CHAPTER TEN

Linear Problems

In this chapter, we discuss procedures for obtaining finite element equations and their solutions in linear two-dimensional boundary value problems. Implementations of boundary conditions are detailed and example problems for steady and unsteady cases are presented. Multivariable simultaneous partial differential equations and simple Stokes flow problems are also included.

10.1 STEADY-STATE PROBLEMS – STANDARD GALERKIN METHODS

10.1.1 TWO-DIMENSIONAL ELLIPTIC EQUATIONS

We have illustrated procedures for constructing finite element equations for one-dimensional problems in Chapters 1 and 8. Extension to two-dimensional cases follows the same general guidelines. The only difference is the appropriate interpolation functions for two-dimensional geometries, specification of Neumann boundary conditions, integration over the domain, and directional variables.

Consider the second order elliptic partial differential equation of the form,

$$R = \nabla^2 u + f(x, y) = 0 \quad \text{in } \Omega \tag{10.1.1}$$

As shown in Chapters 1 and 8, the Standard Galerkin Method (SGM) for (10.1.1) is the inner product of the residual with the test function Φ_α

$$(\Phi_\alpha, R) = \int_\Omega \Phi_\alpha [u_{,ii} + f(x, y)] d\Omega = 0 \tag{10.1.2}$$

Assuming that the variable u is approximated in the form

$$u = \Phi_\alpha u_\alpha \tag{10.1.3}$$

and integrating (10.1.2) by parts we obtain

$$\int_\Gamma \overset{*}{\Phi}_\alpha u_{,i} n_i d\Gamma - \left(\int_\Omega \Phi_{\alpha,i} \Phi_{\beta,i} d\Omega \right) u_\beta + \int_\Omega \Phi_\alpha f(x, y) d\Omega = 0$$

or

$$K_{\alpha\beta} u_\beta = F_\alpha + G_\alpha \tag{10.1.4}$$

where

$$\text{Stiffness matrix} \qquad K_{\alpha\beta} = \int_{\Omega} \Phi_{\alpha,i}\, \Phi_{\beta,i}\, d\Omega \tag{10.1.5a}$$

$$\text{Source vector} \qquad F_{\alpha} = \int_{\Omega} \Phi_{\alpha}\, f(x, y) d\Omega \tag{10.1.5b}$$

$$\text{Neumann boundary vector} \qquad G_{\alpha} = \int_{\Gamma} \overset{*}{\Phi}_{\alpha} u_{,i} n_i\, d\Gamma \tag{10.1.5c}$$

As we noted in the one-dimensional problem, the interpolation function originally defined in the domain is now a function of boundary coordinate Γ in the boundary integral G_{α}, with $\overset{*}{\Phi}_{\alpha}$ indicating the dependency on Γ, not on Ω. It represents the interpolation function describing the way the Neumann data $u_{,i} n_i$ varies along the boundaries. Thus, a suitable form for $\overset{*}{\Phi}_{\alpha}(\Gamma)$ would be the one-dimensional linear interpolation function.

The global forms (10.1.5) can be obtained by the assembly of local forms similarly as in the one-dimensional problems,

$$K_{\alpha\beta} = \bigcup_{e=1}^{E} K_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} \tag{10.1.6a}$$

$$F_{\alpha} = \bigcup_{e=1}^{E} F_{N}^{(e)} \Delta_{N\alpha}^{(e)} \tag{10.1.6b}$$

$$G_{\alpha} = \bigcup_{e=1}^{E} G_{N}^{(e)} \Delta_{N\alpha}^{(e)} \tag{10.1.6c}$$

where

$$K_{NM}^{(e)} = \int_{\Omega} \Phi_{N,i}^{(e)} \Phi_{M,i}^{(e)} d\Omega \tag{10.1.7a}$$

$$F_{N}^{(e)} = \int_{\Omega} \Phi_{N}^{(e)} f(x, y) d\Omega \tag{10.1.7b}$$

$$G_{N}^{(e)} = \int_{\Gamma} \overset{*}{\Phi}_{N}^{(e)} u_{,i} n_i\, d\Gamma \tag{10.1.7c}$$

The source term $f(x, y)$ and the Neumann data $g(\Gamma) = u_{,i} n_i$ can be interpolated as follows:

$$f(x, y) = \Phi_{\alpha}(x, y)\, f_{\alpha}, \qquad f_{\alpha} = [f(x, y)]_{\alpha} \tag{10.1.8a}$$

$$g(\Gamma) = \overset{*}{\Phi}_{\alpha}(\Gamma) g_{\alpha}, \qquad g_{\alpha} = (u_{,i} n_i)_{\alpha} \tag{10.1.8b}$$

These approximations allow the corresponding source term $f(x, y)$ and the Neumann data $u_{,i} n_i$ to be entered directly to the particular node under consideration. Substituting (10.1.8a) and (10.1.8b) into (10.1.5b) and (10.1.5c), respectively, we obtain

$$\begin{aligned} F_{\alpha} &= \left(\int_{\Omega} \Phi_{\alpha} \Phi_{\beta} d\Omega \right) f_{\beta} = C_{\alpha\beta} f_{\beta} = \bigcup_{e=1}^{E} C_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} f_{p}^{(e)} \Delta_{p\beta}^{(e)} \\ &= \bigcup_{e=1}^{E} C_{NM}^{(e)} f_{M}^{(e)} \Delta_{N\alpha}^{(e)} = \bigcup_{e=1}^{E} F_{N}^{(e)} \Delta_{N\alpha}^{(e)} \end{aligned} \tag{10.1.9}$$

and similarly,

$$G_\alpha = \left(\int_\Gamma \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta d\Gamma \right) g_\beta = \overset{*}{C}_{\alpha\beta}\, g_\beta = \bigcup_{e=1}^{E} G_N^{(e)} \Delta_{N\alpha}^{(e)} \tag{10.1.10}$$

where

$$F_N^{(e)} = C_{NM}^{(e)} f_M^{(e)} \tag{10.1.11}$$

$$G_N^{(e)} = \overset{*}{C}{}_{NM}^{(e)} g_M^{(e)} \tag{10.1.12}$$

with

$$C_{NM}^{(e)} = \int_\Omega \Phi_N^{(e)} \Phi_M^{(e)} d\Omega \tag{10.1.13a}$$

$$\overset{*}{C}{}_{NM}^{(e)} = \int_\Gamma \overset{*}{\Phi}{}_N^{(e)}\, \overset{*}{\Phi}{}_M^{(e)} d\Gamma \tag{10.1.13b}$$

For linear variations of $u_{,i} n_i$ for a boundary element of length l, $\overset{*}{\Phi}{}_N^{(e)} = (1 - \Gamma/l, \Gamma/l)$, the integration of (10.1.13b) gives the result,

$$\overset{*}{C}{}_{NM}^{(e)} = \frac{l}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

It is clear that, regardless of the choice of the local finite elements for the domain, whether triangular or quadrilateral, the boundary integral (10.1.13b) can remain independent.

As shown in Section 8.2, the Neumann boundary data interpolation functions $\overset{*}{\Phi}{}_N^{(e)}$ and $\overset{*}{\Phi}_\alpha$ are given by

$$\begin{aligned} &\overset{*}{\Phi}{}_N^{(e)} = \delta\big(\overset{*}{z}{}_N^{(e)} - z_N^{(e)}\big), \qquad \overset{*}{\Phi}{}_N^{(e)}\big(z_M^{(e)}\big) = \delta_{NM} \\ &\overset{*}{\Phi}_\alpha = \delta(\overset{*}{Z}_\alpha - Z_\alpha), \qquad \overset{*}{\Phi}_\alpha(Z_\beta) = \delta_{\alpha\beta} \end{aligned} \tag{10.1.14}$$

implying that $\overset{*}{\Phi}{}_N^{(e)} = 1$ if the Neumann boundary condition is applied at the boundary node N and zero, otherwise. This applies also to $\overset{*}{\Phi}_\alpha$.

The significance and importance of (10.1.14) cannot be overemphasized. Re-examine (10.1.5c), (10.1.6c), (10.1.7c), and (10.1.8b) in conjuction with (10.1.14). The process through these relations indicates that the local Neumann data are passed along across the local adjacent elements normal to the boundary surfaces to ensure the continuity of gradients or "energy balance" (incoming and outgoing normal gradients are cancelled at element boundaries) until the domain edge boundaries are reached, where the Neumann boundary conditions are applied and where the Neumann boundary condition interpolation functions $\overset{*}{\Phi}{}_N^{(e)}$ and $\overset{*}{\Phi}_\alpha$ assume the value of unity if *applied*, zero otherwise. Notice that this logic is established easily and clearly by having constructed the finite element equations in a global form from the beginning, called the "*global approach*," and by seeking the local element contributions in terms of the Boolean matrix algebra afterward. This is contrary to the traditional approach to the finite element formulations, from local to global, called the "*local approach*," in which the passage of Neumann data through element boundary surfaces cannot be defined

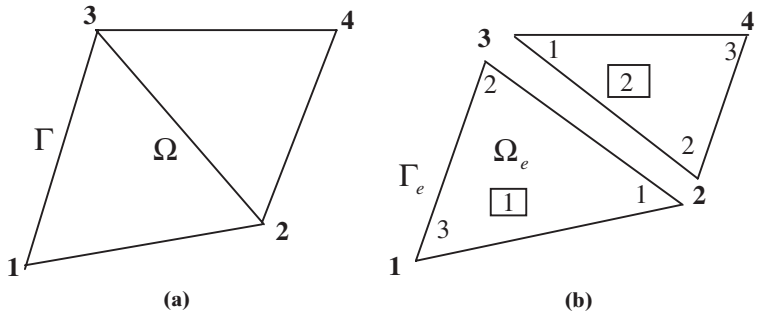


Figure 10.1.1 Finite element discretization. **(a)** Global nodes; **(b)** Local nodes.

easily and automatically. The global approach presented here is in contrast to the finite volume methods in which algebraic equations are generated by physically enforcing the normal gradients across the local element boundary surfaces. The consequences of operations involved in both FEM and FVM, however, are analogous, with the conservation properties maintained in both methods.

The assembly of local elements into a global form follows the same procedure as in the one-dimensional case. To obtain the global matrices $K_{\alpha\beta}$ and F_α, let us consider the two triangular elements in Figure 10.1.1. Although the expansion (10.1.6a) can be performed by summing the repeated indices, we may show such operations by matrix multiplications as follows:

First, we prepare the nodal correspondence table (Table 10.1.1) which indicates the correspondence of the local node with the global node for all elements.

$$
\begin{aligned}
K_{\alpha\beta} &= \bigcup_{e=1}^{E} K^{(e)}_{NM}\Delta^{(e)}_{N\alpha}\Delta^{(e)}_{M\beta} \\
&= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\begin{bmatrix} K^{(1)}_{11} & K^{(1)}_{12} & K^{(1)}_{13} \\ K^{(1)}_{21} & K^{(1)}_{22} & K^{(1)}_{23} \\ K^{(1)}_{31} & K^{(1)}_{32} & K^{(1)}_{33} \end{bmatrix}
\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} K^{(2)}_{11} & K^{(2)}_{12} & K^{(2)}_{13} \\ K^{(2)}_{21} & K^{(2)}_{22} & K^{(2)}_{23} \\ K^{(2)}_{31} & K^{(2)}_{32} & K^{(2)}_{33} \end{bmatrix}
\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

or

$$
K_{\alpha\beta} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} = \begin{bmatrix} K^{(1)}_{33} & K^{(1)}_{31} & K^{(1)}_{32} & 0 \\ K^{(1)}_{13} & K^{(1)}_{11} + K^{(2)}_{22} & K^{(1)}_{12} + K^{(2)}_{21} & K^{(2)}_{23} \\ K^{(1)}_{23} & K^{(1)}_{21} + K^{(2)}_{12} & K^{(1)}_{22} + K^{(2)}_{11} & K^{(2)}_{13} \\ 0 & K^{(2)}_{32} & K^{(2)}_{31} & K^{(2)}_{33} \end{bmatrix} \tag{10.1.15a}
$$

Table 10.1.1 Nodal Correspondence Table

e ⇒ / N ⇓	1	2
1	2	3
2	3	2
3	1	4

* Entries indicate global node numbers corresponding to the local nodes (see Figure 10.1.1)

Similarly,

$$F_\alpha = \bigcup_{e=1}^{E} F_N^{(e)} \Delta_{N\alpha}^{(e)}$$

or

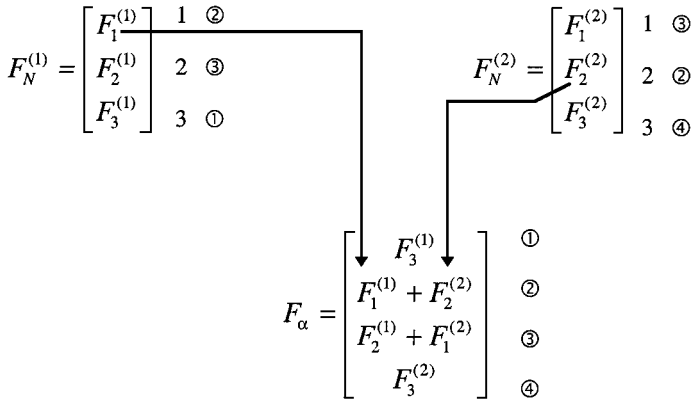
$$F_\alpha = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} F_3^{(1)} \\ F_1^{(1)} + F_2^{(2)} \\ F_2^{(1)} + F_1^{(2)} \\ F_3^{(2)} \end{bmatrix} \tag{10.1.15b}$$

The procedure of assembly implied here requiring determination of Boolean matrices for all elements is quite cumbersome. They are useful and convenient in deriving finite element equations, but are useless in actual performance of assembly operations. Thus, we should avoid Boolean matrices and implement a scheme that can handle complex geometries with a simple algorithm. An intuitive and more convenient approach is schematically shown below.

