W. Klug

The Finite Element Method
Linear Static and Dynamic Finite Element Analysis

Thomas J. R. Hughes
Mary and Gordon Crary Family Professor of Engineering
Chairman of the Division of Mechanics and Computation
Stanford University

DOVER PUBLICATIONS, INC.
Mineola, New York

3

Isoparametric Elements and Elementary Programming Concepts

3.1 PRELIMINARY CONCEPTS

We wish to define the shape functions in such a way that, as the finite element mesh is refined, the approximate Galerkin solution converges to the exact solution. The following question arises: What conditions must the shape functions satisfy so that this property is guaranteed? We shall be content, for the time being, to state sufficient conditions for convergence. These conditions are possessed by the most prevalent and important finite element shape functions. However, we note that convergent elements can be constructed from shape functions which do not satisfy all these requirements. Nevertheless these conditions may be considered basic in that they provide the simplest criteria to ensure convergence for a wide class of problems.

The basic convergence requirements are that the shape functions be

C1. smooth (i.e., at least C^1) on each element interior, Ω^e;
C2. continuous across each element boundary Γ^e; and
C3. complete.

Remarks

1. Conditions C1 and C2 guarantee that first derivatives of the shape functions have, at worst, finite jumps across the element interfaces; see Fig. 3.1.1. This ensures that all integrals necessary for the computation of element arrays (see Sec. 2.11) are well defined, since at most first derivatives appear in the integrands. If we permit finite discontinuities in the shape functions on element boundaries, the derivatives possess delta functions (cf. Sec. 1.10) and we are unable to make sense out of the squares of these quantities that would appear in the stiffness integrands.

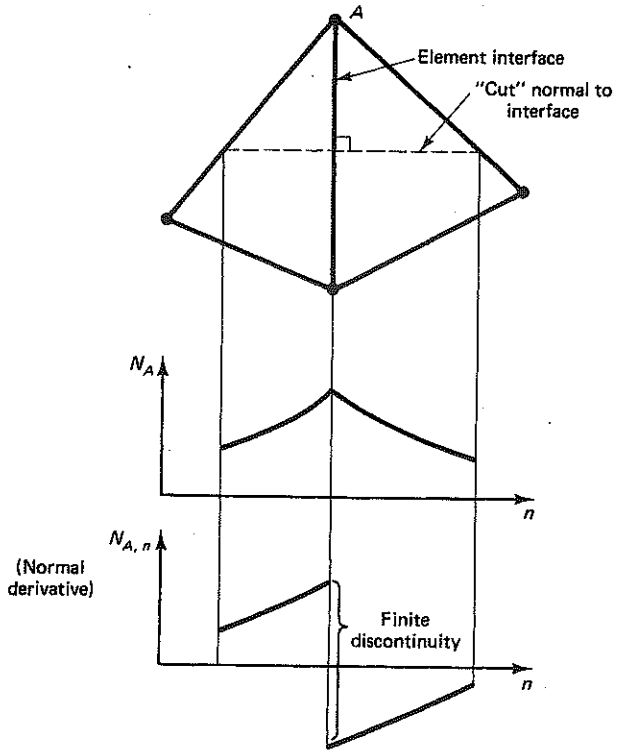


Figure 3.1.1 Example of a shape function that satisfies conditions C1 and C2.

2. Shape functions that satisfy C1 and C2 are of class $C^0(\overline{\Omega})$. Finite elements constructed from $C^0(\overline{\Omega})$ shape functions are often referred to as C^0-***elements***.

3. Theories such as Bernoulli-Euler beam theory, which require higher-order derivatives in the stiffness integrands, necessitate strengthening condition C1 to C^2-continuity on Ω^e and condition C2 to C^1-continuity across Γ^e. Shape functions of this type are of class $C^1(\overline{\Omega})$ and lead to so-called C^1-***elements***. Examples of $C^1(\overline{\Omega})$ shape functions are the Hermite cubics discussed in Sec. 1.16.

4. If the stiffness integrands involve derivatives of order m, Condition C1 should be strengthened to C^m-continuity on Ω^e and Condition C2 should be strengthened to C^{m-1}-continuity across Γ^e. Finite elements that satisfy this property are called ***conforming***, or ***compatible***. (The use of elements that violate this property, ***nonconforming*** or ***incompatible elements***, is, however, common. We say more about this later.)

5. It may be noted that C2 is equivalent to requiring that each function $u^h \in \mathcal{S}^h$ be continuous across Γ^e. In specific cases it is often easier to establish this fact directly rather than prove it for individual shape functions.

Completeness

To illustrate the completeness requirement, let us take the case of heat conduction in which the eth element interpolation function may be written

$$u^h = \sum_{a=1}^{n_{en}} N_a d_a^e \tag{3.1.1}$$

where $d_a^e = u^h(x_a^e)$. Let $n_{sd} = 3$. In this case, the shape functions are said to be ***complete*** if

$$d_a^e = c_0 + c_1 x_a^e + c_2 y_a^e + c_3 z_a^e \tag{3.1.2}$$

implies that

$$u^h(x) = c_0 + c_1 x + c_2 y + c_3 z \tag{3.1.3}$$

where $c_0, \ldots, c_3$ are arbitrary constants. In two dimensions, we omit the z-terms in (3.1.2) and (3.1.3). In words, *completeness requires that the element interpolation function is capable of exactly representing an arbitrary linear polynomial when the nodal degrees of freedom are assigned values in accordance with it.*

Completeness is a plausible requirement as the following argument indicates: As the finite element mesh is further and further refined, the exact solution and its derivatives approach constant values over each element domain. To ensure that these constant values are representable, the shape functions must contain all constant and linear monomials. This argument was originally given in [1] and has subsequently been proved to be the key mathematical idea for proving convergence theorems for finite element approximations (e.g., see [2]).

For elasticity , the requirement is essentially the same. In this case we may write

$$u_i^h = \sum_{a=1}^{n_{en}} N_a d_{ia}^e \tag{3.1.4}$$

where $d_{ia} = u_i^h(x_a)$. The N_a are complete if

$$d_{ia}^e = c_0 + c_1 x_a^e + c_2 y_a^e + c_3 z_a^e \tag{3.1.5}$$

implies

$$u_i^h(x) = c_0 + c_1 x + c_2 y + c_3 z \tag{3.1.6}$$

Remarks

6. In elasticity, the presence of all monomials through linear terms means that an element may exactly represent all rigid motions (i.e., rigid translations and rotations) and constant strain states. We thus often speak of completeness in terms of the ability to represent rigid motions and constant strains (e.g., see [3], p. 33).

7. For theories involving mth derivatives in the stiffness integrands, completeness must be strengthened to mth-order polynomials.

