node is obtained as $x \rightarrow x + \Delta x_1$ while $y = 0$ remains unchanged. The following node has $x \rightarrow x + \Delta x_2$, $y = 0$, and so on until Δx_i are exhausted. We start the second next horizontal row by starting with $x = 0$, $y \rightarrow y + \Delta y_1$ and increasing x until Δx_i are exhausted. We repeat the process until the last node $(n_x + 1)(n_y + 1)$ is reached, i.e., when Δx_i and Δy_i are exhausted simultaneously.

The procedure presented here allows for generating uniform and nonuniform meshes. A mesh is uniform if all Δx_i are equal and all Δy_i are equal; it is nonuniform otherwise. A nonuniform mesh is preferred if it is known in advance that the

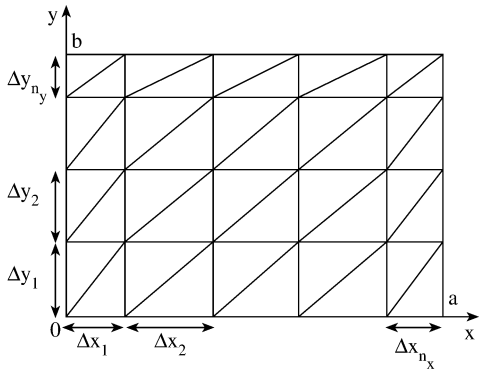


Figure 6.12
Discretization of a rectangular region into a nonuniform mesh.

parameter of interest varies rapidly in some parts of the solution domain. This allows a concentration of relatively small elements in the regions where the parameter changes rapidly, particularly since these regions are often of greatest interest in the solution. Without the preknowledge of the rapid change in the unknown parameter, a uniform mesh can be used. In that case, we set

$$\begin{aligned} \Delta x_1 &= \Delta x_2 = \cdots = h_x \\ \Delta y_1 &= \Delta y_2 = \cdots = h_y \end{aligned} \tag{6.72}$$

where $h_x = a/n_x$ and $h_y = b/n_y$.

In some cases, we also need a list of prescribed nodes. If we assume that all boundary points have prescribed potentials, the number n_p of prescribed node is given by

$$n_p = 2\left(n_x + n_y\right) \tag{6.73}$$

A simple way to obtain the list of boundary points is to enumerate points on the bottom, right, top, and left sides of the rectangular region in that order.

The ideas presented here are implemented in the subroutine GRID in Fig. 6.13. The subroutine can be used for generating a uniform or nonuniform mesh out of a given rectangular region. If a uniform mesh is desired, the required input parameters are a, b, n_x, and n_y. If, on the other hand, a nonuniform mesh is required, we need to supply $n_x, n_y, \Delta x_i, i = 1, 2, \ldots, n_x$, and $\Delta y_j, j = 1, 2, \ldots, n_y$. The output parameters are n_e, n_d, n_p, connectivity list, the global coordinates (x, y) of each node, and the list of prescribed nodes. It is needless to say that subroutine GRID is not useful for a nonrectangular solution region. See the program in Fig. 6.11 as an example on how to use subroutine GRID. A more general program for discretizing a solution region of any shape will be presented in the next section.

```
0001    C*************************************************************
0002    C       SUBROUTINE GRID
0003    C       THIS PROGRAM DIVIDES A RECTANGULAR DOMAIN INTO
0004    C       TRIANGULAR ELEMENTS (NX BY NY NONUNIFORM
0005    C       MESH IN GENERAL)
0006    C       NX & NY ARE THE NO.S OF SUBDIVISION ALONG X & Y AXES
0007    C       NE = NO. OF ELEMENTS IN THE MESH
0008    C       ND = NO. OF NODES IN THE MESH
0009    C       NP = NO. OF BOUNDARY (PRESCRIBED) NODES
0010    C       X(I) & Y(I) ARE GLOBAL COORDINATES OF NODE I
0011    C       DX(I) & DY(I) ARE DISTANCES BETWEEN NODES ALONG X & Y AXES
0012    C       NL(I,J) IS THE LIST OF NODES FOR ELEMENT I, J=1,2,3  ARE
0013    C          LOCAL NUMBERS
0014    C       NDP(I) = LIST OF PRESCRIBED NODES I
0015    C
0016    C     REF:  J. N. REDDY, "AN INTRODUCTION TO THE FINITE ELEMENT
0017    C              METHOD",  NEW YORK: MCGRAW-HILL, 1984, P. 436.
0018
0019            SUBROUTINE GRID(NX,NY,ND,NE,NP)
0020            COMMON X(400),Y(400),DX(50),DY(50),NL(400,3),NDP(400)
0021
0022    C
0023    C   CALCULATE NE, ND, AND NP
0024    C
0025             NE=2*NX*NY
0026             NP = 2*(NX + NY)
0027             NX1=NX + 1
0028             NY1=NY + 1
0029             NXX1=2*NX
0030             NYY1=2*NY
0031             ND=NX1*NY1
0032    C
0033    C   DETERMINE NL(I,J) STARTING FROM LEFT BOTTOM CORNER
0034    C
0035            NL(1,1)=1
0036            NL(1,2)=NX1 + 2
0037            NL(1,3)=NX1 + 1
0038            NL(2,1)=1
0039            NL(2,2)=2
0040            NL(2,3)=NX1 + 2
0041            K=3
0042            DO 50 IY=1,NY
0043            L=IY*NXX1
0044            M=(IY - 1)*NXX1
0045            IF(NX.EQ.1) GO TO 30
0046            DO 20 N=K,L,2
0047            DO 10 I=1,3
0048            NL(N,I)=NL(N-2,I) + 1
0049    10      NL(N+1,I)=NL(N-1,I) + 1
0050    20      CONTINUE
0051    30      IF(NY.EQ.1) GO TO 50
0052            DO 40 I=1,3
0053            NL(L+1,I)=NL(M+1,I) + NX1
0054    40      NL(L+2,I)=NL(M+2,I) + NX1
0055    50      K=L + 3
0056    C
0057    C   DETERMINE X(I) AND Y(I)
0058    C
0059    60      L=0
```

Figure 6.13
Subroutine GRID ***(Continued).***

```
0060            YC=0.0
0061            DO 80 J=1,NY1
0062            XC=0.0
0063            DO 70 I=1,NX1
0064            L=L + 1
0065            X(L)=XC
0066            Y(L)=YC
0067    70      XC=XC + DX(I)
0068    80      YC=YC + DY(J)
0069    C
0070    C   DETERMINE NDP(I)
0071    C
0072             N = 0
0073             DO 90 K=1,NX1
0074             N = N + 1
0075             NDP(N) = N
0076    90       CONTINUE   ! BOTTOM SIDE
0077             DO 100 K=1,NY
0078             N = N+1
0079             NDP(N) = NDP(N-1) + NX1
0080    100      CONTINUE   ! RIGHT SIDE
0081             DO 110 K=1,NX
0082             N = N + 1
0083             NDP(N) = NDP(N-1) - 1
0084    110      CONTINUE   ! TOP SIDE
0085             DO 120 K=1,NY-1
0086             N = N + 1
0087             NDP(N) = NDP(N-1) - NX1
0088    120      CONTINUE   ! LEFT SIDE
0089             WRITE(6,*) NE,ND,NP
0090             DO 130 I=1,NE
0091             WRITE(6,*) I,( NL(I,J),J=1,3)
0092    130      CONTINUE
0093             DO 140 I=1,ND
0094             WRITE(6,*) I,X(I),Y(I)
0095    140      CONTINUE
0096             DO 150 I=1,NP
0097             WRITE(6,*)  NDP(I)
0098    150 CONTINUE
0099            RETURN
0100            END
```

Figure 6.13
(Cont.) **Subroutine GRID.**

6.6 Automatic Mesh Generation II — Arbitrary Domains

As the solution regions become more complex than the ones considered in Section 6.5, the task of developing mesh generators becomes more tedious. A number of mesh generation algorithms (e.g., [21]–[33]) of varying degrees of automation have been proposed for arbitrary solution domains. Reviews of various mesh generation techniques can be found in [34, 35].

The basic steps involved in a mesh generation are as follows [36]:

- subdivide solution region into few quadrilateral blocks,
- separately subdivide each block into elements,

- connect individual blocks.

Each step is explained as follows.

6.6.1 Definition of Blocks

The solution region is subdivided into quadrilateral blocks. Subdomains with different constitutive parameters (σ, μ, ϵ) must be represented by separate blocks. As input data, we specify block topologies and the coordinates at eight points describing each block. Each block is represented by an eight-node quadratic isoparametric element. With natural coordinate system (ζ, η), the x and y coordinates are represented as

$$x(\zeta, \eta) = \sum_{i=1}^{8} \alpha_i(\zeta, \eta)\, x_i \tag{6.74}$$

$$y(\zeta, \eta) = \sum_{i=1}^{8} \alpha_i(\zeta, \eta)\, y_i \tag{6.75}$$

where $\alpha_i(\zeta, \eta)$ is a shape function associated with node i, and (x_i, y_i) are the coordinates of node i defining the boundary of the quadrilateral block as shown in Fig. 6.14. The shape functions are expressed in terms of the quadratic or parabolic isoparametric elements shown in Fig. 6.15. They are given by:

$$\alpha_i = \frac{1}{4}(1 + \zeta\zeta_i)(1 + \eta\eta_i)(\zeta\zeta_i + \eta\eta_i + 1), \qquad i = 1, 3, 5, 7 \tag{6.76}$$

for corner nodes,

$$\alpha_i = \frac{1}{2}\zeta_i^2(1 + \zeta\zeta_i)\left(1 - \eta^2\right) + \frac{1}{2}\eta_i^2(1 + \eta\eta_i + 1)\left(1 - \zeta^2\right), \qquad i = 2, 4, 6, 8 \tag{6.77}$$

for midside nodes. Note the following properties of the shape functions:

(1) They satisfy the conditions

$$\sum_{i=1}^{n} \alpha_i(\zeta, \eta) = 1 \tag{6.78a}$$

$$\alpha_i\left(\zeta_j, \eta_j\right) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{6.78b}$$

(2) They become quadratic along element edges ($\zeta = \pm 1$, $\eta = \pm 1$).

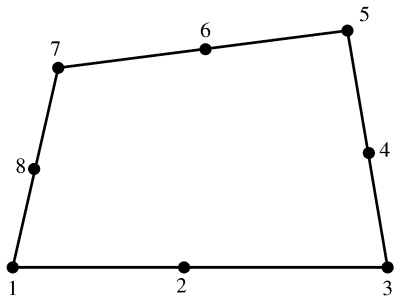


Figure 6.14
Typical quadrilateral block.

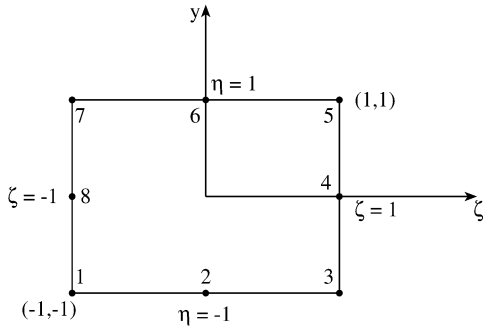


Figure 6.15
Eight-node Serendipity element.

6.6.2 Subdivision of Each Block

For each block, we specify *N DIV X* and *N DIV Y*, the number of element subdivisions to be made in the ζ and η directions, respectively. Also, we specify the weighting factors $(W_\zeta)_i$ and $(W_\eta)_i$ allowing for graded mesh within a block. In specifying *N DIV X*, *N DIV Y*, W_ζ, and W_η care must be taken to ensure that the subdivision along block interfaces (for adjacent blocks) are compatible. We initialize ζ and η to a value of -1 so that the natural coordinates are incremented according to

$$\zeta_i = \zeta_i + \frac{2\left(W_\zeta\right)_i}{W_\zeta^T \cdot F} \tag{6.79}$$

$$\eta_i = \eta_i + \frac{2\left(W_\eta\right)_i}{W_\eta^T \cdot F} \tag{6.80}$$

where

$$W_\zeta^T = \sum_{j=1}^{NDIVX} \left(W_\zeta\right)_j \tag{6.81a}$$

$$W_\eta^T = \sum_{j=1}^{NDIVX} \left(W_\eta\right)_j \tag{6.81b}$$

and

$$F = \begin{cases} 1, & \text{for linear elements} \\ 2, & \text{for quadratic elements} \end{cases}$$

Three element types are permitted: (a) linear four-node quadrilateral elements, (b) linear three-node triangular elements, (c) quadratic eight-node isoparametric elements.

6.6.3 Connection of Individual Blocks

After subdividing each block and numbering its nodal points separately, it is necessary to connect the blocks and have each node numbered uniquely. This is accomplished by comparing the coordinates of all nodal points and assigning the same number to all nodes having identical coordinates. That is, we compare the coordinates of node 1 with all other nodes, and then node 2 with other nodes, etc., until all repeated nodes are eliminated. The listing of the FORTRAN code for automatic mesh generation is shown in Fig. 6.16; it is essentially a modified version of the one in Hinton and Owen [36]. The following example taken from [36] illustrates the application of the code.

Example 6.4

Use the code in Fig. 6.16 to discretize the mesh in Fig. 6.17. □

Solution

The input data for the mesh generation is presented in Table 6.6. The subroutine INPUT reads the number of points (NPOIN) defining the mesh, the number of blocks (NELEM), the element type (NNODE), the number of coordinate dimensions (NDIME), the nodes defining each block, and the coordinates of each node in the mesh. The subroutine GENERATE reads the number of divisions and weighting factors along ζ and η directions for each block. It then subdivides the block into quadrilateral elements. At this point, the whole input data shown in Table 6.6 have been read. The subroutine TRIANGLE divides each four-node quadrilateral element across the shorter diagonal. The subroutine OUTPUT provides the coordinates of the nodes, element topologies, and material property numbers of the generated mesh. For the input data in Table 6.6, the generated mesh with 200 nodes and 330 elements is shown in Fig. 6.18. ■

```
0001    C*************************************************
0002    C THIS PROGRAM PERFORMS A MESH GENERATION OF AN
0003    C ARBITRARY SOLUTION DOMAIN USING A  SYSTEMATIC
0004    C APPROACH. A FEW POINTS ARE  GIVEN TO DETERMINE
0005    C THE  GENERAL  CONFIGURATION  OF  THE REGION.
0006    C THEN THE PROGRAM AUTOMATICALLY GENERATES
0007    C TRIANGULAR OR QUADRILATERAL ELEMENTS
0008    C     REFERENCE: HINTON AND OWEN [36]
0009    C*************************************************
0010          IMPLICIT INTEGER (I-N)
0011          IMPLICIT REAL (A-H,O-Z)
0012          COMMON /MESH1/COORD(1500,2),NL(750,8),MATNO(750),
0013         1 SHAPE(9),NP,NELEM,NTYPE,NDIME,MNODE
0014
0015    C THIS SUBROUTINE ACCEPTS DATA DEFINING THE SOLUTION REGION
0016          CALL INPUT
0017    C THIS SUBROUTINE UNDERTAKES THE MESH SUBDIVISION
0018          CALL GENERATE
0019    C THIS SUBROUTINE SUBDIVIDES INTO TRIANGULAR ELEMENTS
0020          IF(NTYPE.EQ.3)CALL TRIANGLE
0021    C THIS SUBROUTINE OUTPUTS THE GENERATED MESH
0022    C THE SUBROUTINE DOES NOT NEED TO BE CALLED IF A PLOTTING
0023    C SUBROUTINE USED IN DISPLAYING THE OUTPUT
0024          CALL OUTPUT
0025          STOP
0026          END

0001    C*************************************************
0002    C THIS SUBROUTINE ACCEPTS THE DATA WHICH DEFINES THE
0003    C SOLUTION REGION  OUTLINE AND THE MATERIAL ZONES
0004    C NP = NUMBER OF COORDINATE POINTS DEFINING THE
0005    C SOLUTION REGION
0006    C NELEM = NUMBER OF BLOCKS OR ZONES
0007    C NTYPE = THE TYPE OF ELEMENT INTO WHICH THE
0008    C STRUCTURE IS TO BE SUBDIVIDED
0009    C NDIME = THE NUMBER OF COORDINATE DIMENSIONS
0010    C          FOR A PLANE NDIME=2
0011    C NUMEL = BLOCK NUMBER
0012    C (  NL(NUMEL,INODE),INODE=1,NTYPE) )=THE
0013    C BLOCK TOPOLOGY DEFINITION
0014    C MATNO(NUMEL)THE MATERIAL IDENTIFICATION NUMBER:
0015    C INPUT SPECIFICATION FOR EACH BLOCK
0016    C JPOIN = POINT NUMBER
0017    C (COORD(JPOI,IDIME),IDIME=1,NDIME)=X&Y COORDINATES
0018
0019          SUBROUTINE INPUT
0020          COMMON/MESH1/COORD(1500,2),NL(750,8),
0021         1MATNO(750),SHAPE(9),NP,NELEM,NTYPE,NDIME,MNODE
0022          DATA LNODE/8/
0023    C
0024          READ(5,*) NP,NELEM,NTYPE,NDIME
0025          DO 10 IELEM=1,NELEM
0026          READ(5,*) NUMEL,( NL(NUMEL,I), I=1,LNODE ),
0027         1          MATNO(NUMEL)
0028    10    CONTINUE
0029          DO 20 IPOIN=1,NP
0030          READ(5,*)JPOIN, ( COORD(JPOIN,I), I=1,NDIME )
0031    20    CONTINUE
0032          RETURN
0033          END
```