$$K_{NM}^{(1)} = \begin{array}{c} \\ ② \ 1 \\ ③ \ 2 \\ ① \ 3 \end{array} \begin{array}{c} \begin{array}{ccc} ② & ③ & ① \\ 1 & 2 & 3 \end{array} \\ \begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & K_{13}^{(1)} \\ K_{21}^{(1)} & K_{22}^{(1)} & K_{23}^{(1)} \\ K_{31}^{(1)} & K_{32}^{(1)} & K_{33}^{(1)} \end{bmatrix} \end{array} \qquad K_{NM}^{(2)} = \begin{array}{c} \\ ③ \ 1 \\ ② \ 2 \\ ④ \ 3 \end{array} \begin{array}{c} \begin{array}{ccc} ③ & ② & ④ \\ 1 & 2 & 3 \end{array} \\ \begin{bmatrix} K_{11}^{(2)} & K_{12}^{(2)} & K_{13}^{(2)} \\ K_{21}^{(2)} & K_{22}^{(2)} & K_{23}^{(2)} \\ K_{31}^{(2)} & K_{32}^{(2)} & K_{33}^{(2)} \end{bmatrix} \end{array}$$

$$K_{\alpha\beta} = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array} \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{bmatrix} K_{33}^{(1)} & K_{31}^{(1)} & K_{32}^{(1)} & 0 \\ K_{13}^{(1)} & K_{11}^{(1)} + K_{22}^{(2)} & K_{12}^{(1)} + K_{21}^{(2)} & K_{23}^{(2)} \\ K_{23}^{(1)} & K_{21}^{(1)} + K_{12}^{(2)} & K_{22}^{(1)} + K_{11}^{(2)} & K_{13}^{(2)} \\ 0 & K_{32}^{(2)} & K_{31}^{(2)} & K_{33}^{(2)} \end{bmatrix} \end{array} \tag{10.1.15c}$$

Similarly,



(10.1.15d)

Here, the node number with a circle indicates global node. It is seen that the assembled global matrix is obtained by finding the appropriate entries from the local matrices with the local node numbers replaced by the corresponding incident global node numbers. For example, $K_{11}^{(1)}$ of the first element goes to the second row and second column in the global matrix because the local node 1 is incident with the global node 2. Similarly, $K_{12}^{(1)}$ enters in the second row and third column of the global matrix since the global node number 2 is incident with the global node 3. All entries in the same rows and columns are algebraically added together as we move to the second element. The same procedure applies in order to obtain F_α. In this way, we avoid the need to construct the Boolean matrices, and the entire assembly procedure can be programmed very efficiently.

The global load vector may be obtained more conveniently in the form

$$F_\alpha = C_{\alpha\beta} f_\beta$$

in which only $C_{\alpha\beta}$ is assembled from the local contributions with f_β evaluated at global nodes. This will be shown in Example 10.1.2. The assembly of the Neumann boundary data G_α and the method of implementation will be discussed in Section 10.1.2.

Example 10.1.1 Assembly of Two Triangular Elements

Given:

$$K_{NM}^{(e)} = \iint \left(\frac{\partial \Phi_N^{(e)}}{\partial x} \frac{\partial \Phi_M^{(e)}}{\partial x} + \frac{\partial \Phi_N^{(e)}}{\partial y} \frac{\partial \Phi_M^{(e)}}{\partial y} \right) dxdy$$

Required: Calculate $K_{\alpha\beta} = \bigcup_{e=1}^{E} K_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)}$ by assembling two local linear triangular elements (Figure E10.1.1) to a global form and compare the results with a single isoparametric element of Example 9.3.2. for $n = 4$ and $n = 5$.

Solution:

$$K_{NM}^{(e)} = \iint \left(\frac{\partial \Phi_N^{(e)}}{\partial x} \frac{\partial \Phi_M^{(e)}}{\partial x} + \frac{\partial \Phi_N^{(e)}}{\partial y} \frac{\partial \Phi_M^{(e)}}{\partial y} \right) dxdy = A(b_N b_M + c_N c_M)$$

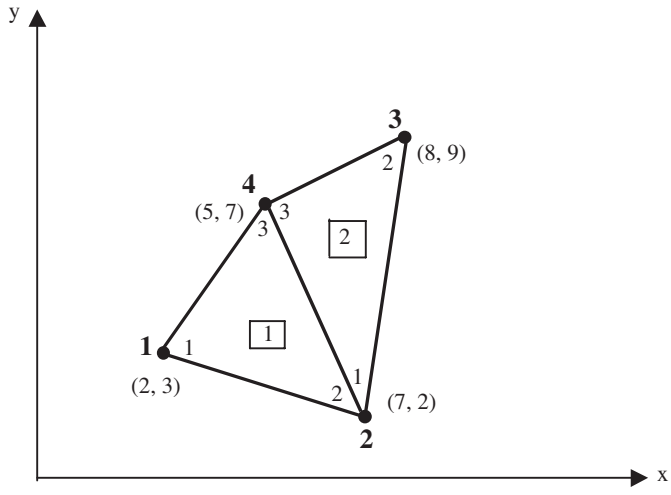


Figure E10.1.1 Assembly of two triangular elements.

where b_N and c_M are given in Example 9.3.1 and A is the triangle area.

$$K_{\alpha\beta} = \begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & 0 & K_{13}^{(1)} \\ & K_{22}^{(1)} + K_{11}^{(2)} & K_{12}^{(2)} & K_{23}^{(1)} + K_{13}^{(2)} \\ & & K_{22}^{(2)} & K_{23}^{(2)} \\ \text{sym} & & & K_{33}^{(1)} + K_{33}^{(2)} \end{bmatrix}$$

$$= \begin{bmatrix} 0.6304 & -0.3043 & 0 & -0.3261 \\ & 0.8856 & 0.1053 & -0.6865 \\ & & 0.7632 & -0.8684 \\ \text{sym} & & & 1.8810 \end{bmatrix}$$

This compares somewhat differently with the single isoparametric element of Example 9.3.2, although diagonal and off-diagonal values are, respectively, about 10% larger and 20% smaller than those for the single quadrilateral isoparametric element (Example 9.3.2). This may influence undoubtedly the solution of differential equations (see Example 10.2.1). It will be shown that the larger diagonal and smaller off-diagonal values in the stiffness matrix result in smaller responses to given boundary input data in general. It is concluded that the triangular element is "stiffer" than the quadrilateral element, as demonstrated in Example 10.2.1. The reason for this behavior is that triangular elements possess only three data points, whereas quadrilateral elements provide four data points allowing an additional degree of freedom and "flexibility."

10.1.2 BOUNDARY CONDITIONS IN TWO DIMENSIONS

(a) Standard Approach

Dirichlet Boundary Conditions. Dirichlet boundary conditions for multidimensional problems can be treated exactly the same as in the case of one-dimensional problems.

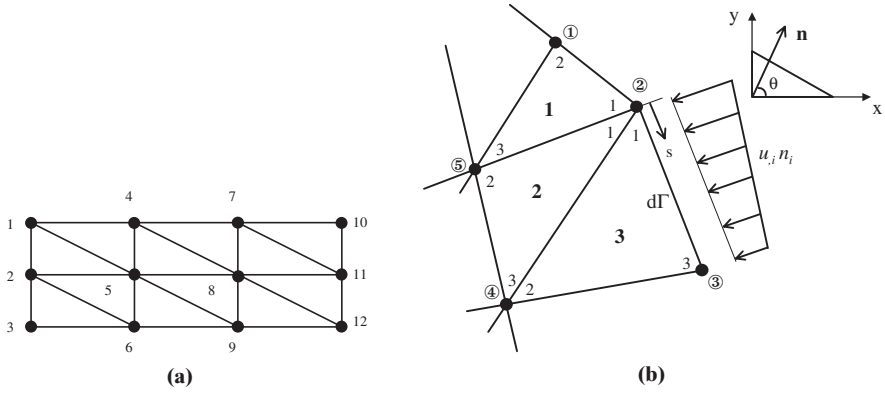


Figure 10.1.2 Boundary conditions. **(a)** Dirichlet boundary conditions ($u_1 = u_2 = u_3 = 2$, $u_4 = u_6 = u_7 = u_9 = u_{10} = u_{11} = u_{12} = 0$). **(b)** Neumann boundary conditions.

That is, the global finite element equations are modified, reflecting the specified Dirichlet data. For example, let us consider that the global finite element equations using either triangular elements or quadrilateral elements have been obtained in the form

$$K_{\alpha\beta} u_\beta = F_\alpha + G_\alpha \tag{10.1.16}$$

where we set $G_\alpha = 0$ because Neumann boundary conditions are not to be specified in this case. Only Dirichlet data are furnished as shown in Figure 10.1.2a. We begin with the assembled global equations,

$$\begin{bmatrix} K_{11} & K_{12} & \cdot & \cdot & \cdot & K_{1\,12} \\ K_{21} & K_{22} & \cdot & \cdot & \cdot & K_{2\,12} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ K_{12\,1} & K_{12\,2} & \cdot & \cdot & \cdot & K_{12\,12} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_{12} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ \cdot \\ F_{12} \end{bmatrix} \tag{10.1.17a}$$

Now, if we apply the Dirichlet boundary conditions in (10.1.17a) as given in Figure 10.1.2a, we obtain

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{55} & 0 & 0 & K_{58} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{85} & 0 & 0 & K_{88} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ F_5 \\ 0 \\ 0 \\ F_8 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \\ 2 \\ 0 \\ -D_5 \\ 0 \\ 0 \\ -D_8 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{10.1.17b}$$

with $D_5 = K_{51}(2) + K_{52}(2) + K_{53}(2)$ and $D_8 = K_{81}(2) + K_{82}(2) + K_{83}(2)$. It is seen that the rows and columns corresponding to the Dirichlet nodes are zero with unity at the diagonal position. The influence of Dirichlet boundary conditions, as imposed here, is reflected in the Dirichlet boundary vector D_α, so that

$$\overline{K}_{\alpha\beta} u_\beta = F_\alpha + D_\alpha \tag{10.1.18}$$

where D_α is given by the second column on the right-hand side with $\overline{K}_{\alpha\beta}$ as modified in (10.1.17) from the given Dirichlet boundary conditions. It is obvious that, if there are so many Dirichlet boundary nodes, then it is convenient to modify the above matrix equations in the form

$$\begin{bmatrix} K_{55} & K_{58} \\ K_{85} & K_{88} \end{bmatrix} \begin{bmatrix} u_5 \\ u_8 \end{bmatrix} = \begin{bmatrix} F_5 \\ F_8 \end{bmatrix} + \begin{bmatrix} -D_5 \\ -D_8 \end{bmatrix} \tag{10.1.19}$$

in which all rows and columns corresponding to Dirichlet boundary nodes are eliminated.

Neumann Boundary Conditions. Neumann boundary conditions are implemented using the integral form of (10.1.5c) with the local contributions coming from adjacent elements to the node at which Neumann data $g_M^{(e)}$ are prescribed in the form (10.1.8b),

$$g_M^{(e)} = (u_{,i} n_i)_M = \left(\frac{\partial u}{\partial x} \cos\theta + \frac{\partial u}{\partial y} \sin\theta \right)_M \tag{10.1.20}$$

as shown in Figure 10.1.2b with the normal angle θ measured counterclockwise from the axis.

Often in boundary value problems, there are instances in which the Dirichlet and Neumann boundary conditions are combined at the same location. For example, consider a heat conduction equation

$$k\nabla^2 T = 0$$

Here, for a resistance layer on the boundary, we specify

$$kT_{,i} n_i + \overline{\alpha}(T - T') = -q \tag{10.1.21}$$

where T, T', $\overline{\alpha}$, and q denote the surface temperature, ambient temperature, heat transfer coefficient, and surface heat flux, respectively. This is referred to as the Cauchy or Robin boundary condition and can be handled by substitution:

$$kT_{,i} n_i = -Q - \overline{\alpha} T$$

with

$$Q = q - \overline{\alpha} T'$$

Thus, we write

$$G_\alpha = \hat{G}_\alpha - \overset{*}{C}_{\alpha\beta} T_\beta \tag{10.1.22}$$

with

$$\hat{G}_\alpha = -\int_\Gamma \overset{*}{\Phi}_\alpha Q d\Gamma = \bigcup_{e=1}^{E} \hat{G}_N^{(e)} \Delta_{N\alpha}^{(e)}, \quad \hat{G}_N^{(e)} = -\int_\Gamma \overset{*}{\Phi}{}_N^{(e)} Q d\Gamma$$

$$\overset{*}{C}_{\alpha\beta} = \int_\Gamma \overline{\alpha}\, \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta d\Gamma = \bigcup_{e=1}^{E} \overset{*}{C}{}_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)}, \quad \overset{*}{C}{}_{NM}^{(e)} = \int_\Gamma \overline{\alpha}\, \overset{*}{\Phi}{}_N^{(e)} \overset{*}{\Phi}{}_M^{(e)} d\Gamma$$

This process then modifies (10.1.4) in the form

$$\left(K_{\alpha\beta} + \overset{*}{C}_{\alpha\beta}\right) T_\beta = F_\alpha + \hat{G}_\alpha \tag{10.1.23}$$

It should be noted that $\overset{*}{C}_{\alpha\beta}$ is activated only if the convection or Cauchy boundary conditions are present. That is, if a global node does not coincide with the boundary node at which the Neumann boundary conditions are prescribed, then $\overset{*}{C}_{\alpha\beta}$ is empty from the definition, $\overset{*}{\Phi}_\alpha(Z_\beta) = \delta_{\alpha\beta}$. It is cautioned that the local boundary surface matrix is (2×2), which is simply added to the local triangular element stiffness matrix (3×3) in correspondence with the nodal incidence along the boundaries.