Example 1

The piecewise linear finite element space introduced in Chapter 1 satisfies the convergence conditions C1–C3. C1 and C2 follow from the definition (see Sec. 1.8). For C3 we proceed as follows: Assume $d_a^e = c_0 + c_1 x_a^e$ and substitute in $u^h = \Sigma_{a=1}^2 N_a d_a^e$, viz.,

$$u^h = \sum_{a=1}^{2} N_a d_a^e$$
$$= \sum_{a=1}^{2} N_a (c_0 + c_1 x_a^e)$$
$$= \left(\sum_{a=1}^{2} N_a \right) c_0 + \left(\sum_{a=1}^{2} N_a x_a^e \right) c_1 \tag{3.1.7}$$

Thus to establish completeness we must show that

$$\sum_{a=1}^{2} N_a = 1 \tag{3.1.8}$$

and

$$\sum_{a=1}^{2} N_a x_a^e = x \tag{3.1.9}$$

By (1.12.5), $N_a(\xi) = (1 + (-1)^a \xi)/2$, from which (3.1.8) follows. Equation (3.1.9) was established previously (see (1.12.6)).

3.2 BILINEAR QUADRILATERAL ELEMENT

This element is attributed to Taig [4]. The rectangular version was proposed earlier by Argyris in [5].

The domain of a straight-edged quadrilateral element is defined by the locations of its four nodal points x_a^e, $a = 1, \ldots, 4$ in the $\mathbb{R}^2$-plane. We assume the nodal points are labeled in ascending order corresponding to the counterclockwise direction (see Fig. 3.2.1). We seek a change of coordinates which maps the given quadrilateral into the biunit square, as depicted in Fig. 3.2.1. The biunit square is sometimes called the ***parent domain.*** The coordinates of a point

$$\boldsymbol{\xi} = \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} \tag{3.2.1}$$

in the biunit square are to be related to the coordinates of a point

$$\boldsymbol{x} = \begin{Bmatrix} x \\ y \end{Bmatrix} \tag{3.2.2}$$

in Ω^e by mappings of the form

$$x(\xi, \eta) = \sum_{a=1}^{4} N_a(\xi, \eta) x_a^e \tag{3.2.3}$$

$$y(\xi, \eta) = \sum_{a=1}^{4} N_a(\xi, \eta) y_a^e \tag{3.2.4}$$

ξ and η are sometimes called the ***natural coordinates.*** A more succinct representation of the above formulas is

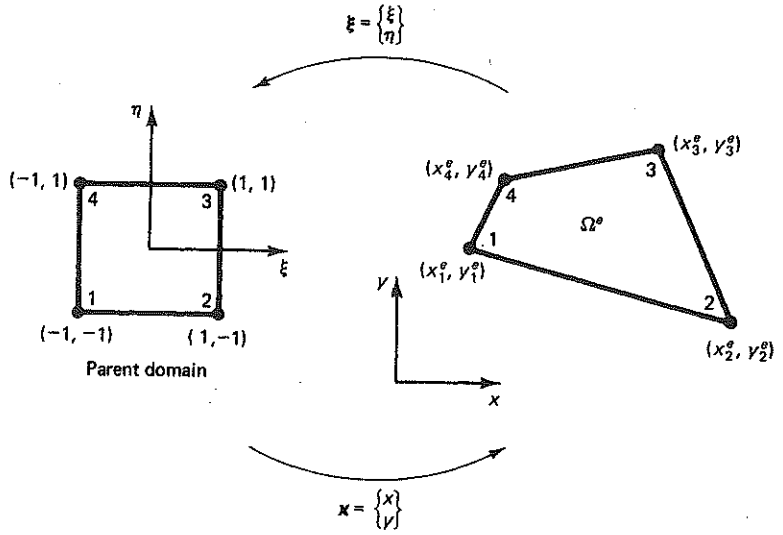


Figure 3.2.1 Bilinear quadrilateral element domain and local node ordering.

$$x(\xi) = \sum_{a=1}^{4} N_a(\xi) x_a^e \tag{3.2.5}$$

We obtain the functions N_a by first *assuming* the "bilinear" expansions

$$x(\xi, \eta) = \alpha_0 + \alpha_1 \xi + \alpha_2 \eta + \alpha_3 \xi\eta \tag{3.2.6}$$

$$y(\xi, \eta) = \beta_0 + \beta_1 \xi + \beta_2 \eta + \beta_3 \xi\eta \tag{3.2.7}$$

where the α's and β's are parameters to be determined; and second, by stipulating that (3.2.6) and (3.2.7) satisfy the (respective) conditions

$$x(\xi_a, \eta_a) = x_a^e \tag{3.2.8}$$

$$y(\xi_a, \eta_a) = y_a^e \tag{3.2.9}$$

where ξ_a and η_a are defined in Table 3.2.1.

TABLE 3.2.1 Nodal coordinates in ξ-space

a	ξ_a	η_a
1	−1	−1
2	1	−1
3	1	1
4	−1	1

Conditions (3.2.8) and (3.2.9) impose restrictions on the functions N_a; namely,

$$N_a(\xi_b) = \delta_{ab} \tag{3.2.10}$$[1]

To see this we can combine (3.2.3) and (3.2.8), viz.,

$$x_b^e = x(\xi_b, \eta_b) = \sum_{a=1}^{4} N_a(\xi_b, \eta_b) x_a^e \tag{3.2.11}$$

which holds only if $N_a(\xi_b, \eta_b) = \delta_{ab}$. This same restriction may be deduced by combining (3.2.4) and (3.2.9). Equations (3.2.6) through (3.2.9) lead to the following matrix equations:

$$\begin{Bmatrix} x_1^e \\ x_2^e \\ x_3^e \\ x_4^e \end{Bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{Bmatrix} \tag{3.2.12}$$

$$\begin{Bmatrix} y_1^e \\ y_2^e \\ y_3^e \\ y_4^e \end{Bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{Bmatrix} \tag{3.2.13}$$

In each case the coefficient matrix is the same. Solving these for the α's and β's, substituting the results into (3.2.6) and (3.2.7), and comparing with (3.2.3) and (3.2.4) reveals that

$$\boxed{N_a(\xi) = N_a(\xi, \eta) = \tfrac{1}{4}(1 + \xi_a \xi)(1 + \eta_a \eta)} \tag{3.2.14}$$

Note that this is precisely the product of one-dimensional shape functions derived previously (see (1.12.5)).

Exercise 1. Verify (3.2.14).

Observe that coordinate lines in ξ-space (i.e., lines such that ξ = constant or η = constant) are mapped into straight lines in x-space. In particular, boundary lines are mapped into boundary lines (see Fig. 3.2.2).

Now we shall *assume* that the element functions are given by similar expansions in terms of the shape functions.

Heat conduction

$$u^h(\xi) = \sum_{a=1}^{4} N_a(\xi) d_a^e \tag{3.2.15}$$

[1]Equation (3.2.10) is sometimes referred to as the ***interpolation property*** since it implies that the assumed expansions (e.g., (3.2.6) and (3.2.7)) interpolate the nodal data (e.g., (3.2.8) and (3.2.9)).

