CHAPTER EIGHT

Introduction to Finite Element Methods

8.1 GENERAL

The finite element theory as applied to one-dimensional problems was discussed in Part One, Preliminaries. In general, finite element methods (FEM) are versatile in applications to multidimensional complex irregular geometries. Initial applications of FEM began with structural analysis in the late 1950s and primarily were based on variational principles. During the early days of the development of FEM, applications were made for simple flow problems, beginning with Zienkiewicz and Cheung [1965], followed by Oden and Wellford [1972], Chung [1978], and Baker [1983], among others. Significant contributions in CFD began with the streamline upwind Petrov-Galerkin (SUPG) methods [Heinrich, Huyakorn, Zienkiewicz, and Mitchell, 1977; Hughes and Brooks, 1982; Hughes, Mallet, and Mizukami, 1986] or streamline diffusion methods (SDM) [Johnson, 1987], Taylor-Galerkin methods (TGM) [Donea, 1984; Löhner, Morgan, and Zienkiewicz, 1985], and *hp* adaptive methods [Oden and Demkowicz, 1991], among many other related works.

New approaches and various alternative methodologies are preponderant in the literature. Efforts are made in this book to simplify and unify some of the terminologies. For example, the original approaches of SUPG or SDM for convection-dominated flows have grown into GLS (Galerkin/least squares) when some changes in the formulation are introduced. It is suggested that all methods related to numerical diffusion test functions be called the generalized Petrov-Galerkin (GPG) methods. Hughes and his co-workers have contributed significantly in the past two decades to the GPG methodologies associated with the problems of convection-dominated flows and shock discontinuities.

Another example is the algorithm arising from the Taylor series expansion such as TGM. Zienkiewicz and his co-workers [Zienkiewicz and Codina, 1995] have applied for the past decade the concept of characteristic Galerkin methods (CGM) which produce results similar to TGM in dealing with convection-dominated problems for both compressible and incompressible flows.

The idea of treating discontinuities developed in the finite difference methods (FDM) flux vector splitting, TVD, and ENO associated with the first and second order upwinding can be utilized in the discontinuous Galerkin methods (DGM) as demonstrated by Oden and his co-workers [Oden, Babuska, and Baumann, 1998]. Clearly, this represents the merit of studying FDM and FEM closely together.

Recall that in FDM we explored solutions for all-speed flows. Among them was the concept of flowfield-dependent variation (FDV) methods [Chung, 1999] as detailed in Section 6.5. This was an attempt to resolve transitions and interactions of various physical properties such as inviscid/viscous, compressible/incompressible, and laminar/turbulent flows. The same approach can be applied to FEM. It can be shown that FDV methods are capable of generating most of the existing computational schemes in both FDM and FEM.

Although the various forms of Galerkin methods constitute the finite element methods in which the test functions are the same as the trial functions, there are other methods where the test functions are different from the trial functions, generally known as the weighted residual methods. Some examples include spectral element methods (SEM), least square methods (LSM), and finite point methods (FPM).

The finite element literature is enriched with mathematical error analysis. Mathematical proofs of convergence, stability, and accuracy are important in the so-called *hp* adaptive methods in which accuracy improves as the mesh is refined and the approximating polynomial degrees are increased in accordance with the flowfield gradients. This subject was developed by Babuska and his co-workers and Oden and his co-workers for the last two decades.

In this chapter, the FEM formulation presented in Chapter 1 will be repeated with more rigorous mathematical notations and expanded into multidimensional problems. Definitions used in error estimates and convergence properties are also introduced in this chapter.

The finite element analysis begins with the interpolation functions of the variables for one-dimensional, two-dimensional, and three-dimensional elements of various geometries with linear and high order approximations, presented in Chapter 9. This will be followed by linear steady and unsteady problems in Chapter 10 and nonlinear problems with convection-dominated flows in Chapter 11.

In Chapters 12 and 13, we present FEM formulations for incompressible flows and compressible flows, respectively. The major issues in CFD as observed in Part Two for FDM are as follows: (1) Difficulties of satisfying the conservation of mass in incompressible flows (incompressibility condition), resulting in checkerboard type pressure oscillations; (2) shock discontinuities in compressible flows; and (3) convection-dominated flows in both incompressible and compressible flows. Mixed methods, penalty methods, and pressure correction methods were developed to cope with the incompressibility condition. On the other hand, the Taylor-Galerkin methods (TGM) and generalized Petrov-Galerkin (GPG) methods have been successful in dealing with shock discontinuities and convection-dominated flows. Recent developments include computational methods capable of analyzing both compressible and incompressible flows by a single formulation and a single computer code using the various schemes extended from TGM, GPG, and FDV (Chapter 13), leading to "all speed flows."

Weighted residual methods including spectral element methods (SEM) and least square methods (LSM) are presented in Chapter 14. Finite point methods (FPM) using only the nodal points without element meshes (meshless methods) are also discussed in this chapter. The finite volume methods (FVM) via FEM are elaborated in Chapter 15.

Finally, in Chapter 16, we examine some of the significant analogies between FDM and FEM. Most of the existing computational schemes in both FDM and FEM are shown to be special cases of the flowfield-dependent variation (FDV) methods. There are many

numerical methods other than the FDM, FEM, and FVM which are based on the standard Eulerian coordinates. They include boundary element methods (BEM), coupled-Eulerian-Lagrangian (CEL) methods, particle-in-cell (PIC) methods, and Monte Carlo methods (MCM). For the sake of completeness, these methods are briefly discussed in Section 16.4.

8.2 FINITE ELEMENT FORMULATIONS

The basic concept of finite element formulations was presented in Chapter 1, for simple one-dimensional problems, using the Galerkin methods. In the Galerkin methods, the variable of the partial differential equation is approximated as a linear combination of the trial (interpolation, shape, or basis) functions. It was shown that local properties were assembled into a global form by superposition intuitively. In this chapter, we demonstrate this process directly from the global form using Boolean algebra, with the local properties then arising indirectly as a consequence.