(b) Lagrange Multipliers Approach

Any boundary condition prescribed at a boundary node may be imposed through Lagrange multipliers. Consider the boundary conditions of the form

$$u_1 = 0 \tag{10.1.24a}$$

$$u_2 = a \tag{10.1.24b}$$

$$u_3 - u_4 = b \tag{10.1.24c}$$

Obviously, if $b = 0$, then the second expression implies $u_3 = u_4$. Otherwise, it represents Neumann boundary conditions $(du/dx)\cos\theta$ or $(du/dy)\sin\theta$, prescribed at the global node Z_3 connected to the adjacent boundary node Z_4. For example, if $du/dx = c$ at Z_3 and the boundary line of length l between Z_3 and Z_4 is inclined an angle of θ from the x axis, then we write

$$\frac{du}{dx} = \frac{u_3 - u_4}{l\cos\theta} = c \tag{10.1.25}$$

or

$$u_3 - u_4 = b \quad \text{with } b = cl\cos\theta$$

Equation (10.1.24) can be written in the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & -1 & 0 & \cdots \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} 0 \\ a \\ b \end{bmatrix} \tag{10.1.26}$$

which may be rearranged as

$$q_{r\alpha}u_\alpha = E_r \tag{10.1.27}$$

with $r = 1, \ldots, m$ (total number of boundary conditions, $m = 3$ in this case) and $\alpha = 1, \ldots, n$ (total number of global nodes). Here, $q_{r\alpha}$ is called the boundary condition matrix. Let us now introduce quantities λ_r, referred to as Lagrange multipliers, and regarded as constraints or forces required to maintain the boundary conditions. Then, the product of (10.1.27) with the Lagrange multiplier λ_r

$$\lambda_r(q_{r\alpha}u_\alpha - E_r) = 0 \tag{10.1.28}$$

may be considered as an invariant or energy required to maintain such boundary conditions.

At this point, we transform the global finite element equation (10.1.16) into a variational energy,

$$\delta I = (K_{\alpha\beta}u_\beta - H_\alpha)\delta u_\alpha = 0 \tag{10.1.29}$$

or

$$\delta I = \delta\left(\frac{1}{2}K_{\alpha\beta}u_\alpha u_\beta - H_\alpha u_\alpha\right) = 0 \tag{10.1.30}$$

for which the stationary condition is given by

$$I = \frac{1}{2}K_{\alpha\beta}u_\alpha u_\beta - H_\alpha u_\alpha \tag{10.1.31}$$

This may be considered as the actual energy contained in the domain. To this we may add (10.1.28),

$$I = \frac{1}{2}K_{\alpha\beta}u_\alpha u_\beta - H_\alpha u_\alpha + \lambda_r(q_{r\alpha}u_\alpha - E_r) \tag{10.1.32}$$

The expression (10.1.32) refers to the total variational energy in equilibrium with the imposed boundary conditions. The variation of (10.1.32) with respect to every u_α and λ_r will lead to the stationary condition

$$\delta I = \frac{\partial I}{\partial u_\alpha}\delta u_\alpha + \frac{\partial I}{\partial \lambda_r}\delta\lambda_r = 0 \tag{10.1.33}$$

Since u_α and λ_r are arbitrary, it is necessary that $\partial I/\partial u_\alpha$ and $\partial I/\partial\lambda_r$ vanish. These conditions yield

$$K_{\alpha\beta}u_\alpha + \lambda_r q_{r\alpha} = H_\alpha$$
$$q_{r\alpha}u_\alpha = E_r$$

Writing these two equations in matrix form, we obtain

$$\begin{bmatrix} K_{\alpha\beta} & q_{r\alpha} \\ q_{r\beta} & 0 \end{bmatrix}\begin{bmatrix} u_\beta \\ \lambda_r \end{bmatrix} = \begin{bmatrix} H_\alpha \\ E_r \end{bmatrix} \tag{10.1.34}$$

which may be expanded with the boundary conditions of (10.1.26) in the form

$$\begin{bmatrix} K_{11} & K_{12} & \cdot & \cdot & \cdot & \cdot & K_{1n} & 1 & 0 & 0 \\ K_{21} & K_{22} & \cdot & \cdot & \cdot & \cdot & K_{2n} & 0 & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & -1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ K_{n1} & K_{n2} & \cdot & \cdot & \cdot & \cdot & K_{nn} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdot & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & \cdot & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \cdot \\ u_n \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \cdot \\ \cdot \\ \cdot \\ H_n \\ E_1 \\ E_2 \\ E_3 \end{bmatrix} \tag{10.1.35}$$

The solution to these equations provides the values of Lagrange multipliers λ_r as well as the unknowns u_α. Here λ_r, interpreted as the boundary forces, assisted in imposing the boundary conditions. Note that the left-hand side matrix (10.1.35) is still symmetric, but matrix rearrangements are required to avoid zeros on the diagonal before a standard equation solver is applied.

Remarks: The Lagrange multiplier approach for implementing boundary conditions is useful if the finite element formulations are performed by means of methods of least squares, moments, or collocation in which the Neumann boundary conditions do not arise naturally since integration by parts is not involved in these methods.

10.1.3 SOLUTION PROCEDURE

In order to illustrate the solution procedure and implementation of both Dirichlet and Neumann boundary conditions, we present the following examples.

Example 10.1.2 Solution of Poisson Equation by Triangular Elements

Given:

$$u_{,ii} = f \quad (i = 1, 2)$$

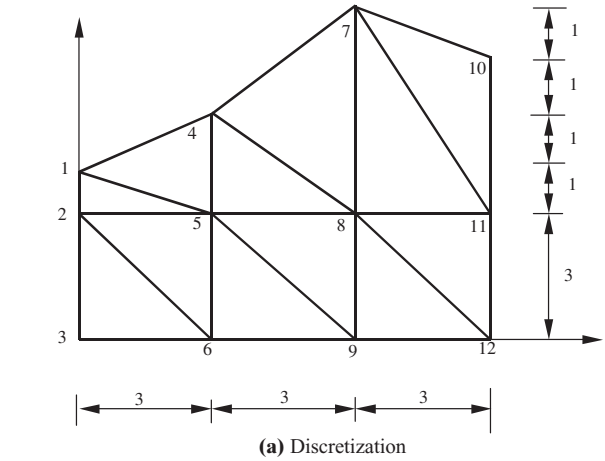
with $f = 4(x^2 + y^2)$, exact solution: $u = 2x^2y^2$.
Consider the geometry (Figure E10.1.2) with Dirichlet boundary conditions:

(1) $u_2 = u_3 = u_6 = u_9 = u_{12} = 0$
(2) $u_{11} = 1,458$
(3) $u_1 = 0, \quad u_4 = 450, \quad u_7 = 3,528, \quad u_{10} = 5,832$

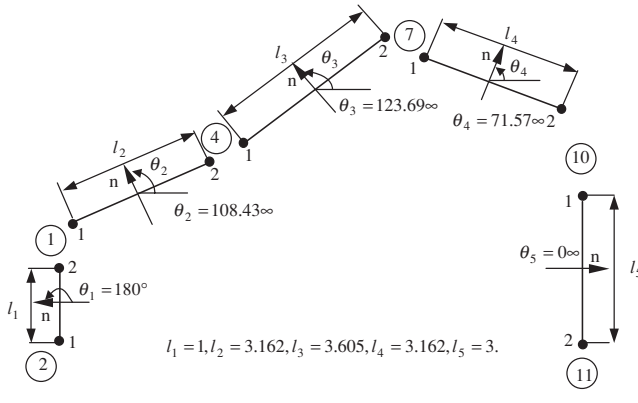
Neumann boundary conditions along nodes 1, 4, 7, and 10:

(4) $\left(\frac{\partial u}{\partial x}\right)_1 = 0, \quad \left(\frac{\partial u}{\partial x}\right)_4 = 300, \quad \left(\frac{\partial u}{\partial x}\right)_7 = 1{,}176, \quad \left(\frac{\partial u}{\partial x}\right)_{10} = 1{,}296,$

$\left(\frac{\partial u}{\partial y}\right)_1 = 0 \quad \left(\frac{\partial u}{\partial y}\right)_4 = 180 \quad \left(\frac{\partial u}{\partial y}\right)_7 = 1008 \quad \left(\frac{\partial u}{\partial y}\right)_{10} = 1{,}944$



(a) Discretization



(b) Neumann boundary elements

Figure E10.1.2 Two-dimensional problem with linear triangular elements.

Required: Using the linear triangular elements, solve the differential equation with the boundary conditions:

(a) – (1), (2), and (3)
(b) – (1), (2), and (4)

Solution:

$$K_{\alpha\beta} u_\beta = F_\alpha + G_\alpha$$

$$K_{\alpha\beta} = \bigcup_{e=1}^{E} K_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)}$$

$$K_{NM}^{(e)} = \int_\Omega \Phi_{N,i}^{(e)} \Phi_{M,i}^{(e)} d\Omega = A(b_N b_M + c_N c_M)$$

$$F_\alpha = -\int_\Omega \Phi_\alpha f d\Omega = -\int_\Omega \Phi_\alpha \Phi_\beta d\Omega f_\beta = -C_{\alpha\beta} f_\beta = -\bigcup_{e=1}^{E} C_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} f_\beta$$

or

$$F_\alpha = -C_{\alpha\beta} f_\beta, \qquad C_{\alpha\beta} = \bigcup_{e=1}^{E} C_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)}, \qquad f_\alpha = [4(x^2 + y^2)]_\alpha$$

The $C_{NM}^{(e)}$ may be determined using (9.3.5) or (9.3.27).
From (9.3.5), we have

$$C_{NM}^{(e)} = \iint (a_N + b_N x + c_N y)(a_M + b_M x + c_M y) dx dy$$

$$C_{11}^{(e)} = A^{(e)} \left[\frac{1}{9} + \frac{1}{12} \left(b_1^2 \alpha + 2 b_1 c_1 \beta + c_1^2 \gamma \right) \right]$$

$$C_{12}^{(e)} = A^{(e)} \left\{ \frac{1}{9} + \frac{1}{12} [b_1 b_2 \alpha + (b_1 c_2 + b_2 c_1)\beta + c_1 c_2 \gamma] \right\}$$

$$C_{13}^{(e)} = A^{(e)} \left\{ \frac{1}{9} + \frac{1}{12} [b_1 b_3 \alpha + (b_1 c_3 + b_3 c_1)\beta + c_1 c_3 \gamma] \right\}$$

$$C_{22}^{(e)} = A^{(e)} \left[\frac{1}{9} + \frac{1}{12} \left(b_2^2 \alpha + 2 b_2 c_2 \beta + c_2^2 \gamma \right) \right]$$

$$C_{23}^{(e)} = A^{(e)} \left\{ \frac{1}{9} + \frac{1}{12} [b_2 b_3 \alpha + (b_2 c_3 + b_3 c_2)\beta + c_2 c_3 \gamma] \right\}$$

$$C_{33}^{(e)} = A^{(e)} \left[\frac{1}{9} + \frac{1}{12} \left(b_3^2 \alpha + 2 b_3 c_3 \beta + c_3^2 \gamma \right) \right]$$

with

$$\alpha = x_1^2 + x_2^2 + x_3^2, \qquad \beta = x_1 y_1 + x_2 y_2 + x_3 y_3, \qquad \gamma = y_1^2 + y_2^2 + y_3^2$$

After some algebra, it can be shown that

$$C_{NM}^{(e)} = \frac{A^{(e)}}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

This result can be obtained easily from (9.3.11 and 9.3.27) using the natural coordinate triangular element.

$$C_{NM}^{(e)} = \int_\Omega \Phi_N^{(e)} \Phi_M^{(e)} d\Omega = \iint \begin{bmatrix} L_1 L_1 & L_1 L_2 & L_1 L_3 \\ L_2 L_1 & L_2 L_2 & L_2 L_3 \\ L_3 L_1 & L_3 L_2 & L_3 L_3 \end{bmatrix} dx dy = \frac{A^{(e)}}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

Thus, the global load vector is calculated from the assembly of $C_{NM}^{(e)}$ matrices for each element into a global form $C_{\alpha\beta}$ to be multiplied by the global nonhomogeneous data f_β determined at each global node.

The Neumann boundary vector G_α can be calculated as follows:

$$G_\alpha = \int_\Gamma \overset{*}{\Phi}_\alpha u_{,i} n_i d\Gamma = \int_\Gamma \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta d\Gamma g_\beta = \bigcup_{e=1}^{E} \overset{*}{C}_{NM}^{(e)} \Delta_{N\alpha}^{(e)} g_M^{(e)} = \bigcup_{e=1}^{E} G_N^{(e)} \Delta_{N\alpha}^{(e)}$$

where

$$\overset{*}{C}{}^{(e)}_{NM} = \int_0^l \overset{*}{\Phi}{}^{(e)}_N \overset{*}{\Phi}{}^{(e)}_M d\Gamma = \frac{l}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Thus

$$G^{(e)}_N = \frac{l}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} g^{(e)}_1 \\ g^{(e)}_2 \end{bmatrix} = \frac{l}{6}\begin{bmatrix} 2g^{(e)}_1 + g^{(e)}_2 \\ g^{(e)}_1 + 2g^{(e)}_2 \end{bmatrix}$$

where $\overset{*}{\Phi}{}^{(e)}_M$ vanishes everywhere except at Neumann boundary nodes. Recall that $\overset{*}{\Phi}{}^{(e)}_M(z_M) = \delta_{NM}$ and thus, $\overset{*}{\Phi}{}^{(e)}_M = 0$ if the boundary node N does not have the Neumann data prescribed, and $\overset{*}{\Phi}{}^{(e)}_M = 1$ if the boundary node N has the Neumann boundary data prescribed.

$$G^{(1)}_N = \frac{l_1}{6}\begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} g^{(1)}_1 \\ g^{(1)}_2 \end{bmatrix} = \frac{l_1}{6}\begin{bmatrix} 0 \\ 2g^{(1)}_2 \end{bmatrix}$$

with $\overset{*}{\Phi}{}^{(1)}_N = 0$, because the Neumann data are not prescribed at the local node 1 for the boundary element 1.