Figure 6.16
FORTRAN code for automatic mesh generation *(Continued)*.

```
0001    C*******************************************************
0002    C THIS SUBROUTINE UNDERTAKES THE SUBDIVISION OF EACH
0003    C BLOCK AND ELIMINATES COMMON NODES ALONG BLOCK INTERFACES
0004    C KBLOC = BLOCK NUMBER
0005    C NDIVX/NDIVY = NUMBER OF ELEMENTS IN THE ZETA/ETA
0006    C DIRECTION INTO WHICH THE BLOCK IS TO BE SUBDIVIDED
0007    C WEITX(IDIVX) AND WEITHY = WEIGHTING FACTORS
0008    C
0009          SUBROUTINE GENERATE
0010          DIMENSION WEITX(40),WEITY(40),TCORD(81,2),
                 TNODS(50,8),
0011         1TMAT0(50),LREPN(350),LASOC(350),LFINN(350),LFASC(350)
0012          COMMON/MESH1/COORD(1500,2),NL(750,8),MATNO(750),
0013         1SHAPE(9),NP,NELEM,NTYPE,NDIME,MNODE
0014          DATA MREPN/350/,MPOIN/1500/,LNODE/8/
0015    C
0016    C INITIALIZATION SECTION
0017    C
0018          DO 10 IREPN=1,MREPN
0019    10    LREPN(IREPN)=0
0020          NPONT=NP
0021          NBLOC=NELEM
0022          NP=0
0023          NELEM=0
0024          MNODE=4
0025          IF(NTYPE.EQ.8)MNODE=8
0026          KNODE=MNODE/4
0027          FNODE=KNODE
0028          DO 20 IPONT=1,NPONT
0029          DO 20 IDIME=1,NDIME
0030    20    TCORD(IPONT,IDIME)=COORD(IPONT,IDIME)
0031          DO 30 IPOIN=1,MPOIN
0032          DO 30 IDIME=1,NDIME
0033    30    COORD(IPOIN,IDIME)=0.0
0034          DO 40 IBLOC=1,NBLOC
0035          TMAT0(IBLOC)=MATNO(IBLOC)
0036          DO 40 INODE=1,LNODE
0037    40    TNODS(IBLOC,INODE)=NL(IBLOC,INODE)
0038    C
0039    C READ AND WRITE BLOCK SUBDIVISION DATA
0040    C
0041          DO 170 IBLOC=1,NBLOC
0042          READ(5,*)KBLOC,NDIVX,NDIVY
0043          READ(5,*) ( WEITX(IDIVX), IDIVX=1,NDIVX )
0044          READ(5,*) ( WEITY(IDIVY), IDIVY=1,NDIVY )
0045    C
0046    C DIVIDE EACH BLOCK INTO ELEMENTS
0047    C
0048          TOTAL=0.0
0049          DO 50 IDIVX=1,NDIVX
0050          IF(WEITX(IDIVX).EQ.0.0)WEITX(IDIVX)=1.0
0051    50    TOTAL=TOTAL+WEITX(IDIVX)
0052          XNORM=2.0/TOTAL
0053          TOTAL=0.0
0054          DO 60 IDIVY=1,NDIVY
0055          IF(WEITY(IDIVY).EQ.0.0)WEITY(IDIVY)=1.0
0056    60    TOTAL=TOTAL+WEITY(IDIVY)
0057          YNORM=2.0/TOTAL
0058          NXTWO=NDIVX*KNODE+1
0059          NYTWO=NDIVY*KNODE+1
0060          IASEY=0
0061          ETASP=-1.0
```

Figure 6.16

(Cont.) **FORTRAN code for automatic mesh generation** ***(Continued).***

```
0062          KWETY=0
0063          DO 160 IYTWO=1,MYTWO
0064          IASEY=IASEY+1
0065          IF(MTYPE.NE.8.AND.IASEY.EQ.3)IASEY=2
0066          IF(MTYPE.EQ.8.AND.IASEY.EQ.4)IASEY=2
0067          IASEX=0
0068          EXISP=-1.0
0069          KWETX=0
0070          DO 130 IXTWO=1,MXTWO
0071          IASEX=IASEX+1
0072          IF(MTYPE.NE.8.AND.IASEX.EQ.3)IASEX=2
0073          IF(MTYPE.EQ.8.AND.IASEX.EQ.4)IASEX=2
0074          MP=MP+1
0075          CALL SHAPEF(EXISP,ETASP)
0076          DO 70 INODE=1,LNODE
0077          JTEMP=TNODS(IBLOC,INODE)
0078          DO 70 IDIME=1,NDIME
0079     70   COORD(MP,IDIME)=COORD(MP,IDIME) +
0080         1           SHAPE(INODE)*TCORD(JTEMP,IDIME)
0081          GO TO (80,90) KNODE
0082     80   IF(IASEX.NE.2.OR.IASEY.NE.2)GO TO 100
0083          MELEM=MELEM+1
0084          JPOIN=MP-MXTWO
0085          ML(MELEM,1)=JPOIN-1
0086          ML(MELEM,2)=JPOIN
0087          ML(MELEM,3)=MP
0088          ML(MELEM,4)=MP-1
0089          MATNO(MELEM)=TMATO(IBLOC)
0090     90   IF(IASEX.NE.3.OR.IASEY.NE.3)GO TO 100
0091          MELEM=MELEM+1
0092          IPOIN=MP-IXTWO-MDIVX+(IXTWO-1)/2
0093          JPOIN=MP-MXTWO-MDIVX-1
0094          ML(MELEM,1)=JPOIN-2
0095          ML(MELEM,2)=JPOIN-1
0096          ML(MELEM,3)=JPOIN
0097          ML(MELEM,4)=IPOIN
0098          ML(MELEM,5)=MP
0099          ML(MELEM,6)=MP-1
0100          ML(MELEM,7)=MP-2
0101          ML(MELEM,8)=IPOIN-1
0102          MATNO(MELEM)=TMATO(IBLOC)
0103     100  CONTINUE
0104          GO TO (110,120),KNODE
0105     110  KWETX=KWETX+1
0106          GO TO 130
0107     120  IF(KONTX.LT.0) KWETX=KWETX + 1
0108          KONTX=KONTX*(-1)
0109     130  EXISP=EXISP+XNORM*WEITX(KWETX)/FNODE
0110          GO TO (140,150),KNODE
0111     140  KWETY=KWETY+1
0112          GO TO 160
0113     150  IF(KONTY.LT.0) KWETY=KWETY + 1
0114          KONTY=KONTY*(-1)
0115     160  ETASP=ETASP+YNORM*WEITY(KWETY)/FNODE
0116     170  CONTINUE
0117     C
0118     C ELIMINATE REPEATED NODES AT BLOCK INTERFACES
0119     C
0120          NREPN=0
0121          DO 210 IPOIN=1,MP
0122          IF(NREPN.EQ.0)GO TO 190
0123          DO 180 IREPN=1,NREPN
0124          IF(IPOIN.EQ.LREPN(IREPN))GO TO 210
```

Figure 6.16
(Cont.) **FORTRAN code for automatic mesh generation** ***(Continued).***

```
0125    180   CONTINUE
0126    190   CONTINUE
0127          LPOIN=IPOIN+1
0128          DO 200 JPONT=LPOIN,NP
0129          TOTAL=ABS(COORD(IPOIN,1)-COORD(JPONT,1)) +
0130         1      ABS(COORD(IPOIN,2)-COORD(JPONT,2))
0131          IF(TOTAL.GT.0.00001)GO TO 200
0132          NREPN=NREPN+1
0133          LREPN(NREPN)=JPONT
0134          LASOC(NREPN)=IPOIN
0135    200   CONTINUE
0136    210   CONTINUE
0137          IF(NREPN.EQ.0)GO TO 360
0138          INDEX=0
0139          DO 240 IPOIN=1,NP
0140          DO 220 IREPN=1,NREPN
0141          IF(LREPN(IREPN).EQ.IPOIN)GO TO 230
0142    220   CONTINUE
0143          GO TO 240
0144    230   INDEX=INDEX+1
0145          LFINN(INDEX)=LREPN(IREPN)
0146          LFASC(INDEX)=LASOC(IREPN)
0147    240   CONTINUE
0148          DO 250 IREPN=1,NREPN
0149          LREPN(IREPN)=LFINN(IREPN)
0150    250   LASOC(IREPN)=LFASC(IREPN)
0151          DO 260 IREPN=1,NREPN
0152          DO 260 IELEM=1,NELEM
0153          DO 260 INODE=1,MNODE
0154          IF(NL(IELEM,INODE).EQ.LREPN(IREPN))
0155         1NL(IELEM,INODE)=LASOC(IREPN)
0156    260   CONTINUE
0157          DO 310 IPOIN=1,NP
0158          DO 270 IREPN=1,NREPN
0159          IF(IPOIN.EQ.LREPN(IREPN)) GO TO 310
0160    270   CONTINUE
0161          IF(IPOIN.LT.LREPN(1))GO TO 310
0162          IDIFF=IPOIN-NREPN
0163          IF(IPOIN.GT.LREPN(NREPN))GO TO 290
0164          DO 280 IREPN=1,NREPN
0165          KREPN=NREPN-IREPN+1
0166    280   IF(IPOIN.LT.LREPN(KREPN))IDIFF=IPOIN-KREPN+1
0167    290   DO 300 IDIME=1,NDIME
0168    300   COORD(IDIFF,IDIME)=COORD(IPOIN,IDIME)
0169    310   CONTINUE
0170          DO 350 IELEM=1,NELEM
0171          DO 350 INODE=1,MNODE
0172          NPOSI=NL(IELEM,INODE)
0173          DO 320 IREPN=1,NREPN
0174          IF(NPOSI.EQ.LREPN(IREPN))GO TO 350
0175    320   CONTINUE
0176          IF(NPOSI.LT.LREPN(1))GO TO 350
0177          IDIFF=NPOSI-NREPN
0178          IF(NPOSI.GT.LREPN(NREPN))GO TO 340
0179          DO 330 IREPN=1,NREPN
0180          KREPN=NREPN-IREPN+1
0181    330   IF(NPOSI.LT.LREPN(KREPN))IDIFF=NPOSI-KREPN+1
0182    340   NL(IELEM,INODE)=IDIFF
0183    350   CONTINUE
0184    360   CONTINUE
0185          NP=NP-NREPN
0186          RETURN
0187          END
```

Figure 6.16

(Cont.) **FORTRAN code for automatic mesh generation.** ***(Continued).***

```
0001    C*************************************************
0002    C THIS SUBROUTINE EVALUATES THE SHAPE FUNCTIONS
0003    C
0004          SUBROUTINE SHAPEF(S,T)
0005          COMMON/MESH1/COORD(1500,2),ML(750,8),
0006         1MATNO(750),SHAPE(9),NP,NELEM,NTYPE,NDIME,MNODE
0007
0008          SHAPE(1)=0.25*(1.0-S)*(1.0-T)*(-S-T-1.0)
0009          SHAPE(2)=0.5*(1.0-S*S)*(1.0-T)
0010          SHAPE(3)=0.25*(1.0+S)*(1.0-T)*(S-T-1.0)
0011          SHAPE(4)=0.5*(1.0-T*T)*(1.0+S)
0012          SHAPE(5)=0.25*(1.0+S)*(1.0+T)*(S+T-1.0)
0013          SHAPE(6)=0.5*(1.0-S*S)*(1.0+T)
0014          SHAPE(7)=0.25*(1.0-S)*(1.0+T)*(-S+T-1.0)
0015          SHAPE(8)=0.5*(1.0-T*T)*(1.0-S)
0016          RETURN
0017          END

0001    C*************************************************
0002    C THIS SUBROUTINE SUBDIVIDES EACH 4-NODED
0003    C QUADRILATERAL ELEMENT INTO TWO TRIANGULAR
0004    C ELEMENTS:THE SUBDIVISION IS DONE ACROSS THE
0005    C SHORTER DIAGONAL
0006    C
0007          SUBROUTINE TRIANGLE
0008          DIMENSION CORDE(4,2),LTEMP(4)
0009          COMMON/MESH1/ COORD(1500,2),ML(750,8),
0010         1MATNO(750),SHAPE(9),NP,NELEM,NTYPE,NDIME,MNODE
0011    C
0012          KOUNT=0
0013          DO 10 IELEM=1,NELEM
0014          NOTAL=NELEM+IELEM
0015          MATNO(NOTAL)=MATNO(IELEM)
0016          DO 10 INODE=1,MNODE
0017    10    ML(NOTAL,INODE)=ML(IELEM,INODE)
0018          DO 40 IELEM=1,NELEM
0019          NOTAL=NELEM+IELEM
0020          DO 20 INODE=1,MNODE
0021          INDEX=ML(NOTAL,INODE)
0022          LTEMP(INODE)=INDEX
0023          DO 20 IDIME=1,NDIME
0024    20    CORDE(INODE,IDIME)=COORD(INDEX,IDIME)
0025          DIAG1=SQRT((CORDE(1,1)-CORDE(3,1))**2 +
0026         1      (CORDE(1,2)-CORDE(3,2))**2)
0027          DIAG2=SQRT((CORDE(2,1)-CORDE(4,1))**2 +
0028         1      (CORDE(2,2)-CORDE(4,2))**2)
0029    C
0030    C DIVIDE ACROSS THE SHORTER DIAGONAL
0031    C
0032          DIFER=DIAG1-DIAG2
0033          IF(DIFER.GT.1.0E-9)GO TO 30
0034          KOUNT=KOUNT+1
0035          ML(KOUNT,1)=LTEMP(1)
0036          ML(KOUNT,2)=LTEMP(2)
0037          ML(KOUNT,3)=LTEMP(3)
0038          MATNO(KOUNT)=MATNO(NOTAL)
0039          KOUNT=KOUNT+1
0040          ML(KOUNT,1)=LTEMP(1)
0041          ML(KOUNT,2)=LTEMP(3)
0042          ML(KOUNT,3)=LTEMP(4)
0043          MATNO(KOUNT)=MATNO(NOTAL)
0044          GO TO 40
```

Figure 6.16

(Cont.) **FORTRAN code for automatic mesh generation.** ***(Continued).***

```
0045    30    KOUNT=KOUNT+1
0046          NL(KOUNT,1)=LTEMP(1)
0047          NL(KOUNT,2)=LTEMP(2)
0048          NL(KOUNT,3)=LTEMP(4)
0049          MATNO(KOUNT)=MATNO(NOTAL)
0050          KOUNT=KOUNT+1
0051          NL(KOUNT,1)=LTEMP(2)
0052          NL(KOUNT,2)=LTEMP(3)
0053          NL(KOUNT,3)=LTEMP(4)
0054          MATNO(KOUNT)=MATNO(NOTAL)
0055    40    CONTINUE
0056          NELEM=2*NELEM
0057          RETURN
0058          END

0001    C************************************************
0002    C THIS SUBROUTINE OUTPUTS THE COORDINATES AND
0003    C ELEMENT TOPOLOGIES OF THE GENERATED MESH
0004    C
0005          SUBROUTINE OUTPUT
0006          COMMON/MESH1/ COORD(1500,2),NL(750,8),
0007         1 MATNO(750),SHAPE(9),NP,NELEM,NTYPE,NDIME,MNODE
0008    C
0009          WRITE(6,*)NP  ! TOTAL NO. OF POINTS
0010          WRITE(6,*)NELEM  ! TOTAL NO. OF ELEMENTS
0011          DO 10 IPOIN=1,NP
0012    10    WRITE(6,*)IPOIN,( COORD(IPOIN,I), I=1,NDIME )
0013          DO 20 IELEM=1,NELEM
0014    20    WRITE(6,*)IELEM,(NL(IELEM,I),I=1,NTYPE),MATNO(IELEM)
0015          RETURN
0016          END
```

Figure 6.16
(Cont.) **FORTRAN code for automatic mesh generation.**

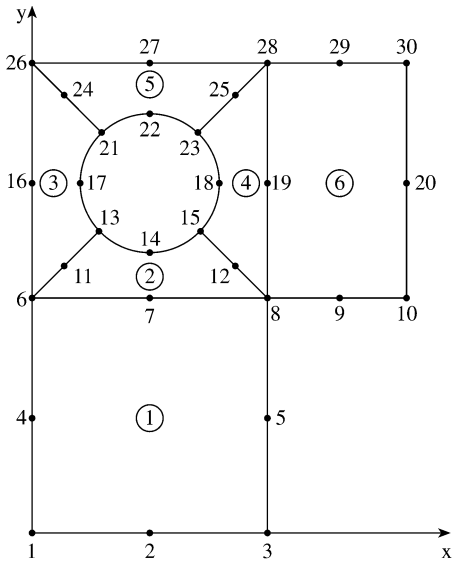


Figure 6.17
Solution region of Example 6.4.