For simplicity of illustration, let us consider a one-dimensional domain as depicted in Figure 8.2.1. Let the domain be divided into subdomains; say two local elements in this example. The end points of elements are called nodes. The finite element model $\overline{\Omega}$

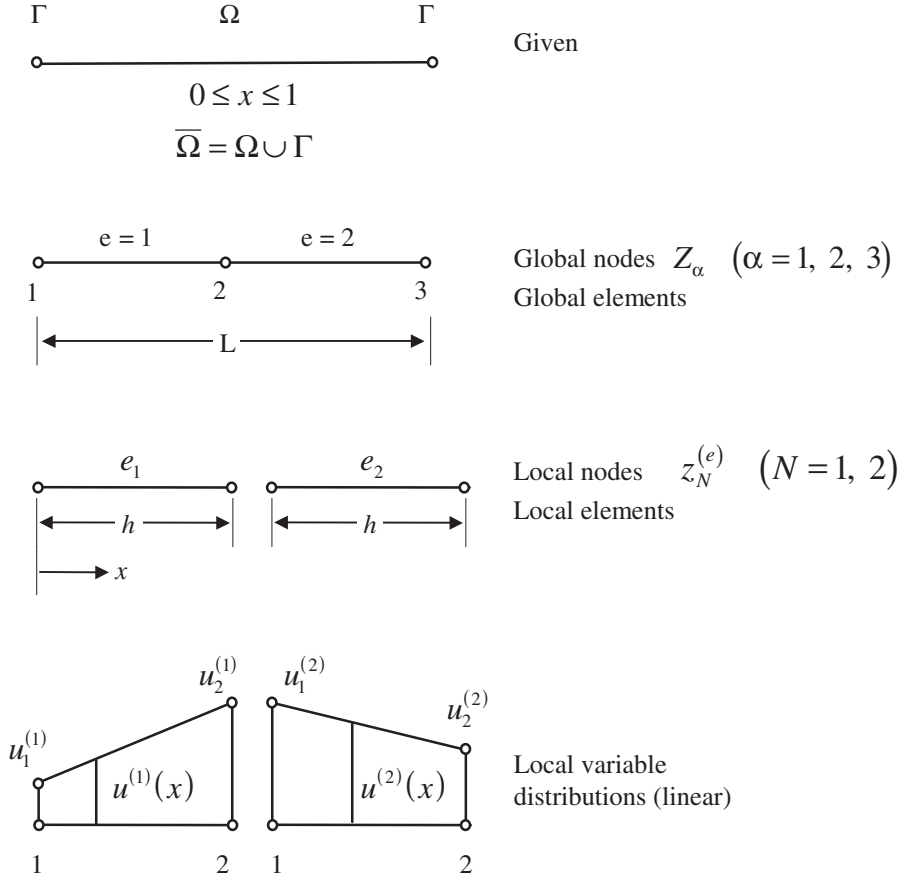

Figure 8.2.1 Finite element approximations.

is expressed as the union of the domain Ω and its boundaries Γ,

$$\overline{\Omega} = \Omega \bigcup \Gamma \tag{8.2.1}$$

We now isolate all elements from the global domain. Each local element $\overline{\Omega}_e$ is identified as

$$\overline{\Omega}_e = \Omega_e \bigcup \Gamma_e$$

The boundaries of this element and the neighboring element are the intersection

$$\Gamma_e \bigcap \Gamma_f = \Gamma_{ef}$$

Thus, the connected finite element model (8.2.1) is the union of all elements

$$\overline{\Omega} = \bigcup_{e=1}^{E} \overline{\Omega}_e \tag{8.2.2}$$

where E is the total number of elements. The global nodes of the connected model $\overline{\Omega}$ and the local nodes of isolated elements are identified by Z_α ($\alpha = 1, 2, 3$, being the number of global nodes) and $z_N^{(e)}$ ($N = 1, 2$, being the number of local nodes) with $e = 1, 2$, being the number of local elements, respectively. They are related as follows:

$$z_1^{(1)} = Z_1, \qquad z_2^{(1)} = Z_2, \qquad z_1^{(2)} = Z_2, \qquad z_2^{(2)} = Z_3,$$

Writing these relations in matrix form yields

$$\begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix}, \qquad \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} \tag{8.2.3a,b}$$

We may express (8.2.3a,b) as

$$z_N^{(e)} = \Delta_{N\alpha}^{(e)} Z_\alpha \quad (N = 1, 2,\ \alpha = 1, 2, 3) \tag{8.2.4}$$

where N is the *free* index capable of producing N number of independent equations corresponding to its range (2 in this case, resulting in two equations) and the *repeated (dummy)* indices α are summed throughout their range (3 in this case, resulting in three terms), known as the *index* notation or *tensor* notation. The symbol $\Delta_{N\alpha}^{(e)}$ is called the Boolean matrix having the property:

$$\Delta_{N\alpha}^{(e)} = \begin{cases} 1 & \text{if the local node } N \text{ corresponds to the global node } \alpha \\ 0 & \text{otherwise} \end{cases}$$

Similarly, we may write

$$Z_\alpha = \Delta_{N\alpha}^{(e)} z_N^{(e)} \tag{8.2.5}$$

where $\Delta_{N\alpha}^{(e)}$ in (8.2.5) is seen to be a transpose of $\Delta_{N\alpha}^{(e)}$ in (8.2.4). This transpose is achieved by the repeated index N in (8.2.5) arising with the first index of the Boolean matrix in contrast to the repeated index α in (8.2.4) arising with the second index of the Boolean matrix. Note that this is typical of index notation, different from the matrix notation.

Inserting (8.2.4) into (8.2.5) yields

$$Z_\alpha = \Delta^{(e)}_{N\alpha}\Delta^{(e)}_{N\beta} Z_\beta \tag{8.2.6}$$

from which we obtain the relation

$$\Delta^{(e)}_{N\alpha}\Delta^{(e)}_{N\beta} = \delta_{\alpha\beta} \tag{8.2.7}$$

where $\delta_{\alpha\beta}$ is the Kronecker delta,

$$\delta_{\alpha\beta} = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta \end{cases}$$

Likewise, substituting (8.2.5) into (8.2.4) gives

$$z^{(e)}_N = \Delta^{(e)}_{N\alpha}\Delta^{(e)}_{M\alpha} z^{(e)}_M \tag{8.2.8}$$

Once again, we obtain

$$\Delta^{(e)}_{N\alpha}\Delta^{(e)}_{M\alpha} = \delta_{NM} \tag{8.2.9}$$

In matrix notation, the above relation shows that

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{8.2.10}$$

The use of Boolean matrix $\Delta^{(e)}_{N\alpha}$ will prove to be convenient in derivations of finite element equations, relating the properties between the local and global systems. However, in actual executions of finite element computations, these Boolean matrices will never be constructed but instead are replaced by computer programs based on local and global node number correspondence.

It should be noted, at this point, that we make use of tensor notation in which a free single index implies the components of a column vector whereas free double indices denote a matrix with its size determined by the ranges of the indices. The free index must match at both sides of the equality sign within an equation. The advantage of using tensor notation in FEM will become obvious as we develop finite element equations more extensively in later chapters.

To obtain the finite element equations, the concept of classical variational or weighted residual methods is used. Toward this end, we require suitable functions for the variable to be approximated *locally* within an element or subdomain. This is in contrast to the classical variational methods or weighted residual methods where the *global* approximating functions are used, in which the satisfaction of boundary conditions is difficult, if not impossible, for complex geometries.

