

Fundamentals for Finite Element Method

CHAPTER OUTLINE HEAD

3.1 Introduction	44
3.2 Strong and Weak Forms: Problem Formulation	44
3.3 Hamilton's Principle: A Weak Formulation	45
3.3.1 Hamilton's Principle.....	45
3.3.2 Minimum Total Potential Energy Principle	47
3.4 FEM Procedure	47
3.4.1 Domain Discretization	47
3.4.2 Displacement Interpolation	49
3.4.3 Standard Procedure for Constructing Shape Functions.....	50
3.4.3.1 <i>On the Inverse of the Moment Matrix</i>	<i>54</i>
3.4.3.2 <i>On the Compatibility of the Shape Functions</i>	<i>54</i>
3.4.3.3 <i>On Other Means of Construct Shape Functions</i>	<i>54</i>
3.4.4 Properties of the Shape Functions	54
3.4.5 Formulation of Finite Element Equations in Local Coordinate System.....	63
3.4.6 Coordinate Transformation	67
3.4.7 Assembly of Global FE Equation	68
3.4.8 Imposition of Displacement Constraints	69
3.4.9 Solving the Global FE Equation	69
3.5 Static Analysis.....	69
3.6 Analysis of Free Vibration (Eigenvalue Analysis).....	69
3.7 Transient Response	71
3.7.1 Central Difference Algorithm	72
3.7.2 Newmark's Method (Newmark, 1959)	75
3.8 Remarks	76
3.8.1 Summary of Shape Function Properties.....	76
3.8.2 Sufficient Requirements for FEM Shape Functions.....	76
3.8.3 Recap of FEM Procedure	77
3.9 Review Questions.....	78

3.1 Introduction

As mentioned in Chapter 1, when using the finite element method (FEM) to solve mechanics problems governed by a set of partial differential equations, the problem domain is first discretized (in a proper manner) into a set of small elements. In each of these elements, the variation/profile/pattern of the displacements is assumed in simple forms to obtain element equations. The equations obtained for each element are then *assembled* together with adjoining elements to form the global finite element equation for the whole problem domain. Equations thus created for the global problem domain can be solved easily for the entire displacement field.

The above-mentioned FEM process does not seem to be a difficult task. However, upon close examination of the process, one would naturally ask a series of questions. How can one simply assume the profile of the solution of the displacement in any simple form? How can one ensure that the governing partial differential equations will be satisfied by the assumed displacement? What should one do when using the assumed profile of the displacement to determine the final displacement field? Yes, one can just simply assume the profile of the solution of the displacements, but a principle has to be followed in order to obtain discretized system equations that can be solved routinely for the final displacement field. The use of such a principle guarantees the *best* (may not be exact) satisfaction of the governing system equation under certain conditions. The following details one of the most important principles, which will be employed to establish the FEM equations for mechanics problems of solids and structures.

Perhaps, for a beginner, full understanding of the details of the equations in this chapter may prove to be a challenging task. It is thus advised that the novice reader just understands the basic concepts involved without digging too much into the equations. It is then recommended to review this chapter again, after going through subsequent chapters, with examples to fully understand the equations. Advanced readers are referred to the FEM handbook (Kardestuncer, 1987) for more complete topics related to FEM.

3.2 Strong and weak forms: problem formulation

The partial differential system equations developed in Chapter 2, such as Eqs. (2.20) and (2.21) with boundary conditions, Eqs. (2.22) and (2.23), are *strong forms* of the governing system of equations for solid mechanics problems. The strong form, in contrast to a *weak form*, requires *strong* continuity on the dependent field variables (the displacements u , v , and w in this case). By “strong,” we mean that the functions used to approximate the field variables have to be differentiable up to the order of the partial differential equations (the strong form of the system equations). Obtaining the exact solution for a strong form of the system equation is usually very difficult for practical engineering problems. The finite difference method (FDM) can be used to solve system equations of the strong form to obtain an approximated solution. However, the method usually works well only for problems with simple and regular geometry and boundary conditions.

A weak form of the system equations is usually created using one of the following widely used methods:

- Energy principles (see, for example, Washizu, 1974; Reddy, 1984; Liu, 2009).
- Weighted residual methods (see, for example, Crandall, 1956; Finlayson and Scriven, 1966; Finlayson, 1972; Zienkiewicz and Taylor, 2000; Liu, 2009; Liu and Zhang, 2013).

The energy principle can be categorized as a special form of the variational principle which is particularly suited to problems of the mechanics of solids and structures. The weighted residual method is a more

general mathematical tool applicable, in principle, for solving all kinds of partial differential equations. Both methods are very easy to understand and apply. This book will demonstrate both methods for creating FEM equations. An energy principle will be used for mechanics problems of solids and structures, and the weighted residual method will be used for formulating the heat transfer problems. It is also equally applicable to use the energy principle for heat transfer problems, and the weighted residual method for solid mechanics problems, and the procedure is very much the same. A comprehensive discussion on the differences and terminologies on energy methods, variation methods, functional analysis approaches, weak form methods, and weakened weak form methods can be found in the mesh-free book by Liu (2009), and the recent book on the smoothed point interpolation method (S-PIM) by Liu and Zhang (2013).

The weak form is often an integral form and requires a weaker continuity on the field variables. Due to the weaker requirement on the field variables, and the integral operation, a formulation based on a weak form usually produces a set of discretized system equations that give much more stable (and hence reliable) and accurate results, especially for problems of complex geometry. Hence, the weak form is preferred by many for obtaining an approximate solution. The FEM is a typical example of successfully using weak form formulations. Using the weak form usually leads to a set of well-behaved algebraic system equations, if the problem domain is discretized properly into elements. As the problem domain can be discretized into different types of elements, the FEM can be applied for many practical engineering problems with most kinds of complex geometry and boundary conditions.

In the following section, Hamilton's principle, which is one of the most powerful energy principles, is introduced for FEM formulation of problems of mechanics of solids and structures. Hamilton's principle is chosen because it is simple and can be used for dynamic problems. The approach adopted in this book is to directly work out the dynamic system equations, after which the static system equations can be easily obtained by simply dropping out the dynamic terms. This can be done because of the simple fact that the dynamic system equations are the general system equations, and the static case can be considered to be just a special case of the dynamic equations.

3.3 Hamilton's principle: A weak formulation

3.3.1 Hamilton's principle

Hamilton's principle is a simple yet powerful tool that can be used to derive discretized dynamic system equations. It states simply that,

*Of all the **admissible** time histories of displacement the most accurate solution makes the Lagrangian functional stationary.*

An *admissible* displacement must satisfy the following conditions:

- a. Compatibility equations
- b. Essential or the kinematic boundary conditions
- c. Conditions at initial (t_1) and final time (t_2).

Condition (a) ensures that the displacements are compatible (continuous) in the problem domain. As will be seen in Chapter 11, there are situations when incompatibility can occur at the edges between elements. Condition (b) ensures that the displacement constraints are satisfied; and condition (c) requires the displacement history to satisfy the constraints at the initial and final times.

Mathematically, Hamilton's principle states:

$$\delta \int_{t_1}^{t_2} L dt = 0 \quad (3.1)$$

The Langrangian functional, L , is obtained using a set of admissible time histories of displacements, and it consists of

$$L = \Pi - T - W_f \quad (3.2)$$

where T is the kinetic energy, Π is the potential energy (for our purposes, it is the elastic strain energy), and W_f is the work done by the external forces. It is clear that the Langrangian functional L is some kind of potential energy of the entire system, and thus the kinetic energy and the work done by the external forces results in deduction in potential energy.

The kinetic energy of the entire problem domain is defined in the integral form

$$T = \frac{1}{2} \int_V \rho \dot{\mathbf{U}}^T \dot{\mathbf{U}} dV \quad (3.3)$$

where V represents the whole volume of the solid, and \mathbf{U} is the set of admissible time histories of displacements.

The strain energy in the entire domain of elastic solids and structures can be expressed as

$$\Pi = \frac{1}{2} \int_V \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \frac{1}{2} \int_V \boldsymbol{\varepsilon}^T \mathbf{c} \boldsymbol{\varepsilon} dV \quad (3.4)$$

where $\boldsymbol{\varepsilon}$ are the strains obtained using the set of admissible time histories of displacements. The work done by the external forces over the set of admissible time histories of displacements can be obtained by

$$W_f = \int_V \mathbf{U}^T \mathbf{f}_b dV + \int_{S_f} \mathbf{U}^T \mathbf{f}_s dS_f \quad (3.5)$$

where S_f represents the surface of the solid on which the surface forces are prescribed (see Figure 2.2).

Hamilton's principle allows one to simply assume any set of displacements, as long as it satisfies the three admissible conditions. The assumed set of displacements will not usually satisfy the strong form of governing system equations unless we are extremely lucky, or the problem is extremely simple and we know the exact solution (in this case we do not need to solve the problem). Application of Hamilton's principle will conveniently guarantee a combination of this assumed set of displacements to produce the most accurate (in the energy norm measure) solution for the system that is governed by the strong form of the system equations.

The power of Hamilton's principle (or any other variational principle) is that it provides freedom of choice, opportunity, and possibility. For practical engineering problems, one usually does not have to pursue the exact solution, which in most cases is usually unobtainable, because we now have a choice of

quite conveniently obtaining a good approximation using Hamilton's principle, by assuming the likely form, pattern or *shape* of the solutions. Hamilton's principle thus provides the foundation for the FEMs. Furthermore, the simplicity of Hamilton's principle (or any other energy principle) manifests itself in the use of scalar energy quantities. Engineers and scientists like working with scalar quantities when it comes to numerical calculations, as they do not need to worry about the direction. All the mathematical tools required to derive final discrete system equations are basic operations of integration, differentiation, and variation, all of which are standard linear operations. Another plus point of Hamilton's principle is that the final discrete system equations produced are usually a set of linear algebraic equations that can be solved using conventional methods and standard computational routines. The following demonstrates how the finite element equations can be established using Hamilton's principle and its simple operations.

3.3.2 Minimum total potential energy principle

For static problems, Hamilton's principle reduces to the well-known minimum total potential energy principle, which may be stated as,

*Of all the **admissible** displacements the most accurate solution makes the total potential energy minimum.*

An admissible displacement must satisfy the following conditions:

- a. Compatibility equations
- b. Essential or the kinematic boundary conditions.

Mathematically, the “most” accurate solution is found using

$$\delta \Pi_t = 0 \quad (3.1a)$$

The total potential energy, Π_t , is obtained using the assume admissible displacement, and it consists of

$$\Pi_t = \Pi - W_f \quad (3.2a)$$

It can be proven for stable materials of a well-posed solid mechanics problem that the stationary point given in (3.1a) is a minimum point of the total potential energy for the solid (see, e.g., [Liu, 2009](#)).

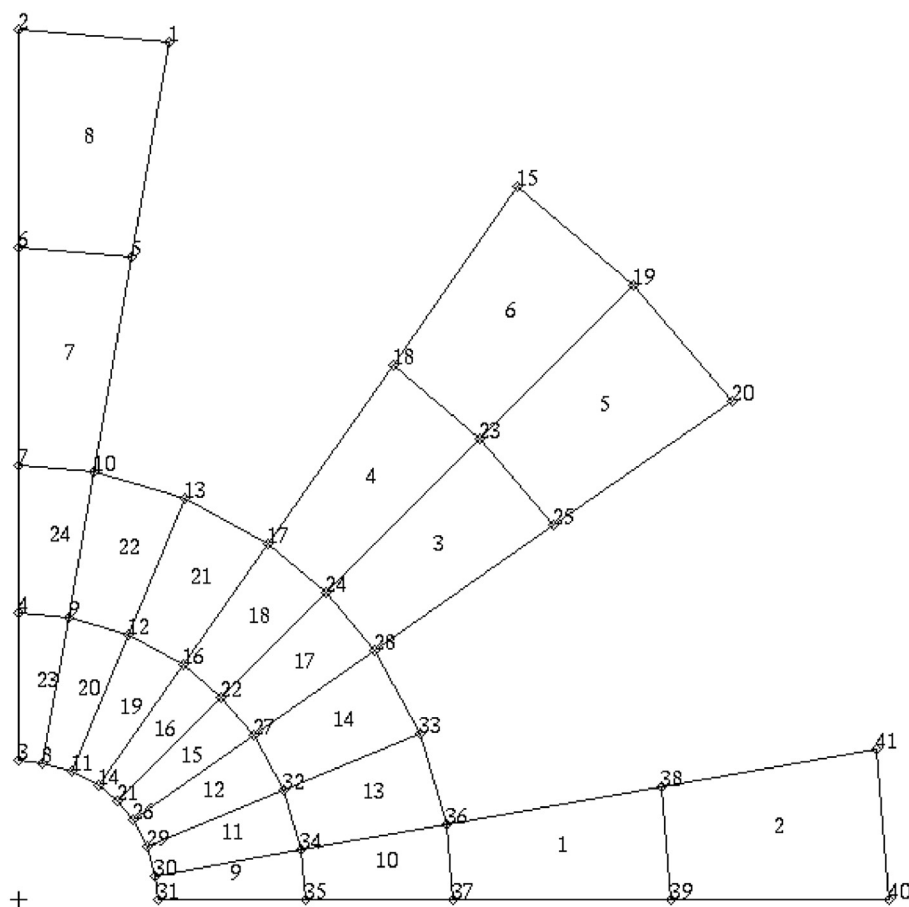
3.4 FEM procedure

The standard FEM procedure can be briefly summarized in the following sections.