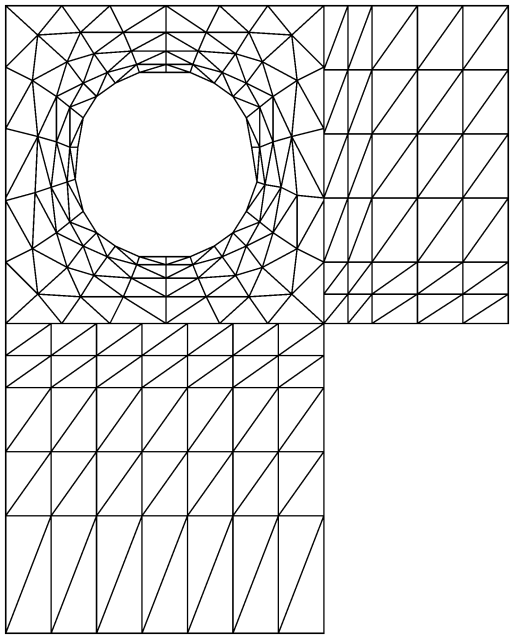


Figure 6.18
The generated mesh corresponding to input data in Table 6.6.

6.7 Bandwidth Reduction

Since most of the matrices involved in FEM are symmetric, sparse, and banded, we can minimize the storage requirements and the solution time by storing only the elements involved in half bandwidth instead of storing the whole matrix. To take the fullest advantage of the benefits from using a banded matrix solution technique, we must make sure that the matrix bandwidth is as narrow as possible.

If we let d be the maximum difference between the lowest and the highest node numbers of any single element in the mesh, we define the semi-bandwidth B (which includes the diagonal term) of the coefficient matrix $[C]$ as

$$B = (d + 1) f \tag{6.82}$$

where f is the number of degrees of freedom (or number of parameters) at each node. If, for example, we are interested in calculating the electric field intensity $\mathbf{E}$ for a three-dimensional problem, then we need E_x, E_y, and E_z at each node, and $f = 3$ in this case. Assuming that there is only one parameter per node,

$$B = d + 1 \tag{6.83}$$

Table 6.6 Input Data for Automatic Mesh Generation for the Solution Region in Fig. 6.17

```
30    6    3          2
1     1    2          3          5          8          7          6          4          1
2     6    7          8          12         15         14         13         11         1
3     6    11         13         17         21         24         26         16         1
4     8    19         28         25         23         18         15         12         1
5     21   22         23         25         28         27         26         24         1
6     8    9          10         20         30         29         28         19         2

1               0.0               0.0
2               2.5               0.0
3               5.0               0.0
4               0.0               2.5
5               5.0               2.5
6               0.0               5.0
7               2.5               5.0
8               5.0               5.0
9               6.5               5.0
10              8.0               5.0
11              0.7196            5.7196
12              4.2803            5.7196
13              1.4393            6.4393
14              2.5               6.0
15              3.5607            6.6493
16              0.0               7.5
17              1.0               7.5
18              4.0               7.5
19              5.0               7.5
20              8.0               7.5
21              1.4393            8.5607
22              2.5               9.0
23              3.5607            8.5607
24              0.7196            9.2805
25              4.2803            9.2805
26              0.0               10.0
27              2.5               10.0
28              2.5               10.0
29              6.5               10.0
30              8.0               10.0
1     7          5
1.0             1.0               1.0                  1.0        1.0        1.0        1.0
2.0             1.0               1.0                  0.5        0.5
2     7          4
1.0             1.0               1.0                  1.0        1.0        1.0        1.0
1.0             0.75              0.5                  0.25
3     4          6
1.0             0.75              0.5                  0.25
1.0             1.0               1.0                  1.0        1.0        1.0
4     6          4
1.0             1.0               2.0        2.0       2.0        2.0
1.0             0.75              0.5        0.25
5     6          4         1.0     1.0        1.0      1.0        1.0        1.0        1.0
0.25            0.5               0.75       1.0
6     5          6
1.0             1.0               2.0        2.0       2.0
1.0             1.0               2.0        2.0       2.0        2.0
```

The semi-bandwidth, which does not include the diagonal term, is obtained from Eq. (6.82) or (6.83) by subtracting one from the right-hand side, i.e., for $f = 1$,

$$B = d \tag{6.84}$$

Throughout our discussion in this section, we will stick to the definition of semi-bandwidth in Eq. (6.84). The total bandwidth may be obtained from Eq. (6.84) as $2B + 1$.

The bandwidth of the global coefficient matrix depends on the node numbering. Hence, to minimize the bandwidth, the node numbering should be selected to minimize d. Good node numbering is usually such that nodes with widely different numbers are widely separated. To minimize d, we must number nodes across the narrowest part of the region.

Consider, for the purpose of illustration, the mesh shown in Fig. 6.19. If the mesh is numbered originally as in Fig. 6.19, we obtain d_e for each element e as

$$d_1 = 2,\ d_2 = 3,\ d_3 = 4,\ d_4 = 5,\ d_5 = 6,\ d_6 = 7 \tag{6.85}$$

From this, we obtain

$$d = \text{ maximum } d_e = 7$$

or

$$B = 7 \tag{6.86}$$

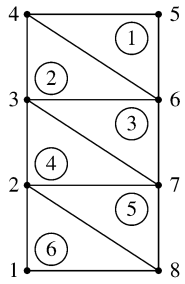


Figure 6.19
Original mesh with $B = 7$.

Alternatively, the semi-bandwidth may be determined from the coefficient matrix, which is obtained by mere inspection of Fig. 6.19 as

$$[C] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \overset{\longleftarrow B=7 \longrightarrow}{\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array}} \\ \begin{bmatrix} x & x & & & & & & x \\ x & x & x & & & & x & x \\ & x & x & x & & x & & \\ & & x & x & x & x & & \\ & & & x & x & x & & \\ & & x & x & x & x & x & \\ & x & & & & x & x & x \\ x & x & & & & & & x \end{bmatrix} \end{array} \tag{6.87}$$

where x indicates a possible nonzero term and blanks are zeros (i.e., $C_{ij} = 0$, indicating no coupling between nodes i and j). If the mesh is renumbered as in Fig. 6.20(a),

$$d_1 = 4 = d_2 = d_3 = d_4 = d_5 = d_6 \tag{6.88}$$

and hence

$$d = \text{ maximum } d_e = 4$$

or

$$B = 4 \tag{6.89}$$

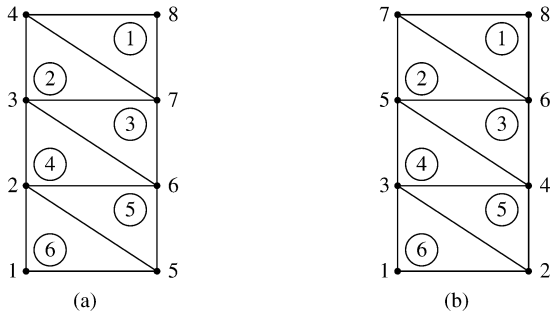


Figure 6.20
Renumbered nodes: (a) $B = 4$, (b) $B = 2$.

Finally, we may renumber the mesh as in Fig. 6.20(b). In this case

$$d_1 = 2 = d_2 = d_3 = d_4 = d_5 = d_6 \tag{6.90}$$

and

$$d = \text{ maximum } d_e = 2 \tag{6.91}$$

or

$$B = 2 \tag{6.92}$$

The value $B = 2$ may also be obtained from the coefficient matrix for the mesh in Fig. 6.20(b), namely,

$$[C] = \begin{array}{c} \\ \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \overset{B=2}{\longleftrightarrow} \\ \mathbf{P} \quad \mathbf{Q} \\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ \left[\begin{array}{cccccccc} x & x & x & & & & & \\ x & x & x & x & & & & \\ x & x & x & x & x & & & \\ & x & x & x & x & & & \\ & & x & x & x & x & x & \\ & & & x & x & x & x & x \\ & & & & x & x & x & x \\ & & & & & x & x & x \end{array}\right] \\ \\ \\ \\ \\ \mathbf{R} \\ \\ \mathbf{S} \end{array} \tag{6.93}$$

From Eq. (6.93), one immediately notices that $[C]$ is symmetric and that terms are clustered in a band about the diagonal. Hence $[C]$ is sparse and banded so that only the data within the area **PQRS** of the matrix need to be stored—a total of 21 terms out of 64. This illustrates the savings in storage by a careful nodal numbering.

For a simple mesh, hand-labeling coupled with a careful inspection of the mesh (as we have done so far) can lead to a minimum bandwidth. However, for a large mesh, a hand-labeling technique becomes a tedious, time-consuming task, which in most cases may not be successful. It is particularly desirable that an automatic relabeling scheme is implemented within a mesh generation program. A number of algorithms have been proposed for bandwidth reduction by automatic mesh renumbering [37]–[40]. A simple, efficient algorithm is found in Collins [37].

6.8 Higher Order Elements

The finite elements we have used so far have been the linear type in that the shape function is of the order one. A higher order element is one in which the shape function or interpolation polynomial is of the order two or more.

The accuracy of a finite element solution can be improved by using finer mesh or using higher order elements or both. A discussion on mesh refinement versus higher order elements is given by Desai and Abel [2]; a motivation for using higher order elements is given by Csendes in [41]. In general, fewer higher order elements are needed to achieve the same degree of accuracy in the final results. The higher order elements are particularly useful when the gradient of the field variable is expected to vary rapidly. They have been applied with great success in solving EM-related problems [4], [41]–[46].

6.8.1 Pascal Triangle

Higher order triangular elements can be systematically developed with the aid of the so-called Pascal triangle given in Fig. 6.21. The family of finite elements generated in this manner with the distribution of nodes illustrated in Fig. 6.22. Note that in higher order elements, some secondary (side and/or interior) nodes are introduced in addition to the primary (corner) nodes so as to produce exactly the right number of nodes required to define the shape function of that order. The Pascal triangle contains terms of the basis functions of various degrees in variables x and y. An arbitrary function $\Phi_i(x, y)$ can be approximated in an element in terms of a complete nth order polynomial as

$$\Phi(x, y) = \sum_{i=1}^{m} \alpha_i \Phi_i \tag{6.94}$$

where

$$m = \frac{1}{2}(n + 1)(n + 2) \tag{6.95}$$

is the number of terms in complete polynomials (also the number of nodes in the triangle). For example, for second order ($n = 2$) or quadratic (six-node) triangular elements,

$$\Phi_e(x, y) = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 \tag{6.96}$$

This equation has six coefficients, and hence the element must have six nodes. It is also complete through the second order terms. A systematic derivation of the interpolation function α for the higher order elements involves the use of the local coordinates.

$$\begin{array}{ccccccccccc}
 & & & & a_1 & & & & & & \\
 & & & a_2x & & & a_3y & & & & \\
 & & a_4x^2 & & a_5xy & & & a_6y^2 & & & \\
 & a_7x^3 & & a_8x^2y & & a_9xy^2 & & & a_{10}y^3 & & \\
a_{11}x^4 & & a_{12}x^3y & & a_{13}x^2y^2 & & a_{14}xy^3 & & & & a_{15}y^4
\end{array}$$

Figure 6.21

The Pascal Triangle. The first row is: (constant, $n = 0$), the second: (linear, $n = 1$), the third: (quadratic, $n = 2$), the fourth: (cubic, $n = 3$), the fifth: (quartic, $n = 4$).

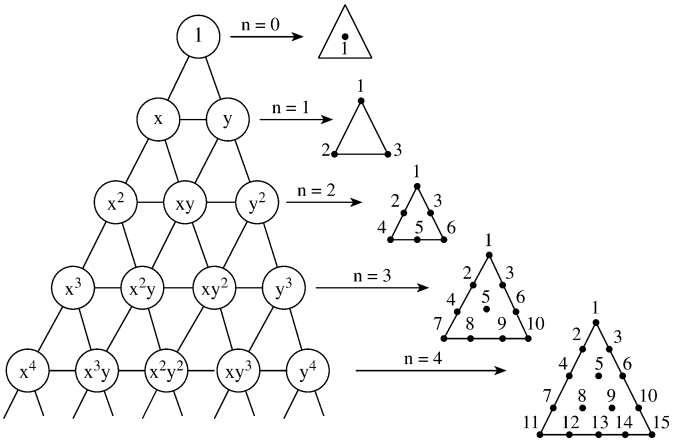


Figure 6.22

The Pascal triangle and the associated polynomial basis function for degree $n = 1$ to 4.

6.8.2 Local Coordinates

The triangular local coordinates (ξ_1, ξ_2, ξ_3) are related to Cartesian coordinates (x, y) as

$$x = \xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 \tag{6.97}$$

$$y = \xi_1 y_1 + \xi_2 y_2 + \xi_3 y_3 \tag{6.98}$$

The local coordinates are dimensionless with values ranging from 0 to 1. By definition, ξ_i at any point within the triangle is the ratio of the perpendicular distance from the point to the side opposite to vertex i to the length of the altitude drawn from vertex i. Thus, from Fig. 6.23 the value of ξ_1 at P, for example, is given by the ratio of the perpendicular distance d from the side opposite vertex 1 to the altitude h of that side, i.e.,

$$\xi_1 = \frac{d}{h} \tag{6.99}$$

Alternatively, from Fig. 6.23, ξ_i at P can be defined as

$$\xi_i = \frac{A_i}{A} \tag{6.100}$$

so that

$$\xi_1 + \xi_2 + \xi_3 = 1 \tag{6.101}$$

since $A_1 + A_2 + A_3 = A$. In view of Eq. (6.100), the local coordinates ξ_i are also called *area coordinates*. The variation of (ξ_1, ξ_2, ξ_3) inside an element is shown in

Fig. 6.24. Although the coordinates ξ_1, ξ_2, and ξ_3 are used to define a point P, only two are independent since they must satisfy Eq. (6.101). The inverted form of Eqs. (6.97) and (6.98) is

$$\xi_i = \frac{1}{2A}\left[c_i + b_i x + a_i y\right] \tag{6.102}$$

where

$$\begin{aligned} a_i &= x_k - x_j \ , \\ b_i &= y_j - y_k \ , \\ c_i &= x_j y_k - x_k y_j \\ A = &\quad \text{area of the triangle} \quad = \frac{1}{2}\left(b_1 a_2 - b_2 a_1\right) \ , \end{aligned} \tag{6.103}$$

and (i, j, k) is an even permutation of (1,2,3). (Notice that a_i and b_i are the same as Q_i and P_i in Eq. (6.34).) The differentiation and integration in local coordinates are carried out using [47]:

$$\frac{\partial f}{\partial \xi_1} = a_2 \frac{\partial f}{\partial x} - b_2 \frac{\partial f}{\partial y} \tag{6.104a}$$

$$\frac{\partial f}{\partial \xi_2} = -a_1 \frac{\partial f}{\partial x} + b_1 \frac{\partial f}{\partial y} \tag{6.104b}$$

$$\frac{\partial f}{\partial x} = \frac{1}{2A}\left(b_1 \frac{\partial f}{\partial \xi_1} + b_2 \frac{\partial f}{\partial \xi_2}\right) \tag{6.104c}$$

$$\frac{\partial f}{\partial y} = \frac{1}{2A}\left(a_1 \frac{\partial f}{\partial \xi_1} + a_2 \frac{\partial f}{\partial \xi_2}\right) \tag{6.104d}$$

$$\iint f \, dS = 2A \int_0^1 \left[\int_0^{1-\xi_2} f\left(\xi_1, \xi_2\right) \, d\xi_1 \right] d\xi_2 \tag{6.104e}$$

$$\iint \xi_1^i \xi_2^j \xi_3^k \, dS = \frac{i!\, j!\, k!}{(i + j + k + 2)!} 2A \tag{6.104f}$$

$$dS = 2A \, d\xi_1 \, d\xi_2 \tag{6.104g}$$

6.8.3 Shape Functions

We may now express the shape function for higher order elements in terms of local coordinates. Sometimes, it is convenient to label each point in the finite elements in Fig. 6.22 with three integers i, j, and k from which its local coordinates (ξ_1, ξ_2, ξ_3) can be found or vice versa. At each point P_{ijk}

$$(\xi_1, \xi_2, \xi_3) = \left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}\right) \tag{6.105}$$

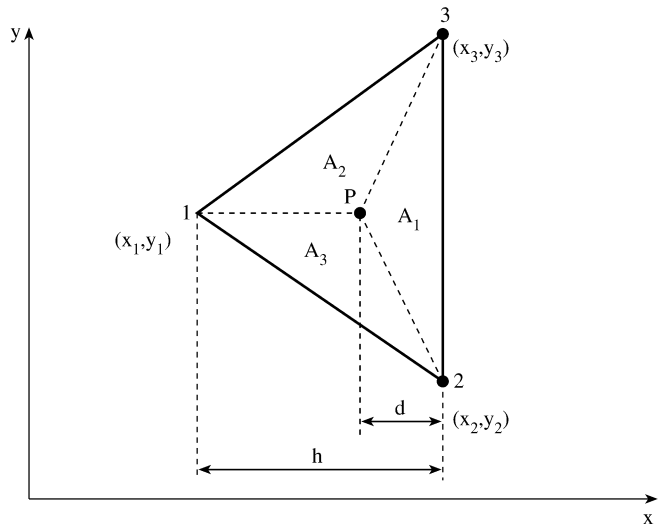


Figure 6.23
Definition of local coordinates.

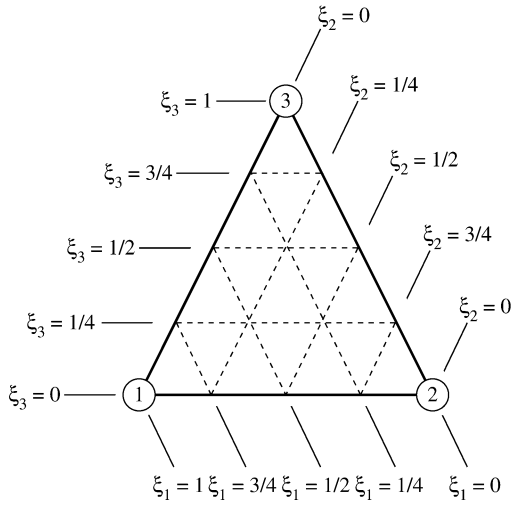


Figure 6.24
Variation of local coordinates.