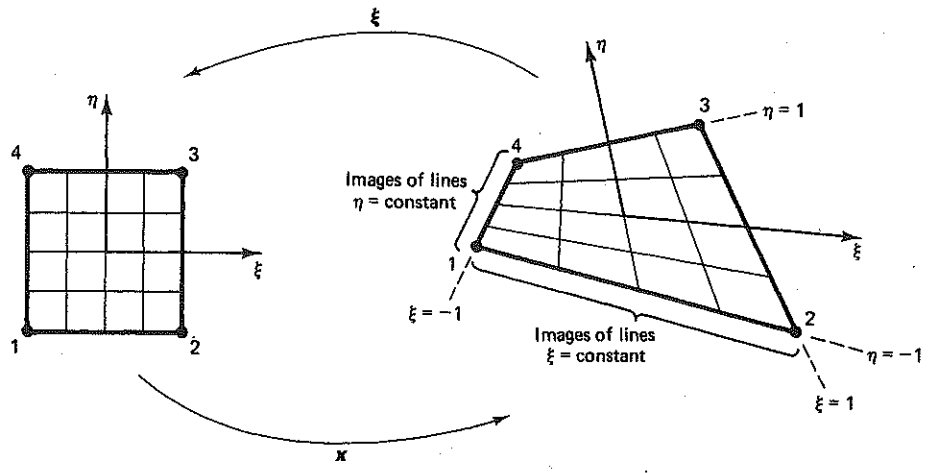


Figure 3.2.2

Elastostatics

$$u_i^h(\boldsymbol{\xi}) = \sum_{a=1}^{4} N_a(\boldsymbol{\xi}) d_{ia}^e \tag{3.2.16}$$

Henceforce, it will be sufficient to consider only (3.2.15). The case of elastostatics may be deduced by viewing (3.2.15) as any one of the n_{sd} component functions of (3.2.16).

(C1) Smoothness on Ω^e. It can be shown that N_a is a smooth function of x and y, if all interior angles formed by two adjacent edges are less than 180°; see Fig. 3.2.3. (Note that N_a is always a smooth function of ξ and η; see (3.2.14). When N_a is viewed as a function of x and y (i.e., $N_a(\xi(x, y), \eta(x, y))$), to ascertain smoothness we must consider the inverse functions $\xi(x, y)$ and $\eta(x, y)$, which may not be well defined if the element is too distorted. Figure 3.2.3 illustrates when this occurs for the four-node element. This issue is discussed from a more general perspective in the following section.)

(C2) Continuity across Γ^e. Let us examine a typical shape function N_a. For the moment, assume $a = 1$ or 2. The behavior of N_a along the edge connecting nodes 1 and 2 may be determined by substituting $\eta = -1$ into (3.2.14). The result is

$$N_a(\xi, -1) = \frac{1 + \xi_a \xi}{2}, \qquad a = 1, 2 \tag{3.2.17}$$

From Table 3.2.1, $\xi_1 = -1$ and $\xi_2 = +1$; thus the right-hand side of (3.2.17) is the familiar linear shape function of the one-dimensional theory (see Fig. 3.2.4). By virtue of the fact that (3.2.17) is typical of all four edges and by (3.2.10), we conclude

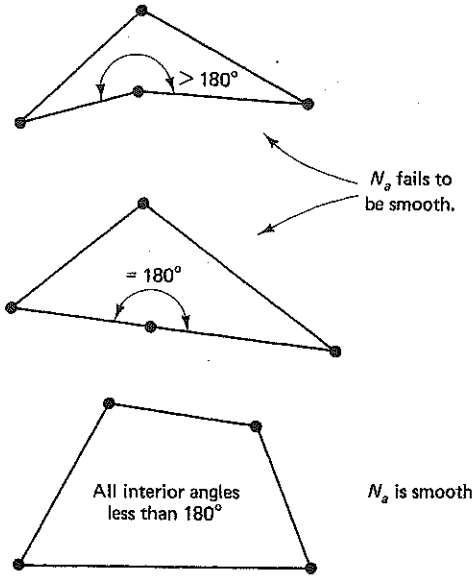


Figure 3.2.3 Smoothness criterion for bilinear shape functions.

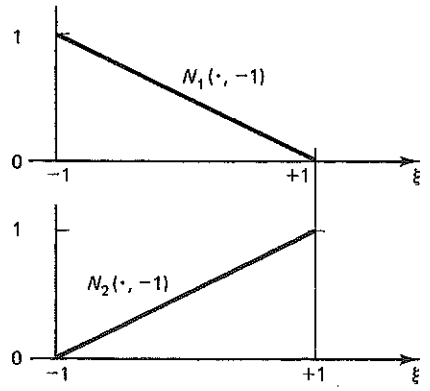


Figure 3.2.4 Bilinear shape functions along edge connecting nodes 1 and 2.

that N_a appears as in Fig. 3.2.5. N_a is said to be a *hyperbolic paraboloid*. Similar considerations for adjacent elements indicate that N_A is continuous and appears as a "tent" whose "pole" emanates from node A; see Fig. 3.2.6 for a typical situation at an internal node. Consequently, $\mathcal{S}^h$ consists of continuous functions as each member is a linear combination of the N_A's (cf. (2.4.3)–(2.4.5)). It is instructive to argue this fact from a slightly different point of view.

Consider the behavior of (3.2.15) along the boundary segment connecting $\boldsymbol{x}_1^e$ to $\boldsymbol{x}_2^e$. By (3.2.17), this may be written

$$u^h(\xi, -1) = \sum_{a=1}^{2} \tfrac{1}{2}(1 + \xi_a \xi) d_a^e \tag{3.2.18}$$

There are two important observations to be made about this result:

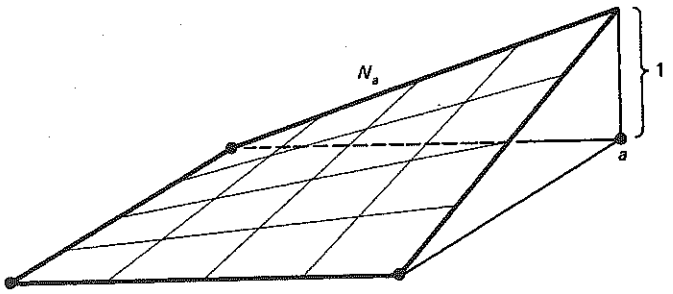


Figure 3.2.5 Bilinear shape function.

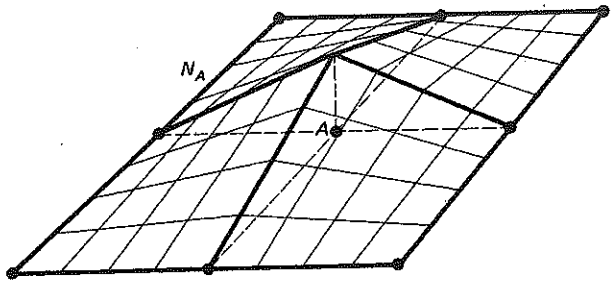


Figure 3.2.6 Global shape function constructed from element bilinear shape functions.