3.4.1 Domain discretization

The solid or structure is divided into N_e elements with a set of N_n nodes. The procedure is often called *meshing*, which is usually performed using so-called pre-processors. This is especially true for complex geometries. [Figure 3.1](#) shows an example of a mesh for a two-dimensional solid.

The pre-processor generates unique numbers for all the elements and nodes for the solid or structure in a proper manner. An element is formed by connecting its nodes in a pre-defined consistent fashion to

**FIGURE 3.1**

Example of a mesh with elements and node properly numbered.

create the *connectivity* of the element. All the elements together form the entire domain of the problem without any gap or overlapping. It is possible to have different types of elements with different numbers of nodes, as long as they are *compatible* (no gaps and overlapping; the admissible condition (a) required by Hamilton's principle) on the boundaries between different elements. In fact, meshes of mixed triangular and quadrilateral elements are frequently used to overcome the difficulties of creating preferable pure quadrilateral elements. The density of the mesh depends upon the accuracy requirement of the analysis and the computational resources available. Generally, a finer mesh will yield results that are more accurate, but will increase the computational cost. As such, the mesh is usually not uniform, with a finer mesh being used in the areas where the displacement gradient is larger or where the accuracy is critical to the analysis. The purpose of the domain discretization is to make it easier in assuming the pattern of the displacement field.

3.4.2 Displacement interpolation

The FEM formulation has to be based on a coordinate system. In formulating FEM equations for elements, it is often convenient to use a *local coordinate system* that is defined for an element in reference to the global coordination system that is usually defined for the entire structure, as shown in [Figure 3.4](#). Based on the *local coordinate system* attached on an element, the displacement within the element is now assumed simply by polynomial interpolation using the displacements at its nodes (or *nodal displacements*) as

$$\mathbf{U}^h(x, y, z) = \sum_{i=1}^{n_d} \mathbf{N}_i(x, y, z) \mathbf{d}_i = \mathbf{N}(x, y, z) \mathbf{d}_e \quad (3.6)$$

where the superscript h stands for approximation, n_d is the number of nodes forming the element, and \mathbf{d}_i is the nodal displacement at the i th node, which is the unknown the analyst wants to find out, and can be expressed in a general form of

$$\mathbf{d}_i = \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n_f} \end{Bmatrix} \begin{matrix} \rightarrow \text{displacement component 1} \\ \rightarrow \text{displacement component 2} \\ \vdots \\ \rightarrow \text{displacement component } n_f \end{matrix} \quad (3.7)$$

where n_f is the number of Degrees Of Freedom (DOF) at a node. For 3D solids, $n_f=3$, and

$$\mathbf{d}_i = \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \begin{matrix} \rightarrow \text{displacement in the } x\text{-direction} \\ \rightarrow \text{displacement in the } y\text{-direction} \\ \rightarrow \text{displacement in the } z\text{-direction} \end{matrix} \quad (3.8)$$

Note that the displacement components can also consist of rotations for structures of beams and plates. The vector \mathbf{d}_e in Eq. (3.6) is the displacement vector for the entire element, and has the form of

$$\mathbf{d}_e = \begin{Bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_{n_d} \end{Bmatrix} \begin{matrix} \rightarrow \text{displacements at node 1} \\ \rightarrow \text{displacements at node 2} \\ \vdots \\ \rightarrow \text{displacements at node } n_d \end{matrix} \quad (3.9)$$

Therefore, the total DOF for the entire element is $n_d \times n_f$.

In Eq. (3.6), \mathbf{N} is a matrix of *shape functions* for the nodes in the element, which are pre-defined to assume the shapes of the displacement variations with respect to the coordinates. It has the general form of

$$\mathbf{N}(x, y, z) = [\mathbf{N}_1(x, y, z) \quad \mathbf{N}_2(x, y, z) \quad \cdots \quad \mathbf{N}_{nd}(x, y, z)] \quad (3.10)$$

$$\begin{array}{ccccccc} \downarrow & & \downarrow & & \cdots & & \downarrow \\ \text{for node 1} & \text{for node 2} & \cdots & & \text{for node } n_d \end{array}$$

where \mathbf{N}_i is a sub-matrix of shape functions for displacement components, which is arranged as

$$\mathbf{N}_i = \begin{bmatrix} N_{i1} & 0 & 0 & 0 \\ 0 & N_{i2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & N_{inf} \end{bmatrix} \quad (3.11)$$

where N_{ik} is the shape function for the k th displacement component, DOF at the i th node. For 3D solids, $n_f = 3$, and often we use $N_{i1} = N_{i2} = N_{i3} = N_i$. Note that it is not always necessary to use the same shape function for all the displacement components at a node. For example, we often use different shape functions for translational and rotational displacements.

Note that this approach of assuming the displacements is often called the *displacement method*. Most of the commercially available FEM packages use this approach. There are, however, FEM approaches that assume the stresses instead, but they will not be covered in this book.

3.4.3 Standard procedure for constructing shape functions

Consider an element with n_d nodes at \mathbf{x}_i ; ($i = 1, 2, \dots, n_d$), where $\mathbf{x}^T = \{x\}$ for one-dimensional problems, $\mathbf{x}^T = \{x \ y\}$ for two-dimensional problems, and $\mathbf{x}^T = \{x \ y \ z\}$ for three-dimensional problems. We should have n_d shape functions for each displacement component for an element. In the following, we consider only one displacement component in the explanation of the standard procedure for constructing the shape functions. The standard procedure is applicable for any other displacement components. First, the displacement component is approximated in the form of a linear combination of n_d linearly-independent basis functions $p_i(\mathbf{x})$, i.e.

$$u^h(\mathbf{x}) = \sum_{i=1}^{n_d} p_i(\mathbf{x}) \alpha_i = \mathbf{p}^T(\mathbf{x}) \boldsymbol{\alpha} \quad (3.12)$$

where u^h is the approximation of the displacement component, $p_i(\mathbf{x})$ is the basis function of monomials in the space coordinates \mathbf{x} , and α_i is the coefficient for the monomial $p_i(\mathbf{x})$. Vector $\boldsymbol{\alpha}$ is defined as

$$\boldsymbol{\alpha}^T = \{\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \cdots \quad \alpha_{n_d}\} \quad (3.13)$$

The $p_i(\mathbf{x})$ in Eq. (3.12) is built with n_d terms of one-dimensional monomials; based on the Pascal's triangle shown in Figure 3.2 for two-dimensional problems; or the well-known Pascal's pyramid shown in Figure 3.3 for three-dimensional problems. A basis of complete order of p in the one-dimensional domain has the form

$$\mathbf{p}^T(x) = \{1 \quad x \quad x^2 \quad x^3 \quad x^4 \quad \dots \quad x^p\} \quad (3.14)$$

A basis of complete order of p in the two-dimensional domain is provided by

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x, y) = \{1 \quad x \quad y \quad xy \quad x^2 \quad y^2 \quad \dots \quad x^p \quad y^p\} \quad (3.15)$$

and that in the three-dimensional domain can be written as

$$\begin{aligned} \mathbf{p}^T(\mathbf{x}) &= \mathbf{p}^T(x, y, z) \\ &= \{1 \quad x \quad y \quad z \quad xy \quad yz \quad zx \quad x^2 \quad y^2 \quad z^2 \quad \dots \quad x^p \quad y^p \quad z^p\} \end{aligned} \quad (3.16)$$

As a general rule, the n_d terms of $p_i(\mathbf{x})$ used in the basis should be selected from the constant term to higher orders symmetrically from the Pascal triangle shown in Figure 3.2 or Figure 3.3. Some higher order terms can be selectively included in the polynomial basis if there is a need in specific circumstances.

The coefficients α_i in Eq. (3.12) can be determined by enforcing the displacements calculated using Eq. (3.12) to be equal to the nodal displacements at the n_d nodes of the element. At node i we can have

$$d_i = \mathbf{p}^T(\mathbf{x}_i)\boldsymbol{\alpha} \quad i = 1, 2, 3, \dots, n_d \quad (3.17)$$

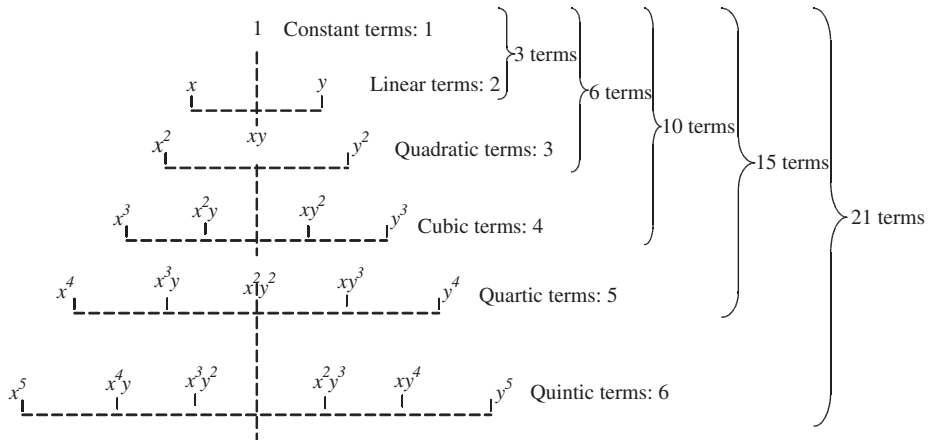


FIGURE 3.2

Pascal triangle of monomials. Two-dimensional case.

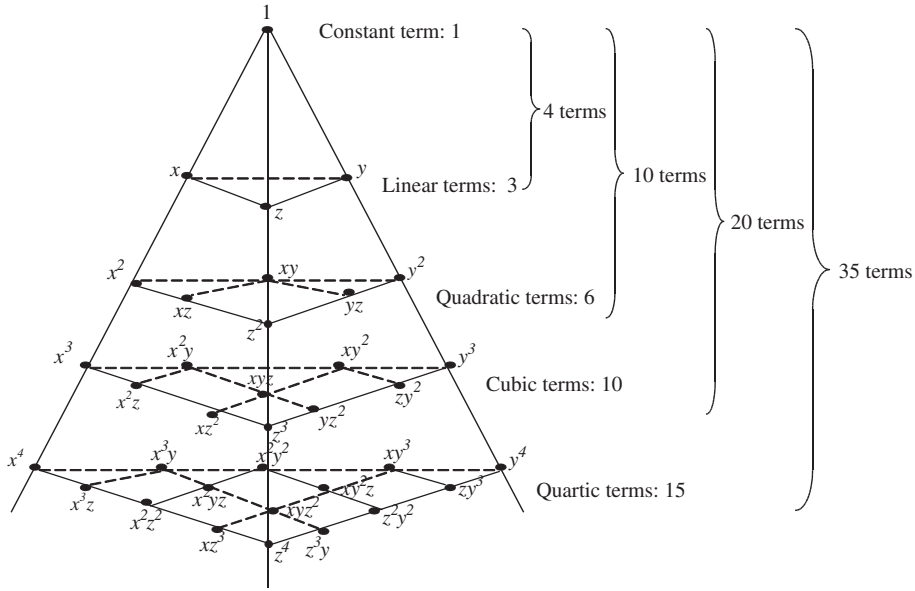


FIGURE 3.3

Pascal pyramid of monomials. Three-dimensional case.

where d_i is the nodal value of u^h at $\mathbf{x} = \mathbf{x}_i$. Equation (3.17) can be written in the following matrix form:

$$\mathbf{d}_e = \mathbf{P}\boldsymbol{\alpha} \quad (3.18)$$

where \mathbf{d}_e is the vector that includes the values of the displacement component at all the n_d nodes in the element:

$$\mathbf{d}_e = \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n_d} \end{Bmatrix} \quad (3.19)$$

and \mathbf{P} is given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \mathbf{p}^T(\mathbf{x}_2) \\ \vdots \\ \mathbf{p}^T(\mathbf{x}_{n_d}) \end{bmatrix} \quad (3.20)$$

which is called the *moment matrix*. The expanded form of \mathbf{P} is

$$\mathbf{P} = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_{n_d}(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_{n_d}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_{n_d}) & p_2(\mathbf{x}_{n_d}) & \cdots & p_{n_d}(\mathbf{x}_{n_d}) \end{bmatrix} \quad (3.21)$$