Suppose that the variable u may be approximated linearly within a local element e, $(0 \leq x \leq h)$, as shown in Figure 1.3.1, Figure 8.2.1, Figure 8.2.2:

$$u^{(e)}(x) = \Phi^{(e)}_N(x) u^{(e)}_N \tag{8.2.11}$$

where $\Phi^{(e)}_N(x)$ are called the local element trial functions [interpolation functions, shape functions, or basis functions as shown in (1.3.3)]. For simplicity, the argument (x) will be omitted in what follows unless confusion is likely to occur. They have the

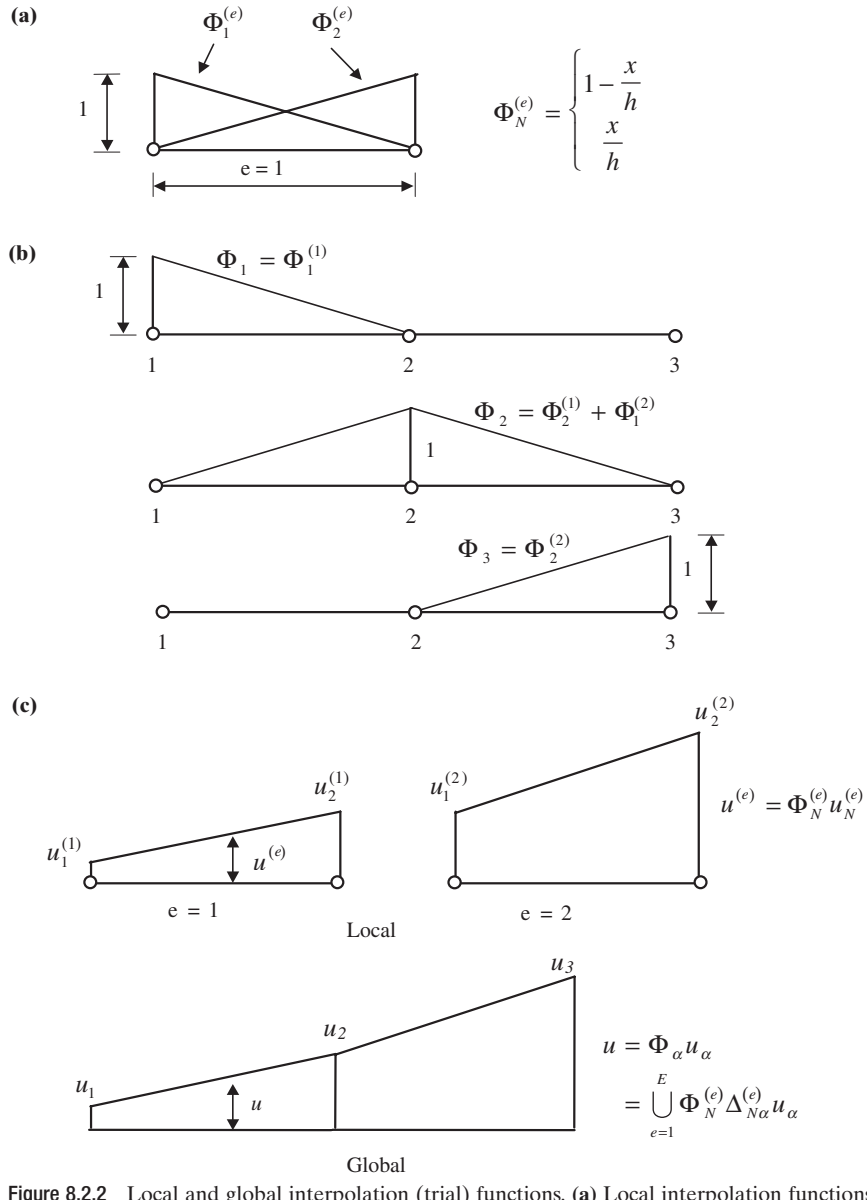

Figure 8.2.2 Local and global interpolation (trial) functions. **(a)** Local interpolation functions. **(b)** Global interpolation functions $\phi_\alpha = \bigcup_{e=1}^{E} \phi_N^{(e)} \Delta_{N\alpha}^{(e)}$. **(c)** Local and global values $(u_N^{(e)}, u_\alpha)$.

properties

$$0 \le \Phi_N^{(e)} \le 1, \qquad \sum_{N=1}^{2} \Phi_N^{(e)} = 1, \qquad \Phi_N^{(e)}\left(z_M^{(e)}\right) = \delta_{NM} \tag{8.2.12}$$

The local nodal values can be related to the global nodal values in a manner similar to (8.2.4):

$$u_N^{(e)} = \Delta_{N\alpha}^{(e)} u_\alpha \tag{8.2.13}$$

Thus, for the total number of elements, E, the global function can be written as the union of all local element contributions:

$$u = \bigcup_{e=1}^{E} u^{(e)} = \bigcup_{e=1}^{E} \Phi_N^{(e)} u_N^{(e)} = \bigcup_{e=1}^{E} \Phi_N^{(e)} \Delta_{N\alpha}^{(e)} u_\alpha \tag{8.2.14a}$$

or

$$u = \Phi_\alpha u_\alpha \tag{8.2.14b}$$

where Φ_α is called the global trial (interpolation, shape, or basis) function,

$$\Phi_\alpha = \bigcup_{e=1}^{E} \Phi_N^{(e)} \Delta_{N\alpha}^{(e)} \tag{8.2.15}$$

with

$$\Phi_\alpha(Z_\beta) = \delta_{\alpha\beta} \tag{8.2.16}$$

It follows from (8.2.15) that the expanded form of (8.2.14) appears as shown in Figure 8.2.2b,c. Note that the union operation in (8.2.14) and (8.2.15) is subject to the constraint (8.2.16). Thus, (8.2.14) through (8.2.16) lead to $u = u_1$ at node 1, $u = u_2$ at node 2, and $u = u_3$ at node 3. The union operation implies a Boolean summing rather than algebraic summing in this process.

With these preliminaries, we are now prepared to revisit the differential equation (1.2.1) for a more formal approach to the finite element solution process. There are two options for the formulation of finite element equations: (a) variational methods and (b) weighted residual methods. In the variational methods, we minimize the variational principle for the governing differential equation, which is a common practice in structural mechanics. Unfortunately, however, variational principles are not available in exact forms for nonlinear fluid mechanics equations in general. Thus, it is logical to seek the weighted residual methods in fluid mechanics where the variational principles are not required. The basic idea of the weighted residual methods is to construct a mathematical process in which the error or the residual of the governing differential equation(s), R (for example, $R = \nabla^2 u$), is minimized to zero. This can be done by forming a subspace spanned by test functions or weighting functions, W_α, and projecting the residual R orthogonally onto this subspace. This process is known as the *inner product* of the test function and the residual, which can be expressed as follows:

$$(W_\alpha, R) = \int_0^1 W_\alpha R dx = 0, \quad 0 < x < 1 \tag{8.2.17}$$

where the test functions W_α are known also as weighting functions. The integral given by (8.2.17) implies that the error at each point in the domain orthogonally projected onto a functional space spanned by the weighting function summed over the entire domain is set equal to zero. This process will provide necessary algebraic equations from which unknowns can be calculated. Thus, the finite element method is sometimes called the projection method.

If the test functions W_α are replaced by the trial functions Φ_α, then the scheme is known as the Galerkin method,

$$(\Phi_\alpha, R) = \int_0^1 \Phi_\alpha R dx = 0 \quad 0 < x < 1 \tag{8.2.18}$$

Note that Φ_α act as trial functions in (8.2.14) but are treated as test functions in (8.2.18). Formulations with test functions different from trial functions such as in the generalized Petrov-Galerkin methods will be discussed in Chapter 11 for nonlinear or convection-dominated flows.