1. The behavior of u^h along $[x_1^e, x_2^e]$ is determined solely by the nodal values of u^h at the nodes x_1^e and x_2^e.
2. The variation of u^h is linear in the natural coordinate ξ (see Fig. 3.2.7).

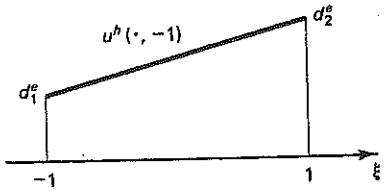


Figure 3.2.7 Graph of u^h along boundary segment joining nodes x_1^e and x_2^e.

The same conclusion can be drawn for any other element of this type whose boundary contains $[x_1^e, x_2^e]$. Thus we see that continuity of u^h is automatic across element interfaces.

(C3) Completeness

$$
\begin{aligned}
u^h &= \sum_{a=1}^{n_{en}} N_a d_a^e \\
&= \sum_{a=1}^{n_{en}} N_a (c_0 + c_1 x_a^e + c_2 y_a^e) \\
&= \left(\sum_{a=1}^{n_{en}} N_a\right) c_0 + \underbrace{\left(\sum_{a=1}^{n_{en}} N_a x_a^e\right)}_{= x \text{ by } (3.2.3)} c_1 + \underbrace{\left(\sum_{a=1}^{n_{en}} N_a y_a^e\right)}_{= y \text{ by } (3.2.4)} c_2
\end{aligned}
$$

Thus it remains only to prove that the shape functions sum to value 1 at each point. This may be done by explicit calculation, viz.,

$$\begin{aligned}\sum_{a=1}^{n_{en}} N_a(\xi, \eta) &= \tfrac{1}{4}(1-\xi)(1-\eta) + \tfrac{1}{4}(1+\xi)(1-\eta) \\ &\quad + \tfrac{1}{4}(1+\xi)(1+\eta) + \tfrac{1}{4}(1-\xi)(1+\eta), \qquad \text{(by (3.2.14))} \\ &= 1 \qquad (3.2.20)\end{aligned}$$

3.3 ISOPARAMETRIC ELEMENTS

The isoparametric concept is generally attributed to Taig [6] and Irons [7].

Let $\square$ denote a parent domain in ξ-space (e.g., the closed, biunit square in $\mathbb{R}^2$).

Definition 1. Let $\boldsymbol{x}$: $\square \to \overline{\Omega}^e$ be of the form

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{a=1}^{n_{en}} N_a(\boldsymbol{\xi})\boldsymbol{x}_a^e \qquad (3.3.1)$$

If the element interpolation function u^h can be written as

$$u^h(\boldsymbol{\xi}) = \sum_{a=1}^{n_{en}} N_a(\boldsymbol{\xi})d_a^e \qquad (3.3.2)$$

the element is said to be ***isoparametric.***

The key point to observe in the definition is that the shape functions which define (3.3.1) *also serve to define* (3.3.2).

The bilinear quadilateral, derived in the previous section, is an example of an isoparametric element.

We shall now discuss some fundamental mathematical properties of mappings. Based on this, we shall argue that convergence condition C1 is generally achieved for isoparametric elements. Furthermore, a criterion, which may be used in a computer program to determine whether or not individual elements satisfy C1, emanates from the discussion.

Convergence Condition C1

Definition 2. A mapping $\boldsymbol{x}$: $\square \to \overline{\Omega}^e \subset \mathbb{R}^{n_{sd}}$ is said to be ***one-to-one*** if for each pair of points $\boldsymbol{\xi}^{(1)}$, $\boldsymbol{\xi}^{(2)} \in \square$ such that $\boldsymbol{\xi}^{(1)} \neq \boldsymbol{\xi}^{(2)}$, then $\boldsymbol{x}(\boldsymbol{\xi}^{(1)}) \neq \boldsymbol{x}(\boldsymbol{\xi}^{(2)})$.

In words, this statement means that two different points of $\square$ do not get mapped into the same point in $\overline{\Omega}^e$.

Definition 3. $\boldsymbol{x} : \square \to \overline{\Omega}^e$ is said to be ***onto*** if $\overline{\Omega}^e = \boldsymbol{x}(\square)$ (i.e., each point in $\overline{\Omega}^e$ is the image of a point in $\square$ under the mapping $\boldsymbol{x}$).

Definition 4. Let $x : \square \to \overline{\Omega}^e$ be a differentiable mapping. The determinant of the derivative, denoted by $j = \det(\partial x/\partial \xi)$, is called the ***Jacobian determinant.***

The Jacobian determinants in two and three dimensions, respectively, are given explicitly by

$$j = \det\begin{bmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{bmatrix} \tag{3.3.3}$$

$$j = \det\begin{bmatrix} x_{,\xi} & x_{,\eta} & x_{,\zeta} \\ y_{,\xi} & y_{,\eta} & y_{,\zeta} \\ z_{,\xi} & z_{,\eta} & z_{,\zeta} \end{bmatrix} \tag{3.3.4}$$

Remark

1. It is a consequence of the inverse function theorem (e.g., see [8]) that if x is

i. one-to-one;
ii. onto;
iii. C^k, $k \geq 1$; and if
iv. $j(\xi) > 0$ for all $\xi \in \square$;

then the inverse mapping $\xi = x^{-1} : \overline{\Omega}^e \to \square$ exists and is C^k.

Proposition 1. Let the mapping defined by (3.3.1) satisfy (i) through (iv). Then the smoothness requirement (C1) is satisfied.

Proof. By virtue of the hypotheses, $N_a = N_a(\xi)$ is also a C^1 function. By Remark 1, $\xi = \xi(x)$ is also C^1. Thus $N_a(x) = N_a(\xi(x))$ is a C^1 function of x. (This last fact may be proved with the aid of the chain rule.) ■

Remarks

2. In practice, the mappings $x : \square \to \overline{\Omega}^e$ usually satisfy (i) through (iv). However, there is one exception of practical importance. It is concerned with the technique of element "degeneration," in which nodes are coalesced. The simplest example of this procedure, in which two nodes of the bilinear quadilateral are coalesced to form a triangle, is presented in the next section. Further examples are presented in subsequent sections. When degeneration is performed, the Jacobian determinant vanishes at certain nodal points within the element. Away from these points it is positive, and the mapping $\xi = \xi(x)$ remains smooth (i.e., C1 is satisfied). For reasons that will be apparent later on, it is not usually required to calculate derivatives at these points.

3. In actual computations the sign of the Jacobian determinant is monitored at special points to ensure condition (iv) is satisfied. If a zero or negative value is encountered, computations are terminated. Generally this is an indication of an input-data error or an overly distorted element.