For two-dimensional polynomial basis functions, we may have

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 & x_1^2 y_1 & x_1 y_1^2 & x_1^3 & \cdots \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 & x_2^2 y_2 & x_2 y_2^2 & x_2^3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n_d} & y_{n_d} & x_{n_d} y_{n_d} & x_{n_d}^2 & y_{n_d}^2 & x_{n_d}^2 y_{n_d} & x_{n_d} y_{n_d}^2 & x_{n_d}^3 & \cdots \end{bmatrix} \quad (3.22)$$

Using Eq. (3.18), and assuming that the inverse of the moment matrix \mathbf{P} exists, we can then have

$$\boldsymbol{\alpha} = \mathbf{P}^{-1} \mathbf{d}_e \quad (3.23)$$

Substituting Eq. (3.23) into Eq. (3.12), we then obtain

$$u^h(\mathbf{x}) = \sum_{i=1}^{n_d} N_i(\mathbf{x}) d_i \quad (3.24)$$

or in matrix form

$$u^h(\mathbf{x}) = \mathbf{N}(\mathbf{x}) \mathbf{d}_e \quad (3.25)$$

where $\mathbf{N}(\mathbf{x})$ is a matrix of shape functions $N_i(\mathbf{x})$ defined by

$$\begin{aligned} \mathbf{N}(\mathbf{x}) &= \mathbf{p}^T(\mathbf{x}) \mathbf{P}^{-1} = \begin{bmatrix} \underbrace{\mathbf{p}^T(\mathbf{x}) \mathbf{P}_1^{-1}}_{N_1(\mathbf{x})} & \underbrace{\mathbf{p}^T(\mathbf{x}) \mathbf{P}_2^{-1}}_{N_2(\mathbf{x})} & \cdots & \underbrace{\mathbf{p}^T(\mathbf{x}) \mathbf{P}_n^{-1}}_{N_n(\mathbf{x})} \end{bmatrix} \\ &= [N_1(\mathbf{x}) \quad N_2(\mathbf{x}) \quad \cdots \quad N_n(\mathbf{x})] \end{aligned} \quad (3.26)$$

where \mathbf{P}_i^{-1} is the i th column of matrix \mathbf{P}^{-1} , and

$$N_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{P}_i^{-1} \quad (3.27)$$

The derivatives of the shape functions can be obtained very easily, as all the functions involved are polynomials. The l th derivative of the shape functions is simply given by

$$N_i^{(l)}(\mathbf{x}) = [\mathbf{p}^{(l)}(\mathbf{x})]^T \mathbf{P}_i^{-1} \quad (3.28)$$

3.4.3.1 On the inverse of the moment matrix

In obtaining Eq. (3.23), we have assumed the existence of the inverse of \mathbf{P} . There could be cases where \mathbf{P}^{-1} does not exist, and the construction of shape functions will fail. The existence of \mathbf{P}^{-1} depends upon (1) the basis function used, and (2) the nodal distribution of the element. The basis functions have to be chosen first from a linearly-independent set of bases, and then the inclusion of the basis terms should be based on the nodal distribution in the element. The discussion in this direction is more involved, and interested readers are referred to a monograph on mesh-free methods by Liu (2009). In this book, we shall only discuss elements whose corresponding moment matrices are invertible.

3.4.3.2 On the compatibility of the shape functions

The other possible failure of the general procedure of shape function construction is the compatibility of the shape functions created. This is often the case for higher order elements created on physical coordinates. This is the main reason why most of the higher order elements are created in the so-called natural coordinate system (see topics on isoparametric elements). The issues related to the compatibility of element shape functions will be addressed in greater detail in Chapter 11. Note that the compatibility conditions can be relaxed if the S-PIM (Liu and Zhang, 2013) is used. This is because the S-PIM is based on the so-called weakened weak form, and the stability and convergence are ensured by the G-space theory.

3.4.3.3 On other means of construct shape functions

There are many other methods for creating shape functions which do not necessarily follow the standard procedure described above. Some of these often used *shortcut* methods will be discussed in later chapters, when we develop different types of elements. These shortcut methods need to make use of the properties of shape functions detailed in the next section.

3.4.4 Properties of the shape functions

PROPERTY 1

Reproduction property and consistency

The consistency of the shape function within the element depends upon the complete orders of the monomial $p_i(\mathbf{x})$ used in Eq. (3.12), and hence is also dependent upon the number of nodes of the element. If the complete order of monomial is k , the shape function is said to possess C^k consistency. To demonstrate, we consider a field given by

$$f(\mathbf{x}) = \sum_j^k p_j(\mathbf{x})\beta_j, \quad ; k \leq n_d \quad (3.29)$$

where $p_j(\mathbf{x})$ are monomials that are included in Eq. (3.12). Such a given field can always be written using Eq. (3.12) using all the basis terms, including those in Eq. (3.29):

$$f(\mathbf{x}) = \sum_j^{n_d} p_j(\mathbf{x})\beta_j = \mathbf{p}^T(\mathbf{x})\boldsymbol{\alpha} \quad (3.30)$$

where

$$\boldsymbol{\alpha}^T = [\beta_1 \quad \beta_2 \quad \dots \quad \beta_k \quad 0 \quad \dots \quad 0] \quad (3.31)$$

Using n nodes in the support domain of \mathbf{x} , we can obtain the vector of nodal function value \mathbf{d}_e as:

$$\mathbf{d}_e = \begin{Bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \\ f_{k+1} \\ \vdots \\ f_n \end{Bmatrix} = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_k(\mathbf{x}_1) & p_{k+1}(\mathbf{x}_1) & p_{n_d}(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_k(\mathbf{x}_2) & p_{k+1}(\mathbf{x}_2) & p_{n_d}(\mathbf{x}_2) \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ p_1(\mathbf{x}_k) & p_2(\mathbf{x}_k) & \cdots & p_k(\mathbf{x}_k) & p_{k+1}(\mathbf{x}_k) & p_{n_d}(\mathbf{x}_k) \\ p_1(\mathbf{x}_{k+1}) & p_2(\mathbf{x}_{k+1}) & \cdots & p_k(\mathbf{x}_{k+1}) & p_{k+1}(\mathbf{x}_{k+1}) & p_{n_d}(\mathbf{x}_{k+1}) \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ p_1(\mathbf{x}_{n_d}) & p_2(\mathbf{x}_{n_d}) & \cdots & p_k(\mathbf{x}_{n_d}) & p_{k+1}(\mathbf{x}_{n_d}) & p_{n_d}(\mathbf{x}_{n_d}) \end{bmatrix} \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \quad (3.32)$$

$$= \mathbf{P}\boldsymbol{\alpha}$$

Substituting Eq. (3.32) into Eq. (3.25), we have the approximation of

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{P}^{-1}\mathbf{d}_e = \mathbf{p}^T(\mathbf{x})\mathbf{P}^{-1}\mathbf{P}\boldsymbol{\alpha} = \mathbf{p}^T(\mathbf{x})\boldsymbol{\alpha} = \sum_j^k p_j(\mathbf{x})\alpha_j = f(\mathbf{x}) \quad (3.33)$$

which is exactly what is given in Eq. (3.30). This proves that any field given by Eq. (3.29) will be exactly reproduced in the element by the approximation using the shape functions, as long as the given field function is included in the basis functions used for constructing the shape functions. This feature of the shape function is in fact also very easy to understand by intuition: Any function given in the form of $f(\mathbf{x}) = \sum_j^k p_j(\mathbf{x})\beta_j$ can be produced exactly by letting $\alpha_j = \beta_j$ ($j = 1, 2, \dots, k$) and $\alpha_j = 0$ ($j = k+1, \dots, n_d$). This can always be done as long as the moment matrix \mathbf{P} is invertible so as to ensure the uniqueness of the solution for $\boldsymbol{\alpha}$.

The proof of the consistency of the shape function implies another important feature of the shape function: That is the *reproduction property*, which states that any function that appears in the basis can be reproduced exactly. This important property can be used for creating fields of special features. To ensure that the shape functions have linear consistency, all one needs to do is include the constant (unit) and linear monomials into the basis. We can make use of the feature of the shape function to compute accurate results for problems by including terms in the basis functions that are good approximations of the problem solution. The difference between consistency and reproduction is:

- Consistency depends upon the complete order of the basis functions
- Reproduction depends upon whatever is included in the basis functions.

PROPERTY 2

Linear independence

Shape functions are linearly-independent. This is because basis functions are of linear independence and \mathbf{P}^{-1} is assumed to exist. The existence of \mathbf{P}^{-1} implies that the shape functions are equivalent to the basis functions in the function space, as shown in Eq. (3.26). Because the basis functions are linearly-independent, the shape functions are hence also linearly-independent. Many FEM users do not pay much attention to this linear independence property; however, it is the foundation for the shape functions to have the delta function property stated below.

Since the elements are independently geometrical (it occupies its own domain), all the nodal shape functions for the entire FEM domain will be independent, which is essential to any FEM model.

PROPERTY 3

Delta function properties

$$N_i(\mathbf{x}_j) = \delta_{ij} = \begin{cases} 1 & i = j, j = 1, 2, \dots, n_d \\ 0 & i \neq j, i, j = 1, 2, \dots, n_d \end{cases} \quad (3.34)$$

where δ_{ij} is the delta function. The delta function property implies that the shape function N_i of an element should be unit at its *home node* i , and vanishes at the *remote nodes* $j = i$ of the element.

The delta function property can be proven easily as follows: because the shape functions $N_i(\mathbf{x})$ are linearly-independent, any vector of length n_d should be uniquely produced by linear combination of these n_d shape functions. Assume that the displacement at node i is $d_i \neq 0$ and the displacements at other nodes are zero, i.e.

$$\mathbf{d}_e = \{0 \ 0 \ \dots \ d_i \ \dots \ 0\}^T \quad (3.35)$$

This vector must be uniquely produced by linear combination of these n_d shape functions. Substituting the above equation into Eq. (3.24), we have at $\mathbf{x} = \mathbf{x}_j$, that

$$u^h(\mathbf{x}_j) = \sum_{k=1}^{n_d} N_k(\mathbf{x}_j) d_k = N_i(\mathbf{x}_j) d_i \quad (3.36)$$

and when $i=j$, we must have

$$u_i = d_i = N_i(\mathbf{x}_i)d_i \quad (3.37)$$

which (since $d_i \neq 0$) implies that

$$N_i(\mathbf{x}_i) = 1 \quad (3.38)$$

This proves the first row of Eq. (3.34). When $i \neq j$, we must have

$$u_j = 0 = N_i(\mathbf{x}_j)d_i \quad (3.39)$$

which (since again $d_i \neq 0$) requires

$$N_i(\mathbf{x}_j) = 0 \quad (3.40)$$

This proves the second row of Eq. (3.34). We can then conclude that the shape functions possess the delta function property, as depicted by Eq. (3.34). Note that there are elements, such as the thin beam and plate elements, whose shape functions may not possess the delta function property (see Section 5.2.1 for details).

PROPERTY 4

Partitions of unity property

Shape functions are partitions of unity:

$$\sum_{i=1}^{n_d} N_i(\mathbf{x}) = 1 \quad (3.41)$$

if the constant is included in the basis. This can be proven easily from the reproduction feature of the shape function. Let $u(\mathbf{x}) = c$, where c is an arbitrary non-zero constant; we should have

$$\mathbf{d}_e = \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n_d} \end{Bmatrix} = \begin{Bmatrix} c \\ c \\ \vdots \\ c \end{Bmatrix} \quad (3.42)$$

which implies the same constant displacement for all the nodes. Substituting the above equation into Eq. (3.24), we obtain

$$u(\mathbf{x}) = c \underset{\text{reproduction}}{=} u^h(\mathbf{x}) \underset{\text{approximation}}{=} \sum_{i=1}^{n_d} N_i(\mathbf{x})d_i = \sum_{i=1}^{n_d} N_i(\mathbf{x})c = c \sum_{i=1}^{n_d} N_i(\mathbf{x}) \quad (3.43)$$

which gives Eq. (3.41). This shows that the partitions of unity of the shape functions in the element allow a constant field or rigid body movement to be reproduced. Note that Eq. (3.41) does not require $0 \leq N_i(\mathbf{x}) \leq 1$.