For the purpose of illustration, let the residual be given by the differential equation (1.2.1a). We then obtain the so-called global Galerkin integral,

$$(\Phi_\alpha, R) = \int_0^1 \Phi_\alpha \left(\frac{d^2u}{dx^2} - 2 \right) dx = 0 \tag{8.2.19}$$

This is in contrast to the local Galerkin integral (1.3.4). Integrate (8.2.19) by parts to arrive at the form known as the *variational equation*,

$$\overset{*}{\Phi}_\alpha \frac{du}{dx}\bigg|_0^1 - \int_0^1 \frac{d\Phi_\alpha}{dx}\frac{du}{dx}dx - \int_0^1 2\Phi_\alpha dx = 0 \tag{8.2.20}$$

where $du/dx|_0^1$ is the global Neumann boundary condition to be specified either at $x=0$ or at $x=1$ if required. Note also that $\overset{*}{\Phi}_\alpha$ indicates the global boundary test function defined only at $x=0$ or $x=1$, which is no longer a continuous function of x [Chung, 1978]. This is because the role of Φ_α is no longer the same at boundaries as in the domain. It is important to realize that $du/dx|_0^1$ in (8.2.20) arises from the one-dimensional assumption of the two-dimensional problem (Figure 8.2.3),

$$\int\int \frac{d^2u}{dx^2}dxdy = \int \frac{du}{dx}dy = \int \frac{du}{dx}\cos\theta d\Gamma = \frac{du}{dx}\cos\theta = \frac{du}{dx}\bigg|_0^1$$

where the integral $\int_\Gamma d\Gamma$ is unity in one dimension, and

$$\frac{du}{dx}\bigg|_{x=1} = \frac{du}{dx}\cos 0^\circ = \frac{du}{dx}(1) \tag{8.2.21a}$$

$$\frac{du}{dx}\bigg|_{x=0} = \frac{du}{dx}\cos 180^\circ = \frac{du}{dx}(-1) \tag{8.2.21b}$$

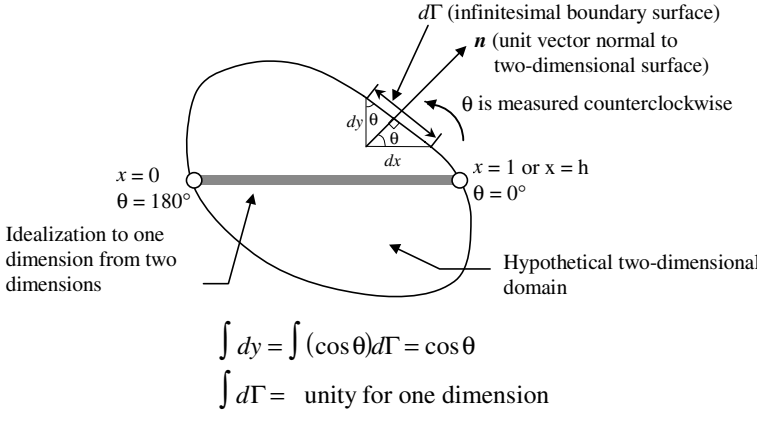

Figure 8.2.3 One-dimensional idealization from hypothetical two-dimensional domain.

The above result is due to the one-dimensional idealization of two-dimensional problems presented in Chapter 10. An important application of the above development is demonstrated for implementation of Neumann boundary conditions at the right or left end nodes in Section 1.5. Substituting (8.2.14b) into (8.2.20) gives

$$\left[\int_0^1 \frac{d\Phi_\alpha}{dx}\frac{d\Phi_\beta}{dx}dx\right]u_\beta = -\int_0^1 2\Phi_\alpha dx + \overset{*}{\Phi}_\alpha \frac{du}{dx}\bigg|_0^1 \tag{8.2.22}$$

We recognize that the left-hand side integral of (8.2.22) represents the first order derivative, known as a *weak form*, reduced (weakened) from the original second order derivative of the governing equation. The solution obtained from this weak form is known as the *weak solution*.

At this time, it is informative to point out that the result similar to (8.2.22) can be obtained using the variational principle approach [Chung, 1978]. The variational principle for the governing differential equation (1.2.1a) is of the form

$$I = \int_0^1 \left[\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2 + 2u\right]dx \tag{8.2.23}$$

In the variational methods, the above integral is minimized with respect to the nodal value of the variable.

$$\delta I = \frac{\partial I}{\partial u_\alpha}\delta u_\alpha = 0$$

Since δu_α is arbitrary, we require

$$\frac{\partial I}{\partial u_\alpha} = 0$$

It can easily be verified that the minimization (differentiation) of (8.2.23) with respect to the nodal values of u_α as indicated above results in (8.2.22) except that the Neumann boundary condition must be manually added. This analogy does not exist in nonlinear fluid mechanics equations, because the integration of the nonlinear convection term by parts can not be carried out in an exact form.

With compact notation, we rewrite (8.2.22) in the form

$$K_{\alpha\beta}u_\beta = F_\alpha + G_\alpha \tag{8.2.24}$$

where $K_{\alpha\beta}$ is the global stiffness, diffusion, or viscosity matrix, F_α is the global load or source vector, and G_α is the global Neumann boundary vector, as deduced from (8.2.22):

$$K_{\alpha\beta} = \int_0^1 \frac{d\Phi_\alpha}{dx}\frac{d\Phi_\beta}{dx}dx = \bigcup_{e=1}^{E}\int_0^h \frac{d\Phi_N^{(e)}}{dx}\frac{d\Phi_M^{(e)}}{dx}dx\,\Delta_{N\alpha}^{(e)}\Delta_{M\beta}^{(e)} = \bigcup_{e=1}^{E} K_{NM}^{(e)}\Delta_{N\alpha}^{(e)}\Delta_{M\beta}^{(e)} \tag{8.2.25}$$

$$F_\alpha = -\int_0^1 2\Phi_\alpha dx = -\bigcup_{e=1}^{E}\int_0^h 2\Phi_N^{(e)}dx\,\Delta_{N\alpha}^{(e)} = \bigcup_{e=1}^{E} F_N^{(e)}\Delta_{N\alpha}^{(e)} \tag{8.2.26a}$$

$$G_\alpha = \overset{*}{\Phi}_\alpha \frac{du}{dx}\bigg|_0^1 = \bigcup_{e=1}^{E} \overset{*}{\Phi}_N^{(e)}\Delta_{N\alpha}^{(e)}\frac{du}{dx}\bigg|_0^h = \bigcup_{e=1}^{E} G_N^{(e)}\Delta_{N\alpha}^{(e)} \tag{8.2.26b}$$

where $\overset{*}{\Phi}{}_N^{(e)}$ is the local Neumann boundary interpolation function,

$$\overset{*}{\Phi}{}_N^{(e)} = \delta(\overset{*}{z}_N - z_N), \qquad \overset{*}{\Phi}{}_N^{(e)}(z_M^{(e)}) = \delta_{NM} \tag{8.2.27}$$

indicating that $\overset{*}{\Phi}{}_N^{(e)}$ is the Dirac delta function at $x = 0$ or $x = h$, being unity at $\overset{*}{z}_N = z_N$ where du/dx is prescribed at node $\overset{*}{z}_N$ and zero elsewhere. This implies that, if the Neumann boundary condition is to be applied to a node, then we set $\overset{*}{\Phi}{}_N^{(e)} = 1$ for that node. Otherwise, we set $\overset{*}{\Phi}{}_N^{(e)} = 0$. Assembled in a global system, we obtain

$$\overset{*}{\Phi}{}_\alpha = \delta(\overset{*}{Z}_\alpha - Z_\alpha), \qquad \overset{*}{\Phi}{}_\alpha(Z_\beta) = \delta_{\alpha\beta} \tag{8.2.28}$$