Hence if a value of Φ, say Φ_{ijk}, is prescribed at each point P_{ijk}, Eq. (6.94) can be written as

$$\Phi(\xi_1, \xi_2, \xi_3) = \sum_{i=1}^{m} \sum_{j=1}^{m-i} \alpha_{ijk}(\xi_1, \xi_2, \xi_3)\, \Phi_{ijk} \tag{6.106}$$

where

$$\alpha_\ell = \alpha_{ijk} = p_i(\xi_1)\, p_j(\xi_2)\, p_k(\xi_3), \quad \ell = 1, 2, \ldots \tag{6.107}$$

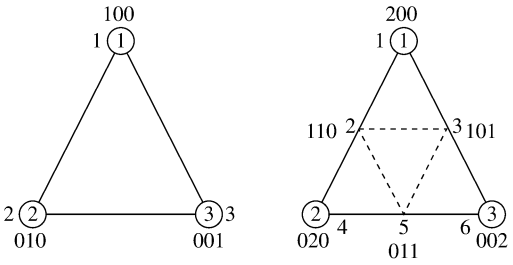
$$p_r(\xi) = \begin{cases} \dfrac{1}{r!} \displaystyle\prod_{t=0}^{r-1} (n\xi - t), & r > 0 \\ 1, & r = 0 \end{cases} \tag{6.108}$$

and $r \in (i, j, k)$. $p_r(\xi)$ may also be written as

$$p_r(\xi) = \frac{(n\xi - r + 1)}{r} p_{r-1}(\xi), \quad r > 0 \tag{6.109}$$

where $p_0(\xi) = 1$.

The relationships between the subscripts $q \in \{1, 2, 3\}$ on ξ_q, $\ell \in \{1, 2, \ldots, m\}$ on α_ℓ, and $r \in (i, j, k)$ on p_r and P_{ijk} in Eqs. (6.107) to (6.109) are illustrated in Fig. 6.25 for n ranging from 1 to 4. Henceforth point P_{ijk} will be written as P_n for conciseness.



(a) n = 1 (b) n = 2

Figure 6.25

Distribution of nodes over triangles for $n = 1$ to 4. The triangles are in standard position *(Continued)*.

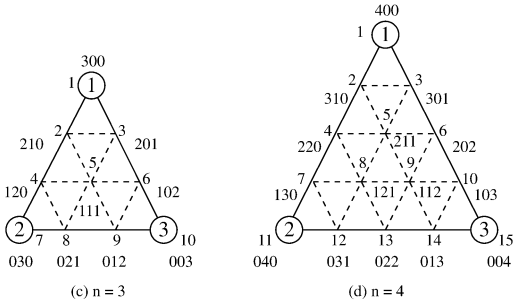


Figure 6.25

(Cont.) **Distribution of nodes over triangles for $n = 1$ to 4. The triangles are in standard position.**

Notice from Eq. (6.108) or Eq. (6.109) that

$$\begin{aligned} p_0(\xi) &= 1 \\ p_1(\xi) &= n\xi \\ p_2(\xi) &= \frac{1}{2}(n\xi - 1)n\xi \\ p_3(\xi) &= \frac{1}{6}(n\xi - 2)(n\xi - 1)n\xi \\ p_4(\xi) &= \frac{1}{24}(n\xi - 3)(n\xi - 2)(n\xi - 1)n\xi, \quad \text{etc} \end{aligned} \tag{6.110}$$

Substituting Eq. (6.110) into Eq. (6.107) gives the shape functions α_ℓ for nodes $\ell = 1, 2, \ldots, m$, as shown in Table 6.7 for $n = 1$ to 4. Observe that each α_ℓ takes the value of 1 at node ℓ and value of 0 at all other nodes in the triangle. This is easily verified using Eq. (6.105) in conjunction with Fig. 6.25.

6.8.4 Fundamental Matrices

The fundamental matrices $[T]$ and $[Q]$ for triangular elements can be derived using the shape functions in Table 6.7. (For simplicity, the brackets [ ] denoting a matrix quantity will be dropped in the remaining part of this section.) In Eq. (6.46), the T matrix is defined as

$$T_{ij} = \iint \alpha_i \alpha_j \, dS \tag{6.46}$$

From Table 6.7, we substitute α_ℓ in Eq. (6.46) and apply Eqs. (6.104f) and (6.104g) to obtain elements of T. For example, for $n = 1$,

$$T_{ij} = 2A \int_0^1 \int_0^{1-\xi_2} \xi_i \xi_j \, d\xi_1 \, d\xi_2$$

Table 6.7 Polynomial Basis Function $\alpha_\ell(\xi_1, \xi_2, \xi_3, \xi_4)$ for First-, Second-, Third-, and Fourth-Order

$n = 1$	$n = 2$	$n = 3$	$n = 4$
$\alpha_1 = \xi_1$	$\alpha_1 = \xi_1(2\xi_1 - 1)$	$\alpha_1 = \frac{1}{2}\xi_1(3\xi_1 - 2)(3\xi_1 - 1)$	$\alpha_1 = \frac{1}{6}\xi_1(4\xi_1 - 3)(4\xi_1 - 2)(4\xi_1 - 1)$
$\alpha_2 = \xi_2$	$\alpha_2 = 4\xi_1\xi_2$	$\alpha_2 = \frac{9}{2}\xi_1(3\xi_1 - 1)\xi_2$	$\alpha_2 = \frac{8}{3}\xi_1(4\xi_1 - 2)(4\xi_1 - 1)\xi_2$
$\alpha_3 = \xi_3$	$\alpha_3 = 4\xi_1\xi_3$	$\alpha_3 = \frac{9}{2}\xi_1(3\xi_1 - 1)\xi_3$	$\alpha_3 = \frac{8}{3}\xi_1(4\xi_1 - 2)(4\xi_1 - 1)\xi_3$
	$\alpha_4 = \xi_2(2\xi_2 - 1)$	$\alpha_4 = \frac{9}{2}\xi_1(3\xi_2 - 1)\xi_2$	$\alpha_4 = 4\xi_1(4\xi_1 - 1)(4\xi_2 - 1)\xi_2$
	$\alpha_5 = 4\xi_2\xi_3$	$\alpha_5 = 27\xi_1\xi_2\xi_3$	$\alpha_5 = 32\xi_1(4\xi_1 - 1)\xi_2\xi_3$
	$\alpha_6 = \xi_3(2\xi_3 - 1)$	$\alpha_6 = \frac{9}{2}\xi_1(3\xi_3 - 1)\xi_3$	$\alpha_6 = 4\xi_1(4\xi_1 - 1)(4\xi_3 - 1)\xi_3$
		$\alpha_7 = \frac{1}{2}\xi_2(3\xi_2 - 2)(3\xi_2 - 1)$	$\alpha_7 = \frac{8}{3}\xi_1(4\xi_2 - 2)(4\xi_2 - 1)\xi_2$
		$\alpha_8 = \frac{9}{2}\xi_2(3\xi_2 - 1)\xi_3$	$\alpha_8 = 32\xi_1(4\xi_2 - 1)\xi_2\xi_3$
		$\alpha_9 = \frac{9}{2}\xi_2(3\xi_3 - 1)\xi_3$	$\alpha_9 = 32\xi_1\xi_2(4\xi_3 - 1)\xi_3$
		$\alpha_{10} = \frac{1}{2}\xi_3(3\xi_3 - 2)(3\xi_3 - 1)$	$\alpha_{10} = \frac{8}{3}\xi_1(4\xi_3 - 2)(4\xi_3 - 1)\xi_3$
			$\alpha_{11} = \frac{1}{6}\xi_2(4\xi_2 - 3)(4\xi_2 - 2)(4\xi_2 - 1)$
			$\alpha_{12} = \frac{8}{3}\xi_2(4\xi_2 - 2)(4\xi_2 - 1)\xi_3$
			$\alpha_{13} = 4\xi_2(4\xi_2 - 1)(4\xi_3 - 1)\xi_3$
			$\alpha_{14} = \frac{8}{3}\xi_2(4\xi_3 - 2)(4\xi_3 - 1)\xi_3$
			$\alpha_{15} = \frac{1}{6}\xi_3(4\xi_3 - 3)(4\xi_3 - 2)(4\xi_3 - 1)$

When $i \neq j$,

$$T_{ij} = \frac{2A(1!)(1!)(0!)}{4!} = \frac{A}{12}\,, \tag{6.111a}$$

when $i = j$,

$$T_{ij} = \frac{2A(2!)}{4!} = \frac{A}{6} \tag{6.111b}$$

Hence

$$T = \frac{A}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \tag{6.112}$$

By following the same procedure, higher order T matrices can be obtained. The T matrices for orders up to $n = 4$ are tabulated in Table 6.8 where the factor A, the area

of the element, has been suppressed. The actual matrix elements are obtained from Table 6.8 by multiplying the tabulated numbers by A and dividing by the indicated common denominator. The following properties of the T matrix are noteworthy:

(a) T is symmetric with positive elements;

(b) elements of T all add up to the area of the triangle, i.e., $\sum_{i}^{m}\sum_{j}^{m} T_{ij} = A$, since by definition $\sum_{\ell=1}^{m} \alpha_\ell = 1$ at any point within the element;

(c) elements for which the two triple subscripts form similar permutations are equal, i.e., $T_{ijk,prq} = T_{ikj,prq} = T_{kij,rpq} = T_{kji,rqp} = T_{jki,qrp} = T_{jik,qpr}$; this should be obvious from Eqs. (6.46) and (6.107).

These properties are not only useful in checking the matrix, they have proved useful in saving computer time and storage. It is interesting to know that the properties are independent of coordinate system [46].

Table 6.8 Table of T Matrix for $n = 1$ to 4 *(Continued)*

$n = 1$ Common denominator: 12

$$\begin{array}{rrr} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{array}$$

$n = 2$ Common denominator: 180

$$\begin{array}{rrrrrr} 6 & 0 & 0 & -1 & -4 & -1 \\ 0 & 32 & 16 & 0 & 16 & -4 \\ 0 & 16 & 32 & -4 & 16 & 0 \\ -1 & 0 & -4 & 6 & 0 & -1 \\ -4 & 16 & 16 & 0 & 32 & 0 \\ -1 & -4 & 0 & -1 & 0 & 6 \end{array}$$

$n = 3$ Common denominator: 6720

$$\begin{array}{rrrrrrrrrr} 76 & 18 & 18 & 0 & 36 & 0 & 11 & 27 & 27 & 11 \\ 18 & 540 & 270 & -189 & 162 & -135 & 0 & -135 & -54 & 27 \\ 18 & 270 & 540 & -135 & 162 & -189 & 27 & -54 & -135 & 0 \\ 0 & -189 & -135 & 540 & 162 & -54 & 18 & 270 & -135 & 27 \\ 36 & 162 & 162 & 162 & 1944 & 162 & 36 & 162 & 162 & 36 \\ 0 & -135 & -189 & -54 & 162 & 540 & 27 & -135 & 270 & 18 \\ 11 & 0 & 27 & 18 & 36 & 27 & 76 & 18 & 0 & 11 \\ 27 & -135 & -54 & 270 & 162 & -135 & 18 & 540 & -189 & 0 \\ 27 & -54 & -135 & -135 & 162 & 270 & 0 & -189 & 540 & 18 \\ 11 & 27 & 0 & 27 & 36 & 18 & 11 & 0 & 18 & 76 \end{array}$$

Table 6.8 *(Cont.)* Table of T Matrix for $n = 1$ to 4

$n = 4$	Common denominator: 56700												
290	160	160	−80	160	−80	0	−160	−160	0	−27	−112	−12	−112	−27
160	2560	1280	−1280	1280	−960	768	256	−256	512	0	512	64	256	−112
160	1280	2560	−960	1280	−1280	512	−256	256	768	−112	256	64	512	0
−80	−1280	−960	3168	384	48	−1280	384	−768	64	−80	−960	48	64	−12
160	1280	1280	384	10752	384	256	−1536	−1536	256	−160	−256	−768	−256	−160
−80	−960	−1280	48	384	3168	64	−768	384	−1280	−12	64	48	−960	−80
0	768	512	−1280	256	64	2560	1280	−256	256	160	1280	−960	512	−112
−160	256	−256	384	−1536	−768	1280	10752	−1536	−256	160	1280	384	256	−160
−160	−256	256	−768	−1536	384	−256	−1536	10752	1280	−160	256	384	1280	160
0	512	768	64	256	−1280	256	−256	1280	2560	−112	512	−960	1280	160
−27	0	−112	−80	−160	−12	160	160	−160	−112	290	160	−80	0	−27
−112	512	256	−960	−256	64	1280	1280	256	512	160	2560	−1280	768	0
−12	64	64	48	−768	48	−960	384	384	−960	−80	−1280	3168	−1280	−80
−112	256	512	64	−256	−960	512	256	1280	1280	0	768	−1280	2560	160
−27	−112	0	−12	−160	−80	−112	−160	160	160	−27	0	−80	160	290

In Eq. (6.14) or Eq. (6.45), elements of $[C]$ matrix are defined by

$$C_{ij} = \iint \left(\frac{\partial \alpha_i}{\partial x} \frac{\partial \alpha_j}{\partial x} + \frac{\partial \alpha_i}{\partial y} \frac{\partial \alpha_j}{\partial y} \right) dS \tag{6.113}$$

By applying Eqs. (6.104a) to (6.104d) to Eq. (6.113), it can be shown that [4, 43]

$$C_{ij} = \frac{1}{2A} \sum_{q=1}^{3} \cot \theta_q \iint \left(\frac{\partial \alpha_i}{\partial \xi_{q+1}} - \frac{\partial \alpha_i}{\partial \xi_{q-1}} \right) \left(\frac{\partial \alpha_j}{\partial \xi_{q+1}} - \frac{\partial \alpha_j}{\partial \xi_{q-1}} \right) dS$$

or

$$\boxed{C_{ij} = \sum_{q=1}^{3} Q_{ij}^{(q)} \cot \theta_q} \tag{6.114}$$

where θ_q is the included angle of vertex $q \in \{1, 2, 3\}$ of the triangle and

$$\boxed{Q_{ij}^{(q)} = \iint \left(\frac{\partial \alpha_i}{\partial \xi_{q+1}} - \frac{\partial \alpha_i}{\partial \xi_{q-1}} \right) \left(\frac{\partial \alpha_j}{\partial \xi_{q+1}} - \frac{\partial \alpha_j}{\partial \xi_{q-1}} \right) d\xi_1 \, d\xi_2} \tag{6.115}$$

We notice that matrix C depends on the triangle shape, whereas the matrices $Q^{(q)}$ do not. The $Q^{(1)}$ matrices for $n = 1$ to 4 are tabulated in Table 6.9. The following properties of Q matrices should be noted:

(a) they are symmetric;

(b) the row and column sums of any Q matrix are zero, i.e., $\sum_{i=1}^{m} Q_{ij}^{(q)} = 0 = \sum_{j=1}^{m} Q_{ij}^{(q)}$ so that the C matrix is singular.

$Q^{(2)}$ and $Q^{(3)}$ are easily obtained from $Q^{(1)}$ by row and column permutations so that the matrix C for any triangular element is constructed easily if $Q^{(1)}$ is known. One approach [48] involves using a rotation matrix R similar to that in Silvester and Ferrari [4], which is essentially a unit matrix with elements rearranged to correspond to one rotation of the triangle about its centroid in a counterclockwise direction. For example, for $n = 1$, the rotation matrix is basically derived from Fig. 6.26 as

$$\boxed{R = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}} \tag{6.116}$$

where $R_{ij} = 1$ node i is replaced by node j after one counterclockwise rotation, or $R_{ij} = 0$ otherwise. Table 6.10 presents the R matrices for $n = 1$ to 4. Note that each

Table 6.9 Table of Q Matrices for $n = 1$ to 4 *(Continued)*

$n = 1$ Common denominator: 2

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

$n = 2$ Common denominator: 6

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & -8 & 0 & 0 & 0 \\ 0 & -8 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -4 & 1 \\ 0 & 0 & 0 & -4 & 8 & -4 \\ 0 & 0 & 0 & 1 & -4 & 3 \end{pmatrix}$$

$n = 3$ Common denominator: 80

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 135 & -135 & -27 & 0 & 27 & 3 & 0 & 0 & -3 \\ 0 & -135 & 135 & 27 & 0 & -27 & -3 & 0 & 0 & 3 \\ 0 & -27 & 27 & 135 & -162 & 27 & 3 & 0 & 0 & -3 \\ 0 & 0 & 0 & -162 & 324 & -162 & 0 & 0 & 0 & 0 \\ 0 & 27 & -27 & 27 & -162 & 135 & -3 & 0 & 0 & 3 \\ 0 & 3 & -3 & 3 & 0 & -3 & 34 & -54 & 27 & -7 \\ 0 & 0 & 0 & 0 & 0 & 0 & -54 & 135 & -108 & 27 \\ 0 & 0 & 0 & 0 & 0 & 0 & 27 & -108 & 135 & -54 \\ 0 & -3 & 3 & -3 & 0 & 3 & -7 & 27 & -54 & 34 \end{pmatrix}$$

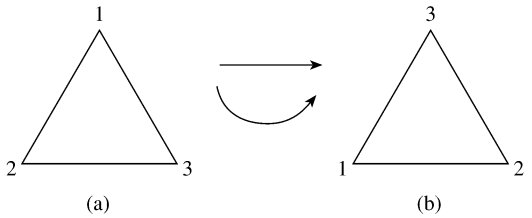


Figure 6.26
One counterclockwise rotation of the triangle in (a) gives the triangle in (b).

row or column of R has only one nonzero element since R is essentially a unit matrix with rearranged elements.