PROPERTY 5

Linear field reproduction

If the first order monomial is included in the basis, the shape functions constructed reproduce the linear field, i.e.

$$\sum_{i=1}^{n_d} N_i(x) x_i = x \quad (3.44)$$

where x_i is the coordinate at node i (and similarly for y and z). This can be proven easily from the reproduction feature of the shape function in exactly the same manner for proving Property 4. Let $u(x) = x$, we should have

$$\mathbf{d}_e = \{x_1 \quad x_2 \quad \dots \quad x_{n_d}\}^T \quad (3.45)$$

Substituting the above equation into Eq. (3.24), we obtain

$$u^h(x) = x = \sum_1^{n_d} N_i(x) x_i \quad (3.46)$$

which is Eq. (3.44).

LEMMA 1

Condition for shape functions being partitions of unity

For a set of shape functions in the general form

$$N_i(\mathbf{x}) = c_{1i} + c_{2i}p_2(\mathbf{x}) + c_{3i}p_3(\mathbf{x}) + \dots + c_{n_d i}p_{n_d}(\mathbf{x}) \quad (3.47)$$

where $p_i(\mathbf{x})$ ($p_1(\mathbf{x}) = 1$, $i = 2, n_d$) is a set of independent base functions, the sufficient and necessary condition for this set of shape functions being partitions of unity is

$$\begin{aligned} C_1 &= 1 \\ C_2 &= C_3 = \dots = C_{n_d} = 0 \end{aligned} \quad (3.48)$$

where

$$C_k = \sum_{i=1}^{n_d} c_{ki} \quad (3.49)$$

PROOF

Using Eq. (3.47), the summation of the shape functions is

$$\begin{aligned} \sum_{i=1}^{n_d} N_i(\mathbf{x}) &= \sum_{i=1}^{n_d} c_{1i} + p_2(\mathbf{x}) \sum_{i=1}^{n_d} c_{2i} + p_3(\mathbf{x}) \sum_{i=1}^{n_d} c_{3i} + \cdots + p_{n_d}(\mathbf{x}) \sum_{i=1}^{n_d} c_{n_d i} \\ &= \underbrace{C_1}_1 + \underbrace{C_2}_0 p_2(\mathbf{x}) + \underbrace{C_2}_0 p_3(\mathbf{x}) + \cdots + \underbrace{C_{n_d}}_0 p_{n_d}(\mathbf{x}) = 0 \end{aligned} \quad (3.50)$$

which proves the sufficient condition. To prove the necessary condition, we argue that, to have the partitions of unity, we have

$$\sum_{i=1}^{n_d} N_i(\mathbf{x}) = C_1 + C_2 p_2(\mathbf{x}) + C_2 p_3(\mathbf{x}) + \cdots + C_{n_d} p_{n_d}(\mathbf{x}) = 1 \quad (3.51)$$

or

$$(C_1 - 1) + C_2 p_2(\mathbf{x}) + C_2 p_3(\mathbf{x}) + \cdots + C_{n_d} p_{n_d}(\mathbf{x}) = 0 \quad (3.52)$$

Because $p_i(\mathbf{x})$ ($p_1(\mathbf{x}) = 1$, $i = 2, n_d$) is a set of independent base functions, the necessary condition for Eq. (3.52) to be satisfied is Eq. (3.48). \square

LEMMA 2***Condition for shape functions being partitions of unity***

Any set of n_d shape functions will automatically satisfy the partitions of unity property if it satisfies:

- *Condition 1:* It is given in a linear combination of the same linearly-independent set of n_d basis functions that contain the constant basis, and the moment matrix defined by Eq. (3.21) is of full rank.
- *Condition 2:* It possesses the delta function property.

PROOF

From Eq. (3.26), we can see that all the n_d shape functions are formed via a combination of the same basis function $p_i(\mathbf{x})$ ($i = 1, 2, \dots, n_d$). This feature, together with the delta function property, can ensure the property of partitions of unity. To prove this, we write a set of shape functions in the general form

$$N_i(\mathbf{x}) = c_{1i} + c_{2i} p_2(\mathbf{x}) + c_{3i} p_3(\mathbf{x}) + \cdots + c_{n_d i} p_{n_d}(\mathbf{x}) \quad (3.53)$$

where we ensured the inclusion of the constant basis of $p_1(\mathbf{x}) = 1$. The other basis function $p_i(\mathbf{x})$ ($i = 2, \dots, n_d$) in Eq. (3.53) can be monomials or any other type of basis functions as long as all the basis functions (including $p_1(\mathbf{x})$) are linearly-independent.

From the Condition 2, the shape functions possess a delta function property that leads to

$$\sum_{i=1}^{n_d} N_i(\mathbf{x}_j) = 1 \quad \text{for } j = 1, 2, \dots, n_d \quad (3.54)$$

Substituting Eq. (3.53) into the previous equations, we have

$$\sum_{i=1}^{n_d} c_{1i} + p_2(\mathbf{x}_j) \sum_{i=1}^{n_d} c_{2i} + p_3(\mathbf{x}_j) \sum_{i=1}^{n_d} c_{3i} + \dots + p_{n_d}(\mathbf{x}_j) \sum_{i=1}^{n_d} c_{n_d i} = 1 \quad (3.55)$$

for $j = 1, 2, \dots, n_d$

or

$$C_1 + p_2(\mathbf{x}_j)C_2 + p_3(\mathbf{x}_j)C_3 + \dots + p_{n_d}(\mathbf{x}_j)C_{n_d} = 1 \quad \text{for } j = 1, 2, \dots, n_d \quad (3.56)$$

Expanding Eq. (3.56) gives

$$\begin{aligned} C_1 + p_2(\mathbf{x}_1)C_2 + p_3(\mathbf{x}_1)C_3 + \dots + p_{n_d}(\mathbf{x}_1)C_{n_d} &= 1 \\ C_1 + p_2(\mathbf{x}_2)C_2 + p_3(\mathbf{x}_2)C_3 + \dots + p_{n_d}(\mathbf{x}_2)C_{n_d} &= 1 \\ \vdots & \\ C_1 + p_2(\mathbf{x}_{n_d})C_2 + p_3(\mathbf{x}_{n_d})C_3 + \dots + p_{n_d}(\mathbf{x}_{n_d})C_{n_d} &= 1 \end{aligned} \quad (3.57)$$

or in the matrix form

$$\begin{bmatrix} 1 & p_2(\mathbf{x}_2) & p_3(\mathbf{x}_2) & \dots & p_{n_d}(\mathbf{x}_2) \\ 1 & p_2(\mathbf{x}_3) & p_3(\mathbf{x}_3) & \dots & p_{n_d}(\mathbf{x}_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & p_2(\mathbf{x}_{n_d}) & p_3(\mathbf{x}_{n_d}) & \dots & p_{n_d}(\mathbf{x}_{n_d}) \end{bmatrix} \begin{Bmatrix} C_1 - 1 \\ C_2 \\ C_3 \\ \vdots \\ C_{n_d} \end{Bmatrix} = 0 \quad (3.58)$$

Note that the coefficient matrix of Eq. (3.58) is the moment matrix that has a full rank (Condition 1); we then have

$$\begin{aligned} C_1 &= 1 \\ C_2 &= C_3 = \dots = C_{n_d} = 0 \end{aligned} \quad (3.59)$$

The use of Lemma 1 proves the partitions of the unity property of shape functions. \square

LEMMA 3***Condition for shape functions being linear field reproduction***

Any set of n_d shape functions will automatically satisfy the linear reproduction property, if it satisfies

- *Condition 1:* It is given in a linear combination of the same linearly-independent set of n_d basis functions that contain the linear basis function, and the moment matrix defined by Eq. (3.21) is of full rank.
- *Condition 2:* It possesses the delta function property.

To prove this, we write a set of shape functions in the following general form of

$$N_i(\mathbf{x}) = c_{1i}p_1(\mathbf{x}) + c_{2i}x + c_{3i}p_3(\mathbf{x}) + \cdots + c_{n_d i}p_{n_d}(\mathbf{x}) \quad (3.60)$$

where we ensure inclusion of the complete linear basis functions of $p_2(\mathbf{x}) = x$. The other basis function $p_i(\mathbf{x})$ ($i = 1, 3, \dots, n_d$) in Eq. (3.53) can be monomials or any other type of basis function as long as all the basis functions are linearly-independent.

Consider a linear field of $u(\mathbf{x}) = x$, we should have the nodal vector as follows:

$$\mathbf{d}_e = \{x_1 \ x_2 \ \dots \ x_{n_d}\}^T \quad (3.61)$$

Substituting the above equation into Eq. (3.24), we obtain

$$\begin{aligned} u^h(\mathbf{x}) &= \sum_{i=1}^{n_d} N_i(\mathbf{x})x_i \\ &= \sum_{i=1}^{n_d} [c_{1i}p_1(\mathbf{x}) + c_{2i}x + c_{3i}p_3(\mathbf{x}) + \cdots + c_{n_d i}p_{n_d}(\mathbf{x})]x_i \\ &= \sum_{i=1}^{n_d} c_{1i}p_1(\mathbf{x})x_i + \sum_{i=1}^{n_d} c_{2i}xx_i + \sum_{i=1}^{n_d} c_{3i}p_3(\mathbf{x})x_i + \cdots + \sum_{i=1}^{n_d} c_{n_d i}p_{n_d}(\mathbf{x})x_i \\ &= \sum_{i=1}^{n_d} c_{1i}p_1(\mathbf{x})x_i + x \sum_{i=1}^{n_d} c_{2i}x_i + p_3(\mathbf{x}) \sum_{i=1}^{n_d} c_{3i}x_i + \cdots + p_{n_d}(\mathbf{x}) \sum_{i=1}^{n_d} c_{n_d i}x_i \\ &= p_1(\mathbf{x})C_{x1} + xC_{x2} + p_3(\mathbf{x})C_{x3} + \cdots + p_{n_d}(\mathbf{x})C_{xn_d} \end{aligned} \quad (3.62)$$

At the n_d nodes of the element, we have n_d equations:

$$\begin{aligned} u^h(\mathbf{x}_1) &= p_1(\mathbf{x})C_{x1} + x_1C_{x2} + p_3(\mathbf{x}_1)C_{x3} + \cdots + p_{n_d}(\mathbf{x}_1)C_{xn_d} \\ u^h(\mathbf{x}_2) &= p_1(\mathbf{x})C_{x1} + x_2C_{x2} + p_3(\mathbf{x}_2)C_{x3} + \cdots + p_{n_d}(\mathbf{x}_2)C_{xn_d} \\ &\vdots \\ u^h(\mathbf{x}_{n_d}) &= p_1(\mathbf{x})C_{x1} + x_{n_d}C_{x2} + p_3(\mathbf{x}_{n_d})C_{x3} + \cdots + p_{n_d}(\mathbf{x}_{n_d})C_{xn_d} \end{aligned} \quad (3.63)$$

Using the delta function property of the shape functions, we have

$$\begin{aligned}
 u^h(\mathbf{x}_j) &= \sum_{i=1}^{n_d} N_i(\mathbf{x}_j) x_i \\
 &= \underbrace{N_1(\mathbf{x}_j)}_0 x_1 + \underbrace{N_2(\mathbf{x}_j)}_0 x_2 + \cdots + \underbrace{N_j(\mathbf{x}_j)}_1 x_j + \cdots + \underbrace{N_{n_d}(\mathbf{x}_j)}_0 x_{n_d} \\
 &= x_j
 \end{aligned} \tag{3.64}$$

Hence, Eq. (3.63) becomes

$$\begin{aligned}
 0 &= p_1(\mathbf{x}) C_{x1} + x_1 (C_{x2} - 1) + p_3(\mathbf{x}_2) C_{x3} + \cdots + p_{n_d}(\mathbf{x}_2) C_{xn_d} \\
 0 &= p_1(\mathbf{x}_3) C_{x1} + x_2 (C_{x2} - 1) + p_3(\mathbf{x}_3) C_{x3} + \cdots + p_{n_d}(\mathbf{x}_3) C_{xn_d} \\
 &\vdots \\
 0 &= p_1(\mathbf{x}_{n_d}) C_{x1} + x_{n_d} (C_{x2} - 1) + p_3(\mathbf{x}_{n_d}) C_{x3} + \cdots + p_{n_d}(\mathbf{x}_{n_d}) C_{xn_d}
 \end{aligned} \tag{3.65}$$

Or in matrix form,

$$\begin{bmatrix} p_1(\mathbf{x}_1) & x_1 & p_3(\mathbf{x}_1) & \cdots & p_{n_d}(\mathbf{x}_1) \\ p_1(\mathbf{x}_1) & x_2 & p_3(\mathbf{x}_2) & \cdots & p_{n_d}(\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_1) & x_{n_d} & p_3(\mathbf{x}_{n_d}) & \cdots & p_{n_d}(\mathbf{x}_{n_d}) \end{bmatrix} \begin{Bmatrix} C_{x1} \\ C_{x2} - 1 \\ C_{x3} \\ \vdots \\ C_{xn_d} \end{Bmatrix} = 0 \tag{3.66}$$

Note that the coefficient matrix of Eq. (3.66) is the moment matrix that has a full rank. We thus have

$$\begin{aligned}
 C_{x1} &= 0 \\
 (C_{x2} - 1) &= 0 \\
 C_{x3} &= \cdots = C_{n_d} = 0
 \end{aligned} \tag{3.67}$$

Substituting the previous equation back into Eq. (3.62), we obtain

$$u^h(x) = \sum_{i=1}^{n_d} N_i(x) x_i = 0 \tag{3.68}$$

This proves the property of linear field reproduction.