In this process, the Neumann boundary conditions are actually enforced between the adjacent elements, with positive and negative gradients cancelled throughout the domain (thus establishing the "energy balance" across the adjacent local element interfaces) until the end point is reached. This is where the actual Neumann boundary conditions are to be physically applied. This process is explicitly demonstrated by having constructed the FEM equations (8.2.24) in a global form instead of beginning with the local form and assembling the element stiffness matrices to a global form afterward. This is contrary to the traditional FEM formulations shown in other textbooks.

The global stiffness matrix (1.3.8), source vector (1.3.9), and Neumann boundary vector (1.3.10) are now assembled from the local element properties as

$$\begin{aligned} K_{\alpha\beta} &= \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} = \begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & 0 \\ K_{21}^{(1)} & K_{22}^{(1)} + K_{11}^{(2)} & K_{12}^{(2)} \\ 0 & K_{21}^{(2)} & K_{22}^{(2)} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \frac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{1}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \frac{1}{h} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \end{aligned} \tag{8.2.29}$$

$$F_\alpha = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} F_1^{(1)} \\ F_2^{(1)} + F_1^{(2)} \\ F_2^{(2)} \end{bmatrix} = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} h \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} h \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -h \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \tag{8.2.30}$$

$$\begin{aligned} G_\alpha &= \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} = \begin{bmatrix} G_1^{(1)} \\ G_2^{(1)} + G_1^{(2)} \\ G_2^{(2)} \end{bmatrix} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \overset{*}{\Phi}{}_1^{(1)} \\ \overset{*}{\Phi}{}_2^{(1)} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \overset{*}{\Phi}{}_1^{(2)} \\ \overset{*}{\Phi}{}_2^{(2)} \end{bmatrix} \right\} \frac{du}{dx} \cos\theta \\ &= \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} \frac{du}{dx} \cos\theta = \begin{bmatrix} \overset{*}{\Phi}{}_1 \\ \overset{*}{\Phi}{}_2 \\ \overset{*}{\Phi}{}_3 \end{bmatrix} \frac{du}{dx} \cos\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \frac{du}{dx} \cos\theta \end{aligned} \tag{8.2.31}$$

with $\overset{*}{\Phi}_1 = \overset{*}{\Phi}_2 = \overset{*}{\Phi}_3 = 0$ indicating that the Neumann boundary conditions are not to be applied to any of the global nodes for the solution of (1.2.1a,b). This implies that, if the Neumann boundary conditions are not applied, then the Neumann boundary vector is zero even if the gradient du/dx is not zero. Recall that in Section 1.3 the assembly of local properties into a global form was achieved intuitively. This has now been verified with a mathematical rigor of Boolean matrices. In practice, however, these Boolean matrices are never constructed, but they are replaced by computer codes based on the nodal correspondence between global and local nodes as detailed in (10.1.15c,d).

For multidimensional problems, the formulation of the finite element equations is carried out similarly as in one-dimensional problems. For example, let us examine the Poisson equation,

$$R = \nabla^2 u - f = 0 \tag{8.2.32}$$

The corresponding finite element equation takes the form

$$\int_\Omega \Phi_\alpha (u_{,ii} - f) d\Omega = 0 \tag{8.2.33}$$

$$\int_\Gamma \overset{*}{\Phi}_\alpha u_{,i} n_i d\Gamma - \int_\Omega \Phi_{\alpha,i} u_{,i} d\Omega - \int_\Omega \Phi_\alpha f d\Omega = 0 \tag{8.2.34}$$

or

$$K_{\alpha\beta} u_\beta = F_\alpha + G_\alpha \tag{8.2.35}$$

with

$$K_{\alpha\beta} = \int_\Omega \Phi_{\alpha,i} \Phi_{\beta,i} d\Omega = \bigcup_{e=1}^{E} \int_\Omega \Phi^{(e)}_{N,i} \Phi^{(e)}_{M,i} d\Omega \Delta^{(e)}_{N\alpha} \Delta^{(e)}_{M\beta} \tag{8.2.36}$$

$$F_\alpha = \int_\Omega \Phi_\alpha f d\Omega = \bigcup_{e=1}^{E} \int_\Omega \Phi^{(e)}_N f d\Omega \Delta^{(e)}_{N\alpha} \tag{8.2.37}$$

$$G_\alpha = \int_\Gamma \overset{*}{\Phi}_\alpha u_{,i} n_i d\Gamma = \bigcup_{e=1}^{E} \int_\Gamma \overset{*}{\Phi}{}^{(e)}_N u_{,i} n_i d\Gamma \Delta^{(e)}_{N\alpha} \tag{8.2.38}$$

For two-dimensional problems, trial and test functions, $\Phi^{(e)}_N$, are functions of x and y and thus the Neumann boundary test functions, $\overset{*}{\Phi}{}^{(e)}_N$, are functions of one dimension around the boundary contour. This will require the numerical integration around the boundaries. Step-by-step details of assembly for applications to multidimensional geometries will be presented in Chapter 10.

Before we proceed further, we must recognize the special mathematical and physical implications of the expression given by (8.2.34). This is the variational equation or the *weak* form of the original governing equation (8.2.32), which is the two-dimensional form of (8.2.20). Physically, if the residual (8.2.32) represents the force, then the integral given by (8.2.33) implies the energy contained in the domain Ω. Once integrated by parts

as in (8.2.34), the consequence implies the energy balance between the domain Ω and the boundary surface Γ containing the Neumann boundary conditions (normal gradients of u). Thus, the physical consequence of the variational equation (or energy) allows us to add any number of physical constraints in variational forms. These constraints can be those terms playing a role of numerical diffusion (viscosity) as necessary. Many of the recent developments of FEM take advantage of this variational concept, which we shall discuss in greater detail in later chapters.