Once the R is known, we obtain

$$\boxed{Q^{(2)} = RQ^{(1)}\,R^t} \tag{6.117a}$$

$$\boxed{Q^{(3)} = RQ^{(2)}\,R^t} \tag{6.117b}$$

where R^t is the transpose of R.

Table 6.9 *(Cont.)* Table of Q Matrices for $n = 1$ to 4

$n = 4$	Common denominator: 1890													
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3968	−3968	−1440	0	1440	640	0	0	−640	−80	0	0	0	80
0	−3968	3968	1440	0	−1440	−640	0	0	640	80	0	0	0	−80
0	−1440	1440	4632	−5376	744	−1248	768	768	−288	80	−128	96	−128	80
0	0	0	−5376	10752	−5376	1536	−1536	−1536	1536	−160	256	−192	256	−160
0	1440	−1440	744	−5376	4632	−288	768	768	−1248	80	−128	96	−128	80
0	640	−640	−1248	1536	−288	3456	−4608	1536	−384	240	−256	192	−256	80
0	0	0	768	−1536	768	−4608	10752	−7680	1536	−160	256	−192	256	−160
0	0	0	768	−1536	768	1536	−7680	10752	−4608	−160	256	−192	256	−160
0	−640	640	−288	1536	−1248	−384	1536	−4608	3456	80	−256	192	−256	240
0	−80	80	80	−160	80	240	−160	−160	80	705	−1232	884	−464	107
0	0	0	−128	256	−128	−256	256	256	−256	−1232	3456	−3680	1920	−464
0	0	0	96	−192	96	192	−192	−192	192	884	−3680	5592	−3680	884
0	0	0	−128	256	−128	−256	256	256	−256	−464	1920	−3680	3456	−1232
0	80	−80	80	−160	80	80	−160	−160	240	107	−464	884	−1232	705

Table 6.10 *R* Matrix for $n = 1$ to 4

$n = 1$

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

$n = 2$

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

$n = 3$

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

$n = 4$

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

Example 6.5

For $n = 2$, calculate $Q^{(1)}$ and obtain $Q^{(2)}$ from $Q^{(1)}$ using Eq. (6.117a). □

Solution

By definition,

$$Q_{ij}^{(1)} = \iint \left(\frac{\partial \alpha_i}{\partial \xi_2} - \frac{\partial \alpha_i}{\partial \xi_3}\right)\left(\frac{\partial \alpha_j}{\partial \xi_2} - \frac{\partial \alpha_j}{\partial \xi_3}\right) d\xi_1\, d\xi_2$$

For $n = 2$, $i, j = 1, 2, \ldots, 6$, and α_i are given in terms of the local coordinates in Table 6.7. Since $Q^{(1)}$ is symmetric, only some of the elements need be calculated. Substituting for α_ℓ from Table 6.7 and applying Eqs. (6.104e) and (6.104f), we obtain

$$\begin{aligned}
Q_{1j} &= 0, \quad j = 1 \text{ to } 6\,, \\
Q_{i1} &= 0, \quad i = 1 \text{ to } 6\,, \\
Q_{22} &= \frac{1}{2A}\iint (4\xi_1)^2\, d\xi_1\xi_2 = \frac{8}{6}\,, \\
Q_{23} &= \frac{1}{2A}\iint (4\xi_1)\,(-4\xi_1)\, d\xi_1\xi_2 = -\frac{8}{6}\,, \\
Q_{24} &= \frac{1}{2A}\iint (4\xi_1)\,(4\xi_1 - 1)\, d\xi_1\xi_2 = 0 = Q_{26}\,, \\
Q_{25} &= \frac{1}{2A}\iint (4\xi_1)\,(4\xi_3 - 4\xi_2)\, d\xi_1\xi_2 = 0\,, \\
Q_{33} &= \frac{1}{2A}\iint (-4\xi_1)^2\, d\xi_1\xi_2 = \frac{8}{6}\,, \\
Q_{34} &= \frac{1}{2A}\iint (-4\xi_1)\,(4\xi_2 - 1)\, d\xi_1\xi_2 = 0 = Q_{36}\,, \\
Q_{35} &= \frac{1}{2A}\iint (-4\xi_1)\,(4\xi_3 - 4\xi_2)\, d\xi_1\xi_2 = 0\,, \\
Q_{44} &= \frac{1}{2A}\iint (4\xi_2 - 1)^2\, d\xi_1\xi_2 = \frac{3}{6}\,, \\
Q_{45} &= \frac{1}{2A}\iint (4\xi_2 - 1)\,(4\xi_3 - 4\xi_2)\, d\xi_1\xi_2 = -\frac{4}{6}\,, \\
Q_{46} &= \frac{1}{2A}\iint (4\xi_2 - 1)\,(4\xi_3 - 1)\,(-1)\, d\xi_1\xi_2 = \frac{1}{6}\,, \\
Q_{55} &= \frac{1}{2A}\iint (4\xi_3 - 4\xi_2)^2\, d\xi_1\xi_2 = \frac{8}{6}\,, \\
Q_{56} &= \frac{1}{2A}\iint (4\xi_3 - 4\xi_2)\,(-1)\,(4\xi_3 - 1)\, d\xi_1\xi_2 = -\frac{4}{6}\,, \\
Q_{66} &= \frac{1}{2A}\iint (-1)\,(4\xi_3 - 1)^2\, d\xi_1\xi_2 = \frac{3}{6}
\end{aligned}$$

Hence

$$Q^{(1)} = \frac{1}{6}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & -8 & 0 & 0 & 0 \\ 0 & -8 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -4 & 1 \\ 0 & 0 & 0 & -4 & 8 & -4 \\ 0 & 0 & 0 & 1 & -4 & 3 \end{bmatrix}$$

We now obtain $Q^{(2)}$ from

$$\begin{aligned} Q^{(2)} &= RQ^{(1)}R^t \\ &= \frac{1}{6}R\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & -8 & 0 & 0 & 0 \\ 0 & -8 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -4 & 1 \\ 0 & 0 & 0 & -4 & 8 & -4 \\ 0 & 0 & 0 & 1 & -4 & 3 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \frac{1}{6}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -8 & 0 & 0 & 8 & 0 \\ 0 & 8 & 0 & 0 & -8 & 0 \\ 1 & 0 & -4 & 0 & 0 & 3 \\ -4 & 0 & 8 & 0 & 0 & -4 \\ 3 & 0 & 4 & 0 & 0 & 1 \end{bmatrix} \\ Q^{(2)} &= \frac{1}{6}\begin{bmatrix} 3 & 0 & -4 & 0 & 0 & 1 \\ 0 & 8 & 0 & 0 & -8 & 0 \\ -4 & 0 & 8 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -8 & 0 & 0 & 8 & 0 \\ 1 & 0 & -4 & 0 & 0 & 3 \end{bmatrix} \quad \blacksquare \end{aligned}$$

6.9 Three-Dimensional Elements

The finite element techniques developed in the previous sections for two-dimensional elements can be extended to three-dimensional elements. One would expect three-dimensional problems to require a large total number of elements to achieve an accurate result and demand a large storage capacity and computational time. For the sake of completeness, we will discuss the finite element analysis of Helmholtz's equation in three dimensions, namely,

$$\nabla^2\Phi + k^2\Phi = g \tag{6.118}$$

We first divide the solution region into tetrahedral or hexahedral (rectangular prism) elements as in Fig. 6.27. Assuming a four-node tetrahedral element, the function Φ

is represented within the element by

$$\Phi_e = a + bx + cy + dz \tag{6.119}$$

The same applies to the function g. Since Eq. (6.119) must be satisfied at the four nodes of the tetrahedral elements,

$$\Phi_{ei} = a + bx_i + cy_i + dz_i, \qquad i = 1, \ldots, 4 \tag{6.120}$$

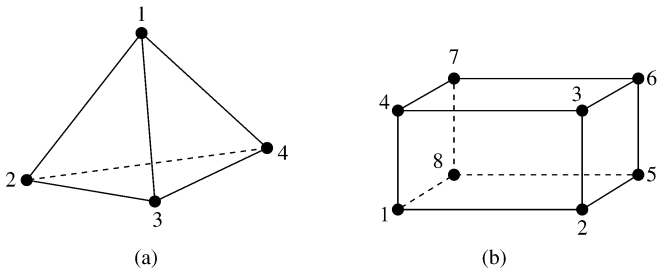


Figure 6.27
Three-dimensional elements: (a) Four-node or linear-order tetrahedral, (b) eight-node or linear-order hexahedral.

Thus we have four simultaneous equations (similar to Eq. (6.5)) from which the coefficients $a, b, c,$ and d can be determined. The determinant of the system of equations is

$$\det = \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix} = 6v \,, \tag{6.121}$$

where v is the volume of the tetrahedron. By finding $a, b, c,$ and d, we can write

$$\boxed{\Phi_e = \sum_{i=1}^{4} \alpha_i(x, y)\Phi_{ei}} \tag{6.122}$$

where

$$\alpha_1 = \frac{1}{6v} \begin{vmatrix} 1 & x & y & z \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix} \,, \tag{6.123a}$$

$$\alpha_2 = \frac{1}{6v} \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x & y & z \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix} \,, \tag{6.123b}$$

with α_3 and α_4 having similar expressions. For higher order approximation, the matrices for αs become large in size and we resort to local coordinates. Another motivation for using local coordinates is the existence of integration equations which simplify the evaluation of the fundamental matrices T and Q.

For the tetrahedral element, the local coordinates are ξ_1, ξ_2, ξ_3, and ξ_4, each perpendicular to a side. They are defined at a given point as the ratio of the distance from that point to the appropriate apex to the perpendicular distance from the side to the opposite apex. They can also be interpreted as volume ratios, i.e., at a point P

$$\xi_i = \frac{v_i}{v} \tag{6.124}$$

where v_i is the volume bound by P and face i. It is evident that

$$\sum_{i=1}^{4} \xi_i = 1 \tag{6.125a}$$

or

$$\xi_4 = 1 - \xi_1 - \xi_2 - \xi_3 \tag{6.125b}$$

The following properties are useful in evaluating integration involving local coordinates [47]:

$$dv = 6v \, d\xi_1 \, d\xi_2 \, d\xi_3 \, , \tag{6.126a}$$

$$\iiint f dv = 6v \int_0^1 \left[\int_0^{1-\xi_3} \left(\int_0^{1-\xi_2-\xi_3} f \, d\xi_1 \right) d\xi_2 \right] d\xi_3 \, , \tag{6.126b}$$

$$\iiint \xi_1^i \xi_2^j \xi_3^k \xi_4^\ell \, dv = \frac{i!j!k!\ell!}{(i+j+k+\ell+3)!} 6v \tag{6.126c}$$

In terms of the local coordinates, an arbitrary function $\Phi(x, y)$ can be approximated within an element in terms of a complete nth order polynomial as

$$\boxed{\Phi_e(x, y) = \sum_{i=1}^{m} \alpha_i(x, y) \Phi_{ei}} \tag{6.127}$$

where $m = \frac{1}{6}(n+1)(n+2)(n+3)$ is the number of nodes in the tetrahedron or the number of terms in the polynomial. The terms in a complete three-dimensional polynomial may be arrayed as shown in Fig. 6.28.

Each point in the tetrahedral element is represented by four integers i, j, k, and ℓ which can be used to determine the local coordinates $(\xi_1, \xi_2, \xi_3, \xi_4)$. That is at $P_{ijk\ell}$,

$$(\xi_1, \xi_2, \xi_3, \xi_4) = \left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}, \frac{\ell}{n} \right) \tag{6.128}$$

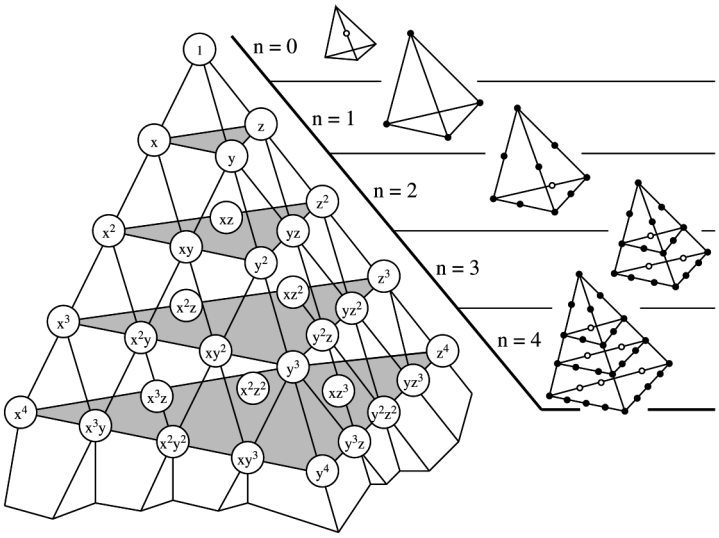


Figure 6.28
Pascal tetrahedron and associated array of terms.

Hence at each node,

$$\alpha_q = \alpha_{ijk\ell} = p_i\,(\xi_1)\; p_j\,(\xi_2)\; p_k\,(\xi_3)\; p_\ell\,(\xi_4)\;\;, \tag{6.129}$$

where $q = 1, 2, \ldots, m$ and p_r is defined in Eq. (6.108) or (6.109). The relationship between the node numbers q and $ijk\ell$ is illustrated in Fig. 6.29 for the second order tetrahedron ($n = 2$). The shape functions obtained by substituting Eq. (6.108) into Eq. (6.129) are presented in Table 6.11 for $n = 1$ to 3.

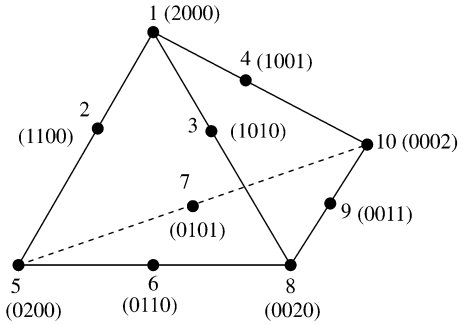


Figure 6.29
Numbering scheme for second-order tetrahedron.

The expressions derived from the variational principle for the two-dimensional problems in Sections 6.2 to 6.4 still hold except that the fundamental matrices $[T]$

Table 6.11 Shape Functions $\alpha_q(\xi_1, \xi_2, \xi_3, \xi_4)$ for $n = 1$ to 3

$n = 1$	$n = 2$	$n = 3$
$\alpha_1 = \xi_1$	$\alpha_1 = \xi_1(2\xi_2 - 1)$	$\alpha_1 = \frac{1}{2}\xi_1(3\xi_1 - 2)(3\xi_1 - 1)$
$\alpha_2 = \xi_2$	$\alpha_2 = 4\xi_1\xi_2$	$\alpha_2 = \frac{9}{2}\xi_1(3\xi_1 - 1)\xi_2$
$\alpha_3 = \xi_3$	$\alpha_3 = 4\xi_1\xi_3$	$\alpha_3 = \frac{9}{2}\xi_1(3\xi_1 - 1)\xi_3$
$\alpha_4 = \xi_4$	$\alpha_4 = 4\xi_1\xi_4$	$\alpha_4 = \frac{9}{2}\xi_1(3\xi_1 - 1)\xi_4$
	$\alpha_5 = \xi_2(2\xi_2 - 1)$	$\alpha_5 = \frac{9}{2}\xi_1(3\xi_3 - 1)\xi_2$
	$\alpha_6 = 4\xi_2\xi_3$	$\alpha_6 = 27\xi_1\xi_2\xi_3$
	$\alpha_7 = 4\xi_2\xi_4$	$\alpha_7 = 27\xi_1\xi_2\xi_4$
	$\alpha_8 = \xi_2(2\xi_3 - 1)$	$\alpha_8 = \frac{9}{2}\xi_1(3\xi_3 - 1)\xi_3$
	$\alpha_9 = 4\xi_3\xi_4$	$\alpha_9 = 27\xi_1\xi_3\xi_4$
	$\alpha_{10} = \xi_4(2\xi_4 - 1)$	$\alpha_{10} = \frac{9}{2}\xi_1(3\xi_4 - 1)\xi_4$
		$\alpha_{11} = \frac{1}{2}\xi_2(3\xi_2 - 1)(3\xi_2 - 2)$
		$\alpha_{12} = \frac{9}{2}\xi_2(3\xi_2 - 1)\xi_3$
		$\alpha_{13} = \frac{9}{2}\xi_2(3\xi_2 - 1)\xi_4$
		$\alpha_{14} = \frac{9}{2}\xi_2(3\xi_3 - 1)\xi_3$
		$\alpha_{15} = 27\xi_2\xi_3\xi_4$
		$\alpha_{16} = \frac{9}{2}\xi_2(3\xi_3 - 1)\xi_3$
		$\alpha_{17} = \frac{1}{2}\xi_3(3\xi_3 - 1)(3\xi_3 - 2)$
		$\alpha_{18} = \frac{9}{2}\xi_3(3\xi_3 - 1)\xi_4$
		$\alpha_{19} = \frac{9}{2}\xi_3(3\xi_4 - 1)\xi_4$
		$\alpha_{20} = \frac{1}{2}\xi_4(3\xi_4 - 1)(3\xi_4 - 2)$

and $[Q]$ now involve triple integration. For Helmholtz equation (6.56), for example, Eq. (6.68) applies, namely,

$$\left[C_{ff} - k^2 T_{ff}\right] \Phi_f = 0 \tag{6.130}$$

except that

$$\begin{aligned} C_{ij}^{(e)} &= \int_v \nabla\alpha_i \cdot \nabla\alpha_j \, dv \\ &= \int_v \left(\frac{\partial\alpha_i}{\partial x}\frac{\partial\alpha_j}{\partial x} + \frac{\partial\alpha_i}{\partial y}\frac{\partial\alpha_j}{\partial y} + \frac{\partial\alpha_i}{\partial z}\frac{\partial\alpha_j}{\partial z} \right) dv \,, \end{aligned} \tag{6.131}$$

$$T_{ij}^{(e)} = \int_v \alpha_i \alpha_j \, dv = v \iiint \alpha_i \alpha_j \, d\xi_1 \, d\xi_2 \, d\xi_3 \tag{6.132}$$

For further discussion on three-dimensional elements, one should consult Silvester and Ferrari [4]. Applications of three-dimensional elements to EM-related problems can be found in [49]–[53].