The delta function property (Property 3) ensures convenient imposition of the essential boundary conditions (admissible condition (b) required by Hamilton's principle), because the nodal displacement at a node is independent of that at any other nodes. The constraints can often be written in the form of a so-called *Single Point Constraint* (SPC). If the displacement at a node is fixed, all one needs to do is to remove corresponding rows and columns without affecting the other rows and columns.

The proof of Property 4 gives a convenient way to confirm the partitions of unity property of shape functions. As long as the constant (unit) basis is included in the basis functions, the shape functions constructed are partitions of unity. Properties 4 and 5 are essential for the FEM to pass the standard *patch test*, used for decades in the FEM for validating the elements. In the standard patch test, the patch is meshed with a number of elements, with at least one interior node. Linear displacements are then enforced on the boundary (edges) of the patch. A successful patch test requires the FEM solution to produce the linear displacement (or constant strain) field at any interior node. Therefore, the property of reproduction of a linear field of shape function provides the foundation for passing the patch test. Note that the property of reproducing the linear field of the shape function does not guarantee successful patch tests, as there could be other sources of numerical error, such as numerical integration, which can cause failure.

Lemma 1 seems to be redundant, since we already have Property 4. However, Lemma 1 is a very convenient property to use for checking the property of partitions of unity of shape functions that are constructed using other *shortcut* methods, rather than the standard procedure. Using Lemma 1, one only needs to make sure whether the shape functions satisfy Eq. (3.48).

Lemma 2 is another very convenient property to use for checking the property of partitions of unity of shape functions. Using Lemma 2, we only need to make sure that the constructed n_d shape functions are of the delta function property, and they are linear combinations of the same n_d basis functions that are linearly-independent and contain the constant basis function. The conformation of full rank of the moment matrix of the basis functions can sometimes be difficult. In this book, we usually assume that the rank is full for the normal elements, as long as the basis functions are linearly-independent. In usual situations, one will not be able to obtain the shape functions if the rank of the moment matrix is not full. If we somehow obtained the shape functions successfully, we can usually be sure that the rank of the corresponding moment matrix is full.

Lemma 3 is a very convenient property to use for checking the property of linear field reproduction of shape functions. Using Lemma 3, we only need to make sure that the constructed n_d shape functions are of the delta function property, and they are linear combinations of the same n_d basis functions that are linearly-independent and contain the linear basis function.

3.4.5 Formulation of finite element equations in local coordinate system

Once the shape functions are constructed, the finite element (FE) equation for an element can be formulated using the following process. By substituting the interpolation of the nodes, Eq. (3.6), and the strain–displacement equation, say Eq. (2.5), into the strain energy term (Eq. (3.4)), we have

$$\Pi = \frac{1}{2} \int_{V_e} \boldsymbol{\epsilon}^T \mathbf{c} \boldsymbol{\epsilon} \, dV = \frac{1}{2} \int_{V_e} \mathbf{d}_e^T \mathbf{B}^T \underbrace{\mathbf{c}}_{\mathbf{e}} \mathbf{B} \mathbf{d}_e \, dV = \frac{1}{2} \mathbf{d}_e^T \left(\int_{V_e} \mathbf{B}^T \mathbf{c} \mathbf{B} \, dV \right) \mathbf{d}_e \quad (3.69)$$

where the subscript e stands for the element. Note that the volume integration over the global domain has been changed to that over the elements. This can be done because we assume that the assumed displacement field satisfies the compatibility condition (see Section 3.3) on all the edges between the elements. Otherwise, techniques discussed in Chapter 11 are needed. In Eq. (3.69), \mathbf{B} gives the strain in the element using the element nodal displacements, and is often called the strain–displacement matrix or simply *strain matrix*. It is defined by

$$\mathbf{B} = \mathbf{L}\mathbf{N} \quad (3.70)$$

where \mathbf{L} is the differential operator that is defined for different problems in Chapter 2. This implies that the strain matrix is determined once the shape functions are obtained. For 3D solids, \mathbf{L} is given by Eq. (2.7). By denoting

$$\mathbf{k}_e = \int_{V_e} \mathbf{B}^T \mathbf{c} \mathbf{B} dV \quad (3.71)$$

which is called the element stiffness matrix, Eq. (3.69) can be rewritten as

$$\Pi = \frac{1}{2} \mathbf{d}_e^T \mathbf{k}_e \mathbf{d}_e \quad (3.72)$$

Note that the stiffness matrix \mathbf{k}_e is symmetrical, because

$$[\mathbf{k}_e]^T = \int_{V_e} [\mathbf{B}^T \mathbf{c} \mathbf{B}]^T dV = \int_{V_e} \mathbf{B}^T \mathbf{c}^T [\mathbf{B}^T]^T dV = \int_{V_e} \mathbf{B}^T \mathbf{c} \mathbf{B} dV = \mathbf{k}_e \quad (3.73)$$

which shows that the transpose of matrix \mathbf{k}_e is itself. In deriving the above equation, $\mathbf{c} = \mathbf{c}^T$ has been employed. Making use of the symmetry of the stiffness matrix, only half of the terms in the matrix need to be evaluated and stored.

By substituting Eq. (3.6) into Eq. (3.3), the kinetic energy can be expressed as

$$T = \frac{1}{2} \int_{V_e} \rho \dot{\mathbf{U}}^T \dot{\mathbf{U}} dV = \frac{1}{2} \int_{V_e} \rho \dot{\mathbf{d}}_e^T \mathbf{N}^T \mathbf{N} \dot{\mathbf{d}}_e dV = \frac{1}{2} \dot{\mathbf{d}}_e^T \left(\int_{V_e} \rho \mathbf{N}^T \mathbf{N} dV \right) \dot{\mathbf{d}}_e \quad (3.74)$$

By denoting

$$\mathbf{m}_e = \int_{V_e} \rho \mathbf{N}^T \mathbf{N} dV \quad (3.75)$$

which is called the mass matrix of the element, Eq. (3.74) can be rewritten as

$$T = \frac{1}{2} \dot{\mathbf{d}}_e^T \mathbf{m}_e \dot{\mathbf{d}}_e \quad (3.76)$$

It is obvious that the element mass matrix is also symmetrical.

Finally, to obtain the work done by the external forces, Eq. (3.6) is substituted into Eq. (3.5):

$$\begin{aligned}
 W_f &= \int_{V_e} \mathbf{d}_e^T \mathbf{N}^T \mathbf{f}_b dV + \int_{S_e} \mathbf{d}_e^T \mathbf{N}^T \mathbf{f}_s dS \\
 &= \mathbf{d}_e^T \left(\underbrace{\int_{V_e} \mathbf{N}^T \mathbf{f}_b dV}_{\mathbf{F}_b} \right) + \mathbf{d}_e^T \left(\underbrace{\int_{S_e} \mathbf{N}^T \mathbf{f}_s dS}_{\mathbf{F}_s} \right)
 \end{aligned} \tag{3.77}$$

where the surface integration is performed only for elements on the force boundary of the problem domain. By denoting

$$\mathbf{F}_b = \int_{V_e} \mathbf{N}^T \mathbf{f}_b dV \tag{3.78}$$

and

$$\mathbf{F}_s = \int_{S_e} \mathbf{N}^T \mathbf{f}_s dS \tag{3.79}$$

Eq. (3.77) can then be rewritten as

$$W_f = \mathbf{d}_e^T \mathbf{F}_b + \mathbf{d}_e^T \mathbf{F}_s = \mathbf{d}_e^T \mathbf{f}_e \tag{3.80}$$

\mathbf{F}_b and \mathbf{F}_s are the nodal forces acting on the nodes of the elements, which are equivalent to the body forces and surface forces applied on the element in terms of the work done on a virtual displacement. These two nodal force vectors can then be added up to form the total nodal force vector \mathbf{f}_e :

$$\mathbf{f}_e = \mathbf{F}_b + \mathbf{F}_s \tag{3.81}$$

Substituting Eqs. (3.72), (3.76), and (3.80) into Lagrangian functional L (Eq. (3.2)), we have

$$L = -\frac{1}{2} \dot{\mathbf{d}}_e^T \mathbf{m}_e \dot{\mathbf{d}}_e + \frac{1}{2} \mathbf{d}_e^T \mathbf{k}_e \mathbf{d}_e - \mathbf{d}_e^T \mathbf{f}_e \tag{3.82}$$

Applying Hamilton's principle (Eq. (3.1)), we have

$$\delta \int_{t_1}^{t_2} \left(-\frac{1}{2} \dot{\mathbf{d}}_e^T \mathbf{m}_e \dot{\mathbf{d}}_e + \frac{1}{2} \mathbf{d}_e^T \mathbf{k}_e \mathbf{d}_e - \mathbf{d}_e^T \mathbf{f}_e \right) dt = 0 \tag{3.83}$$

Note that the variation and integration operators are interchangeable, hence we obtain

$$\int_{t_1}^{t_2} (-\delta \dot{\mathbf{d}}_e^T \mathbf{m}_e \dot{\mathbf{d}}_e + \delta \mathbf{d}_e^T \mathbf{k}_e \mathbf{d}_e - \delta \mathbf{d}_e^T \mathbf{f}_e) dt = 0 \quad (3.84)$$

To explicitly illustrate the process of deriving Eq. (3.84) from Eq. (3.83), we use a two-degree of freedom system as an example. Here, we show the procedure for deriving the second term in Eq. (3.84):

$$\begin{aligned} \delta \left(\frac{1}{2} \mathbf{d}_e^T \mathbf{k}_e \mathbf{d}_e \right) &= \delta \left(\frac{1}{2} \{d_1 \ d_2\} \begin{bmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} \right) \\ &= \frac{1}{2} \delta \left(\{d_1 k_{11} + d_2 k_{12} \ d_1 k_{12} + d_2 k_{22}\} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} \right) \\ &= \frac{1}{2} \delta \left(d_1^2 k_{11} + 2d_1 d_2 k_{12} + d_2^2 k_{22} \right) \\ &= \frac{1}{2} \left[\frac{\partial (d_1^2 k_{11} + 2d_1 d_2 k_{12} + d_2^2 k_{22})}{\partial d_1} \delta d_1 + \frac{\partial (d_1^2 k_{11} + 2d_1 d_2 k_{12} + d_2^2 k_{22})}{\partial d_2} \delta d_2 \right] \\ &= (d_1 k_{11} + d_2 k_{12}) \delta d_1 + (d_1 k_{12} + d_2 k_{22}) \delta d_2 \\ &= \{\delta d_1 \ \delta d_2\} \begin{Bmatrix} d_1 k_{11} + d_2 k_{12} \\ d_1 k_{12} + d_2 k_{22} \end{Bmatrix} = \{\delta d_1 \ \delta d_2\} \begin{bmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} = \delta \mathbf{d}_e^T \mathbf{k}_e \mathbf{d}_e \end{aligned}$$

In Eq. (3.84), the variation and differentiation with time are also interchangeable, i.e.,

$$\delta \dot{\mathbf{d}}_e^T = \delta \left(\frac{d \mathbf{d}_e^T}{dt} \right) = \frac{d}{dt} (\delta \mathbf{d}_e^T) \quad (3.85)$$

Hence, by substituting Eq. (3.85) into Eq. (3.84), and integrating the first term by parts, we obtain

$$\int_{t_1}^{t_2} \delta \dot{\mathbf{d}}_e^T \mathbf{m}_e \dot{\mathbf{d}}_e dt = \underbrace{\delta \mathbf{d}_e^T \mathbf{m}_e \dot{\mathbf{d}}_e}_{=0} \Big|_{t_1}^{t_2} - \int_{t_1}^{t_2} \delta \mathbf{d}_e^T \mathbf{m}_e \ddot{\mathbf{d}}_e dt = - \int_{t_1}^{t_2} \delta \mathbf{d}_e^T \mathbf{m}_e \ddot{\mathbf{d}}_e dt \quad (3.86)$$