Unfortunately, the Galerkin methods described in (8.2.33–8.2.35) lead to unstable and inaccurate solutions in fluid dynamics equations in which the flow is convection-dominated. In this case, we must use the methods of weighted residual (MWR) with test functions W_α chosen differently from the trial functions Φ_α such that

$$(W_\alpha,\, R) = \int_\Omega W_\alpha R d\Omega = 0 \tag{8.2.39}$$

Thus, the determination of the most suitable test functions W_α remains the crucial task in order to be successful in dealing with convection-dominated flows. The most commonly used test functions are the Galerkin test functions Φ_α plus the numerical diffusion test functions Ψ_α. In this case, the finite element equations are of the form,

$$((\Phi_\alpha + \Psi_\alpha),\, R) = \int_\Omega (\Phi_\alpha + \Psi_\alpha) R d\Omega = 0 \tag{8.2.40}$$

Here, the numerical diffusion test functions Ψ_α play a role of numerical viscosities, equivalent to those used in FDM formulations. Some specific applications include streamline upwind Petrov-Galerkin (SUPG) methods, Taylor-Galerkin methods (TGM), generalized Petrov-Galerkin (GPG) methods, characteristic Galerkin methods (CGM), discontinuous Galerkin methods (DGM), and flowfield-dependent variation (FDV) methods, discussed in Chapters 11 through 13.

For multidimensional time-dependent problems, $R_j = \frac{\partial \mathrm{v}_j}{\partial t} + \mathrm{v}_{j,i}\mathrm{v}_i - \mathrm{v}_{j,ii} - f_j$, the general approach is to construct a double inner product of space and time in the form,

$$(\hat{W}(\xi), ((\Phi_\alpha + \Psi_\alpha),\, R_j)) = \int_\xi \hat{W}(\xi) \int_\Omega (\Phi_\alpha + \Psi_\alpha) R_j d\Omega d\xi = 0 \tag{8.2.41}$$

where $\hat{W}(\xi)$ is the temporal test function approximating the temporal variation between the discrete time steps with ξ being the nondimensional time variable. Note that the temporal approximation used here is independent of and discontinuous from the spatial approximations. Details on transient time-dependent problems with and without convection will be presented in Chapters 10 through 14 for linear and nonlinear cases.

8.3 DEFINITIONS OF ERRORS

Definitions of errors and error estimates for finite element methods have been well developed since the early 1970s. Finite element computational errors are defined in various norms. The most frequently used error norms are the pointwise error, L_2 norm error, and energy norm error. These error norms are the special cases of the more

rigorous and general norm, called the Sobolev space norm, which can then be simplified into more meaningful and practical error definitions.

Sobolev Space (W_p^m) Norm Error

Let us define the global node error e_α as

$$e_\alpha = u_\alpha - \hat{u}_\alpha \tag{8.3.1}$$

where u_α and $\hat{u}_\alpha$ denotes the finite element approximate solution and exact solution, respectively. Then, the Sobolev space norm error is defined as

$$\|e\|_{W_p^m} = \left\{ \int \left[e^p + \left(\frac{de}{dx}\right)^p + \left(\frac{d^2e}{dx^2}\right)^p + \cdots + \left(\frac{d^me}{dx^m}\right)^p \right] dx \right\}^{\frac{1}{p}} \tag{8.3.2}$$

where m denotes the highest order of the weak derivatives of the $2m$th governing equation and p represents the power to which the derivatives are raised. Here, weak derivatives refer to the order $m, m-1, \ldots 0$. The Sobolev space (W_p^m) is defined as the functional space which includes all weak derivatives with p integrable functions, $0 \le p \le \infty$.

Hilbert Space (H^m) Norm Error

The Hilbert space (H^m) is the Sobolev space (W_p^m) with p equal to 2, $H^m = W_2^m$. Thus

$$\|e\|_{H^m} = \|e\|_{W_2^m} = \left\{ \int \left[e^2 + \left(\frac{de}{dx}\right)^2 + \left(\frac{d^2e}{dx^2}\right)^2 + \cdots + \left(\frac{d^me}{dx^m}\right)^2 \right] dx \right\}^{\frac{1}{2}} \tag{8.3.3}$$

It is seen that the Hilbert space is the square integrable function ($p = 2$) complete in the inner product space.

Energy Norm Error

The energy norm error, $\|e\|_E$ is a special case of the Hilbert space norm error H^m in the $2m$th order differential equation. Thus, for the fourth order equation ($m = 2$), we have

$$\|e\|_E = \|e\|_{H^2} = \|e\|_{W_2^2} = \left\{ \int \left[e^2 + \left(\frac{de}{dx}\right)^2 + \left(\frac{d^2e}{dx^2}\right)^2 \right] dx \right\}^{\frac{1}{2}} \tag{8.3.4}$$

Notice that, for the second order differential equation ($m = 1$), we write

$$\|e\|_E = \|e\|_{H^1} = \|e\|_{W_2^1} = \left\{ \int \left[e^2 + \left(\frac{de}{dx}\right)^2 \right] dx \right\}^{\frac{1}{2}} \tag{8.3.5}$$

which can be written in terms of nodal errors e_α with $e = \Phi_\alpha e_\alpha$,

$$\|e\|_E = \left\{ \int \left[\Phi_\alpha \Phi_\beta + \frac{d\Phi_\alpha}{dx}\frac{d\Phi_\beta}{dx} \right] dx e_\alpha e_\beta \right\}^{\frac{1}{2}} \tag{8.3.6}$$

Here, as usual, the global interpolation functions are obtained by means of assembly of the local interpolation functions $\Phi_N^{(e)}$.

L_2 Space Norm Error

The L_2 space arises from the Banach space (L_p) with $p = 2$, equivalent to the Hilbert space (H^m) with $m = 0$. Thus

$$\|e\|_{L_2} = \|e\|_{H^0} = \|e\|_{W_2^0} = \left(\int e^2 dx \right)^{\frac{1}{2}} \tag{8.3.7}$$

in which no rates of change of errors are involved.

p-Norm (Banach Space Norm) Error

The Banach space (L_p) is defined as the complete normed linear space such that

$$\|e\|_{L_p} = \left(\int e^p dx \right)^{\frac{1}{p}}$$

For $p = 1$ and $p = \infty$, we obtain L_1 and L_∞ norms, respectively,

$$\|e\|_{L_1} = \int e dx = \sum_{j=1}^{n} (|e_1| + |e_2| + \cdots + |e_n|) \tag{8.3.8}$$

$$\|e\|_{L_\infty} = \max_j |e_j| \tag{8.3.9}$$

It should be noted that the L_2 norm is a special case of the Banach space norm $(p = 2)$, and is one of the most widely used error norm. Other norms of Banach space (other than $p = 1, 2, \infty$) are seldom used in practice.

Pointwise Error or Root Mean Square (RMS) Error

This is the simplest form of an error definition given by

$$\|e\|_{\text{RMS}} = \left(\sum e^2 \right)^{\frac{1}{2}} = (e_\alpha e_\alpha)^{\frac{1}{2}} \tag{8.3.10}$$

Here the percent error may be defined as

$$\|e\|_{\%} = \frac{\|e\|_{\text{RMS}}}{\left(\sum u^2\right)^{\frac{1}{2}}} = \left(\frac{e_\alpha e_\alpha}{u_\beta u_\beta} \right)^{\frac{1}{2}} \tag{8.3.11}$$

Note that there is no integral involved in this approach, thus it is called the pointwise error, or often known as the root mean square (RMS) error.