$$G^{(2)}_N = \frac{l_2}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} g^{(2)}_1 \\ g^{(2)}_2 \end{bmatrix} = \frac{l_2}{6}\begin{bmatrix} 2g^{(2)}_1 + g^{(2)}_2 \\ g^{(2)}_1 + 2g^{(2)}_2 \end{bmatrix}$$

$$G^{(3)}_N = \frac{l_3}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} g^{(3)}_1 \\ g^{(3)}_2 \end{bmatrix} = \frac{l_3}{6}\begin{bmatrix} 2g^{(3)}_1 + g^{(3)}_2 \\ g^{(3)}_1 + 2g^{(3)}_2 \end{bmatrix}$$

$$G^{(4)}_N = \frac{l_4}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} g^{(4)}_1 \\ g^{(4)}_2 \end{bmatrix} = \frac{l_4}{6}\begin{bmatrix} 2g^{(4)}_1 + g^{(4)}_2 \\ g^{(4)}_1 + 2g^{(4)}_2 \end{bmatrix}$$

$$G^{(5)}_N = \frac{l_5}{6}\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} g^{(5)}_1 \\ g^{(5)}_2 \end{bmatrix} = \frac{l_5}{6}\begin{bmatrix} 2g^{(5)}_1 \\ 0 \end{bmatrix}$$

with $\overset{*}{\Phi}{}^{(5)}_2 = 0$ and

$$g^{(1)}_2 = \left(\frac{\partial u}{\partial x}\cos\theta + \frac{\partial u}{\partial y}\sin\theta\right)^{(1)}_2 = \left(\frac{\partial u}{\partial x}\right)^{(1)}_2 (-1) = 0$$

$$g^{(2)}_1 = \left(\frac{\partial u}{\partial x}\right)^{(2)}_1 (-0.316) + \left(\frac{\partial u}{\partial y}\right)^{(2)}_1 (0.948) = 0$$

$$g^{(2)}_2 = \left(\frac{\partial u}{\partial x}\right)^{(2)}_2 (-0.316) + \left(\frac{\partial u}{\partial y}\right)^{(2)}_2 (0.948)$$

$$= (300)(-0.316) + (180)(0.948) = 75.84$$

Similarly,

$$g_1^{(3)} = -16.74, \quad g_2^{(3)} = 185.97, \quad g_1^{(4)} = 1,328.2, \quad g_2^{(4)} = 2,254.4,$$

$$g_1^{(5)} = \left(\frac{\partial u}{\partial x}\right)_1^{(5)}(1) = 1,296$$

$$G_\alpha = \begin{bmatrix} G_1 \\ G_4 \\ G_7 \\ G_{10} \end{bmatrix} = \frac{1}{6}\begin{bmatrix} \ell_1 2g_2^{(1)} + \ell_2\left(2g_1^{(2)} + g_2^{(2)}\right) \\ \ell_2\left(g_1^{(2)} + 2g_2^{(2)}\right) + \ell_3\left(2g_1^{(3)} + g_2^{(3)}\right) \\ \ell_3\left(g_1^{(3)} + 2g_2^{(3)}\right) + \ell_4\left(2g_1^{(4)} + g_2^{(4)}\right) \\ \ell_4\left(g_1^{(4)} + 2g_2^{(4)}\right) + \ell_5\left(2g_1^{(5)}\right) \end{bmatrix} = \begin{bmatrix} 40.00 \\ 172.03 \\ 2,802.05 \\ 4,372.02 \end{bmatrix}$$

with $G_\alpha = 0$ elsewhere. The sum of $F_\alpha + G_\alpha$ is given by

$$F_\alpha + G_\alpha = -\begin{bmatrix} 113.50 \\ 134.00 \\ 27.00 \\ 629.00 \\ 609.50 \\ 216.00 \\ 1673.50 \\ 2008.00 \\ 648.00 \\ 613.50 \\ 1652.00 \\ 810.00 \end{bmatrix} + \begin{bmatrix} 40.00 \\ 0.00 \\ 0.00 \\ 172.03 \\ 0.00 \\ 0.00 \\ 2802.05 \\ 0.00 \\ 0.00 \\ 4372.02 \\ 0.00 \\ 0.00 \end{bmatrix}$$

Note that G_α is obtained by an assembly of local data $g_M^{(e)}$. However for F_α, it is preferable to construct the $C_{\alpha\beta}$ matrix independent of local data $f_M^{(e)}$ and use the global data f_α instead.

The solution is carried out, and the results are shown in Table E10.1.1. It is seen that the solution for the Neumann data is less accurate than for the Dirichlet data. It can be shown that accuracy improves with mesh refinements. This is demonstrated in Section 10.4.1 for isoparametric elements.

10.1.4 STOKES FLOW PROBLEMS

Stokes flows or creeping flows occur in highly viscous, slowly moving fluids and are characterized by the conservation of mass and momentum. For a steady state, the governing equations take the form

$$\nabla \cdot \mathbf{v} = 0 \tag{10.1.36a}$$

$$-\mu\nabla^2\mathbf{v} + \nabla p - \rho\mathbf{F} = 0 \tag{10.1.36b}$$

Although these equations are still linear (note that convective terms are absent), their solutions may not be easy to obtain because the enforcement of incompressibility

Table E10.1.1 Computed Results for Example 10.1.2

(a) Dirichlet Problem with the Boundary Conditions (1), (2), and (3)

Node	Exact Solution	FEM Solution	% Error
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	450.00	450.00	0.00
5	162.00	110.72	−31.66
6	0.00	0.00	0.00
7	3528.00	3528.00	0.00
8	648.00	508.92	−21.46
9	0.00	0.00	0.00
10	5832.00	5832.00	0.00
11	1458.00	1458.00	0.00
12	0.00	0.00	0.00

(b) Neumann Problem with the Boundary Conditions (1), (2), and (4)

Node	Exact Solution	FEM Solution	% Error
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	450.00	392.33	−12.82
5	162.00	79.57	−50.88
6	0.00	0.00	0.00
7	3528.00	3264.54	−7.47
8	648.00	458.15	−29.30
9	0.00	0.00	0.00
10	5832.00	5031.26	−13.73
11	1458.00	1458.00	0.00
12	0.00	0.00	0.00

conditions (conservation of mass) is difficult. As a result, the computed pressure, p, may be spurious and oscillatory, known as checkerboard type oscillations.

To cope with these difficulties, many methods have been reported in the literature [Carey and Oden, 1986; Zienkiewicz and Taylor, 1991]. Among them are the mixed methods and penalty methods, which are presented below.

Mixed Methods

The momentum equation has the second derivative of velocity ($\mathbf{v}\ \varepsilon\ H^2$) and first derivative of pressure ($p\ \varepsilon\ H^1$). In order to enforce the mass conservation (incompressibility condition) we must use an appropriate function for the pressure consistent with the functional space for the velocity. This is known as the "consistency condition" or "LBB condition" after Ladyzhenskaya [1969], Babuska [1973], and Brezzi [1974]. This condition requires that the trial function for pressure in the momentum equation and

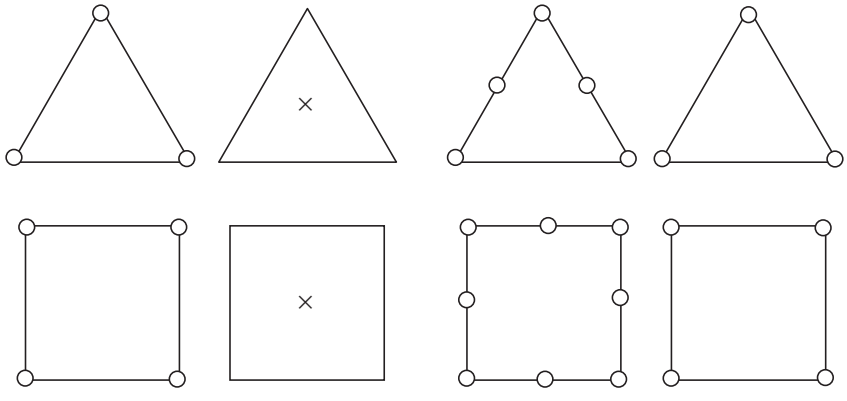


Figure 10.1.3 Mixed methods with triangles and quadrilaterals. **(a)** Mixed interpolation with constant pressure. **(b)** Mixed interpolation with linear pressure.

the test function for the continuity equation be chosen one order lower than the test function for the momentum equation and trial function for the velocity in the continuity equation, respectively.

Based on these requirements, the SGM equations of (10.1.36a,b) are of the form

$$\begin{bmatrix} A_{\alpha\beta ik} & B_{\alpha\beta i} \\ B_{\alpha\beta k} & 0 \end{bmatrix} \begin{bmatrix} \mathrm{v}_{\beta k} \\ p_\beta \end{bmatrix} = \begin{bmatrix} F_{\alpha i} \\ 0 \end{bmatrix} + \begin{bmatrix} G_{\alpha i} \\ 0 \end{bmatrix} \tag{10.1.37}$$

If pressure is interpolated as constant (pressure node at the center of an element) and velocity as a linear function (velocity defined at corner nodes), then such element becomes over-constrained (known as "locking element") (Figure 10.1.3a). To avoid this situation, we may use linear pressure and quadratic velocity interpolations (Figure 10.1.3b). However, experience has shown that further improvements are needed in order to expedite convergence toward acceptable solutions. This subject will be elaborated in Chapter 12.

Penalty Methods

Penalty methods are designed such that the continuity equation which actually represents a constraint condition can be eliminated from the solution process. This is achieved by setting

$$p = -\lambda \boldsymbol{\nabla} \cdot \mathbf{v} \tag{10.1.38}$$

where λ is the penalty parameter, equivalent to the Lagrange multiplier. The idea is to set λ equal to a large number ($\lambda \to \infty$) in the hope that $\boldsymbol{\nabla} \cdot \mathrm{v} \approx 0$ as seen from

$$\boldsymbol{\nabla} \cdot \mathbf{v} + \frac{p}{\lambda} \cong 0 \tag{10.1.39}$$

Substituting (10.1.38) into (10.1.36b), we obtain

$$-\mu \boldsymbol{\nabla}^2 \mathrm{v} - \lambda \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathrm{v}) - \rho \mathbf{F} = 0 \tag{10.1.40}$$

Here λ is seen to act as dilatational viscosity. It is now clear that pressure is eliminated from the solution of (10.1.40) in which the mass conservation is enforced through (10.1.39). Once the velocity components are calculated from (10.1.40), then pressure is calculated by means of (10.3.38).

Unfortunately, however, the solution of (10.1.40) is difficult because the penalty term dominates as λ becomes large, which is analogous to the over-constraint in the mixed methods. In other words, the consistency condition is violated. To cope with this difficulty, the finite element equation integral term involving the penalty function (pressure term) is given a special treatment by means of "reduced" Gaussian quadrature numerical integration. Specifically, we under-integrate the penalty term one point less than the shear viscosity term. For example, one point Gaussian quadrature rule for the penalty term is performed against the two-point rule for the shear viscosity term of a linear element. Similarly, a two-point rule for the penalty term against a three-point rule for the shear viscosity term of a quadratic element is recommended, and so on.

Once again, the mixed methods and penalty methods represent relatively earlier developments. They are being replaced by more efficient and advanced techniques to be discussed in Chapter 12 for incompressible viscous flows.

10.2 TRANSIENT PROBLEMS – GENERALIZED GALERKIN METHODS

10.2.1 PARABOLIC EQUATIONS

To describe the time-dependent behavior, we may use either the continuous space-time (CST) method or the discontinuous space-time (DST) method. In the CST method, continuous interpolation functions in both space and time are used so that

$$u(\mathbf{x}, t) = \Phi_\alpha(\mathbf{x}, t) u_\alpha \tag{10.2.1}$$

Alternatively, the DST method allows separation of variables between the spatial and temporal domains,

$$u(\mathbf{x}, t) = \Phi_\alpha(\mathbf{x}) u_\alpha(t) \tag{10.2.2}$$

This requires interpolations of $\Phi_\alpha(\mathbf{x})$ in the spatial domain and the nodal values $u_\alpha(t)$ for the temporal domain.

The disadvantage of the CST method is the increase in computational dimension requiring the finite element in time. For this reason, our discussions in the sequel will be limited to the DST method, in which a time marching procedure is followed.

Consider a parabolic equation or the time-dependent differential equation in the form

$$R = \frac{\partial u(\mathbf{x}, t)}{\partial t} - \nabla^2 u(\mathbf{x}, t) - f(\mathbf{x}, t) = 0 \tag{10.2.3}$$

Let the nondimensional temporal variable be given by

$$\xi = t/\Delta t \tag{10.2.4}$$

where t and Δt denote time and a small time step, respectively.

In the past, the so-called semidiscrete method was used, in which the SGM equation for (10.2.3) is written as

$$(\Phi_\alpha, R) = \int_\Omega \Phi_\alpha \left(\frac{\partial u}{\partial t} - u_{,ii} - f \right) d\Omega = 0$$

where the time derivative of u is approximated by finite differences. Instead, our approach in DST is to seek a temporal test function independently and discontinuously from the spatial test function.

The DST method consists of first constructing the inner product of the residual (10.2.3) with the spatial test function $\Phi_\alpha(x)$ over the spatial domain and, subsequently, constructing another inner product of the resulting residual with the temporal weighting function or test function $\hat{W}(\xi)$ over the temporal domain. These steps lead to

$$(\hat{W}(\xi), (\Phi_\alpha, R)) = \int_0^1 \hat{W}(\xi) \left[\int_\Omega \Phi_\alpha \left(\frac{\partial u}{\partial t} - u_{,ii} - f \right) d\Omega \right] d\xi = 0 \tag{10.2.5}$$

which represents the SGM with DST approximations. The double projections of the residual onto the subspaces spanned by spatial and temporal test functions are referred to as the generalized Galerkin Method (GGM) as opposed to SGM. As noted in (8.2.41), the temporal weighting function $\hat{W}(\xi)$ is independent of and discontinuous from the spatial approximations.