Note that in deriving Eq. (3.86) as above, the condition $\delta \mathbf{d}_e = 0$ at t_1 and t_2 have been used, which leads to the vanishing of the first term on the right-hand side. This is because the initial condition at t_1 and final condition at t_2 have to be satisfied for any \mathbf{d}_e (admissible conditions (c) required by Hamilton's principle), and no variation at t_1 and t_2 is allowed. Substituting Eq. (3.86) into Eq. (3.84) leads to

$$\int_{t_1}^{t_2} \delta \mathbf{d}_e^T (\mathbf{m}_e \ddot{\mathbf{d}}_e + \mathbf{k} \mathbf{d}_e - \mathbf{f}_e) dt = 0 \quad (3.87)$$

To have the integration in Eq. (3.87) as zero for an arbitrary integrand, the integrand itself has to vanish, i.e.,

$$\delta \mathbf{d}_e^T (\mathbf{m}_e \ddot{\mathbf{d}}_e + \mathbf{k} \mathbf{d}_e - \mathbf{f}_e) = 0 \quad (3.88)$$

Due to the arbitrary nature of the variation of the displacements, the only insurance for Eq. (3.88) to be satisfied is

$$\mathbf{k}_e \mathbf{d}_e + \mathbf{m}_e \ddot{\mathbf{d}}_e = \mathbf{f}_e \quad (3.89)$$

Equation (3.89) is the FEM equation for an element, while \mathbf{k}_e and \mathbf{m}_e are the *stiffness* and *mass* matrices for the element, and \mathbf{f}_e is the element force vector of the total external forces acting on the nodes of the element. All these element matrices and vectors can be obtained simply by integration for the given shape functions of displacements.

3.4.6 Coordinate transformation

The element equation given by Eq. (3.89) is often formulated based on the local coordinate system defined on an element. In general, the structure is divided into many elements with different orientations (see Figure 3.4). To assemble all the element equations to form the global system equations, a coordinate transformation has to be performed for each element in order to convert its element equation in reference to the *global coordinate system* which is defined on the whole structure.

The coordinate transformation gives the relationship between the displacement vector \mathbf{d}_e based on the local coordinate system and the displacement vector \mathbf{D}_e for the same element, but based on the global coordinate system. It is therefore a transformation of a vector between two (local and global) coordinate systems, which can be written in general in the form of

$$\mathbf{d}_e = \mathbf{T} \mathbf{D}_e \quad (3.90)$$

where \mathbf{T} is the transformation matrix that can have different forms depending upon the type of element, and will be discussed in detail in later chapters when different elements are formulated. The transformation matrix can also be applied to the force vectors between the local and global coordinate systems:

$$\mathbf{f}_e = \mathbf{T} \mathbf{F}_e \quad (3.91)$$

in which \mathbf{F}_e stands for the force vector at node i on the global coordinate system. Substitution of Eq. (3.90) into Eq. (3.89) leads to the element equation based on the global coordinate system:

$$\mathbf{K}_e \mathbf{D}_e + \mathbf{M}_e \ddot{\mathbf{D}}_e = \mathbf{F}_e \quad (3.92)$$

where

$$\mathbf{K}_e = \mathbf{T}^T \mathbf{k}_e \mathbf{T} \quad (3.93)$$

$$\mathbf{M}_e = \mathbf{T}^T \mathbf{m}_e \mathbf{T} \quad (3.94)$$

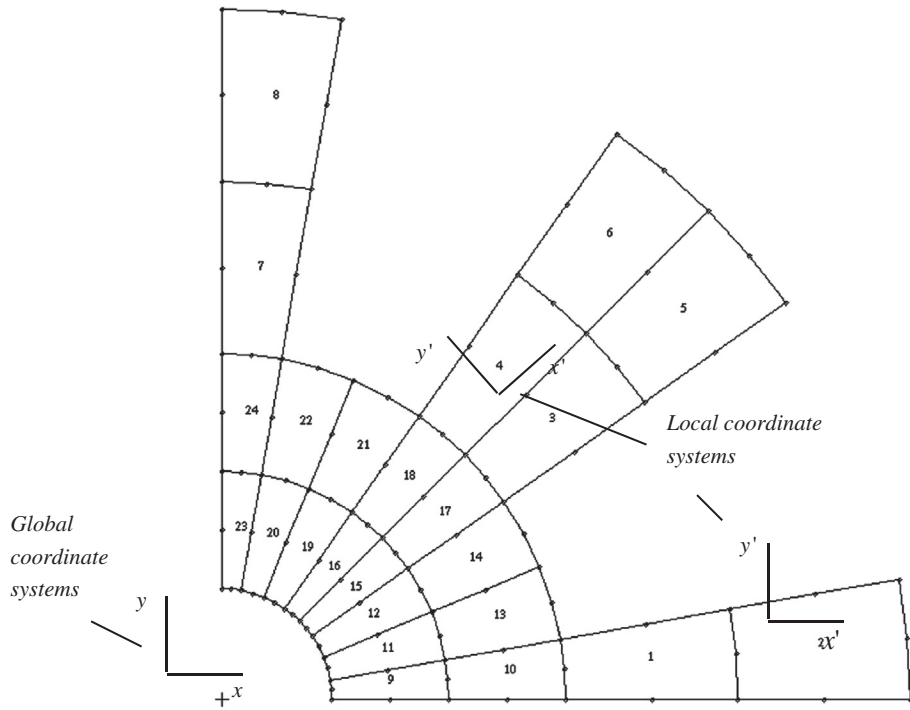


FIGURE 3.4

Local and global coordinate systems.

$$\mathbf{F}_e = \mathbf{T}^T \mathbf{f}_e \quad (3.95)$$

In the cases when the element equations are formulated based on the global coordinate system (for example for the linear triangular or tetrahedral elements), the above coordinate transformation is not necessary, or the \mathbf{T} matrix becomes an identity matrix.

3.4.7 Assembly of global FE equation

The FE equations for all the individual elements can be assembled together to form the global FE system equation:

$$\mathbf{K}\mathbf{D} + \mathbf{M}\ddot{\mathbf{D}} = \mathbf{F} \quad (3.96)$$

where \mathbf{K} and \mathbf{M} are the global stiffness and mass matrix, \mathbf{D} is a vector of all the displacements at all the nodes in the entire problem domain, and \mathbf{F} is a vector of all the equivalent nodal force vectors. The process of assembly is one of simply adding up the contributions from all the elements connected at a node. The detailed process will be demonstrated in Chapter 4 using example problems. It may be noted here that the assembly of the global matrices can be skipped by combining assembling with the

procedure of equation solving. This means that the assembling of a term in the global matrix is done only when the equation solver is operating on this term. Such a procedure can drastically reduce the need for storage during the computation.

3.4.8 Imposition of displacement constraints

The global stiffness matrix \mathbf{K} in Eq. (3.96) does not usually have a full rank, because displacement constraints (supports) are not yet imposed, and it is non-negative definite or positive semi-definite. Physically, an unconstrained solid or structure is capable of performing rigid movements. Therefore, if the solid or structure is free of support, Eq. (3.96) gives the behavior that includes the rigid body dynamics, if it is subjected to dynamic forces. If the external forces applied are static, the displacements cannot be uniquely determined from Eq. (3.96) for any given force vector. It is physically meaningless to try to determine the static displacements of an unconstrained solid or structure that can move freely.

For constrained solids and structures, the constraints can be imposed by simply removing the rows and columns corresponding to the constrained nodal displacements. We shall demonstrate this method in example problems in later chapters. After such treatments of constraints (and if the constraints are sufficient), the stiffness matrix \mathbf{K} in Eq. (3.96) will be of full rank, and will be Positive Definite (PD). Since we have already proven that \mathbf{K} is symmetric, \mathbf{K} is of a symmetric positive definite property.

3.4.9 Solving the global FE equation

By solving the global FE equation, displacements at the nodes can be obtained. The strain and stress in any element can then be retrieved using Eq. (3.6) in Eqs. (2.5) and (2.8).

3.5 Static analysis

Static analysis involves the solving of Eq. (3.96) without the term with the global mass matrix, \mathbf{M} . Hence, as discussed, the static system of equations takes the form

$$\mathbf{K}\mathbf{D} = \mathbf{F} \quad (3.97)$$

There are numerous methods and algorithms to solve the above matrix equation. The methods often used are Gauss elimination or LU decompositions for small systems, and iterative methods for large systems. These methods are all routinely available in any math library of any computer system.

3.6 Analysis of free vibration (eigenvalue analysis)

For a structural system with a total DOF of N , the stiffness matrix \mathbf{K} and mass matrix \mathbf{M} in Eq. (3.96) have a dimension of $N \times N$. By solving the above equation we can obtain the displacement field, and the stress and strain can then be calculated. The question now is how to solve this equation, as N is usually very large for practical engineering structures. One way to solve this equation is by using the so-called direct integration method discussed in the next section. An alternative way of solving Eq. (3.96) is by using the so-called modal analysis (or mode superposition) technique. In this technique,

we first have to solve the homogenous equation of Eq. (3.96). The homogeneous equation is when we consider the case of $\mathbf{F}=0$, therefore it is also called *free vibration* analysis, as the system is free of external forces. For a solid or structure that undergoes a free vibration, the discretized system equation Eq. (3.96) becomes

$$\mathbf{K}\mathbf{D} + \mathbf{M}\ddot{\mathbf{D}} = 0 \quad (3.98)$$

The solution for the free vibration problem can be assumed as

$$\mathbf{D} = \phi \exp(i\omega t) \quad (3.99)$$

where ϕ is the amplitude of the nodal displacement, ω is the (angular) frequency of the free vibration, and t is the time. By substituting Eq. (3.99) into Eq. (3.98), we obtain

$$[\mathbf{K} - \omega^2 \mathbf{M}]\phi = 0 \quad (3.100)$$

or

$$[\mathbf{K} - \lambda \mathbf{M}]\phi = 0 \quad (3.101)$$

where

$$\lambda = \omega^2 \quad (3.102)$$

Equation (3.100), (or (3.101)) is called the *eigenvalue equation*. To have a non-zero solution for ϕ , the determinate of the matrix must vanish:

$$\det[\mathbf{K} - \lambda \mathbf{M}] = |\mathbf{K} - \lambda \mathbf{M}| = 0 \quad (3.103)$$

The expansion of the above equation will lead to a polynomial of λ of order N . This polynomial equation will have N roots, $\lambda_1, \lambda_2, \dots, \lambda_N$, called *eigenvalues*, which relate to the *natural frequencies* of the system by Eq. (3.100). The natural frequency is a very important characteristic of the structure carrying dynamic loads. It has been found that if a structure is excited by a load with a frequency of one of the structure's natural frequencies, the structure can undergo extremely violent vibration, which often leads to catastrophic failure of the structural system. Such a phenomenon is called *resonance*. Therefore, an eigenvalue analysis has to be performed in designing a structural system that is to be subjected to dynamic loadings.

By substituting an eigenvalue λ_i back into the eigenvalue equation, Eq. (3.101), we have

$$[\mathbf{K} - \lambda_i \mathbf{M}]\phi = 0 \quad (3.104)$$

which is a set of algebraic equations. Solving the above equation for ϕ , a vector denoted by ϕ_i can then be obtained. This vector corresponding to the i th eigenvalue λ_i is called the i th *eigenvector* that satisfies the following equation:

$$[\mathbf{K} - \lambda_i \mathbf{M}]\phi_i = 0 \quad (3.105)$$

An eigenvector ϕ_i corresponds to a *vibration mode* that gives the shape of the vibrating structure of the i th mode. Therefore, analysis of the eigenvalue equation also gives very important information on possible vibration modes experienced by the structure when it undergoes a vibration. Vibration modes of a structure are therefore another important characteristic of the structure. Mathematically, the eigenvectors can be used to construct the displacement fields. It has been found that using a few of the lowest modes can obtain very accurate results for many engineering problems. Modal analysis techniques have been developed to take advantage of these properties of natural modes.

In Eq. (3.101), since the mass matrix \mathbf{M} is symmetric positive definite and the stiffness matrix \mathbf{K} is symmetric and either positive or positive semi-definite, the eigenvalues are all real and either positive or zero. It is possible that some of the eigenvalues may coincide. The corresponding eigenvalue equation is said to have multiple eigenvalues. If there are m coincident eigenvalues, the eigenvalue is said to be of a multiplicity of m . For an eigenvalue of multiplicity m , there are m vectors satisfying Eq. (3.105).