Matrix Norms

Matrix norms are an important concept in determining the computational stability of the finite element equations such as in (8.2.35) in terms of the so-called *condition number*. To demonstrate this concept, we write (8.2.35) in the matrix form

$$\mathbf{Ku} = \mathbf{F} \tag{8.3.12}$$

If $\mathbf{K}$ is an $n \times n$ matrix and $\mathbf{u}$ any vector with n components, then there exists a constant c such that

$$\|\mathbf{Ku}\| \leq c\|\mathbf{F}\| \tag{8.3.13}$$

where $\mathbf{u} \neq \mathbf{0}, \|\mathbf{u}\| > 0$, and the constant c is given by

$$c \geq \frac{\|\mathbf{Ku}\|}{\|\mathbf{u}\|} \tag{8.3.14}$$

The smallest c is known as the matrix norm of $\mathbf{K}$, denoted by $\|\mathbf{K}\|$.

$$\|\mathbf{K}\| \leq \max \frac{\|\mathbf{Ku}\|}{\|\mathbf{u}\|} \tag{8.3.15}$$

with the matrix norm being calculated from

$$\|\mathbf{K}\|_{L_1} = \max \sum_{\beta} |\mathbf{K}_{\alpha\beta}|, \qquad \|\mathbf{K}\|_{L_2} = (K_{\alpha\beta} K_{\beta\alpha})^{1/2}, \qquad \|\mathbf{K}\|_{L_\infty} = \max \sum_{\alpha} |\mathbf{K}_{\alpha\beta}|$$

Combining (8.3.13) and (8.3.15), we obtain

$$\|\mathbf{Ku}\| \leq \|\mathbf{K}\|\|\mathbf{u}\| \tag{8.3.16}$$

If we define the condition number N as

$$N(\mathbf{K}) = \|\mathbf{K}\|\|\mathbf{K}^{-1}\| \tag{8.3.17}$$

the following theorem can be established.

Theorem: A linear system of equations given by (8.3.12) is said to be *well-conditioned* if the condition number as defined in (8.3.17) is small.

Proof: It follows from (8.3.12) and (8.3.16) that $\|\mathbf{F}\| \leq \|\mathbf{K}\|\|\mathbf{u}\|$. Let $\mathbf{F} \neq \mathbf{0}, \mathbf{u} \neq \mathbf{0}$. Then, we have

$$\frac{1}{\|\mathbf{u}\|} \leq \frac{\|\mathbf{K}\|}{\|\mathbf{F}\|} \tag{8.3.18}$$

Let the residual be given by

$$\mathbf{R} = \mathbf{K}(\mathbf{u} - \hat{\mathbf{u}}) \tag{8.3.19}$$

Combining (8.3.16) and (8.3.19) leads to

$$\|\mathbf{u} - \hat{\mathbf{u}}\| = \|\mathbf{K}^{-1}\mathbf{R}\| \leq \|\mathbf{K}^{-1}\|\|\mathbf{R}\| \tag{8.3.20}$$

From (8.3.18) and (8.3.20) we obtain

$$\frac{\|\mathbf{u} - \hat{\mathbf{u}}\|}{\|\mathbf{u}\|} \leq \frac{1}{\|\mathbf{u}\|}\|\mathbf{K}^{-1}\|\|\mathbf{R}\| \leq \frac{\|\mathbf{K}\|}{\|\mathbf{F}\|}\|\mathbf{K}^{-1}\|\|\mathbf{R}\| = N(\mathbf{K})\frac{\|\mathbf{R}\|}{\|\mathbf{F}\|} \tag{8.3.21}$$

This proves that a small relative error results from the small condition number with the system being well-conditioned. Otherwise, the system is ill-conditioned.

Example 8.3.1

Given:

$$\mathbf{e} = \begin{bmatrix} 1 \\ -2 \\ -3 \\ 2 \end{bmatrix}$$

Required: Find the vector norms in L_1, L_2, L_∞.

Solution: $\|\mathbf{e}\|_{L_1} = 8; \quad \|\mathbf{e}\|_{L_2} = \sqrt{18}; \quad \|\mathbf{e}\|_{L\infty} = 3$

Example 8.3.2

Given:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 10 & 0 \\ 1 & 1 & 5 & 1 \\ 0 & 1 & 5 & 1 \\ 0 & 0 & 5 & 1 \end{bmatrix}$$

Required: Find the matrix norms in L_1, L_2, L_∞.

Solution: $\|\mathbf{K}\|_{L_1} = \max\{1, 2, 25, 3\} = 25; \|\mathbf{K}\|_{L_2} = \sqrt{181}; \|\mathbf{K}\|_{L_\infty} = \max\{10, 8, 7, 6\} = 10$

Typical convergence properties are shown in Figure 8.3.1. It is seen in Figure 8.3.1a that convergence is achieved at the point N and that further refinements or the increase of polynomial degrees do not affect the exact solution. The convergence to the exact solution depends on the so-called *mesh parameter*. The mesh parameter h is defined as "diameter" of the largest element in a given domain. For one-dimensional problems, it is simply the length h of the domain with $0 < h < 1$. Let e_1 and e_2 be the errors for the mesh parameters h_1 and h_2, respectively. Assume that reduction of mesh parameters results in the increase of the order p of the rate of convergence. This relation may be written in the form (Figure 8.3.1b)

$$\frac{\|e_1\|}{\|e_2\|} = \left(\frac{h_1}{h_2}\right)^p \tag{8.3.22}$$

Taking the natural logarithm on both sides, we obtain

$$p = \frac{\ln \|e_1\| - \ln \|e_2\|}{\ln h_1 - \ln h_2} \tag{8.3.23}$$

where the magnitude of p is indicative of the rate of convergence of the finite element solution to the exact solution. In plotting the computed results to examine the convergence, one may choose at least three different mesh parameters. They should be chosen in the range where convergence to the exact solution has not been achieved as illustrated in points 1, 2, and 3 of Figure 8.3.1a,b. The slope p is seen to be a straight line with accuracy increasing with a steeper slope. If the mesh parameter is chosen too small beyond convergence, the slope p will become horizontal ($p = 0$), such as points 4, 5, and 6 in Figure 8.3.1a. If computational round-off errors are accumulated due to the