Convergence Condition (C3)

Proposition 2. If $\sum_{a=1}^{n_{en}} N_a = 1$, then completeness condition C3 is satisfied for isoparametric elements.

Proof. (We shall prove the assertion for the three-dimensional case.)

$$
\begin{aligned}
u^h &= \sum_{a=1}^{n_{en}} N_a d_a^e && \text{(by (3.3.2))} \\
&= \sum_{a=1}^{n_{en}} N_a (c_0 + c_1 x_a^e + c_2 y_a^e + c_3 z_a^e) \\
&= c_0 \left(\sum_{a=1}^{n_{en}} N_a \right) + c_1 \left(\sum_{a=1}^{n_{en}} N_a x_a^e \right) + c_2 \left(\sum_{a=1}^{n_{en}} N_a y_a^e \right) + c_3 \left(\sum_{a=1}^{n_{en}} N_a z_a^e \right) \\
&= c_0 \left(\sum_{a=1}^{n_{en}} N_a \right) + c_1 x + c_2 y + c_3 z && \text{(by (3.3.1))}
\end{aligned}
$$

The condition that the shape functions sum to one, assumed in Proposition 2, is easily checked on a case-by-case basis. ■

The only remaining convergence condition is C2, the continuity requirement on Γ^e. This condition can be verified once the construction of the global shape functions from the element shape functions is explicated. It happens that if this procedure is done in the "obvious" way, continuity is achieved. In the sequel we shall consider this issue on a case-by-case basis.

Summary. The importance of the isoparametric concept is that the three basic convergence conditions are virtually automatic. In addition, isoparametric elements may be designed to take on convenient shapes for practical analysis, including curved boundaries, and lend themselves to concise computer implementation.

3.4 LINEAR TRIANGULAR ELEMENT; AN EXAMPLE OF "DEGENERATION"

The linear triangle was one of the first finite elements conceived (see Courant [9] and Turner et al. [10]) and is still widely used. In the structural mechanics literature it is often referred to as the constant stress/strain triangle, or simply the CST. We shall derive it from the bilinear quadrilateral by coalescing nodes 3 and 4; see Fig. 3.4.1. Specifically, we set $\boldsymbol{x}_4^e = \boldsymbol{x}_3^e$ in (3.2.5) and define new shape functions for the triangle N_a', $a = 1, 2, 3$, as follows:

$$
\begin{aligned}
\boldsymbol{x} &= \sum_{a=1}^{4} N_a \boldsymbol{x}_a^e \\
&= N_1 \boldsymbol{x}_1^e + N_2 \boldsymbol{x}_2^e + (N_3 + N_4) \boldsymbol{x}_3^e = \sum_{a=1}^{3} N_a' \boldsymbol{x}_a^e
\end{aligned} \tag{3.4.1}
$$

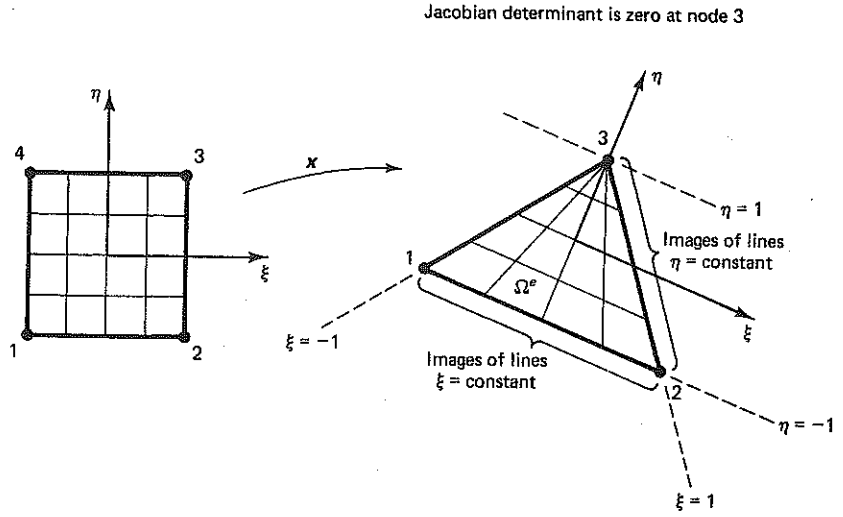


Figure 3.4.1

where

$$N'_a = \begin{cases} N_a = \frac{1}{4}[1 + (-1)^a \xi](1 - \eta) & a = 1, 2 \\ N_3 + N_4 = \frac{1}{2}(1 + \eta) & a = 3 \end{cases} \tag{3.4.2}$$

The shape functions are illustrated in Fig. 3.4.2; they are "planes" above Ω^e, and so the derivatives with respect to coordinates x and y are constants. In this case note that (3.4.1) is not one-to-one, since the boundary segment $\xi \in [-1, 1]$, $\eta = 1$ in $\boldsymbol{\xi}$-space is mapped into the point $\boldsymbol{x}_3^e$ in $\boldsymbol{x}$-space. In fact, the Jacobian determinant is zero at $\boldsymbol{x}_3^e$. However, as we have already seen, the derivatives with respect to x and y are well behaved. Thus condition C1 is satisfied.

The continuity condition C2 may be inferred form Fig. 3.4.2. A typical global shape function is illustrated in Fig. 3.4.3. Sometimes these functions are called *pyramids,* for obvious reasons. An alternative verification of continuity may be attained by evaluating the element interpolation function

$$u^h = \sum_{a=1}^{3} N'_a d_a^e \tag{3.4.3}$$

along the boundaries, viz.,

(*Side* 1, 2)

$$u^h(\xi, -1) = \tfrac{1}{2}(1 - \xi)d_1^e + \tfrac{1}{2}(1 + \xi)d_2^e \tag{3.4.4}$$

(*Side* 2, 3)

$$u^h(1, \eta) = \tfrac{1}{2}(1 - \eta)d_2^e + \tfrac{1}{2}(1 + \eta)d_3^e \tag{3.4.5}$$

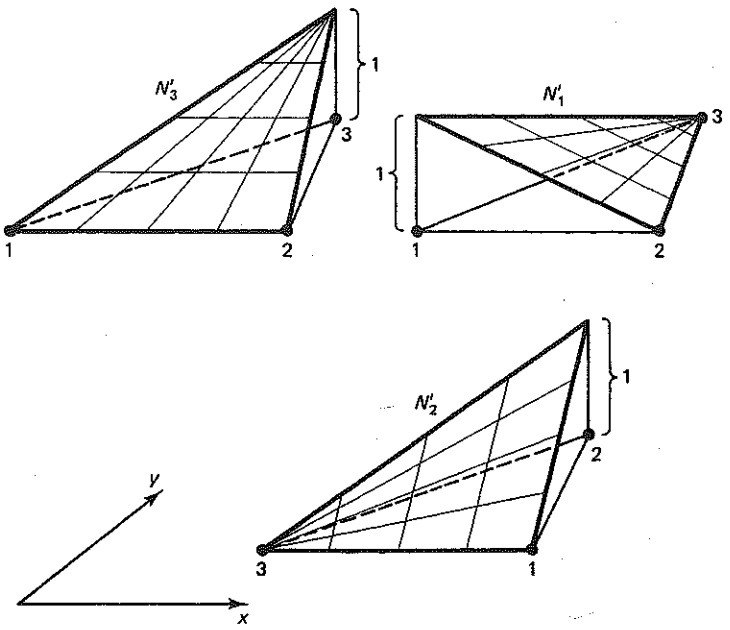


Figure 3.4.2 Element shape functions for the linear triangle.