Methods for the effective computation of the eigenvalues and eigenvectors for an eigenvalue equations system like Eq. (3.100) or Eq. (3.101) are outside the scope of this book. Intensive research has been conducted to date, and many sophisticated algorithms are already well established and readily available in the open literature, and routinely in computational libraries. The commonly used methods are (see, for example, Petyt, 1990):

- Jacobi's method;
- Given's method and householder's method;
- the bisection method (using Sturm sequences);
- inverse iteration;
- QR method;
- subspace iteration;
- Lanczos' method.

3.7 Transient response

Structural systems are very often subjected to *transient* excitation. A transient excitation is a highly dynamic time-dependent force exerted on the solid or structure, such as earthquake, impact, and shocks. The discrete governing equation system for such a structure is still Eq. (3.96), but it often requires a different solver from that used in the eigenvalue analysis. The widely used method is the so-called *direct integration method*.

The direct integration method basically uses the *finite difference method* (FDM) for time stepping to solve Eq. (3.96). There are two main types of direct integration method: *implicit* and *explicit*. Implicit methods are generally more efficient for a relatively slow phenomenon, and explicit methods are more efficient for a very fast phenomenon, such as impact and explosion. The literature on the various algorithms available for solving transient problems is vast. This section introduces the basic idea of time stepping used in FDMs, which are employed in solving transient problems.

Before discussing the equations used for the time stepping techniques, the general system equation for a structure considering damping effects is written as

$$\mathbf{K}\mathbf{D} + \mathbf{C}\dot{\mathbf{D}} + \mathbf{M}\ddot{\mathbf{D}} = \mathbf{F} \quad (3.106)$$

where $\dot{\mathbf{D}}$ is the vector of velocity components, and \mathbf{C} is the matrix of damping coefficients for the solid/structure that are determined experimentally. The damping coefficients are often expressed simply as proportions of the mass and stiffness matrices, called *proportional damping* (e.g. Petyt, 1990; Clough and Penzien, 1975). For a proportional damping system, \mathbf{C} can be simply determined in the form

$$\mathbf{C} = c_K \mathbf{K} + c_M \mathbf{M} \quad (3.107)$$

where c_K and c_M are determined by experiments.

3.7.1 Central difference algorithm

We first re-write the system equation (3.106) in the form

$$\mathbf{M}\ddot{\mathbf{D}} = \mathbf{F} - \underbrace{[\mathbf{C}\dot{\mathbf{D}} + \mathbf{K}\mathbf{D}]}_{\mathbf{F}^{\text{int}}} = \mathbf{F} - \mathbf{F}^{\text{int}} = \mathbf{F}^{\text{residual}} \quad (3.108)$$

where $\mathbf{F}^{\text{residual}}$ is the *residual* force vector, and

$$\mathbf{F}^{\text{int}} = [\mathbf{C}\dot{\mathbf{D}} + \mathbf{K}\mathbf{D}] \quad (3.109)$$

is defined as the *internal* force at time t . The acceleration, $\ddot{\mathbf{D}}$, can be simply obtained by

$$\ddot{\mathbf{D}} = \mathbf{M}^{-1} \mathbf{F}^{\text{residual}} \quad (3.110)$$

In practice, the above equation does not usually require the solving of the matrix equation, since lumped masses are usually used which form a diagonal mass matrix (Petyt, 1990). The solution to Eq. (3.110) is thus trivial, and the matrix equation is the set of independent equations for each degree of freedom i as follows:

$$d_i = \frac{f_i^{\text{residual}}}{m_i} \quad (3.111)$$

where f_i^{residual} is the residual force, and m_i is the lumped mass corresponding to the i th DOF. We now introduce the following finite central difference equations:

$$\mathbf{D}_{t+\Delta t} = 2(\Delta t)\dot{\mathbf{D}}_t + \mathbf{D}_{t-\Delta t} \quad (3.112)$$

$$\dot{\mathbf{D}}_{t+\Delta t} = 2(\Delta t)\ddot{\mathbf{D}}_t + \dot{\mathbf{D}}_{t-\Delta t} \quad (3.113)$$

$$\ddot{\mathbf{D}}_t = \frac{1}{(\Delta t)^2} (\mathbf{D}_{t+\Delta t} - 2\mathbf{D}_t + \mathbf{D}_{t-\Delta t}) \quad (3.114)$$

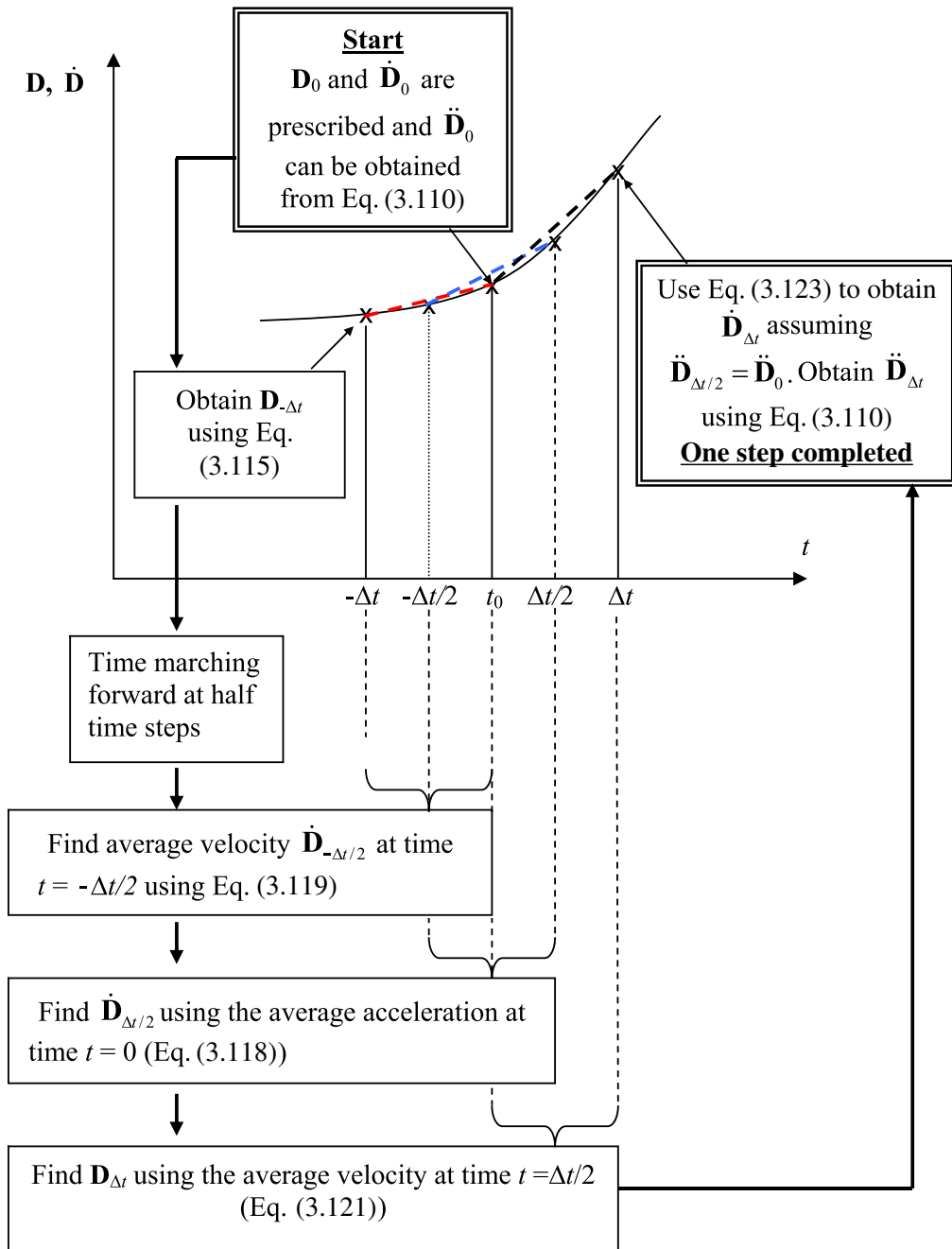


FIGURE 3.5

Time marching in the central difference algorithm: explicitly advancing in time.

By eliminating $\mathbf{D}_{t+\Delta t}$ from Eqs. (3.112) and (3.114), we have

$$\mathbf{D}_{t-\Delta t} = \mathbf{D}_t - (\Delta t)\dot{\mathbf{D}}_t + \frac{(\Delta t)^2}{2}\ddot{\mathbf{D}}_t \quad (3.115)$$

To explain the time stepping procedure, refer to Figure 3.5, which shows an arbitrary plot of either displacement or velocity against time. The time stepping/marching procedure in the central difference method starts at $t=0$, and computes the acceleration $\ddot{\mathbf{D}}_0$ using Eq. (3.110):

$$\ddot{\mathbf{D}}_0 = \mathbf{M}^{-1}\mathbf{F}_0^{\text{residual}} \quad (3.116)$$

For given initial conditions, \mathbf{D}_0 and $\dot{\mathbf{D}}_0$ are known. Substituting \mathbf{D}_0 , $\dot{\mathbf{D}}_0$, and $\ddot{\mathbf{D}}_0$ into Eq. (3.115), we find $\mathbf{D}_{-\Delta t}$. Considering a half of the time step and using the central difference Eqs. (3.112) and (3.113), we have

$$\mathbf{D}_{t+\Delta t/2} = (\Delta t)\dot{\mathbf{D}}_t + \mathbf{D}_{t-\Delta t/2} \quad (3.117)$$

$$\dot{\mathbf{D}}_{t+\Delta t/2} = (\Delta t)\ddot{\mathbf{D}}_t + \dot{\mathbf{D}}_{t-\Delta t/2} \quad (3.118)$$

The velocity, $\dot{\mathbf{D}}_{-\Delta t/2}$ at $t = -\Delta t/2$ can be obtained by Eq. (3.117) by performing the central differencing at $t = -\Delta t/2$ and using values of $\mathbf{D}_{-\Delta t}$ and \mathbf{D}_0 :

$$\dot{\mathbf{D}}_{-\Delta t/2} = \frac{\mathbf{D}_0 - \mathbf{D}_{-\Delta t}}{(\Delta t)} \quad (3.119)$$

After this, Eq. (3.118) is used to compute $\dot{\mathbf{D}}_{\Delta t/2}$ using $\ddot{\mathbf{D}}_0$ and $\dot{\mathbf{D}}_{-\Delta t/2}$:

$$\dot{\mathbf{D}}_{\Delta t/2} = (\Delta t)\ddot{\mathbf{D}}_0 + \dot{\mathbf{D}}_{-\Delta t/2} \quad (3.120)$$

Then, Eq. (3.117) is used once again to compute $\mathbf{D}_{\Delta t}$ using $\dot{\mathbf{D}}_{-\Delta t/2}$ and \mathbf{D}_0 :

$$\mathbf{D}_{\Delta t} = (\Delta t)\dot{\mathbf{D}}_{\Delta t/2} + \mathbf{D}_0 \quad (3.121)$$

Once $\mathbf{D}_{\Delta t}$ is determined, Eq. (3.118) at $t = \Delta t/2$ can be used to obtain $\dot{\mathbf{D}}_{\Delta t}$ by assuming that the acceleration is constant over the step Δt

$$\ddot{\mathbf{D}}_{t+\Delta t/2} \approx \ddot{\mathbf{D}}_t \quad (3.122)$$

and using $\dot{\mathbf{D}}_0$ (that is the prescribed initial velocity). We have,

$$\dot{\mathbf{D}}_{\Delta t} = (\Delta t) \underbrace{\ddot{\mathbf{D}}_{\Delta t/2}}_{\approx \ddot{\mathbf{D}}_0} + \dot{\mathbf{D}}_0 = (\Delta t)\ddot{\mathbf{D}}_0 + \dot{\mathbf{D}}_0 \quad (3.123)$$

At the next step in time, $\ddot{\mathbf{D}}_{t+\Delta t}$ is again computed using Eq. (3.110). The above process is then repeated. The time marching is continued until it reaches the final desired time.

Note that in the above process, the solutions (displacement, velocity, and acceleration) are obtained without solving any matrix form of system equation, but repeatedly using Eqs. (3.110), (3.117), (3.118), and (3.122). The central difference method is therefore an *explicit method*. The time marching in explicit methods is therefore extremely fast, and the coding is also very straightforward. It is particularly suited for simulating highly nonlinear, large deformation, contact, and extremely fast events of mechanics.