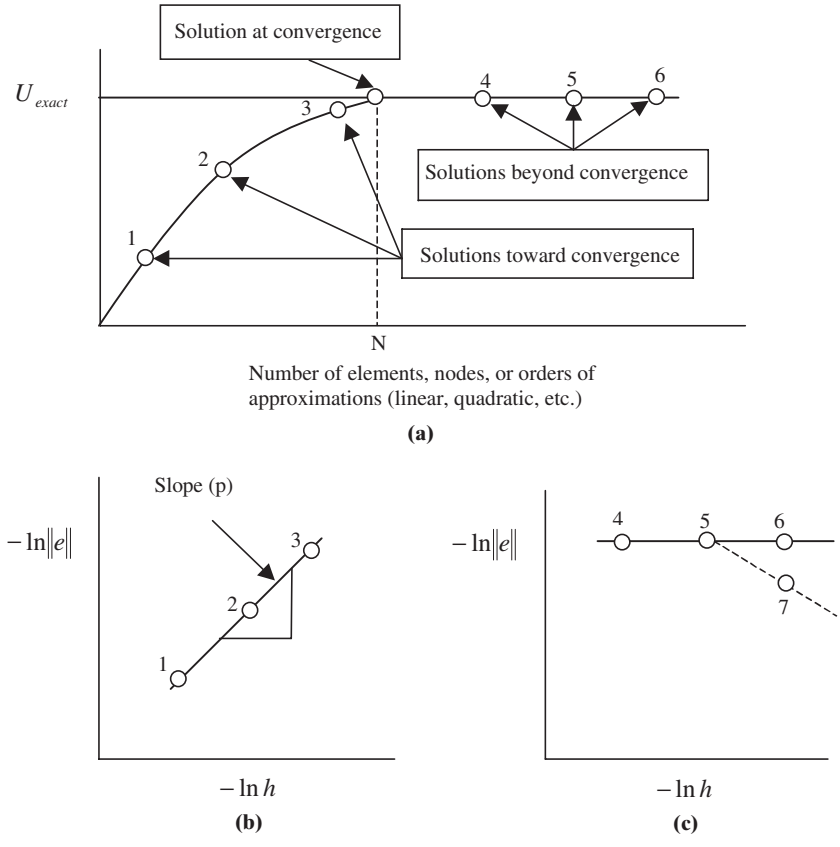

Figure 8.3.1 Convergence toward and beyond exact solutions. Notice that the order (p) of rate of convergence becomes horizontal ($p = 0$) for the solutions beyond convergence but may turn negative ($p < 0$) due to round-off errors. **(a)** Solutions vs. refinements. **(b)** Solutions toward convergence. **(c)** Solutions beyond convergence.

limitation of the computer through no fault of the computational methodology itself, then the slope may tend to deviate from the horizontal line (point 7 in Figure 8.3.1b). This is not an indication of deterioration of accuracy or rate of convergence, but rather it is meaningless to show the rate of convergence beyond the point at which convergence has already been achieved.

In recent years, error estimates particularly in the adaptive *h-p* methods have been studied extensively by Babuska and his co-workers [Babuska and Guo, 1988] and Oden and his co-workers [Oden and Demkowicz, 1991], among others. Some discussions on this topic will be presented in Chapter 19.

8.4 SUMMARY

In this chapter, we revisited Chapter 1 and reintroduced the finite element theory with more rigorous mathematical foundations as applied to multidimensional problems. Definitions of errors in terms of various functional norms and convergence vs. errors have also been presented.

Notations used in this book are designed in such a way that the beginner can understand the procedure of formulations and computer programming more easily, using tensorial indices. This is in contrast to most of the journal papers or other CFD books in which direct tensors or matrices are used. They are simple in writing, but confusing to the beginner and inconvenient for computer programming. To alleviate these difficulties, tensor notations with indices are used throughout this book.

Tensors with indices, although cumbersome to write, reveal the precise number of equations and exact number of terms in an equation. From this information, all inner and outer do-loops in the computer programming can be constructed easily, facilitating the multiplication of matrix and vector quantities with specified sizes precisely and explicitly defined.

If indices are not balanced, then the reader is warned that derivations of the equations are in error and are possibly in violation of the physical laws. In this case, the computer programmer is immediately reminded that it is not possible to proceed with incorrect indexing of do-loops. Moreover, a tensor represents the concept of invariance of physical properties with the frame of reference, safeguarding the physical laws, constitutive equations, and subsequently the computational processes as well.

Instead of constructing finite element equations in a local form which are then assembled into a global form as shown in Section 1.3, it is convenient to perform global formulations from the beginning so that flow physics can be accommodated in a global form easily in the development of complex finite element equations. The direct global formulation of finite element equations will be followed for the rest of this book.

REFERENCES

Babuska, I. and Guo, B. Q. [1988]. The *h-p* version of the finite element method for domains with curved boundaries. *SIAM J. Num. Anal.*, 25, 4, 837–61.

Babuska, I., Szabo, B. A., and Katz, I. N. [1981]. The p-version of the finite element method. *SIAM J. Num. Anal.*, 18, 512–45.

Baker, A. J. [1983]. *Finite Element Computational Fluid Mechanics*. New York: Hemisphere, McGraw-Hill.

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

———. [1999]. Transitions and interactions of inviscid/viscous, compressible/incompressible and laminar/turbulent flows. *Int. J. Num. Meth. Fl.*, 31, 223–46.

Donea, J. [1984]. A Taylor-Galerkin method for convective transport problems. *Int. J. Num. Meth. Eng.*, 20, 101–19.

Heinrich, J. C., Huyakorn, P. S., Zienkiewicz, O. C., and Mitchell, A. R. [1977]. An upwind finite element scheme for two-dimensional convective transport equation. *Int. J. Num. Meth. Eng.*, 11, 1, 131–44.

Hughes, T. J. R. and Brooks, A. N. [1982]. A theoretical framework for Petrov-Galerkin methods with discontinuous weighting functions: application to the streamline upwind procedure. In R. H. Gallagher et al. (eds). *Finite Elements in Fluids*, London: Wiley.

Hughes, T., Mallet, M. and Mizukami, A. [1986]. A new finite element formulation for computational fluid dynamics I. Beyond SUPG, *Comp. Meth. Appl. Mech. Eng.*, 54, 341–55.

Johnson, C. [1987]. Numerical Solution of Partial Differential Equations on the Element Method. student litteratur, Lund, Sweden.

Löhner, R., Morgan, K., and Zienkiewicz, O. C. [1985]. An adaptive finite element procedure for compressible high speed flows. *Comp. Meth. Appl. Mech. Eng.*, 51, 441–65.
Oden, J. T., Babuska, I., and Baumann, C. E. [1998]. A discontinuous hp finite element methods for diffusion problems. *J. Comp. Phy.*, 146, 491–519.
Oden, J. T. and Demkowicz, L. [1991]. h-p adaptive finite element methods in computational fluid dynamics. *Comp. Meth. Appl. Mech. Eng.*, 89 (1–3): 1140.
Oden, J. T. and L. C. Wellford, Jr. [1972]. Analysis of viscous flow by the finite element method. *AIAA J.*, 10, 1590–99.
Zienkiewicz, O. C. and Cheung, Y. K. [1965]. Finite elements in the solution of field problems. *The Engineer*, 507–10.
Zienkiewicz, O. C. and Codina, R. [1995]. A general algorithm for compressible and incompressible flow–Part I. Characteristic-based scheme. *Int. J. Num. Meth. Fl.*, 20, 869–85.