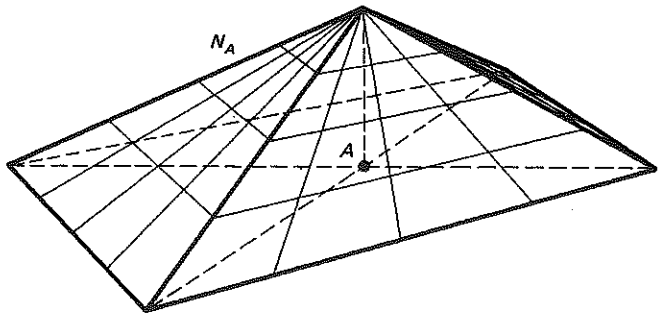


Figure 3.4.3 Global shape function constructed from linear triangular element shape functions.

(*Side* 3, 1)

$$u^h(-1, \eta) = \tfrac{1}{2}(1 + \eta)d_3^e + \tfrac{1}{2}(1 - \eta)d_1^e \tag{3.4.6}$$

As is clear from (3.4.4)–(3.4.6), the behavior is linear along each edge. As this is the same for all contiguous elements, continuity is assured. We may also freely mix linear triangles and bilinear quadrilaterals in the same mesh and achieve continuity, since their edge behavior is identical.

The completeness condition C3 is automatic since we have employed the isoparametric concept in defining (3.4.3).

Exercise 1. Compute the Jacobian determinant at $\xi = \eta = 0$ for the element shown in Fig. 3.4.4. Plot the result as a function of $x_1^e \in [-2, 2]$. Comment.

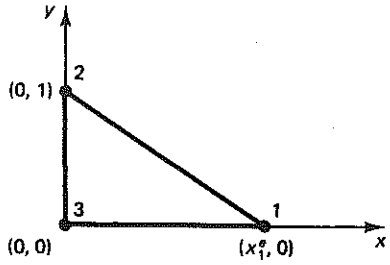


Figure 3.4.4

Summary. The linear triangular element may be obtained from the bilinear quadrilateral by *degeneration*. Computer implementation may be performed along similar lines once the bilinear quadrilateral is programmed.

3.5 TRILINEAR HEXAHEDRAL ELEMENT

The trilinear hexahedral element is the basic element for three-dimensional analysis. It is a straightforward generalization of the bilinear quadrilateral presented in Sec. 3.2. The domain Ω^e (see Fig. 3.5.1) is the image of the biunit cube in ξ-space under the trilinear mapping

$$x(\xi) = \alpha_0 + \alpha_1\xi + \alpha_2\eta + \alpha_3\zeta + \alpha_4\xi\eta + \alpha_5\eta\zeta + \alpha_6\xi\zeta + \alpha_7\xi\eta\zeta \tag{3.5.1}$$

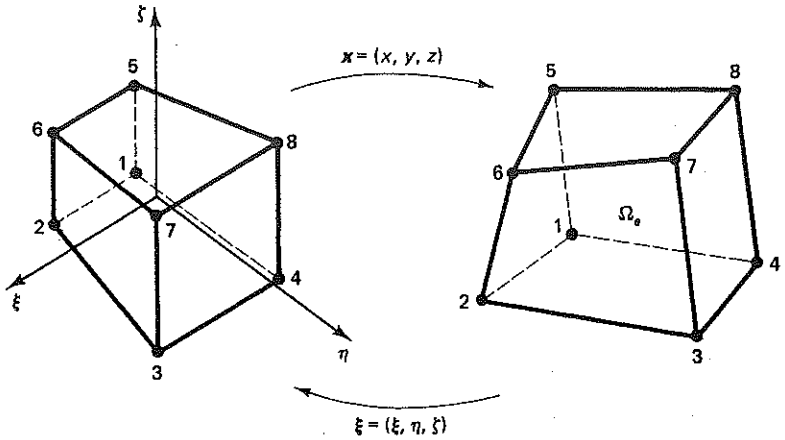


Figure 3.5.1 Trilinear hexahedral element domain and local node ordering.

with corresponding expressions for $y(\xi)$ and $z(\xi)$. The coefficients $\alpha_0, \ldots, \alpha_7$ are determined by the conditions

$$x(\xi_a) = x_a^e, \qquad a = 1, \ldots, 8 \tag{3.5.2}$$

(The nodal coordinates in ξ-space are defined in Table 3.5.1.) This gives rise to a system of linear algebraic equations, viz.,

$$\begin{bmatrix} 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \end{bmatrix} \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{Bmatrix} = \begin{Bmatrix} x_1^e \\ x_2^e \\ x_3^e \\ x_4^e \\ x_5^e \\ x_6^e \\ x_7^e \\ x_8^e \end{Bmatrix} \tag{3.5.3}$$

TABLE 3.5.1 Nodal coordinates in ξ-space

a	ξ_a	η_a	ζ_a
1	−1	−1	−1
2	1	−1	−1
3	1	1	−1
4	−1	1	−1
5	−1	−1	1
6	1	−1	1
7	1	1	1
8	−1	1	1

The matrix may be inverted with the aid of the computer. Solving for the α's and substituting in (3.5.3) yields

$$x(\xi) = \sum_{a=1}^{8} N_a(\xi) x_a^e \tag{3.5.4}$$

where

$$\boxed{N_a(\xi, \eta, \zeta) = \tfrac{1}{8}(1 + \xi_a\xi)(1 + \eta_a\eta)(1 + \zeta_a\zeta)} \tag{3.5.5}$$

with similar expressions for $y(\xi)$ and $z(\xi)$. Observe that the right-hand side of (3.5.5) consists of the product of the one-dimensional shape functions derived in Sec. 1.12.

We define u^h by invoking the isoparametric concept, i.e.,

$$u^h(\xi) = \sum_{a=1}^{8} N_a(\xi) d_a^e \tag{3.5.6}$$

The continuity of the interpolation functions may be seen by observing the behavior of (3.5.6) along a face of Ω^e. For example, if $\zeta = -1$, (3.5.6) becomes

$$u^h(\xi, \eta, -1) = \sum_{a=1}^{4} \tfrac{1}{4}(1 + \xi_a \xi)(1 + \eta_a \eta) d_a^e \tag{3.5.7}$$

As can be seen from (3.5.7), u^h has hyperbolic paraboloidal variation on the face $\zeta = -1$ and is uniquely defined by the four nodal values associated with that face. The same conclusion can be drawn for any other element of this type which shares this face, and so the continuity of u^h is assured. If the element domain, Ω^e, is not too distorted, the shape functions will be smooth functions of $\boldsymbol{x}$. The completeness condition is guaranteed by virtue of the isoparametric hypothesis.