6.10 Finite Element Methods for Exterior Problems

Thus far in this chapter, the FEM has been presented for solving interior problems. To apply the FEM to exterior or unbounded problems such as open-type transmission lines (e.g., microstrip), scattering, and radiation problems poses certain difficulties. To overcome these difficulties, several approaches [54]–[82] have been proposed, all of which have strengths and weaknesses. We will consider three common approaches: the infinite element method, the boundary element method, and absorbing boundary condition.

6.10.1 Infinite Element Method

Consider the solution region shown in Fig. 6.30(a). We divide the entire domain into a near field (n.f.) region, which is bounded, and a far field (f.f.) region, which is unbounded. The n.f. region is divided into finite triangular elements as usual, while the f.f. region is divided into *infinite elements*. Each infinite elements shares two nodes with a finite element. Here we are mainly concerned with the infinite elements.

Consider the infinite element in Fig. 6.30(b) with nodes 1 and 2 and radial sides intersecting at point (x_o, y_o). We relate triangular polar coordinates (ρ, ξ) to the global Cartesian coordinates (x, y) as [62]

$$\begin{aligned} x &= x_o + \rho\left[(x_1 - x_o) + \xi\,(x_2 - x_1)\right] \\ y &= y_o + \rho\left[(y_1 - y_o) + \xi\,(y_2 - y_1)\right] \end{aligned} \tag{6.133}$$

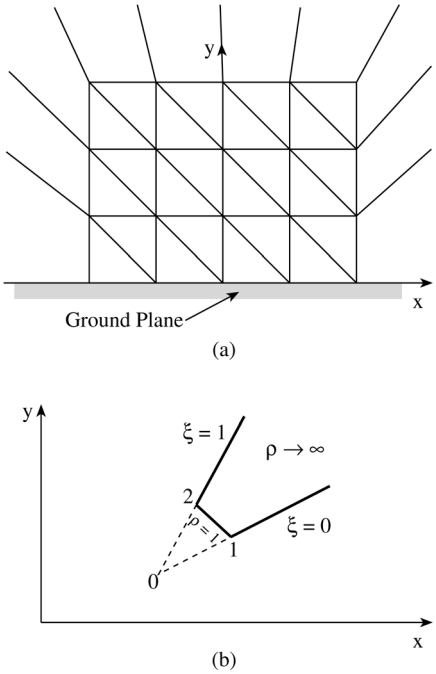


Figure 6.30
(a) Division of solution region into finite and infinite elements; (b) typical infinite element.

where $1 \leq \rho < \infty$, $0 \leq \xi \leq 1$. The potential distribution within the element is approximated by a linear variation as

$$V = \frac{1}{\rho}[V_1(1-\xi) + V_2\xi]$$

or

$$V = \sum_{i=1}^{2} \alpha_i V_i \tag{6.134}$$

where V_1 and V_2 are potentials at nodes 1 and 2 of the infinite elements, α_1 and α_2 are the interpolation or shape functions, i.e.,

$$\alpha_1 = \frac{1-\xi}{\rho}, \quad \alpha_2 = \frac{\xi}{\rho} \tag{6.135}$$

The infinite element is compatible with the ordinary first order finite element and satisfies the boundary condition at infinity. With the shape functions in Eq. (6.135), we can obtain the $[C^{(e)}]$ and $[T^{(e)}]$ matrices. We obtain solution for the exterior problem by using a standard finite element program with the $[C^{(e)}]$ and $[T^{(e)}]$ matrices of the infinite elements added to the $[C]$ and $[T]$ matrices of the n.f. region.

6.10.2 Boundary Element Method

A comparison between the finite element method (FEM) and the method of moments (MOM) is shown in Table 6.12. From the table, it is evident that the two methods have properties that complement each other. In view of this, hybrid methods have been proposed. These methods allow the use of both MOM and FEM with the aim of exploiting the strong points in each method.

Table 6.12 Comparison Between Method of Moments and Finite Element Method [83]

Method of Moments	Finite Element Method
Conceptually easy	Conceptually involved
Requires problem-dependent Green's functions	Avoids difficulties associated with singularity of Green's functions
Few equations; $O(n)$ for 2-D, $O(n^2)$ for 3-D	Many equations; $O(n^2)$ for 2-D, $O(n^3)$ for 3-D
Only boundary is discretized	Entire domain is discretized
Open boundary easy	Open boundary difficult
Fields by integration	Fields by differentiation
Good representation of far-field condition	Good representation of boundary conditions
Full matrices result	Sparse matrices result
Nonlinearity, inhomogeneity difficult	Nonlinearity, inhomogeneity easy

One of these hybrid methods is the so-called boundary element method (BEM). It is a finite element approach for handling exterior problems [68]–[80]. It basically involves obtaining the integral equation formulation of the boundary value problem [84], and solving this by a discretization procedure similar to that used in regular finite element analysis. Since the BEM is based on the boundary integral equivalent to the governing differential equation, only the surface of the problem domain needs to be modeled. Thus the dimension of the problem is reduced by one as in MOM. For 2-D problems, the boundary elements are taken to be straight line segments, whereas for 3-D problems, they are taken as triangular elements. Thus the shape or interpolation functions corresponding to subsectional bases in the MOM are used in the finite element analysis.

6.10.3 Absorbing Boundary Conditions

To apply the finite element approach to open region problems such as for scattering or radiation, an artificial boundary is introduced in order to bound the region and limit the number of unknowns to a manageable size. One would expect that as the boundary approaches infinity, the approximate solution tends to the exact one. But the closer the boundary to the radiating or scattering object, the less computer memory is required. To avoid the error caused by this truncation, an *absorbing boundary condition* (ABC)

is imposed on the artificial boundary S, as typically portrayed in Fig. 6.31. The ABC minimizes the nonphysical reflections from the boundary. Several ABCs have been proposed [85]–[91]. The major challenge of these ABCs is to bring the truncation boundary as close as possible to the object without sacrificing accuracy and to absorb the outgoing waves with little or no reflection. A popular approach is the PML-based ABC discussed in Section 3.8.3 for FD-TD. The finite element technique is used in enforcing the condition as a tool for mesh truncation [87].

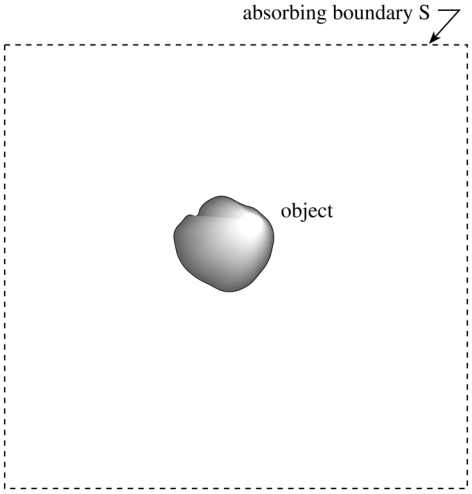


Figure 6.31
A radiating (or scattering) object surrounded by an absorbing boundary.

Another popular ABC derived Bayliss, Gunzburger, and Turkel (BGT) employs asymptotic analysis [91]. For example, for the solution of a three-dimensional problem, an expansion of the scalar Helmholtz equation is [90]:

$$\Phi(r,\theta,\phi) = \frac{e^{-jkr}}{kr} \sum_{i=0}^{\infty} \frac{F_i(\theta,\phi)}{(kr)^i} \tag{6.136}$$

The sequence of BGT operators is obtained by the recursion relation

$$\begin{aligned} B_1 &= \left(\frac{\partial}{\partial r} + jk + \frac{1}{r}\right) \\ B_m &= \left(\frac{\partial}{\partial r} + jk + \frac{2m-1}{r}\right) B_{m-1}\,, \quad m = 2, 3, \ldots \end{aligned} \tag{6.137}$$

Since Φ satisfies the higher-order radiation condition

$$B_m \Phi = O\left(1/r^{2m+1}\right) \tag{6.138}$$

imposing the mth-order boundary condition

$$B_m \Phi = 0 \quad \text{on } S \tag{6.139}$$

will compel the solution Φ to match the first $2m$ terms of the expansion in Eq. (6.136). Equation (6.139) along with other appropriate equations is solved for Φ using the finite element method.

6.11 Concluding Remarks

An introduction to the basic concepts and applications of the finite element method has been presented. It is by no means an exhaustive exposition of the subject. However, we have given the flavor of the way in which the ideas may be developed; the interested reader may build on this by consulting the references. Several introductory texts have been published on FEM. Although most of these texts are written for civil or mechanical engineers, the texts by Silvester and Ferrari [4], Chari and Silvester [41], Steele [92], Hoole [93], and Itoh [94] are for electrical engineers.

Due to its flexibility and versatility, the finite element method has become a powerful tool throughout engineering disciplines. It has been applied with great success to numerous EM-related problems. Such applications are:

- transmission line problems [95]–[97],
- optical and microwave waveguide problems [8]–[17], [92]–[103],
- electric machines [41], [104]–[106],
- scattering problems [71, 72, 75, 107, 108],
- human exposition to EM radiation [109]–[112], and
- others [113]–[116].

Applications of the FEM to time-dependent phenomena can be found in [108], [117]–[126].

For other issues on FEM not covered in this chapter, one is referred to introductory texts on FEM such as [2, 4, 36, 41, 47], [92]–[94], [126]–[133]. The issue of edge elements and absorbing boundary are covered in [126]. Estimating error in finite element solution is discussed in [52, 124, 125]. The reader may benefit from the numerous finite element codes that are commercially available. An extensive description of these systems and their capabilities can be found in [127, 134]. Although the codes were developed for one field of engineering or the other, they can be applied to problems in a different field with little or no modification.

References

[1] R. Courant, "Variational methods for the solution of problems of equilibrium and vibrations," *Bull. Am. Math. Soc.,* vol. 49, 1943, pp. 1–23.

[2] C.S. Desai and J.F. Abel, *Introduction to the Finite Element Method: A Numerical Approach for Engineering Analysis*. New York: Van Nostrand Reinhold, 1972.

[3] M.N.O. Sadiku, "A simple introduction to finite element analysis of electromagnetic problems," *IEEE Trans. Educ.,* vol. 32, no. 2, May 1989, pp. 85–93.

[4] P.P. Silvester and R.L. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge: Cambridge University Press, 3rd ed., 1996.

[5] O.W. Andersen, "Laplacian electrostatic field calculations by finite elements with automatic grid generation," *IEEE Trans. Power App. Syst.,* vol. PAS-92, no. 5, Sept./Oct. 1973, pp. 1485–1492.

[6] S. Nakamura, *Computational Methods in Engineering and Science*. New York: John Wiley, 1977, pp. 446, 447.

[7] B.S. Garbow, *Matrix Eigensystem Routine—EISPACK Guide Extension*. Berlin: Springer-Verlag, 1977.

[8] S. Ahmed and P. Daly, "Finite-element methods for inhomogeneous waveguides," *Proc. IEEE,* vol. 116, no. 10, Oct. 1969, pp. 1661–1664.

[9] Z.J. Csendes and P. Silvester, "Numerical solution of dielectric loaded waveguides: I—Finite-element analysis," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-18, no. 12, Dec. 1970, pp. 1124–1131.

[10] Z.J. Csendes and P. Silvester, "Numerical solution of dielectric loaded waveguides: II—Modal approximation technique," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-19, no. 6, June 1971, pp. 504–509.

[11] M. Hano, "Finite-element analysis of dielectric-loaded waveguides," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-32, no. 10, Oct. 1984, pp. 1275–1279.

[12] A. Konrad, "Vector variational formulation of electromagnetic fields in anisotropic media," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-24, Sept. 1976, pp. 553–559.

[13] M. Koshiba, et al., "Improved finite-element formulation in terms of the magnetic field vector for dielectric waveguides," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-33, no. 3, March 1985, pp. 227–233.

[14] M. Koshiba, et al., "Finite-element formulation in terms of the electric-field vector for electromagnetic waveguide problems," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-33, no. 10, Oct. 1985, pp. 900–905.

[15] K. Hayata, et al., "Vectorial finite-element method without any spurious solutions for dielectric waveguiding problems using transverse magnetic-field component," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-34, no. 11, Nov. 1986.

[16] K. Hayata, et al., "Novel finite-element formulation without any spurious solutions for dielectric waveguides," *Elect. Lett.,* vol. 22, no. 6, March 1986, pp. 295, 296.

[17] S. Dervain, "Finite element analysis of inhomogeneous waveguides," Masters thesis, Department of Electrical and Computer Engineering, Florida Atlantic University, Boca Raton, April 1987.

[18] J.R. Winkler and J.B. Davies, "Elimination of spurious modes in finite element analysis," *J. Comp. Phys.,* vol. 56, no. 1, Oct. 1984, pp. 1–14.

[19] M.N.O. Sadiku, et al., "A further introduction to finite element analysis of electromagnetic problems," *IEEE Trans. Educ.,* vol. 34, no. 4, Nov. 1991, pp. 322–329.

[20] The IMSL Libraries: Problem-solving software systems for numerical FORTRAN programming, IMSL, Houston, TX, 1984.

[21] M. Kono, "A generalized automatic mesh generation scheme for finite element method," *Inter. J. Num. Method Engr.,* vol. 15, 1980, pp. 713–731.

[22] J.C. Cavendish, "Automatic triangulation of arbitrary planar domains for the finite element method," *Inter. J. Num. Meth. Engr.,* vol. 8, 1974, pp. 676–696.

[23] A.O. Moscardini, et al., "AGTHOM—Automatic generation of triangular and higher order meshes," *Inter. J. Num. Meth. Engr.,* vol. 19, 1983, pp. 1331–1353.

[24] C.O. Frederick, et al., "Two-dimensional automatic mesh generation for structured analysis," *Inter. J. Num. Meth. Engr.,* vol. 2, no. 1, 1970, pp. 133–144.

[25] E.A. Heighway, "A mesh generation for automatically subdividing irregular polygon into quadrilaterals," *IEEE Trans. Mag.,* vol. MAG-19, no. 6, Nov. 1983, pp. 2535–2538.

[26] C. Kleinstreuer and J.T. Holdeman, "A triangular finite element mesh generator for fluid dynamic systems of arbitrary geometry," *Inter. J. Num. Meth. Engr.,* vol. 15, 1980, pp. 1325–1334.

[27] A. Bykat, "Automatic generation of triangular grid I—subdivision of a general polygon into convex subregions. II—Triangulation of convex polygons," *Inter. J. Num. Meth. Engr.,* vol. 10, 1976, pp. 1329–1342.

[28] N.V. Phai, "Automatic mesh generator with tetrahedron elements," *Inter. J. Num. Meth. Engr.,* vol. 18, 1982, pp. 273–289.

[29] F.A. Akyuz, "Natural coordinates systems—an automatic input data generation scheme for a finite element method," *Nuclear Engr. Design,* vol. 11, 1970, pp. 195–207.

[30] P. Girdinio, et al., "New developments of grid optimization by the grid iteration method," in Z.J. Csendes (ed.), *Computational Electromagnetism*. New York: North-Holland, 1986, pp. 3–12.

[31] M. Yokoyama, "Automated computer simulation of two-dimensional elastrostatic problems by finite element method," *Inter. J. Num. Meth. Engr.,* vol. 21, 1985, pp. 2273–2287.

[32] G.F. Carey, "A mesh-refinement scheme for finite element computations," *Comp. Meth. Appl. Mech. Engr.,* vol. 7, 1976, pp. 93–105.

[33] K. Preiss, "Checking the topological consistency of a finite element mesh," *Inter. J. Meth. Engr.,* vol. 14, 1979, pp. 1805–1812.

[34] H. Kardestuncer (ed.), *Finite Element Handbook*. New York: McGraw-Hill, 1987, pp. 4.191–4.207.

[35] W.C. Thacker, "A brief review of techniques for generating irregular computational grids," *Inter. J. Num. Meth. Engr.,* vol. 15, 1980, pp. 1335–1341.