Substituting (10.2.2) into (10.2.5) yields

$$\int_0^1 \hat{W}(\xi) \left[A_{\alpha\beta} \frac{\partial u_\beta(t)}{\partial t} + K_{\alpha\beta} u_\beta(t) - H_\alpha \right] d\xi = 0 \tag{10.2.6}$$

where we may define

Mass Matrix

$$A_{\alpha\beta} = \int_\Omega \Phi_\alpha \Phi_\beta d\Omega \tag{10.2.7}$$

Stiffness Matrix

$$K_{\alpha\beta} = \int_\Omega \Phi_{\alpha,i} \Phi_{\beta,i} d\Omega \tag{10.2.8}$$

$$H_\alpha = F_\alpha + G_\alpha \tag{10.2.9}$$

with

Source Vector $$F_\alpha = \int_\Omega \Phi_\alpha f d\Omega$$

Neumann Boundary Vector $$G_\alpha = \int_\Gamma \overset{*}{\Phi}_\alpha u_{,i} n_i d\Gamma.$$

If linear variations of $u_\alpha(t)$ are assumed within a small time step, we may write

$$u_\alpha(t) = \hat{\Phi}_m(\xi) u_\alpha^m \quad (m = 1, 2) \tag{10.2.10}$$

where the temporal trial functions may be derived from the standard one-dimensional configuration,

$$\hat{\Phi}_1 = 1 - \xi, \qquad \hat{\Phi}_2 = \xi$$

Thus,

$$u_\alpha(t) = (1-\xi)u_\alpha^n + \xi u_\alpha^{n+1} \tag{10.2.11}$$

in which $m = 1$ and $m = 2$ are replaced by the time steps n and $n+1$, respectively.

Differentiating (10.2.11) with respect to time, we obtain

$$\frac{\partial u_\alpha(t)}{\partial t} = \frac{\partial u_\alpha(\xi)}{\partial \xi}\frac{\partial \xi}{\partial t} = \frac{1}{\Delta t}\left(u_\alpha^{n+1} - u_\alpha^n\right) \tag{10.2.12}$$

which is identical to the forward finite difference of $\partial u(t)/\partial t$. Substituting (10.2.12) into (10.2.6) yields

$$\left[A_{\alpha\beta} + \eta\Delta t\, K_{\alpha\beta}\right] u_\beta^{n+1} = \left[A_{\alpha\beta} - (1-\eta)\Delta t\, K_{\alpha\beta}\right] u_\beta^n + \Delta t\, H_\alpha \tag{10.2.13}$$

where H_α may be regarded as the forcing function. If H_α is time dependent, then it may be expanded in a manner similar to u_α given in (10.2.11).

$$H_\alpha = (1-\xi)\, H_\alpha^n + \xi\, H_\alpha^{n+1}$$

Temporal Parameter

We define η as the temporal parameter,

$$\eta = \frac{\displaystyle\int_0^1 \hat{W}(\xi)\xi\, d\xi}{\displaystyle\int_0^1 \hat{W}(\xi)\, d\xi} \tag{10.2.14}$$

Evaluation of the temporal parameter requires an explicit form for the temporal test function $\hat{W}(\xi)$ as introduced in Zienkiewicz and Taylor [1991]. Some of the examples for $\hat{W}(\xi)$ and the corresponding temporal parameters are shown in Table 10.2.1. A glance at the temporal parameters suggested above reveals that they remain in the range

$$0 \le \eta \le 1$$

Equation (10.2.13) may be written in the form

$$D_{\alpha\beta} u_\beta^{n+1} = Q_\alpha^n \tag{10.2.15}$$

Table 10.2.1 Temporal Parameters for Parabolic Equations

$\hat{W}(\xi)$	η
$1-\xi$	1/3
ξ	2/3
1	1/2
$\delta(\xi-0)$	0
$\delta(\xi-1/2)$	1/2
$\delta(\xi-1)$	1

with

$$D_{\alpha\beta} = A_{\alpha\beta} + \eta\Delta t\, K_{\alpha\beta}$$
$$Q_\alpha^n = [A_{\alpha\beta} - (1-\eta)\Delta t\, K_{\alpha\beta}]u_\beta^n + \Delta t\, H_\alpha$$

Notice that, to solve (10.2.15), we must first apply the boundary conditions in a manner similar to that used in the steady-state problems. Initial conditions can be specified in Q_α^n. Initially, $n = 0$, and $u_\alpha^{(1)}$ for the first step is calculated from $Q_\alpha^{(0)}$. Then $u_\beta^{(2)}$ for the second time step will be calculated from $u_\alpha^{(1)}$ substituted into $Q_\alpha^{(1)}$, thus continuously marching in time until the desired time has been reached. An adequate choice of the temporal parameter η and the time step Δt is regarded as crucial to the success of the analysis. To this end, we examine the two cases in which $\eta = 0$ and $\eta \neq 0$, corresponding to the *explicit* scheme and the *implicit* scheme, respectively. Notice that $\eta = 1/2$ corresponds to the so-called Crank-Nicolson scheme (Section 4.3.2).

Explicit Scheme

The explicit scheme refers to the case $\eta = 0$. Rewrite (10.2.13) in the form

$$u_\alpha^{n+1} = A_{\alpha\gamma}^{-1}\left[(A_{\gamma\beta} - \Delta t\, K_{\gamma\beta})u_\beta^n + \Delta t\, H_\gamma\right] \tag{10.2.16}$$

and assume that errors are generated each time step, giving ε_α^n and ε_α^{n+1} corresponding to u_α^n and u_α^{n+1}, respectively, such that

$$u_\alpha^{n+1} + \varepsilon_\alpha^{n+1} = A_{\alpha\gamma}^{-1}\left[(A_{\gamma\beta} - \Delta t\, K_{\gamma\beta})\left(u_\beta^n + \varepsilon_\beta^n\right) + \Delta t\, H_\gamma\right] \tag{10.2.17}$$

Subtracting (10.2.16) from (10.2.17) yields

$$\varepsilon_\alpha^{n+1} = g_{\alpha\gamma}\varepsilon_\gamma^n \tag{10.2.18}$$

where $g_{\alpha\gamma}$ is the amplification matrix

$$g_{\alpha\gamma} = \delta_{\alpha\gamma} - A_{\alpha\delta}^{-1}K_{\delta\gamma}\Delta t \tag{10.2.19}$$

For stable solutions, we must assure that errors at the nth step do not grow toward the $(n+1)$th step; that is,

$$\left|\varepsilon_\alpha^{n+1}\right| \leq \left|\varepsilon_\alpha^n\right|$$

This requirement can be met when

$$|g_{\alpha\gamma}| = |\delta_{\alpha\gamma} - A_{\alpha\delta}^{-1}K_{\gamma\delta}\Delta t| \leq |\delta_{\alpha\gamma}| = 1 \tag{10.2.20}$$

Thus, in view of (10.2.19) and (10.2.20), and setting

$$\varepsilon_\alpha^{n+1} = \lambda\varepsilon_\alpha^n \tag{10.2.21}$$

we write

$$(g_{\alpha\gamma} - \lambda\delta_{\alpha\gamma})\varepsilon_\gamma^n = 0 \tag{10.2.22}$$

The stability of the solution of (10.2.16) can be assured if each and every eigenvalue λ_α of the amplification matrix $g_{\alpha\gamma}$ is made smaller than unity,

$$|\lambda_\alpha| \leq 1$$

The largest eigenvalue, called the *spectral radius*, governs the stability. Since there exists a bound for Δt outside of which stability can no longer be maintained, the explicit scheme is said to be *conditionally stable*.

Implicit Scheme

The implicit scheme arises for $\eta \neq 0$ in (10.2.13). Solving for u_α^{n+1}, we obtain

$$u_\alpha^{n+1} = (A_{\alpha\gamma} + \eta\Delta t\, K_{\alpha\gamma})^{-1}\{[A_{\gamma\beta} - (1-\eta)\Delta t\, K_{\gamma\beta}]u_\beta^n + \Delta t\, H_\gamma\} \tag{10.2.23}$$

The amplification matrix becomes

$$g_{\alpha\beta} = E_{\alpha\gamma}^{-1} D_{\gamma\beta}$$

with

$$E_{\alpha\gamma} = A_{\alpha\gamma} + \eta\Delta t\, K_{\alpha\gamma}$$
$$D_{\gamma\beta} = A_{\gamma\beta} - (1-\eta)\Delta t\, K_{\gamma\beta}$$

For all values of Δt, it is seen that we have $g_{\alpha\beta} \leq \delta_{\alpha\beta}$, and the implicit scheme is *unconditionally stable*.

To study the stability behavior of (10.2.23) let us examine one-dimensional linear finite element approximation of (10.2.23) with three nodes,

$$\frac{1}{6}\left(\Delta u_{j-1}^{n+1} + 4\Delta u_j^{n+1} + \Delta u_{j+1}^{n+1}\right) + \eta D\left(-u_{j-1}^{n+1} + 2u_j^{n+1} - u_{j+1}^{n+1}\right)$$
$$= -D\left(-u_{j-1}^n + 2u_j^n - u_{j+1}^n\right) \tag{10.2.24}$$

with $\Delta u_j^{n+1} = u_j^{n+1} - u_j^n$, $h = \Delta x$, and D being the nondimensional convergence parameter.

$$D = \nu\frac{\Delta t}{\Delta x^2}$$

The combined spatial and temporal response of the amplitude u^n may be written as

$$u_j^n = e^{ikx}e^{\omega t} = e^{ikj\Delta x}e^{ckn\Delta t} = e^{ikj\Delta x}g^n \tag{10.2.25}$$

where $g = e^{ck\Delta t}$ is the amplification factor, with k and c being the wave number and wave velocity, respectively. Thus,

$$\Delta u_j^{n+1} = e^{ikj\Delta x}(g-1)g^n \tag{10.2.26}$$

Substituting (10.2.25) and (10.2.26) into (10.2.24) leads to

$$e^{ikj\Delta x}g^n\left\{(g-1)\left[\frac{1}{6}(e^{-i\theta}+4+e^{i\theta}) + \eta D(-e^{-i\theta}+2-e^{i\theta})\right] + D(-e^{-i\theta}+2-e^{i\theta})\right\} = 0$$

with

$$\theta = k\Delta x$$

or

$$g = 1 + \frac{2D\sin^2\left(\frac{\theta}{2}\right)}{-\frac{1}{3} - \frac{1}{6}\cos\theta + \eta D(\cos\theta - 1)}$$

For $\theta \to 0$, the amplification factor takes the form

$$g = 1 - D\theta^2$$

It is seen that stability is maintained for $g < 1$ or

$$D\theta^2 > 0$$

which shows that the stability is proportional to the square of the phase angle.

10.2.2 HYPERBOLIC EQUATIONS

Consider the hyperbolic equation in the form

$$R = \frac{\partial^2 u}{\partial t^2} - u_{,ii} - f(x, y) = 0 \tag{10.2.27}$$

in which the time dependent term is of the second order. Proceeding in a manner similar to the parabolic equation, we write the DST/GGM equations as

$$(\hat{W}(\xi), (\Phi_\alpha, R)) = \int \hat{W}(\xi)(A_{\alpha\beta}\ddot{u}_\beta + K_{\alpha\beta}u_\beta - H_\alpha)d\xi = 0 \tag{10.2.28}$$

In order to handle the second order derivative of u with respect to time, we must provide at least quadratic trial functions for u_α,

$$u_\alpha = \hat{\Phi}_m u_\alpha^m \quad (m = 1, 2, 3)$$

Here, $\hat{\Phi}_m$ may be defined in $0 < \xi < 1$ or $-1 < \xi < 1$ as follows:

For $0 < \xi < 1$	For $-1 < \xi < 1$
$\hat{\Phi}_1 = 2\left(\xi - \frac{1}{2}\right)(\xi - 1)$	$\hat{\Phi}_1 = \frac{1}{2}\xi(\xi - 1)$
$\hat{\Phi}_2 = -4\xi(\xi - 1)$	$\hat{\Phi}_2 = 1 - \xi^2$
$\hat{\Phi}_3 = 2\xi\left(\xi - \frac{1}{2}\right)$	$\hat{\Phi}_3 = \frac{1}{2}\xi(\xi + 1)$

Using the interval $-1 < \xi < 1$, since this interval is more convenient for integration, we obtain

$$\ddot{u}_\alpha = \frac{\partial}{\partial t}\dot{u}_\alpha = \frac{\partial \dot{u}_\alpha}{\partial \xi}\frac{\partial \xi}{\partial t} = \frac{\partial}{\partial \xi}\frac{\partial u_\alpha}{\partial \xi}\left(\frac{\partial \xi}{\partial t}\right)^2 = \frac{1}{\Delta t^2}(u_\alpha^{n-1} - 2u_\alpha^n + u_\alpha^{n+1}) \tag{10.2.29}$$

which is identical to the finite difference form for the second derivative of u_α.

Defining the temporal parameters η and ζ in the form

$$\eta = \frac{\frac{1}{2}\int_{-1}^{1} \hat{W}\xi(1 + \xi)d\xi}{\int_{-1}^{1} \hat{W}d\xi}, \qquad \zeta = \frac{\int_{-1}^{1} \hat{W}\left(\xi + \frac{1}{2}\right)d\xi}{\int_{-1}^{1} \hat{W}d\xi} \tag{10.2.30}$$

the recursive finite element equation takes the form

$$\begin{aligned}(A_{\alpha\beta} + \eta\Delta t^2 K_{\alpha\beta})u_\beta^{n+1} = &\left[2A_{\alpha\beta} - \left(\frac{1}{2} - 2\eta + \zeta\right)\Delta t^2 K_{\alpha\beta}\right]u_\beta^n \\ &- \left[A_{\alpha\beta} + \left(\frac{1}{2} + \eta - \zeta\right)\Delta t^2 K_{\alpha\beta}\right]u_\beta^{n-1} + \Delta t^2 H_\alpha\end{aligned} \tag{10.2.31}$$

Table 10.2.2 Temporal Parameters for Hyperbolic Equations

$\hat{W}(\xi)$	η	ζ
$\delta(\xi+1)$	0	1/2
$\delta(\xi-0)$	0	1/2
$\delta(\xi-1)$	1	3/2
$1, 0 \le \xi \le 1$	1/6	1/2
$1+\xi, -1 \le \xi \le 0$	4/5	3/2
$1-\xi, -1 \le \xi \le 0$	1/12	1/2
$-\xi, 0 \le \xi \le 1$	1/4	1/2
ξ	1/4	1/2
$1-\xi^2$	1/10	1/2
$(1/2)\xi(1+\xi)$	4/5	3/2

Once again, $\eta = 0$ and $\eta = 1$ lead to the explicit and implicit schemes, respectively. Various values for $\hat{W}$, and the corresponding temporal parameters η and ζ, are presented in Table 10.2.2.