The hexahedron, or "brick" as it is often referred to, is the most generally useful shape in modeling three-dimensional domains. However, it is often convenient to have a wedge-shaped element at our disposal. The technique of degeneration may be employed to derive a wedge-shaped element from the hexahedron. Specifically, we coalesce nodes 3 and 4, and 7 and 8 (see Fig. 3.5.2). In this case (3.5.4) degenerates to

$$x = \sum_{a=1}^{3} N'_a x_a^e + \sum_{a=5}^{7} N'_a x_a^e \tag{3.5.8}$$

where

$$N'_a = \begin{cases} N_a & a = 1, 2, 5, 6 \\ N_a + N_{a+1} & a = 3, 7 \end{cases} \tag{3.5.9}$$

The Jacobian determinant becomes zero along the edge connecting nodes 3 and 7, but the x-derivatives of the shape functions are well behaved (cf. Sec. 3.4).

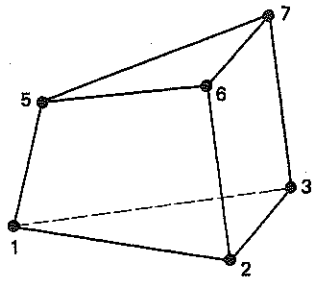


Figure 3.5.2 Wedge-shaped element formed by degenerating the eight-node hexahedral element.

A further coalescing of nodes 5, 6 and 7 yields the well-known linear tetrahedral element (see Fig. 3.5.3). Equation (3.5.8) becomes

$$x = \sum_{a=1}^{3} N'_a x^e_a + N''_5 x^e_5 \tag{3.5.10}$$

where

$$N''_5 = N'_5 + N'_6 + N'_7 \tag{3.5.11}$$

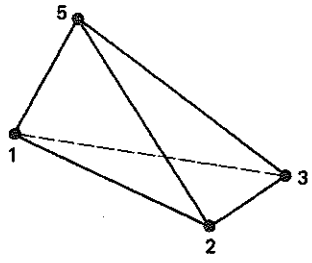


Figure 3.5.3 The linear tetrahedral element formed by degenerating the wedge-shaped element.

It can be shown that the x-derivatives of the shape functions are constants in this case (cf. Sec. 3.4). The linear tetrahedron was first proposed by Gallagher et al. [11].

3.6 HIGHER-ORDER ELEMENTS; LAGRANGE POLYNOMIALS

Thus far we have considered elements for which the shape functions are constructed from products of the linear one-dimensional shape functions. In this section we begin consideration of *higher-order elements*. These elements are often capable of more accurate representations than the *linear* elements considered previously and also allow more faithful representations of element domains, as the boundary edges and surfaces may be curved. At the same time they are generally more expensive than the basic linear elements. Thus the cost-effectiveness of the various elements is often in dispute. It appears that the optimal choice is very problem-dependent, so no single element is to be preferred exclusively.

Systematic derivation of certain higher-order isoparametric elements may be performed with the aid of the one-dimensional Lagrange polynomials. We first present the definition of the Lagrange polynomials and then, by way of examples, indicate the construction for multidimensional elements.

Lagrange Polynomials

The Lagrange polynomials are defined by

$$l_a^{n_{en}-1}(\xi) = \frac{\prod_{\substack{b=1 \\ b\neq a}}^{n_{en}} (\xi - \xi_b)}{\prod_{\substack{b=1 \\ b\neq a}}^{n_{en}} (\xi_a - \xi_b)}$$

($l_a^{n_{en}-1}$: Order of the polynomial; a: node)

$$= \frac{(\xi - \xi_1)(\xi - \xi_2)\cdots(\xi - \xi_{a-1})(\xi - \xi_{a+1})\cdots(\xi - \xi_{n_{en}})}{(\xi_a - \xi_1)(\xi_a - \xi_2)\cdots(\xi_a - \xi_{a-1})(\xi_a - \xi_{a+1})\cdots(\xi_a - \xi_{n_{en}})}$$

(a-term omitted; a-term omitted)

(3.6.1)

If the order of the polynomial is not important we omit the superscript and simply write l_a. It is simple to see from (3.6.1) that $l_a(\xi_a) = 1$ and, if $b \neq a$, $l_a(\xi_b) = 0$. In other words

$$l_a(\xi_b) = \delta_{ab} \tag{3.6.2}$$

We shall define the shape functions of an n_{en}-noded element in one dimension by the relation

$$N_a = l_a^{n_{en}-1} \tag{3.6.3}$$

The ξ_a's, as usual, denote the locations of the nodes in ξ-space. Equation (3.6.2) guarantees satisfaction of the interpolation property.

Example 1

Consider the two-noded, one-dimensional element presented in Sec. 1.12. We shall derive the shape functions by way of (3.6.3). From (3.6.3), we have that

$$l_1^1(\xi) = \frac{(\xi - \xi_2)}{(\xi_1 - \xi_2)} = \frac{(\xi - 1)}{-1 - (+1)} = \frac{1}{2}(1 - \xi) \tag{3.6.4}$$

$$l_2^1(\xi) = \frac{(\xi - \xi_1)}{(\xi_2 - \xi_1)} = \frac{(\xi + 1)}{1 - (-1)} = \frac{1}{2}(1 + \xi) \tag{3.6.5}$$

As the reader may see, (3.6.4) and (3.6.5) are the familiar shape functions of the one-dimensional, linear element.

Example 2

We shall now use (3.6.3) to derive the shape functions for a *quadratic* three-node element. We assume that the nodes are located at -1, 0, and $+1$ in ξ-space (see Fig.

3.6.1). The shape functions are given by

$$N_1(\xi) = l_1^2(\xi) = \frac{(\xi - \xi_2)(\xi - \xi_3)}{(\xi_1 - \xi_2)(\xi_1 - \xi_3)} = \frac{\xi(\xi - 1)}{(-1)(-2)} = \frac{1}{2}\xi(\xi - 1) \tag{3.6.6}$$

$$N_2(\xi) = l_2^2(\xi) = \frac{(\xi - \xi_1)(\xi - \xi_3)}{(\xi_2 - \xi_1)(\xi_2 - \xi_3)} = \frac{(\xi + 1)(\xi - 1)}{(1)(-1)} = 1 - \xi^2 \tag{3.6.7}$$

$$N_3(\xi) = l_3^2(\xi) = \frac{(\xi - \xi_1)(\xi - \xi_2)}{(\xi_3 - \xi_1)(\xi_3 - \xi_2)} = \frac{(\xi + 1)\xi}{(2)(1)} = \frac{1}{2}\xi(\xi + 1) \tag{3.6.8}$$

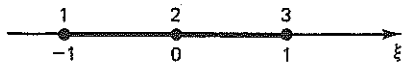


Figure 3.6.1 Node locations in ξ-space for the quadratic, three-node element.