The central difference method, like most explicit methods, is *conditionally stable*. This means that if the time step, Δt , becomes too large to exceed a critical time step, Δt_{cr} , then the computed solution will become unstable and might grow without limit. The critical time step Δt_{cr} should be the time taken for the fastest stress wave in the solids/structure to across the smallest element in the mesh. Therefore, the time steps used in the explicit methods are typically 100 to 1000 times smaller than those used with implicit methods (that are outlined in the next subsection). The need to use a small time step, and especially its dependence on the smallest element size, makes the explicit codes lose out to implicit codes for some of the problems, especially for those of slow dynamic events.

3.7.2 Newmark's method (Newmark, 1959)

Newmark's method is the most widely used implicit algorithm. The example software used in this book, ABAQUS, also uses the Newmark's method as its implicit solver except that the equilibrium equation defined in Eq. (3.106) is modified with the introduction of an operator defined by Hilber et al. (1978). In this book, we will introduce the standard Newmark's method as follows. It is first assumed that

$$\mathbf{D}_{t+\Delta t} = \mathbf{D}_t + (\Delta t)\dot{\mathbf{D}}_t + (\Delta t)^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{D}}_t + \beta \ddot{\mathbf{D}}_{t+\Delta t} \right] \quad (3.124)$$

$$\dot{\mathbf{D}}_{t+\Delta t} = \dot{\mathbf{D}}_t + (\Delta t) \left[(1 - \gamma) \ddot{\mathbf{D}}_t + \gamma \ddot{\mathbf{D}}_{t+\Delta t} \right] \quad (3.125)$$

where β and γ are constants chosen by the analyst. Equations (3.124) and (3.125) are then substituted into the system equation, Eq. (3.106) to give

$$\begin{aligned} & \mathbf{K} \left\{ \mathbf{D}_t + (\Delta t)\dot{\mathbf{D}}_t + (\Delta t)^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{D}}_t + \beta \ddot{\mathbf{D}}_{t+\Delta t} \right] \right\} \\ & + \mathbf{C} \left\{ \dot{\mathbf{D}}_t + (\Delta t) \left[(1 - \gamma) \ddot{\mathbf{D}}_t + \gamma \ddot{\mathbf{D}}_{t+\Delta t} \right] \right\} + \mathbf{M} \ddot{\mathbf{D}}_{t+\Delta t} = \mathbf{F}_{t+\Delta t} \end{aligned} \quad (3.126)$$

If we group all the terms involving $\ddot{\mathbf{D}}_{t+\Delta t}$ on the left and shift the remaining terms to the right, we can write

$$\mathbf{K}_{cm} \ddot{\mathbf{D}}_{t+\Delta t} = \mathbf{F}_{t+\Delta t}^{\text{residual}} \quad (3.127)$$

where

$$\mathbf{K}_{\text{cm}} = [\mathbf{K}\beta(\Delta t)^2 + \mathbf{C}\gamma\Delta t + \mathbf{M}] \quad (3.128)$$

and

$$\begin{aligned} \mathbf{F}_{t+\Delta t}^{\text{residual}} = & \mathbf{F}_{t+\Delta t} - \mathbf{K} \left\{ \mathbf{D}_t + (\Delta t)\dot{\mathbf{D}}_t + (\Delta t)^2 \left(\frac{1}{2} - \beta \right) \ddot{\mathbf{D}}_t \right\} \\ & - \mathbf{C} \left\{ \dot{\mathbf{D}}_t + (\Delta t)(1 - \gamma)\ddot{\mathbf{D}}_t \right\} \end{aligned} \quad (3.129)$$

The accelerations $\ddot{\mathbf{D}}_{t+\Delta t}$ can then be obtained by solving the matrix system equation, Eq. (3.127):

$$\ddot{\mathbf{D}}_{t+\Delta t} = \mathbf{K}_{\text{cm}}^{-1} \mathbf{F}_{t+\Delta t}^{\text{residual}} \quad (3.130)$$

Note that the above equation involves matrix inversion, and hence it is analogous to solving a matrix equation. This makes it an implicit method, meaning that one needs to solve a set of linear algebraic equations to obtain a solution at the current time step.

The algorithm normally starts with a prescribed initial velocity and displacements, \mathbf{D}_0 and $\dot{\mathbf{D}}_0$. The initial acceleration $\ddot{\mathbf{D}}_0$ can then be obtained by substituting \mathbf{D}_0 and $\dot{\mathbf{D}}_0$ into Eq. (3.106), if $\ddot{\mathbf{D}}_0$ is not prescribed initially. Then Eq. (3.130) can be used to obtain the acceleration at the next time step, $\ddot{\mathbf{D}}_{\Delta t}$. The displacements $\mathbf{D}_{\Delta t}$ and velocities $\dot{\mathbf{D}}_{\Delta t}$ can then be calculated using Eqs. (3.124) and (3.125), respectively. The procedure then repeats to march forward in time until arriving at the final desired time. Because at each time step, the matrix system Eq. (3.127) has to be solved, which can be very time consuming, the implicit algorithm is a very slow time stepping process.

Newmark's method, like most implicit methods, is *unconditionally stable* if $\gamma \geq 0.5$ and $\beta \geq (2\gamma + 1)^2/16$. Unconditionally stable methods are those in which the size of the time step, Δt , will not affect the stability of the solution, but rather it is governed by accuracy considerations. The unconditionally stable property allows the implicit algorithms to use significantly larger time steps when the external excitation is of a slow time variation. When the external excitation changes too fast with time, the time step needs to be sufficiently small to capture the time variation features of the excitation.

3.8 Remarks

3.8.1 Summary of shape function properties

The properties of the shape functions are summarized in Table 3.1.

3.8.2 Sufficient requirements for FEM shape functions

Properties 3 and 4 are the minimum requirements for shape functions workable for the FEM. In mesh-free methods (Liu, 2009), Property 3 is not a necessary condition for shape functions. Property 5 is a sufficient requirement for shape functions workable for the FEM for solid mechanics problems. It is possible for shape functions that do not possess Property 5 to produce convergent FEM solutions. In this book, however, we generally require all the FEM shape functions to satisfy Properties 3, 4, and 5.

Table 3.1 List of properties of shape functions.

Item	Name	Significance
Property 1	Reproduction property and consistency	Ensures shape functions produce all the functions that can be formed using basis functions used to create the shape functions. It is useful for constructing shape functions with desired accuracy and consistency in displacement field approximation.
Property 2	Shape functions are linearly-independent	Ensures the shape functions are qualified for a unique approximation of the (displacement) field function. It is often viewed as the default property for any FEM model.
Property 3	Delta function properties	Facilitates an easy imposition of essential boundary conditions. This is a minimum requirement for shape functions workable for the standard FEM.
Property 4	Partitions of unity property	Enables the shape functions to produce the rigid body movements. This is a minimum requirement for shape functions for the standard FEM models.
Property 5	Linear field reproduction	Ensures shape functions produce the linear displacement field. This is a sufficient requirement for shape functions capable of passing the patch test, and hence enabling FEM solution to converge to any smooth displacement solution for solid mechanics problems.
Lemma 1	Condition for shape functions being partitions of unity	Provides an alternative tool for checking the property of partitions of unity of shape functions.
Lemma 2	Condition for shape functions being partitions of unity	Provides an alternative tool for checking the property of partitions of unity of shape functions.
Lemma 3	Condition for shape functions being linear field reproduction	Provides an alternative tool for checking the property of linear field reproduction of shape functions.
<i>Property 2 is the default property for any FEM model; Property 3, 4, and 5 are the sufficient requirements for FEM shape functions.</i>		

These three requirements are called the sufficient requirements in this book for FEM shape functions; they are the *delta function property*, *partitions of unity*, and *linear field reproduction*.

3.8.3 Recap of FEM procedure

In FEMs, the displacement field \mathbf{U} is expressed by displacements at *nodes* using *shape functions* defined over *elements*. Once the shape functions are found, the mass matrix and force vector can be obtained using Eqs. (3.75), (3.78), (3.79), and (3.81). The stiffness matrix can also be obtained using Eq. (3.71), once the shape functions and the strain matrix have been found. Therefore, to develop FE equations for any type of structure components, all one needs to do is formulate the shape function N and then establish the strain matrix \mathbf{B} . The other procedures are very much the same. Hence, in the following

chapters, the focus will be mainly on the derivation of the shape function and strain matrix for various types of elements for solids and structural components.

3.9 Review questions

1. What is the difference between the strong and weak forms of system equations?
2. What are the conditions that a summed displacement has to satisfy in order to apply the Hamilton's principle?
3. Briefly describe the standard steps involved in the FEM.
4. Do we have to discretize the problem domain in order to apply the Hamilton's principle?
5. What is the purpose of dividing the problem domain into elements?
6. For a function defined in $[0, l]$ as

$$f(x) = a_0 + a_1x + a_2x^2$$

where l is a given constant, x is an independent variable, and a_i ($i=0,1,2$) are variable constants (but independent of x). Show that:

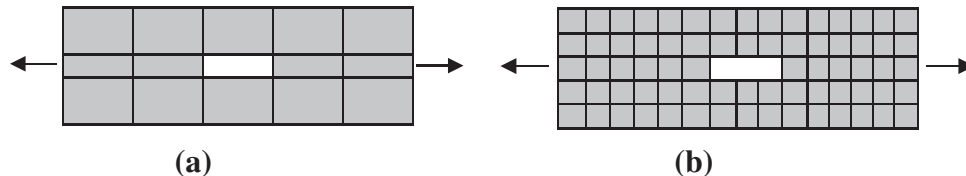
- a. The variation operator and definite integral operator are exchangeable, i.e.,

$$\int_0^l [\delta f(x)] dx = \delta \left[\int_0^l f(x) dx \right]$$

- b. The variation operator and the differential operator are exchangeable, i.e.,

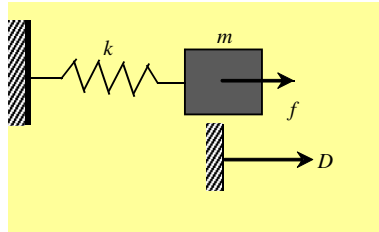
$$\delta \frac{df(x)}{dx} = \frac{d}{dx} [\delta f(x)]$$

7. What are the properties of a shape function? Can we use shape functions that do not possess these properties?

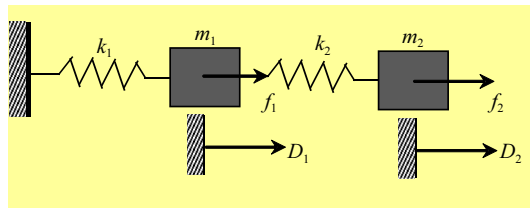


8. Answer the following questions with reference to the two meshes, (a) and (b), shown above:
 - a. Which mesh will yield more accurate results?
 - b. Which will be more computationally expensive?
 - c. Suggest a way of meshing which will yield relatively accurate results and at the same time be less computationally expensive than B?
9. Why is there a need to perform coordinate transformation for each element?

10. Describe how element matrices can be assembled together to form the global system matrix.



11. Consider a 1 DOF system with a rigid block of mass m that can move only in the horizontal direction, as shown in the above figure. The system is at the static equilibrium:
- Give the expressions for the strain potential energy, work done by external forces, and the total potential energy.
 - Derive the equilibrium equations for the system using the minimum potential energy principle.
 - Show that the total potential energy of the 1DOF system is at a minimum at the equilibrium position.
12. Consider a 1DOF system with a rigid block of mass m that can move only in the horizontal direction, as shown in the previous figure. The system is at a dynamic equilibrium. Derive the dynamic equilibrium equation for this system.



13. Consider a 2DOF system with two rigid blocks that can move only in the horizontal direction, as shown in the above figure. The system is at the static equilibrium:
- Give the expressions for the strain potential energy, work done by external forces, and the total potential energy.
 - Derive the equilibrium equations for the system using the minimum potential energy principle.
14. Consider again a 2DOF system with two rigid blocks that can move only in the horizontal direction, as shown in the previous figure. The system is at a dynamic equilibrium.
- Give the expressions for the strain potential energy, work done by external forces, and the total potential energy.
 - Derive the equilibrium equations for the system using the minimum potential energy principle.
15. Consider again the simplest problem of a continuum 1D bar of uniform cross-section, as shown in Figure 2.18. The bar is fixed at the left end and is of length l and section area A . It is subjected to a uniform body force f_x and a concentrated force F_s at the right end. The Young's modulus of the material is E . Using the method of minimum potential energy, obtain the distribution of the displacement for the following cases:
- $f_x = 0$ and $F_s = \text{constant}$;
 - $f_x = \text{constant}$ and $F_s = \text{constant}$.