[36] E. Hinton and D.R.J. Owen, *An Introduction to Finite Element Computations*. Swansea, UK: Pineridge Press, 1979, pp. 247, 328–346.

[37] R.J. Collins, "Bandwidth reduction by automatic renumbering," *Inter. J. Num. Meth. Engr.,* vol. 6, 1973, pp. 345–356.

[38] E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices," *ACM Nat. Conf.,* San Francisco, 1969, pp. 157–172.

[39] G.A. Akhras and G. Dhatt, "An automatic node relabelling scheme for minimizing a matrix or network bandwidth," *Inter. J. Num. Meth. Engr.,* vol. 10, 1976, pp. 787–797.

[40] F.A. Akyuz and S. Utku, "An automatic node-relabelling scheme for bandwidth minimization of stiffness matrices," *J. Amer. Inst. Aero. Astro.,* vol. 6, no. 4, 1968, pp. 728–730.

[41] M.V.K. Chari and P.P. Silvester (eds.), *Finite Elements for Electrical and Magnetic Field Problems*. Chichester: John Wiley, 1980, pp. 125–143.

[42] P. Silvester, "Construction of triangular finite element universal matrices," *Inter. J. Num. Meth. Engr.,* vol. 12, 1978, pp. 237–244.

[43] P. Silvester, "High-order polynomial triangular finite elements for potential problems," *Inter. J. Engr. Sci.,* vol. 7, 1969, pp. 849–861.

[44] G.O. Stone, "High-order finite elements for inhomogeneous acoustic guiding structures," *IEEE Trans. Micro. Theory Tech.,* vol. MTT-21, no. 8, Aug. 1973, pp. 538–542.

[45] A. Konrad, "High-order triangular finite elements for electromagnetic waves in anistropic media," *IEEE Trans. Micro. Theory Tech.,* vol. MTT-25, no. 5, May 1977, pp. 353–360.

[46] P. Daly, "Finite elements for field problems in cylindrical coordinates," *Inter. J. Num. Meth. Engr.,* vol. 6, 1973, pp. 169–178.

[47] C.A. Brebbia and J.J. Connor, *Fundamentals of Finite Element Technique.* London: Butterworth, 1973, pp. 114–118, 150–163, 191.

[48] M. Sadiku and L. Agba, "News rules for generating finite elements fundamental matrices," *Proc. IEEE Southeastcon,* 1989, pp. 797–801.

[49] R.L. Ferrari and G.L. Maile, "Three-dimensional finite element method for solving electromagnetic problems," *Elect. Lett.,* vol. 14, no. 15, 1978, pp. 467, 468.

[50] M. de Pourcq, "Field and power-density calculation by three-dimensional finite elements," *IEEE Proc.,* vol. 130, Pt. H, no. 6, Oct. 1983, pp. 377–384.

[51] M.V.K. Chari, et al., "Finite element computation of three-dimensional electrostatic and magnetostatic field problems," *IEEE Trans. Mag.,* vol. MAG-19, no. 16, Nov. 1983, pp. 2321–2324.

[52] O.A. Mohammed, et al., "Validity of finite element formulation and solution of three dimensional magnetostatic problems in electrical devices with applications to transformers and reactors," *IEEE Trans. Pow. App. Syst.,* vol. PAS-103, no. 7, July 1984, pp. 1846–1853.

[53] J.S. Savage and A.F. Peterson, "Higher-order vector finite elements for tetrahedral cells," *IEEE Trans. Micro. Theo. Tech.,* vol. 44, no. 6, June 1996, pp. 874–879.

[54] J.F. Lee and Z.J. Cendes, "Transfinite elements: a highly efficient procedure for modeling open field problems," *Jour. Appl. Phys.,* vol. 61, no. 8, April 1987, pp. 3913–3915.

[55] B.H. McDonald and A. Wexler, "Finite-element solution of unbounded field problems," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-20, no. 12, Dec. 1972, pp. 841–847.

[56] P.P. Silvester, et al., "Exterior finite elements for 2-dimensional field problems with open boundaries," *Proc. IEEE,* vol. 124, no. 12, Dec. 1977, pp. 1267–1270.

[57] S. Washisu, et al., "Extension of finite-element method to unbounded field problems," *Elect. Lett.,* vol. 15, no. 24, Nov. 1979, pp. 772–774.

[58] P. Silvester and M.S. Hsieh, "Finite-element solution of 2-dimensional exterior-field problems," *Proc. IEEE,* vol. 118, no. 12, Dec. 1971, pp. 1743–1747.

[59] Z.J. Csendes, "A note on the finite-element solution of exterior-field problems," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-24, no. 7, July 1976, pp. 468–473.

[60] T. Corzani, et al., "Numerical analysis of surface wave propagation using finite and infinite elements," *Alta Frequenza,* vol. 51, no. 3, June 1982, pp. 127–133.

[61] O.C. Zienkiewicz, et al., "Mapped infinite elements for exterior wave problems," *Inter. J. Num. Meth. Engr.,* vol. 21, 1985.

[62] F. Medina, "An axisymmetric infinite element," *Int. J. Num. Meth. Engr.,* vol. 17, 1981, pp. 1177–1185.

[63] S. Pissanetzky, "A simple infinite element," *Int. J. Comp. Math. Elect. Engr.,* (COMPEL), vol. 3, no. 2, 1984, pp. 107–114.

[64] Z. Pantic and R. Mittra, "Quasi-TEM analysis of microwave transmission lines by the finite-element method," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-34, no. 11, Nov. 1986, pp. 1096–1103.

[65] K. Hayata, et al., "Self-consistent finite/infinite element scheme for unbounded guided wave problems," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-36, no. 3, Mar. 1988, pp. 614–616.

[66] P. Petre and L. Zombory, "Infinite elements and base functions for rotationally symmetric electromagnetic waves," *IEEE Trans. Ant. Prog.,* vol. 36, no. 10, Oct. 1988, pp. 1490, 1491.

[67] Z.J. Csendes and J.F. Lee, "The transfinite element method for modeling MMIC devices," *IEEE Trans. Micro. Theo. Tech.* vol. 36, no. 12, Dec. 1988, pp. 1639–1649.

[68] K.H. Lee, et al., "A hybrid three-dimensional electromagnetic modeling scheme," *Geophys.,* vol. 46, no. 5, May 1981, pp. 796–805.

[69] S.J. Salon and J.M. Schneider, "A hybrid finite element-boundary integral formulation of Poisson's equation." *IEEE Trans. Mag.,* vol. MAG-17, no. 6, Nov. 1981, pp. 2574–2576.

[70] S.J. Salon and J. Peng, "Hybrid finite-element boundary-element solutions to axisymmetric scalar potential problems," in Z.J. Csendes (ed.), *Computational Electromagnetics*. New York: North-Holland/Elsevier, 1986, pp. 251–261.

[71] J.M. Lin and V.V. Liepa, "Application of hybrid finite element method for electromagnetic scattering from coated cylinders," *IEEE Trans. Ant. Prop.,* vol. 36, no. 1, Jan. 1988, pp. 50–54.

[72] J.M. Lin and V.V. Liepa, "A note on hybrid finite element method for solving scattering problems," *IEEE Trans. Ant. Prop.,* vol. 36, no. 10, Oct. 1988, pp. 1486–1490.

[73] M.H. Lean and A. Wexler, "Accurate field computation with boundary element method," *IEEE Trans. Mag.,* vol. MAG-18, no. 2, Mar. 1982, pp. 331-335.

[74] R.F. Harrington and T.K. Sarkar, "Boundary elements and the method of moments," in C.A. Brebbia, et al. (eds.), *Boundary Elements*. Southampton: CML Publ., 1983, pp. 31–40.

[75] M.A. Morgan, et al., "Finite element-boundary integral formulation for electromagnetic scattering," *Wave Motion,* vol. 6, no. 1, 1984, pp. 91–103.

[76] S. Kagami and I. Fukai, "Application of boundary-element method to electromagnetic field problems," *IEEE Trans. Micro. Theo. Tech.,* vol. 32, no. 4, Apr. 1984, pp. 455–461.

[77] Y. Tanaka, et al., "A boundary-element analysis of TEM cells in three dimensions," *IEEE Trans. Elect. Comp.,* vol. EMC-28, no. 4, Nov. 1986, pp. 179–184.

[78] N. Kishi and T. Okoshi, "Proposal for a boundary-integral method without using Green's function," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-35, no. 10, Oct. 1987, pp. 887–892.

[79] D.B. Ingham, et al., "Boundary integral equation analysis of transmission-line singularities," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-29, no. 11, Nov. 1981, pp. 1240–1243.

[80] S. Washiru, et al., "An analysis of unbounded field problems by finite element method," *Electr. Comm. Japan,* vol. 64-B, no. 1, 1981, pp. 60–66.

[81] T. Yamabuchi and Y. Kagawa, "Finite element approach to unbounded Poisson and Helmholtz problems using hybrid-type infinite element," *Electr. Comm. Japan,* Pt. I, vol. 68, no. 3, 1986, pp. 65–74.

[82] K.L. Wu and J. Litva, "Boundary element method for modelling MIC devices," *Elect. Lett.,* vol. 26, no. 8, April 1990, pp. 518–520.

[83] M.N.O. Sadiku and A.F. Peterson, "A comparison of numerical methods for computing electromagnetic fields," *Proc. of IEEE Southeastcon,* April 1990, pp. 42–47.

[84] P.K. Kythe, *An Introduction to Boundary Element Methods.* Boca Raton, FL: CRC Press, 1995, p. 2.

[85] J.M. Jin et al., "Fictitious absorber for truncating finite element meshes in scattering," *IEEE Proc. H,* vol. 139, Oct. 1992, pp. 472–476.

[86] R. Mittra and O. Ramahi, "Absorbing bounding conditions for direct solution of partial differential equations arising in electromagnetic scattering problems," in M.A. Morgan (ed.), *Finite Element and Finite Difference Methods in Electromagnetics*. New York: Elsevier, 1990, pp. 133–173.

[87] U. Pekel and R. Mittra, "Absorbing boundary conditions for finite element mesh truncation," in T. Itoh et al. (eds.), *Finite Element Software for Microwave Engineering*. New York: John Wiley & Sons, 1996, pp. 267–312.

[88] U. Pekel and R. Mittra, "A finite element method frequency domain application of the perfectly matched layer (PML) concept," *Micro. Opt. Technol. Lett.,* vol. 9, pp. 117–122.

[89] A. Boag and R. Mittra, "A numerical absorbing boundary condition for finite difference and finite element analysis of open periodic structures," *IEEE Trans. Micro. Theo. Tech.,* vol. 43, no. 1 Jan. 1995, pp. 150–154.

[90] P.P. Silvester and G. Pelosi (eds.), *Finite Elements for Wave Electromagnetics: Methods and Techniques*. New York: IEEE Press, 1994, pp. 351–490.

[91] A.M. Bayliss, M. Gunzburger, and E. Turkel, "Boundary conditions for the numerical solution of elliptic equation in exterior regions," *SIAM Jour. Appl. Math.,* vol. 42, 1982, pp. 430–451.

[92] C.W. Steele, *Numerical Computation of Electric and Magnetic Fields*. New York: Van Nostrand Reinhold, 1987.

[93] S.R. Hoole, *Computer-aided Analysis and Design of Electromagnetic Devices*. New York: Elsevier, 1989.

[94] T. Itoh (ed.), *Numerical Technique for Microwave and Millimeterwave Passive Structure*. New York: John Wiley, 1989.

[95] R.L. Khan and G.I. Costache, "Finite element method applied to modeling crosstalk problems on printed circuits boards," *IEEE Trans. Elect. Comp.,* vol. 31, no. 1, Feb. 1989, pp. 5–15.

[96] P. Daly, "Upper and lower bounds to the characteristic impedance of transmission lines using the finite method," *Inter. J. Comp. Math. Elect. Engr.,* (COMPEL), vol. 3, no. 2, 1984, pp. 65–78.

[97] A. Khebir, et al., "An absorbing boundary condition for quasi-TEM analysis of microwave transmission lines via the finite element method," *J. Elect. Waves Appl.,* vol. 4, no. 2, 1990, pp. 145–157.

[98] N. Mabaya, et al., "Finite element analysis of optical waveguides," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-29, no. 6, June 1981, pp. 600–605.

[99] M. Ikeuchi, et al., "Analysis of open-type dielectric waveguides by the finite-element iterative method," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-29, no. 3, Mar. 1981, pp. 234–239.

[100] C. Yeh, et al., "Single model optical waveguides," *Appl. Optics,* vol. 18, no. 10, May 1979, pp. 1490–1504.

[101] J. Katz, "Novel solution of 2-D waveguides using the finite element method," *Appl. Optics,* vol. 21, no. 15, Aug. 1982, pp. 2747–2750.

[102] B.A. Rahman and J.B. Davies, "Finite-element analysis of optical and microwave waveguide problems," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-32, no. 1, Jan. 1984, pp. 20–28.

[103] X.Q. Sheng and S. Xu, "An efficient high-order mixed-edge rectangular-element method for lossy anisotropic dielectric waveguide," *IEEE Micro. Theo. Tech.,* vol. 45, no. 7, July 1997, pp. 1009–1013.

[104] C.B. Rajanathan, et al., "Finite-element analysis of the Xi-core leviator," *IEEE Proc.,* vol. 131, Pt. A, no. 1, Jan. 1984, pp. 62–66.

[105] T.L. Ma and J.D. Lavers, "A finite-element package for the analysis of electromagnetic forces and power in an electric smelting furnace," *IEEE Trans. Indus. Appl.,* vol. IA-22, no. 4, July/Aug. 1986, pp. 578–585.

[106] C.O. Obiozor and M.N.O. Sadiku, "Finite element analysis of a solid rotor induction motor under stator winding effects," *Proc. IEEE Southeastcon,* 1991, pp. 449–453.

[107] J.L. Mason and W.J. Anderson, "Finite element solution for electromagnetic scattering from two-dimensional bodies," *Inter. J. Num. Meth. Engr.,* vol. 21, 1985, pp. 909–928.

[108] A.C. Cangellaris, et al., "Point-matching time domain finite element methods for electromagnetic radiation and scattering," *IEEE Trans. Ant. Prop.,* vol. AP35, 1987, pp. 1160–1173.

[109] A. Chiba, et al., "Application of finite element method to analysis of induced current densities inside human model exposed to 60 Hz electric field," *IEEE Trans. Power App. Sys.,* vol. PAS-103, no. 7, July 1984, pp. 1895–1902.

[110] Y. Yamashita and T. Takahashi, "Use of the finite element method to determine epicardial from body surface potentials under a realistic torso model," *IEEE Trans. Biomed. Engr.,* vol. BME-31, no. 9, Sept. 1984, pp. 611–621.

[111] M.A. Morgan, "Finite element calculation of microwave absorption by the cranial structure," *IEEE Trans. Biomed. Engr.,* vol. BME-28, no. 10, Oct. 1981, pp. 687–695.

[112] D.R. Lynch, et al., "Finite element solution of Maxwell's equation for hyperthermia treatment planning," *J. Comp. Phys.* vol. 58, 1985, pp. 246–269.

[113] J.R. Brauer, et al., "Dynamic electric fields computed by finite elements," *IEEE Trans. Ind. Appl.,* vol. 25, no. 6, Nov./Dec. 1989, pp. 1088–1092.

[114] C.H. Chen and C.D. Lien, "A finite element solution of the wave propagation problem for an inhomogeneous dielectric slab," *IEEE Trans. Ant. Prop.,* vol. AP-27, no. 6, Nov. 1979, pp. 877–880.

[115] T.L.W. Ma and J.D. Lavers, "A finite-element package for the analysis of electromagnetic forces and power in an electric smelting furnace," *IEEE Trans. Ind. Appl.,* vol. IA-22, no. 4, July/Aug., 1986, pp. 578–585.

[116] M.A. Kolbehdari and M.N.O. Sadiku, "Finite element analysis of an array of rods or rectangular bars between ground," *Jour. Franklin Inst.,* vol. 335B, no. 1, 1998, pp. 97–107.

[117] V. Shanka, "A time-domain, finite-volume treatment for Maxwell's equations," *Electromagnetics,* vol. 10, 1990, pp. 127–145.

[118] J.H. Argyris and D.W. Scharpf, "Finite elements in time and space," *Nucl. Engr. Space Des.,* vol. 10, no. 4, 1969, pp. 456–464.

[119] I. Fried,"Finite-element analysis of time-dependent phenomena," *AIAA J.,* vol. 7, no. 6, 1969, pp. 1170–1173.

[120] O.C. Zienkiewicz and C.J. Pareth, "Transient field problems: two-dimensional and three-dimensional analysis by isoparametric finite elements," *Inter. J. Num. Meth. Engr.,* vol. 2, 1970, pp. 61–71.

[121] J.H. Argyris and A.S.L. Chan, "Applications of finite elements in space and time," *Ingenieur-Archiv,* vol. 41, 1972, pp. 235–257.

[122] J.C. Bruch and G. Zyvoloski, "Transient two-dimensional heat conduction problems solved by the finite element method," *Inter. J. Num. Meth. Engr.,* vol. 8, 1974, pp. 481–494.

[123] B. Swartz and B. Wendroff, "The relative efficiency of finite difference and finite element methods I: hyperbolic problems and splines," *SIAM J. Numer. Anal.,* vol. 11, no. 5, Oct. 1974.

[124] J. Cushman, "Difference schemes or element schemes," *Int. J. Num. Meth. Engr.,* vol. 14, 1979, pp. 1643–1651.