For highly oscillatory motions, quadratic approximations may be inadequate and cubic approximations are required for acceptable accuracy. Cubic variations can be formulated using the Lagrange polynomials for $-1 \le \xi \le 1$ so that u_α and $\ddot{u}_\alpha$ take the forms

$$u_\alpha = -\frac{9}{16}\left(\xi+\frac{1}{3}\right)\left(\xi-\frac{1}{3}\right)(\xi-1)u_\alpha^{n-2} + \frac{27}{16}(\xi+1)\left(\xi-\frac{1}{3}\right)(\xi-1)u_\alpha^{n-1} - \frac{27}{16}(\xi+1)\left(\xi+\frac{1}{3}\right)(\xi-1)u_\alpha^{n} + \frac{9}{16}(\xi+1)\left(\xi+\frac{1}{3}\right)\left(\xi-\frac{1}{3}\right)u_\alpha^{n+1}$$

and

$$\ddot{u}_\alpha = \frac{1}{\Delta t^2}\left[-\frac{9}{16}(6\xi-2)u_\alpha^{n-2} + \frac{27}{16}\left(6\xi-\frac{2}{3}\right)u_\alpha^{n-1} - \frac{27}{16}\left(6\xi+\frac{2}{3}\right)u_\alpha^{n} + \frac{9}{16}(6\xi+2)u_\alpha^{n+1}\right]$$

Substituting the above into (10.2.28), we arrive at

$$\begin{aligned}
&\left[A_{\alpha\beta}\frac{9}{16}(6\gamma+2) + \Delta t^2\frac{9}{16}\left(\eta+\zeta-\frac{1}{9}\gamma-\frac{1}{9}\right)K_{\alpha\beta}\right]u_\beta^{n+1} \\
&+\left[A_{\alpha\beta}\frac{27}{16}\left(-6\gamma-\frac{2}{3}\right) + \Delta t^2\frac{27}{16}\left(-\eta-\frac{1}{3}\zeta+\gamma+\frac{1}{3}\right)K_{\alpha\beta}\right]u_\beta^{n} \\
&+\left[A_{\alpha\beta}\frac{27}{16}\left(6\gamma-\frac{2}{3}\right) + \Delta t^2\frac{27}{16}\left(\eta-\frac{1}{3}\zeta-\gamma+\frac{1}{3}\right)K_{\alpha\beta}\right]u_\beta^{n-1} \\
&+\left[A_{\alpha\beta}\frac{9}{16}(-6\gamma+2) + \Delta t^2\frac{9}{16}\left(-\eta+\zeta+\frac{1}{9}\gamma-\frac{1}{9}\right)K_{\alpha\beta}\right]u_\beta^{n-2} \\
&-\Delta t^2(F_\alpha+G_\alpha) = 0
\end{aligned} \tag{10.2.32}$$

with

$$\eta = \frac{\int_{-1}^{1} \hat{W}(\xi)\xi^3 d\xi}{\int_{-1}^{1} \hat{W} d\xi}, \qquad \zeta = \frac{\int_{-1}^{1} \hat{W}(\xi)\xi^2 d\xi}{\int_{-1}^{1} \hat{W} d\xi}, \qquad \gamma = \frac{\int_{-1}^{1} \hat{W}(\xi)\xi d\xi}{\int_{-1}^{1} \hat{W} d\xi}$$

Appropriate choices of $\hat{W}(\xi)$ will lead to a variety of integration formulas.

Using the Newton backward difference (Chung, 1975), it can be shown that the cubic approximations may also be given as

$$[11A_{\alpha\beta} + 6\Delta t(1-\theta)K_{\alpha\beta}]v_\beta^{n+1} + [-18A_{\alpha\beta} + 6\Delta t\theta K_{\alpha\beta}]v_\beta^n + A_{\alpha\beta}(9v_\beta^{n-1} - 2v_\beta^{n-2}) - 6\Delta t H_\beta = 0 \tag{10.2.33}$$

where $0 \le \theta \le 1$.

10.2.3 MULTIVARIABLE PROBLEMS

The finite element formulation of multivariable problems which occur in two- or three-dimensional problems may be best handled using tensors. Let us consider a differential equation of the form

$$\frac{\partial \mathbf{v}}{\partial t} - \nabla^2 \mathbf{v} - \nabla(\nabla \cdot \mathbf{v}) - \mathbf{f} = 0 \tag{10.2.34a}$$

or

$$R_i = \frac{\partial v_i}{\partial t} - v_{i,jj} - v_{j,ji} - f_i = 0 \tag{10.2.34b}$$

where the variables v_i may be approximated spatially as

$$v_i = \Phi_\alpha v_{\alpha i} \quad (i = 1, 2) \text{ for 2-D} \tag{10.2.35}$$

Note that $v_{\alpha i}$ implies v_i at the global node α. The GGM equations for (10.2.34b) become

$$(\hat{W}(\xi), (\Phi_\alpha, R_i)) = \int_\xi \hat{W}(\xi)\left[\int_\Omega \Phi_\alpha\left(\frac{\partial v_i}{\partial t} - v_{i,jj} - v_{j,ji} - f_i\right)d\Omega\right]d\xi = 0 \tag{10.2.36}$$

which yields

$$\int_\xi \hat{W}(\xi)\left[A_{\alpha\beta}\delta_{ik}\dot{v}_{\beta k} + \left(K^{(1)}_{\alpha i\beta k} + K^{(2)}_{\alpha j\beta j}\delta_{ik}\right)v_{\beta k} - F_{\alpha i} - G_{\alpha i}\right]d\xi = 0$$

where

$$A_{\alpha\beta} = \int_\Omega \Phi_\alpha \Phi_\beta d\Omega = \bigcup_{e=1}^{E} \int_\Omega \Phi_N^{(e)} \Phi_M^{(e)} d\Omega \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} = \bigcup_{e=1}^{E} A_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)}$$

$$K^{(1)}_{\alpha i\beta k} = \int_\Omega \Phi_{\alpha,i} \Phi_{\beta,k} d\Omega = \bigcup_{e=1}^{E} \int_\Omega \Phi_{N,i}^{(e)} \Phi_{M,k}^{(e)} d\Omega \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} = \bigcup_{e=1}^{E} K_{Ni\,Mk}^{(1)(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)}$$

$$K^{(2)}_{\alpha j\beta j} = \int_\Omega \Phi_{\alpha,j} \Phi_{\beta,j} d\Omega = \bigcup_{e=1}^{E} \int_\Omega \Phi_{N,j}^{(e)} \Phi_{M,j}^{(e)} d\Omega \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} = \bigcup_{e=1}^{E} K_{Nj\,Mj}^{(2)(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)}$$

$$F_{\alpha i} = \int_\Omega \Phi_\alpha \Phi_\beta d\Omega \delta_{ik} f_{\beta k} = C_{\alpha\beta}\delta_{ik} f_{\beta k} = \bigcup_{e=1}^{E} C_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} \delta_{ik} f_{\beta k}$$

$$C_{NM}^{(e)} = \int_\Omega \Phi_N^{(e)} \Phi_M^{(e)} d\Omega$$

$$G_{\alpha i} = \int_\Gamma \overset{*}{\Phi}(\mathrm{v}_{i,j} n_j + \mathrm{v}_{j,j} n_i) d\Gamma = \bigcup_{e=1}^{E} G_{Ni}^{(e)} \Delta_{N\alpha}^{(e)}$$

For the case of Figure E10.1.2, we have

$$G_{\alpha i}^{(e)} = \bigcup_{e=1}^{E} \int_\Gamma \overset{*}{\Phi}_N^{(e)} \overset{*}{\Phi}_M^{(e)} d\Gamma \delta_{ik} g_{Mk}^{(e)} \Delta_{N\alpha}^{(e)} = \bigcup_{e=1}^{E} \overset{*}{C}_{NM}^{(e)} \delta_{ik} g_{Mk}^{(e)} \Delta_{N\alpha}^{(e)}$$

$$= \frac{l}{6} \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix} \begin{bmatrix} g_{11}^{(e)} \\ g_{12}^{(e)} \\ g_{21}^{(e)} \\ g_{22}^{(e)} \end{bmatrix} = \frac{l}{6} \begin{bmatrix} 2g_{11}^{(e)} + g_{21}^{(e)} \\ 2g_{12}^{(e)} + g_{22}^{(e)} \\ g_{11}^{(e)} + 2g_{21}^{(e)} \\ g_{12}^{(e)} + 2g_{22}^{(e)} \end{bmatrix}$$

where

$$g_{M1}^{(e)} = (2\mathrm{v}_{1,1} + \mathrm{v}_{2,2}) n_1 + \mathrm{v}_{1,2} n_2$$

$$g_{M2}^{(e)} = \mathrm{v}_{2,1} n_1 + (\mathrm{v}_{1,1} + 2\mathrm{v}_{2,2}) n_2$$

With linear temporal approximations, the global finite element equations take the form

$$\left[A_{\alpha\beta}\delta_{ik} + \eta\Delta t\left(K_{\alpha i\beta k}^{(1)} + K_{\alpha j\beta j}^{(2)}\delta_{ik}\right)\right]\mathrm{v}_{\beta k}^{n+1} = \left[A_{\alpha\beta}\delta_{ik} - (1-\eta)\Delta t\left(K_{\alpha i\beta k}^{(1)} + K_{\alpha j\beta j}^{(2)}\delta_{ik}\right)\right]\mathrm{v}_{\beta k}^{n} + \Delta t(F_{\alpha i} + G_{\alpha i}) \tag{10.2.37}$$

The solution of (10.2.37) will proceed similarly as a single variable problem except that the multivariables $\mathrm{v}_{\beta k}$ are to be solved simultaneously.

10.2.4 AXISYMMETRIC TRANSIENT HEAT CONDUCTION

Consider the transient heat conduction, without convection, in an axisymmetric geometry,

$$\rho c_p \frac{\partial T}{\partial t} - k\left(\frac{\partial^2 T}{\partial r^2} + \frac{\partial^2 T}{\partial z^2} + \frac{1}{r}\frac{\partial T}{\partial r}\right) = 0 \tag{10.2.38}$$

where ρ, c_p, T, k, and r are the density, specific heat at constant pressure, temperature, coefficient of thermal conductivity, and radius of a cylindrical geometry, respectively.

The generalized Galerkin finite element formulation of (10.2.38) leads to

$$\int_0^1 \hat{W}(\xi)\left\{\int_0^{2\pi}\iint \Phi_\alpha\left[\rho c_p \frac{\partial T}{\partial t} - k\left(\frac{\partial^2 T}{\partial r^2} + \frac{\partial^2 T}{\partial z^2} + \frac{1}{r}\frac{\partial T}{\partial r}\right)\right] r d\theta dr dz\right\} d\xi = 0 \tag{10.2.39}$$

Here, the partial integration of the term containing $\partial^2 T/\partial r^2$ in (10.2.39) becomes

$$\int_0^{2\pi}\iint \Phi_\alpha \frac{\partial^2 T}{\partial r^2} r\,d\theta\,dr\,dz = 2\pi\left(\int \overset{*}{\Phi}_\alpha \frac{\partial T}{\partial r} r\,dz - \iint \frac{\partial \Phi_\alpha}{\partial r}\frac{\partial T}{\partial r} r\,dr\,dz - \iint \Phi_\alpha \frac{\partial T}{\partial r} dr\,dz\right)$$

Thus, after canceling out the $\partial T/\partial r$ terms, we have

$$\int_0^1 \hat{W}(\xi)(A_{\alpha\beta}\dot{T}_\beta + K_{\alpha\beta}T_\beta - G_\alpha)d\xi = 0 \tag{10.2.40}$$

where, for isoparametric quadrilateral elements, with $r = \Phi_\gamma r_\gamma$, we obtain

$$A_{\alpha\beta} = 2\pi\int_{-1}^1\int_{-1}^1 \rho c_p \Phi_\alpha \Phi_\beta \Phi_\gamma r_\gamma |J|\,d\xi d\eta$$

Here, $d\xi$ refers to the isoparametric coordinates rather than the nondimensional time,

$$K_{\alpha\beta} = 2\pi\left\{\int_{-1}^1\int_{-1}^1 k\left(\frac{\partial \Phi_\alpha}{\partial r}\frac{\partial \Phi_\beta}{\partial r} + \frac{\partial \Phi_\alpha}{\partial z}\frac{\partial \Phi_\beta}{\partial z}\right)\Phi_\gamma r_\gamma |J| d\xi d\eta\right\}$$

$$G_\alpha = 2\pi\int_\Gamma \overset{*}{\Phi}_\alpha k T_{,i} n_i r\,d\Gamma = 2\pi\int_\Gamma -\overset{*}{\Phi}_\alpha \overline{\alpha}(T - T')r\,d\Gamma$$

$$= 2\pi\left[\int_\Gamma -\overline{\alpha}\,\overset{*}{\Phi}_\alpha\,\overset{*}{\Phi}_\beta\, r\,d\Gamma\, T_\beta + \int_\Gamma \overline{\alpha}\,\overset{*}{\Phi}_\alpha T' r\,d\Gamma\right] = \overset{*}{K}_{\alpha\beta}T_\beta + \overset{*}{G}_\alpha$$

where we set

$$-kT_{,i}n_i = \overline{\alpha}(T - T')$$

with $\overline{\alpha}$ and T' being defined as the heat transfer coefficient and ambient temperature, respectively. Here, $\overset{*}{K}_{\alpha\beta}$ is the convection boundary stiffness matrix representing the contribution of ambient temperature toward the boundary surface:

$$\overset{*}{K}_{\alpha\beta} = 2\pi\int_\Gamma \overset{*}{\Phi}_\alpha\,\overset{*}{\Phi}_\beta\,\overset{*}{\Phi}_\gamma\, r_\gamma\, d\Gamma$$

$$\overset{*}{G}_\alpha = 2\pi\int_\Gamma T'\overline{\alpha}\,\overset{*}{\Phi}_\alpha\,\overset{*}{\Phi}_\gamma\, r_\gamma\, d\Gamma$$

where $\overset{*}{K}_{\alpha\beta}$ should be combined with $K_{\alpha\beta}$ but its contribution is restricted only to the convection boundary nodes along the surface of convection boundaries as shown in (10.1.23). Thus,

$$\int_0^1 \hat{W}(\xi)(A_{\alpha\beta}\dot{T}_\beta + (K_{\alpha\beta} + \overset{*}{K}_{\alpha\beta})T_\beta - \overset{*}{G}_\alpha)d\xi = 0 \tag{10.2.41}$$

This ordinary differential equation will then be integrated over the temporal domain as in Section 10.2.1.

10.3 SOLUTIONS OF FINITE ELEMENT EQUATIONS

Solutions of simultaneous algebraic equations are carried out by using either direct or iterative methods. The direct methods yield answers in a finite number of operations (Section 4.2.7). They include Gauss elimination, Thomas algorithm, etc., which are suitable for linear equations. The iterative methods [Saad, 1996] include Gauss-Seidel methods, relaxation methods, conjugate gradient methods (CGM), and generalized minimal residual (GMRES) methods, among others. Here, solutions are obtained through a number of iterative steps, accuracy being increased with an increase of iterations. These methods are suitable for nonlinear as well as linear equations.

For a large system of equations, it is expected that the assembly of element stiffness matrices into a global form would take a prohibitive amount of computer time. This can be avoided by the so-called element-by-element (EBE) solution scheme [Hughes, Levit, and Winget, 1983; Carey and Jiang, 1986; Wathen, 1989, etc.]. In this approach, we replace the matrix assembly process by vector operations. This will be presented in Section 10.3.2.

The coverage of solution methods for algebraic equations in general is beyond the scope of this book. However, we select the conjugate gradient method (CGM) as one of the most popular schemes in CFD and present its brief description, followed by the EBE approach for finite element equations.

10.3.1 CONJUGATE GRADIENT METHODS (CGM)

Let us consider the global system of finite element equations in the form

$$K_{\alpha\beta} U_\beta = F_\alpha \tag{10.3.1}$$

The iterative solution by the conjugate gradient methods (CGM) can be obtained, using the following steps:

(1) Assume initial values $U_\alpha^{(r)}$

(2) Determine the residual $E_\alpha^{(r)}$

$$E_\alpha^{(r)} = F_\alpha - K_{\alpha\beta} U_\beta^{(r)} \tag{10.3.2}$$

(3) Define the auxiliary variables $P_\alpha^{(r)}$

$$P_\alpha^{(r)} = E_\alpha^{(r)}$$

(4) Compute rth iteration residual

$$\overline{E}_\alpha^{(r)} = K_{\alpha\beta} P_\beta^{(r)} \tag{10.3.3}$$

(5) Compute the correction factor $a^{(r)}$

$$a^{(r)} = \frac{E_\alpha^{(r)} P_\alpha^{(r)}}{\overline{E}_\beta^{(r)} P_\beta^{(r)}} \tag{10.3.4}$$

(6) Compute the solution $U_\alpha^{(r+1)}$

$$U_\alpha^{(r+1)} = U_\alpha^{(r)} + a^{(r)} P_\alpha^{(r)} \tag{10.3.5}$$

(7) Compute the residual $E_\alpha^{(r+1)}$

$$E_\alpha^{(r+1)} = E_\alpha^{(r)} - a^{(r)}\overline{E}_\alpha^{(r)} \tag{10.3.6}$$

(8) Compute the correction factor $b^{(r+1)}$

$$b^{(r+1)} = \frac{E_\alpha^{(r+1)} E_\alpha^{(r+1)}}{E_\beta^{(r)} E_\beta^{(r)}} \tag{10.3.7}$$

(9) Define the auxiliary variables $P_\alpha^{(r+1)}$

$$P_\alpha^{(r+1)} = E_\alpha^{(r+1)} + b^{(r+1)} P_\alpha^{(r)} \tag{10.3.8}$$

(10) Return to Step 4 and repeat the process until convergence.

If the matrix $K_{\alpha\beta}$ is nonsymmetric, then it is possible to symmetrize $K_{\alpha\beta}$ by multiplying the transpose of the stiffness matrix in (10.3.1) as follows:

$$[K]^T[K][U] = [K]^T[F]$$

or

$$K_{\gamma\alpha}K_{\gamma\beta}U_\beta = K_{\gamma\alpha}F_\gamma$$

This can be written in the form

$$A_{\alpha\beta}U_\beta = \overline{F}_\alpha \tag{10.3.9}$$

with

$$A_{\alpha\beta} = K_{\gamma\alpha}K_{\gamma\beta}, \qquad \overline{F}_\alpha = K_{\gamma\alpha}F_\gamma$$

The same procedure as given in Steps 1 through 10 above can be applied to (10.3.9). However, this will require extremely large operations and we may avoid them by constructing the product of the transpose of the stiffness matrix and the auxiliary variables as follows:

(1) Start with the initial guess $U_\alpha^{(o)}$

(2) $E_\alpha^{(o)} = K_{\gamma\alpha}(F_\gamma - K_{\gamma\beta}U_\beta)$

(3) $P_\alpha^{(r)} = E_\alpha^{(r)}$

(4) $\overline{E}_\alpha^{(r)} = K_{\gamma\alpha}K_{\gamma\beta}P_\beta^{(r)}$

(5) $a^{(r)} = \dfrac{E_\alpha^{(r)} P_\alpha^{(r)}}{\overline{E}_\beta^{(r)} P_\beta^{(r)}}$

(6) $U_\alpha^{(r+1)} = U_\alpha^{(r)} + a^{(r)} P_\alpha^{(r)}$

(7) $E_\alpha^{(r+1)} = E_\alpha^{(r)} - a^{(r)}\overline{E}_\alpha^{(r)}$

(8) $b^{(r+1)} = \dfrac{E_\alpha^{(r+1)} E_\alpha^{(r+1)}}{E_\beta^{(r)} E_\beta^{(r)}}$

(9) $P_\alpha^{(r+1)} = E_\alpha^{(r+1)} + b^{(r)} P_\alpha^{(r)}$

(10) Return to step (4) and repeat the process until convergence.

Example 10.3.1

Given: Consider a system of algebraic equations of the form,

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -2 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}$$

Required: Solve using the CGM algorithm and compare with the exact solution: $U_1 = 1, U_2 = 1, U_3 = 1$.

Solution:

(1) Assume $U_\alpha^{(o)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

(2) $E_\alpha^{(o)} = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -2 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}$

(3) $P_\alpha^{(o)} = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}$

(4) $\overline{E}_\alpha^{(o)} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -2 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

(5) $a^{(o)} = \dfrac{0+1+1}{0+0-1} = -2$

(6) $U_\alpha^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + (-2) \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix}$

(7) $E_\alpha^{(1)} = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix} - (-2) \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$

(8) $b^{(1)} = \dfrac{4+1+1}{0+1+1} = 3$

(9) $P_\alpha^{(1)} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} + (3) \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \\ -2 \end{bmatrix}$

(10) $\overline{E}_\alpha^{(1)} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -2 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -4 \\ -2 \end{bmatrix} = \begin{bmatrix} 6 \\ -6 \\ 6 \end{bmatrix}$

(11) $a^{(1)} = \dfrac{4+4-2}{12+24-12} = \dfrac{6}{24} = 0.25$

(12) $U_\alpha^{(2)} = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix} + (0.25) \begin{bmatrix} 2 \\ -4 \\ -2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \\ 1.5 \end{bmatrix}$

Repeating another cycle of iteration, we obtain

$$U_\alpha^{(3)} = \begin{bmatrix} 1.0002 \\ 1 \\ 0.9998 \end{bmatrix}$$

The next step (7) shows the residual $E_\alpha^{(3)}$ to be zero and the exact answers, $U_1 = U_2 = U_3 = 1$, are obtained.

If the stiffness matrix $K_{\alpha\beta}$ is nonsymmetric or nonlinear, then the procedure for (10.3.9) can be used. It is expected that convergence toward the exact solution will be much slower. The GMRES methods suitable for CFD equations will be covered in Section 11.5.2.

10.3.2 ELEMENT-BY-ELEMENT (EBE) SOLUTIONS OF FEM EQUATIONS

A large system of equations is encountered when the number of finite element nodes increases in order to improve accuracy. The assembly of element stiffness matrices into a global form and solutions may occupy a large portion of computing time. To avoid this inconvenience, we shall examine the so-called element-by-element (EBE) approach [Hughes et al., 1983; Carey and Jiang, 1986; Wathen, 1989, etc.], in which the assembly of entire stiffness matrices is eliminated. The EBE methods using the Jacobi-iteration and conjugate gradient methods are described below.

Let us consider the global finite element equations of the form,

$$K_{\alpha\beta} U_\beta = F_\alpha \tag{10.3.10}$$

The global stiffness matrix $K_{\alpha\beta}$ can be split into the diagonal components $D_{\alpha\beta}$ and the off-diagonal matrix $N_{\alpha\beta}$ as follows:

$$K_{\alpha\beta} = D_{\alpha\beta} + N_{\alpha\beta} \tag{10.3.11}$$

leading to

$$(D_{\alpha\beta} + N_{\alpha\beta}) U_\beta = F_\alpha \tag{10.3.12}$$

or

$$D_{\alpha\beta} U_\beta^{(r+1)} \cong F_\alpha^{(r)} - N_{\alpha\beta} U_\beta^{(r)} \tag{10.3.13}$$

where the diagonal matrix and off-diagonal matrix are allowed to be associated with the iteration steps of U_β at $(r+1)$ and (r), respectively. Subtracting $D_{\alpha\beta} U_\beta^{(r)}$ from the left- and right-hand sides of (10.3.13), we obtain

$$D_{\alpha\beta}\left(U_\beta^{(r+1)} - U_\beta^{(r)}\right) = F_\alpha^{(r)} - (N_{\alpha\beta} + D_{\alpha\beta}) U_\beta^{(r)} \tag{10.3.14}$$

or

$$U_\alpha^{(r+1)} = U_\alpha^{(r)} - D_{\alpha\beta}^{-1}\left(\overline{F}_\beta^{(r)} - F_\beta^{(r)}\right) \tag{10.3.15}$$

with the diagonal matrix playing the role of the preconditioning matrix and

$$\overline{F}_{\alpha}^{(r)} = (N_{\alpha\beta} + D_{\alpha\beta})U_{\beta}^{(r)} = K_{\alpha\beta}U_{\beta}^{(r)} = \bigcup_{e=1}^{E} \overline{F}_{N}^{(e)} \Delta_{N\alpha}^{(e)}$$

$$\overline{F}_{N}^{(e)} = K_{NM}^{(e)} U_{M}^{(e)} \tag{10.3.16}$$

It is clearly seen that the assembly of the stiffness matrix has been replaced by the element-by-element basis as a column vector, identical to the assembly of the source vector $F_{\alpha}^{(r)}$ such as in (10.1.15b). Thus, the solution of (10.3.10) is obtained as

$$\begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ \vdots \end{bmatrix}^{(r+1)} = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ \vdots \end{bmatrix}^{(r)} - \begin{bmatrix} (\overline{F}_1 - F_1)/D_{11} \\ (\overline{F}_2 - F_2)/D_{22} \\ \vdots \\ \vdots \end{bmatrix}^{(r)} \tag{10.3.17}$$

In order to increase convergence and accuracy, it is necessary to implement a standard relaxation process in the form

$$U = \xi U^{(r+1)} + (1 - \xi)U^{(r)}$$

with $0 < \xi < 1$ or preferably $\xi = 0.8$. The procedure described above resembles the Jacobi iteration method and, thus, this scheme is called the EBE Jacobi method [Hughes et al., 1983].

The EBE scheme may be incorporated into any high-accuracy iterative equation solver. For example, let us consider the conjugate gradient method. Here, we may adopt the following step-by-step procedure.

(1) Assume initial values $U_{\alpha}^{(r)}$.

(2) Compute the residual $E_{\alpha}^{(r)}$

$$E_{\alpha}^{(r)} = F_{\alpha} - K_{\alpha\beta}U_{\beta}^{r} = F_{\alpha} - \overline{F}_{\alpha} \tag{10.3.18}$$

with

$$\overline{F}_{\alpha} = \bigcup_{e=1}^{E} \overline{F}_{N}^{(e)} \Delta_{N\alpha}^{(e)}$$

$$\overline{F}_{N}^{(e)} = K_{NM}^{(e)} U_{M}^{(e)}$$

(3) Set the residual as the auxiliary variables $P_{\alpha}^{(r)}$

$$P_{\alpha}^{(r)} = E_{\alpha}^{(r)} \tag{10.3.19}$$

(4) Determine the rth iteration residual $E_{\alpha}^{(r)}$ as

$$\overline{E}_{\alpha}^{(r)} = K_{\alpha\beta}P_{\beta}^{(r)} = \bigcup_{e=1}^{E} H_{N}^{(e)} \Delta_{N\alpha}^{(e)} \tag{10.3.20}$$

with

$$H_{N}^{(e)} = K_{NM}^{(e)} P_{M}^{(r)}$$

(5) Determine the correction factor $a^{(r)}$

$$a^{(r)} = \frac{E_\alpha^{(r)} P_\alpha^{(r)}}{\overline{E}_\beta^{(r)} P_\beta^{(r)}} \tag{10.3.21}$$

(6) Solve $U_\alpha^{(r+1)}$

$$U_\alpha^{(r+1)} = U_\alpha^{(r)} + a^{(r)} P_\alpha^{(r)} \tag{10.3.22}$$

(7) Determine the residual $E_\alpha^{(r+1)}$

$$E_\alpha^{(r+1)} = E_\alpha^{(r)} - a^{(r)} \overline{E}_\alpha^{(r)} \tag{10.3.23}$$

(8) Compute the correction factor $b^{(r+1)}$

$$b^{(r+1)} = \frac{E_\alpha^{(r+1)} E_\alpha^{(r+1)}}{E_\beta^{(r)} E_\beta^{(r)}} \tag{10.3.24}$$

(9) Determine the auxiliary variables $P_\alpha^{(r+1)}$

$$P_\alpha^{(r+1)} = E_\alpha^{(r+1)} + b^{(r+1)} P_\alpha^{(r)} \tag{10.3.25}$$

(10) Return to (4) and repeat until convergence.