[125] A.J. Baker and M.O. Soliman, "Utility of a finite element solution algorithm for initial-value problems," *J. Comp. Phys.,* vol. 32, 1979, pp. 289–324.

[126] J.N. Reddy, *An Introduction to the Finite Element Method.* New York: McGraw-Hill, 2nd ed., 1993, pp. 293–403.

[127] C.A. Brebbia (ed.), *Applied Numerical Modelling.* New York: John Wiley, 1978, pp. 571–586.

[128] C.A. Brebbia (ed.), *Finite Element Systems: A Handbook.* Berlin: Springer-Verlag, 1985.

[129] O.C. Zienkiewicz, *The Finite Element Method.* New York: McGraw-Hill, 1977.

[130] A.J. Davies, *The Finite Element Method: A First Approach.* Oxford: Clarendon, 1980.

[131] C. Martin and G.F. Carey, *Introduction to Finite Element Analysis: Theory and Application.* New York: McGraw-Hill, 1973.

[132] T.J. Chung, *Finite Element Analysis in Fluid Dynamics.* New York: McGraw-Hill, 1978.

[133] D.H. Norris and G de. Vries, *An Introduction to Finite Element Analysis.* New York: Academic Press, 1978.

[134] T. Itoh et al. (eds.), *Finite Element Software for Microwave Engineering*. New York: John Wiley & Sons, 1996.

[135] J.L. Volakis et al., *Finite Element Method for Electromagnetics*. New York: IEEE Press, 1998.

[136] G.R. Buchanan, *Finite Element Analysis* (Schaum's Outline). New York: McGraw-Hill, 2nd ed., 1995.

[137] J. Jin, *The Finite Element Method in Electromagnetics*. New York: John Wiley & Sons, 1993.

[138] R. Thatcher, "Assessing the error in a finite element solution," *IEEE Trans. Micro. Theo. Tech.,* vol. MTT-30, no. 6, June 1982, pp. 911–914.

[139] J. Penman and M. Grieve, "Self-adaptive finite-element techniques for the computation of inhomogeneous Poissonian fields," *IEEE Trans. Micro. Theo. Tech.,* vol. 24, no. 6, Nov./Dec. 1998, pp. 1042–1049.

Problems

6.1 For the triangular elements in Fig. 6.32, determine the element coefficient matrices.

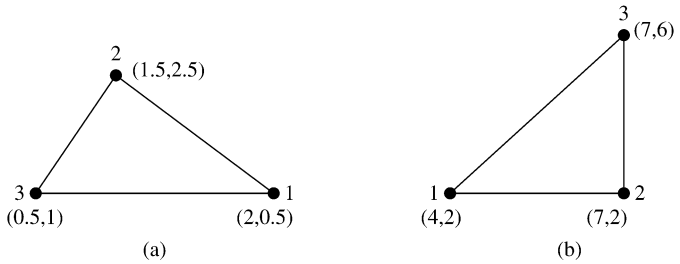


Figure 6.32
For Problem 6.1.

6.2 Find the coefficient matrix for the two-element mesh of Fig. 6.33. Given that $V_2 = 10$ and $V_4 = -10$, determine V_1 and V_3.

6.3 Determine the shape functions α_1, α_2, and α_3 for the element in Fig. 6.34.

6.4 Consider the mesh shown in Fig. 6.35. The shaded region is conducting and has no finite elements. Calculate the global elements $C_{3,10}$ and $C_{3,3}$.

6.5 With reference to the finite element in Fig. 6.36, calculate the energy per unit length associated with the element.

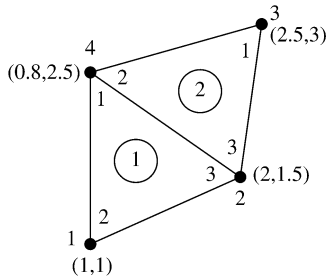


Figure 6.33
For Problem 6.2.

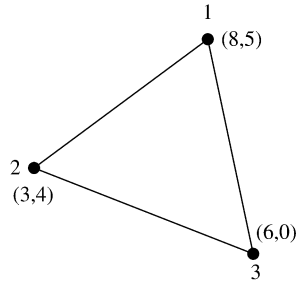


Figure 6.34
For Problem 6.3.

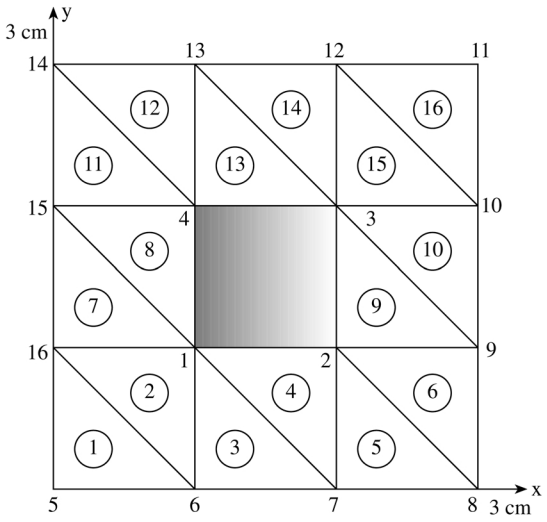


Figure 6.35
For Problem 6.4.

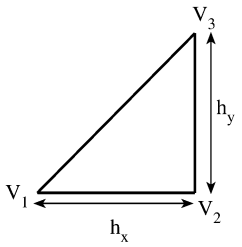


Figure 6.36
For Problem 6.5.

6.6 Consider the element whose sides are parallel to the x and y axis, as shown in Fig. 6.37. Verify that the potential distribution within the elements can be expressed as

$$V(x, y) = \alpha_1 V_1 + \alpha_2 V_2 + \alpha_3 V_3 + \alpha_4 V_4$$

where V_i are the nodal potentials and α_i are local interpolating functions defined as

$$\alpha_1 = \frac{(x - x_2)(y - y_4)}{(x_1 - x_2)(y_1 - y_4)}$$
$$\alpha_2 = \frac{(x - x_1)(y - y_3)}{(x_2 - x_1)(y_2 - y_3)}$$
$$\alpha_3 = \frac{(x - x_4)(y - y_2)}{(x_3 - x_4)(y_3 - y_2)}$$
$$\alpha_4 = \frac{(x - x_3)(y - y_1)}{(x_4 - x_3)(y_4 - y_1)}$$

6.7 The cross section of an infinitely long rectangular trough is shown in Fig. 6.38; develop a program using FEM to find the potential at the center of the cross section. Take $\epsilon_r = 4.5$.

6.8 Solve the problem in Example 3.3 using the finite element method.

6.9 Modify the program in Fig. 6.10 to calculate the electric field intensity **E** at any point in the solution region.

6.10 The program in Fig. 6.10 applies the iteration method to determine the potential at the free nodes. Modify the program and use the band matrix method to determine the potential. Test the program using the data in Example 6.2.

6.11 A grounded rectangular pipe with the cross section in Fig. 6.39 is half-filled with hydrocarbons ($\epsilon = 2.5\epsilon_o$, $\rho_o = 10^{-5}$ C/m^3). Use FEM to determine the potential along the liquid-air interface. Plot the potential versus x.

6.12 Solve the problem in Example 3.4 using the finite element method.

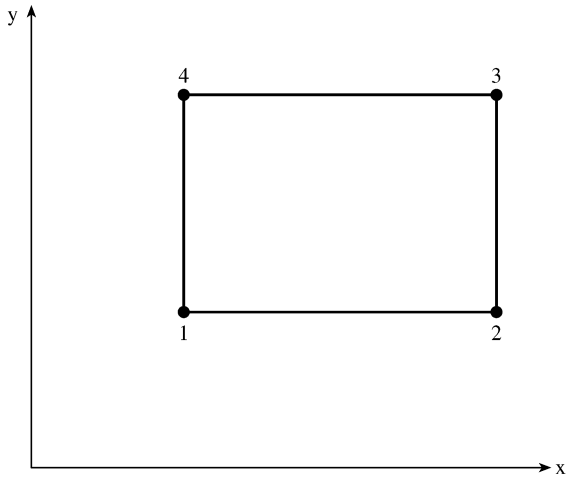


Figure 6.37
For Problem 6.6.

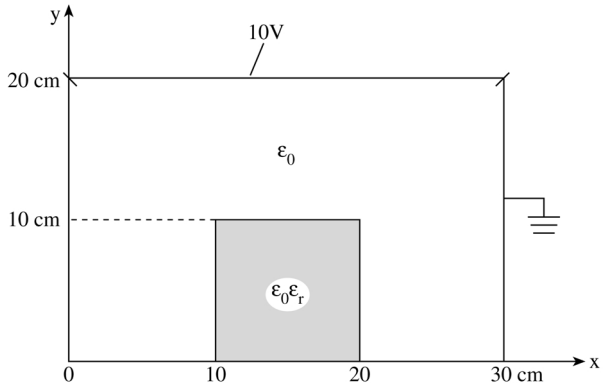


Figure 6.38
For Problem 6.7.

6.13 The cross section of an isosceles right-triangular waveguide is discretized as in Fig. 6.40. Determine the first 10 TM cutoff wavelengths of the guide.

6.14 Using FEM, determine the first 10 cutoff wavelengths of a rectangular waveguide of cross section 2 cm by 1 cm. Compare your results with exact solution. Assume the guide is air-filled.

6.15 Use the mesh generation program in Fig. 6.16 to subdivide the solution regions in Fig. 6.41. Subdivide into as many triangular elements as you choose.

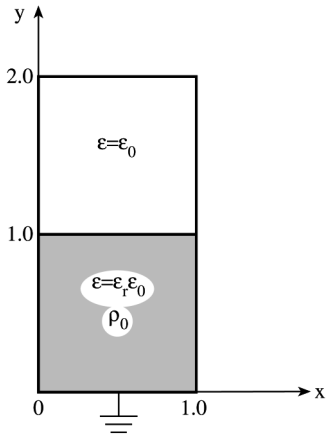


Figure 6.39
For Problem 6.11.

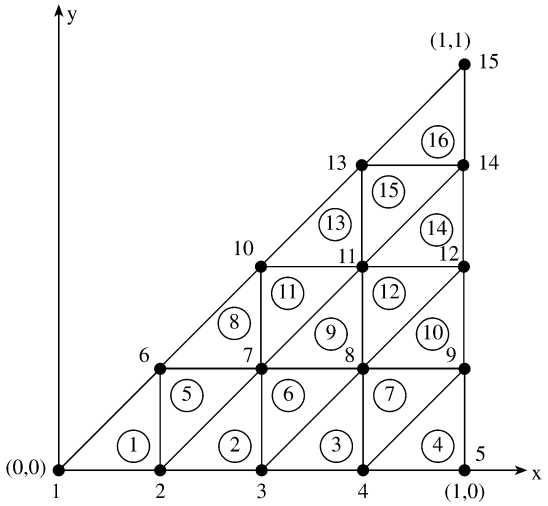


Figure 6.40
For Problem 6.13.

6.16 Determine the semi-bandwidth of the mesh shown in Fig. 6.42. Renumber the mesh so as to minimize the bandwidth.

6.17 Find the semi-bandwidth B of the mesh in Fig. 6.43. Renumber the mesh to minimize B and determine the new value of B.

6.18 Rework Problem 3.18 using the FEM.

Hint: After calculating V at all free nodes with ϵ lumped with C_{ij}, use Eq. (6.19)

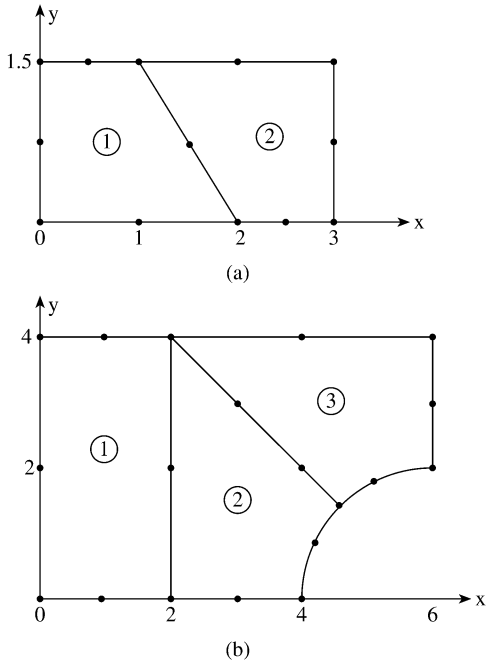


Figure 6.41
For Problem 6.15.

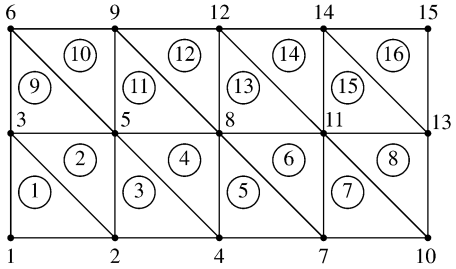


Figure 6.42
For Problem 6.16.

to calculate W, i.e.,

$$W = \frac{1}{2}[V]^t[C][V]$$

Then find the capacitance from

$$C = \frac{2W}{V_d^2}$$

where V_d is the potential difference between inner and outer conductors.

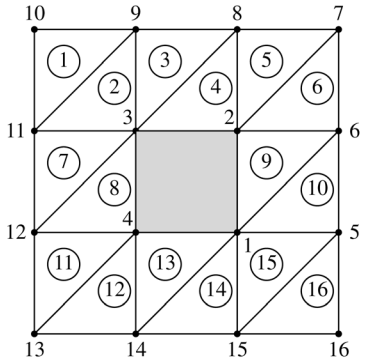


Figure 6.43
For Problem 6.17.

6.19 Verify the interpolation functions for the six-node quadratic triangular element.

6.20 Using the area coordinates (ξ_1, ξ_2, ξ_3) for the triangular element in Fig. 6.3, evaluate:

(a) $\int_S x\, dS$,
(b) $\int_S x\ dS$,
(c) $\int_S xy\, dS$

6.21 Evaluate the following integrals:

(a) $\int_S \alpha_2^3\, dS$,
(b) $\int_S \alpha_1\alpha_5\ dS$,
(c) $\int_S \alpha_1\alpha_2\alpha_3\, dS$

6.22 Evaluate the shape functions $\alpha_1, \ldots, \alpha_6$ for the second-order elements in Fig. 6.44.

6.23 Derive matrix T for $n = 2$.

6.24 By hand calculation, obtain $Q^{(2)}$ and $Q^{(3)}$ for $n = 1$ and $n = 2$.

6.25 The $D^{(q)}$ matrix is an auxilliary matrix used along with the T matrix to derive other fundamental matrices. An element of D is defined in [43] as the partial derivative of α_i with respect to ξ_q evaluated at node P_j, i.e.,

$$D_{ij}^{(q)} = \left.\frac{\partial \alpha_i}{\partial \xi_q}\right|_{P_i}, \qquad i, j = 1, 2, \ldots, m$$

where $q \in \{1, 2, 3\}$. For $n = 1$ and 2, derive $D^{(1)}$. From $D^{(1)}$, derive $D^{(2)}$ and $D^{(3)}$.

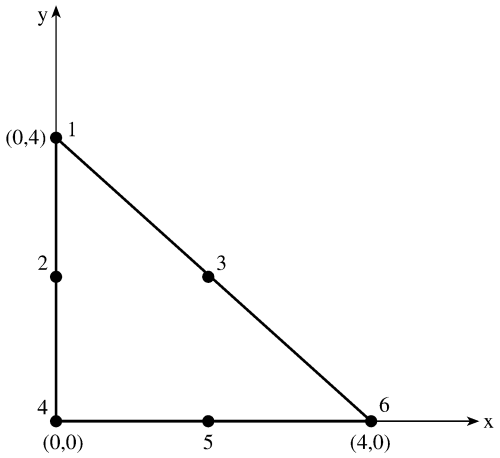


Figure 6.44
For Problem 6.22.

6.26 (a) The matrix $K^{(pq)}$ can be defined as

$$K_{ij}^{(pq)} = \iint \frac{\partial \alpha_i}{\partial \xi_p} \frac{\partial \alpha_j}{\partial \xi_q} \, dS$$

where $p, q = 1, 2, 3$. Using the $D^{(q)}$ matrix of the previous problem, show that

$$K^{(pq)} = D^{(p)} T D^{(q)^t}$$

where t denotes transposition.

(b) Show that the $Q^{(q)}$ matrix can be written as

$$Q^{(q)} = \left[D^{(q+1)} - D^{(q-1)} \right] T \left[D^{(q+1)} - D^{(q-1)} \right]^t$$

Use this formula to derive $Q^{(1)}$ for $n = 1$ and 2.

6.27 Verify the interpolation function for the 10-node tetrahedral element.

6.28 Using the volume coordinates for a tetrahedron, evaluate

$$\int z^2 \, dv$$

Assume that the origin is located at the centroid of the tetrahedron.

6.29 Obtain the T matrix for the first-order tetrahedral element.

6.30 For the tetrahedral cell, obtain the matrix M whose elements are defined by

$$M_{ij} = \frac{1}{v} \int_v \xi_i \xi_j \, dv$$

6.31 For the two-dimensional problem, the BGI sequence of operators are defined by the recurrence relation

$$B_m = \left(\frac{\partial}{\partial \rho} + jk + \frac{4m-3}{2\rho} \right) B_{m-1}$$

where $B_o = 1$. Obtain B_1 and B_2.