For time-dependent and nonlinear problems, procedures similar to those above can be used. In order to expedite the convergence, however, appropriate preconditioning processes are important. These and other topics on the equation solvers such as GMRES and the EBE algorithms will be presented in Section 11.5.

10.4 EXAMPLE PROBLEMS

10.4.1 SOLUTION OF POISSON EQUATION WITH ISOPARAMETRIC ELEMENTS

In this example, we repeat Example 10.1.2 using 6 and 24 bilinear (4 node) isoparametric elements by removing the diagonals (Figure 10.4.1.1). Use the three-point Gaussian

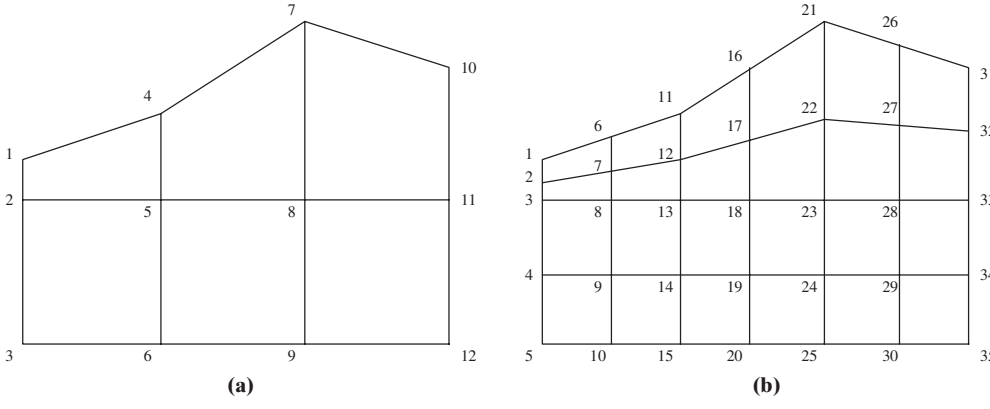


Figure 10.4.1.1 Meshes for Example 10.4.1.1. **(a)** Six bilinear isoparametric element system. **(b)** Twenty-four bilinear isoparametric element system.

quadrature integration. The solution procedure is as follows:

$$K_{\alpha\beta} = \bigcup_{e=1}^{E} K_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)}$$

$$K_{NM}^{(e)} = \int_{\Omega} \Phi_{N,i}^{(e)} \Phi_{M,i}^{(e)} d\Omega = \sum_{p=1}^{n} \sum_{q=1}^{n} w_p w_q k_{NM}(\xi_p, \eta_q)$$

$$F_{\alpha} = C_{\alpha\beta} f_{\beta} = \bigcup_{e=1}^{E} C_{NM}^{(e)} \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} f_{\beta} = \bigcup_{e=1}^{E} \left[\sum_{p=1}^{n} \sum_{q=1}^{n} w_p w_q c_{NM}(\xi_p, \eta_q) \right] \Delta_{N\alpha}^{(e)} \Delta_{M\beta}^{(e)} f_{\beta}$$

It is obvious that no local evaluation of the load vector is necessary and it is convenient to leave $f_{\beta} = [4(x^2 + y^2)]_{\beta}$ in the global form, unlike the Neumann boundary vector which was evaluated in the local level and assembled into a global form.

The Neumann boundary vector remains the same as in the case of triangular elements, and is independent of the Gaussian quadrature integration. If desired, however, the Neumann boundary vector may be rederived from the one-dimensional isoparametric (natural) coordinate. The results would be the same.

The Neumann boundary vector G_{α} for the six-element problem is the same as in Example 10.1.2, although the load vector F_{α} is different due to the different integration scheme. The summary of results is given in Table E10.4.1.1.

The following conclusions are drawn from Examples 10.1.2 and 10.1.3.

(1) The six isoparametric elements provide higher accuracy than twelve triangular elements. At interior nodes (5 and 8), triangular elements give answers smaller than the exact solutions, whereas the isoparametric elements lead to larger values, indicating that triangular elements are stiffer than the isoparametric elements as seen in Examples 10.1.2 and 10.1.3.
(2) In the coarse grid system, the Neumann problem is not as accurate as in the Dirichlet problem.

10.4.2 PARABOLIC PARTIAL DIFFERENTIAL EQUATION IN TWO DIMENSIONS

Consider the two-dimensional linear partial differential equation of the form

$$\frac{\partial u}{\partial t} - \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - f_x = 0$$

$$\frac{\partial \mathrm{v}}{\partial t} - \nu \left(\frac{\partial^2 \mathrm{v}}{\partial x^2} + \frac{\partial^2 \mathrm{v}}{\partial y^2} \right) - f_y = 0$$

with

$$f_x = -\frac{1}{(1+t)^2} - 2\nu y, \qquad f_y = -\frac{1}{(1+t)^2} - 2\nu x$$

Table E10.4.1.1 Computed Results for Example 10.4.1.1

(a) Dirichlet Data (6 elements)

Node	Exact Solution	FEM Solution	% Error
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	450.00	450.00	0.00
5	162.00	197.05	21.64
6	0.00	0.00	0.00
7	3528.00	3528.00	0.00
8	648.00	667.45	3.00
9	0.00	0.00	0.00
10	5832.00	5832.00	0.00
11	1458.00	1458.00	0.00
12	0.00	0.00	0.00

(b) Neumann Data (6 elements)

Node	Exact Solution	FEM Solution	% Error
1	0.00	−28.99	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	450.00	339.18	24.63
5	162.00	130.63	19.36
6	0.00	0.00	0.00
7	3528.00	3221.45	8.69
8	648.00	601.47	7.18
9	0.00	0.00	0.00
10	5832.00	5697.71	2.30
11	1458.00	1458.00	0.00
12	0.00	0.00	0.00

(c) Dirichlet Data (24 elements)

Node	Exact Solution	FEM Solution	% Error
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	91.13	91.13	0.00
7	63.28	65.68	3.79
8	40.50	44.09	8.86
9	10.13	12.10	19.47
10	0.00	0.00	0.00
11	450.00	450.00	0.00
12	288.00	287.79	.07
13	162.00	170.18	5.05
14	40.50	44.51	9.90
15	0.00	0.00	0.00
16	1458.00	1458.00	0.00
17	820.13	830.87	1.31
18	364.50	379.19	4.03
19	91.13	94.76	3.99
20	0.00	0.00	0.00
21	3528.00	3528.00	0.00
22	1800.00	1812.86	.71
23	648.00	648.80	.12
24	162.00	163.41	.87
25	0.00	0.00	0.00
26	4753.13	4753.13	0.00
27	2538.28	2530.26	.32
28	1012.50	1005.26	.71
29	253.13	252.50	.25
30	0.00	0.00	0.00
31	5832.00	5832.00	0.00
32	3280.50	3280.50	0.00
33	1458.00	1458.00	0.00
34	364.50	364.50	0.00
35	0.00	0.00	0.00

(d) Neumann Data (24 elements)

Node	Exact Solution	FEM Solution	% Error
1	0.00	−3.69	
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	91.12	70.60	22.52
7	63.28	49.13	22.36
8	40.50	31.82	21.43
9	10.12	6.51	35.67
10	0.00	0.00	0.00
11	450.00	409.87	8.92
12	288.00	257.40	10.63
13	162.00	148.10	8.58
14	40.50	34.16	15.65
15	0.00	0.00	0.00
16	1458.00	1392.81	4.47
17	820.12	781.97	4.65
18	364.50	349.83	4.02
19	91.12	81.24	10.84
20	0.00	0.00	0.00
21	3528.00	3381.90	4.14
22	1800.00	1746.66	2.96
23	648.00	615.65	4.99
24	162.00	150.18	7.30
25	0.00	0.00	0.00
26	4753.12	4659.78	1.96
27	2538.28	2449.15	3.51
28	1012.50	983.92	2.82
29	253.12	244.03	3.59
30	0.00	0.00	0.00
31	5832.00	5586.42	4.21
32	3280.50	3280.50	0.00
33	1458.00	1458.00	0.00
34	364.50	364.50	0.00
35	0.00	0.00	0.00

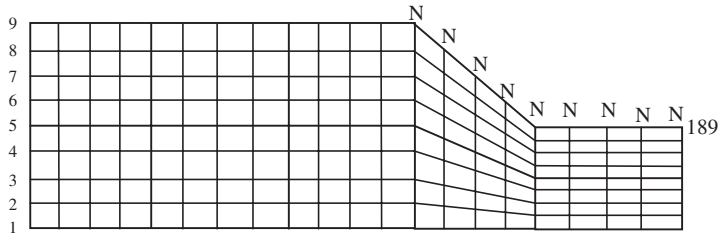


Figure 10.4.2.1 Geometry and discretization for Section 10.4.2 with N representing the Neumann boundary conditions. Dirichlet and Neumann boundary conditions are prescribed from the exact solution.

Exact Solution:

$$u = \frac{1}{1+t} + x^2 y, \quad \mathrm{v} = \frac{1}{1+t} + xy^2$$

Required: Solve the above partial differential equations using GGM for the coarse, intermediate, and fine meshes with the Dirichlet and Neumann boundary data as shown in Figure 10.4.2.1. Set $\nu = 1$, $\Delta t = 10^{-4}$, $\eta = 1/2$ Set $u = \mathrm{v} = 0$ initially at all interior nodes and observe convergence behavior.

Solution: The steady state is reached at $t \cong 0.25$ and 0.4 for u and v, respectively, at $x = 4.5$ and $y = 0.75$ to the almost exact steady-state values as shown in Figure 10.4.2.2. In Section 11.6.4, the results with nonlinear convection terms will be presented, demonstrating the solution convergence as a function of grid refinements.

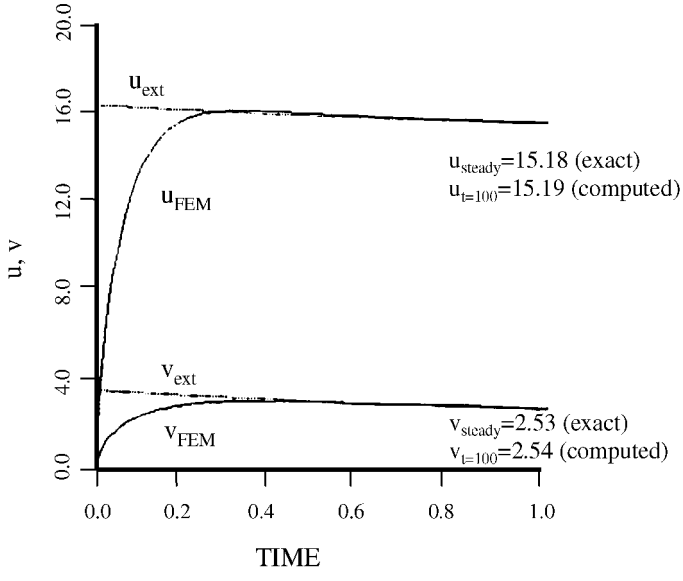


Figure 10.4.2.2 Convergence history of u and v($\nu = 1.0$, $\Delta t = 0.01$, $x = 4.5$ and $y = 0.75$).

10.5 SUMMARY

In this chapter, we have shown the basic computational procedures involved in finite element calculations for linear partial differential equations, using the standard Galerkin methods (SGM). Assembly of multidimensional finite element equations into a global form and various approaches to implementations of both Dirichlet and Neumann boundary conditions are demonstrated. Furthermore, we have described the mixed methods and penalty methods in order to satisfy the incompressibility condition involved in the Stokes flow.

In dealing with time-dependent problems, formulations with the generalized Galerkin methods (GGM) for parabolic and hyperbolic partial differential equations are presented. In particular, it was shown that temporal approximations can be provided independently and discontinuously from spatial approximations.

Solution procedures of finite element equations in general and solution approaches using element-by-element assembly techniques in particular are also elaborated. It is shown that, by means of the element-by-element (EBE) vector operations, the formulation of entire stiffness matrix array can be avoided.

Note that convective or nonlinear terms are not included in this chapter, which constitute one of the most important aspects of fluid dynamics, both physically and numerically. This is the subject of the next chapter.

REFERENCES

Babuska, I. [1973]. The finite element method with Lagrange multipliers. *Num. Math.*, 20, 179–92.

Brezzi, F. [1974]. On the existence, uniqueness and approximation of saddle point problems arising from Lagrangian multiplier, RAIRO, Ser. *Rouge Anal. Numer.*, R-2, 129–51.

Carey, G. F. and Jiang, B. [1986]. Element-by-element linear and nonlinear solution schemes. *Comm. Appl. Num. Meth.*, 2, 103–53.

Carey, G. F. and Oden, J. T. [1986]. Finite Elements, Fluid Dynamics. Englewood Cliffs, NJ: Prentice Hall.

Chung, T. J. [1975]. Convergence and stability of nonlinear finite elements. *AIAA J.*, 13, 7, 963–66.

Hughes, T. J. R., Levit, I., and Winget, J. [1983]. An element-by-element implicit algorithm for heat conduction. *ASCE J. Eng. Mech. Div.*, 74, 271–87.

Ladyszhenskaya, O. A. [1969]. *The Mathematical Theory of Viscous Incompressible Flow*. New York: Gordon and Breach.

Saad, Y. [1996]. *Iterative Methods for Sparse Systems*. Boston: PWS Publishing.

Wathen, A. J. [1989]. An analysis of some element-by-element techniques. *Comp. Meth. Appl. Mech. Eng.*, 74, 271–87.

Zienkiewicz, O. C. and Taylor, R. L. [1991]. *The Finite Element Methods*, Vol. 2. New York: McGraw-Hill.