These functions are depicted in Fig. 3.6.2.

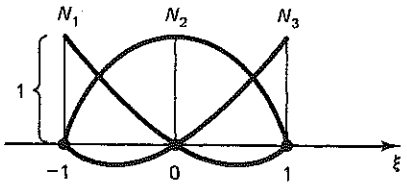


Figure 3.6.2 Shape functions for the quadratic, three-node element.

Exercise 1. Employ (3.6.1) to obtain the shape functions for the *cubic* four-noded (i.e., $n_{en} = 4$) element shown in Fig. 3.6.3. Sketch the results.

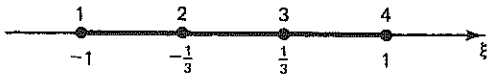


Figure 3.6.3 Node locations in ξ-space for the cubic, four-node element.

Example 3

The shape functions for the bilinear quadrilateral may be set up by taking products of the first order, Lagrange polynomials. The formula is

$$N_a(\xi, \eta) = l_b^1(\xi) l_c^1(\eta) \tag{3.6.9}$$

where the relationship among the indices is given in Table 3.6.1.

TABLE 3.6.1

a	b	c
1	1	1
2	2	1
3	2	2
4	1	2

The procedure is schematically illustrated in Fig. 3.6.4.

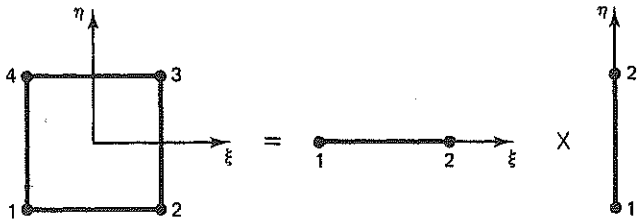


Figure 3.6.4 Schematic relationship between the linear, one-dimensional shape functions and the bilinear, two-dimensional shape functions.

Explicating (3.6.9), we have that

$$N_1(\xi, \eta) = l_1^1(\xi) l_1^1(\eta) = \tfrac{1}{4}(1 - \xi)(1 - \eta) \tag{3.6.10}$$

$$N_2(\xi, \eta) = l_2^1(\xi) l_1^1(\eta) = \tfrac{1}{4}(1 + \xi)(1 - \eta) \tag{3.6.11}$$

$$N_3(\xi, \eta) = l_2^1(\xi) l_2^1(\eta) = \tfrac{1}{4}(1 + \xi)(1 + \eta) \tag{3.6.12}$$

$$N_4(\xi, \eta) = l_1^1(\xi) l_2^1(\eta) = \tfrac{1}{4}(1 - \xi)(1 + \eta) \tag{3.6.13}$$

An analogous procedure may be employed to obtain the trilinear, three-dimensional shape functions of the previous section.

Example 4

Higher-order, two- and three-dimensional elements may be derived by taking products of Lagrange polynomials. We shall illustrate this idea by setting up the nine-node, two-dimensional Lagrange element. The element shape functions are the products of quadratic Lagrange polynomials. The setup is illustrated in Fig. 3.6.5. Note that the element domain Ω^e may have curved edges. The formula for the shape functions is

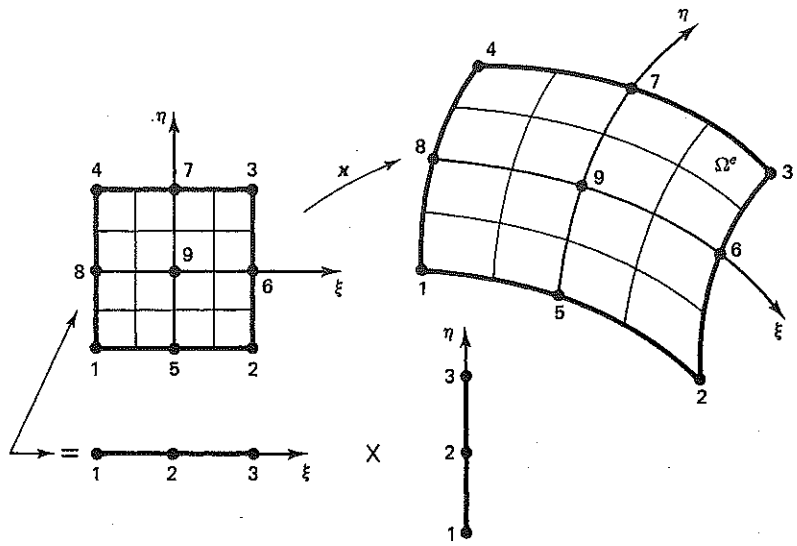


Figure 3.6.5 Nine-node, two-dimensional Lagrange element.

$$N_a(\xi, \eta) = l_b^2(\xi) l_c^2(\eta) \tag{3.6.14}$$

where the relationship among the indices is given in Table 3.6.2.

TABLE 3.6.2

a	b	c
1	1	1
2	3	1
3	3	3
4	1	3
5	2	1
6	3	2
7	2	3
8	1	2
9	2	2

Typical amongst (3.6.14) are

$$\begin{pmatrix}\text{Corner}\\ \text{node}\end{pmatrix} \qquad N_1(\xi, \eta) = l_1^2(\xi) l_1^2(\eta) = \tfrac{1}{4}\xi\eta(\xi - 1)(\eta - 1) \tag{3.6.15}$$

$$\begin{pmatrix}\text{Midside}\\ \text{node}\end{pmatrix} \qquad N_5(\xi, \eta) = l_2^2(\xi) l_1^2(\eta) = \tfrac{1}{2}\eta(1 - \xi^2)(\eta - 1) \tag{3.6.16}$$

$$\begin{pmatrix}\text{Middle}\\ \text{node}\end{pmatrix} \qquad N_9(\xi, \eta) = l_2^2(\xi) l_2^2(\eta) = (1 - \xi^2)(1 - \eta^2) \tag{3.6.17}$$

These functions are shown in Fig. 3.6.6; N_9 is frequently referred to as the ***bubble function***, for obvious reasons.

Remark

Elements for which the shape functions are formed from products of one-dimensional Lagrange polynomials are called ***Lagrange elements***.

Exercise 2. Use the results of Exercise 1 to derive the shape functions for the 16-node, two-dimensional Lagrange element.

Exercise 3. Use the quadratic one-dimensional shape functions to derive the shape functions for the triquadratic 27-node three-dimensional element.

Exercise 4. Use linear and quadratic shape functions to derive shape functions for the six-node quadrilateral depicted in Fig. 3.6.7.

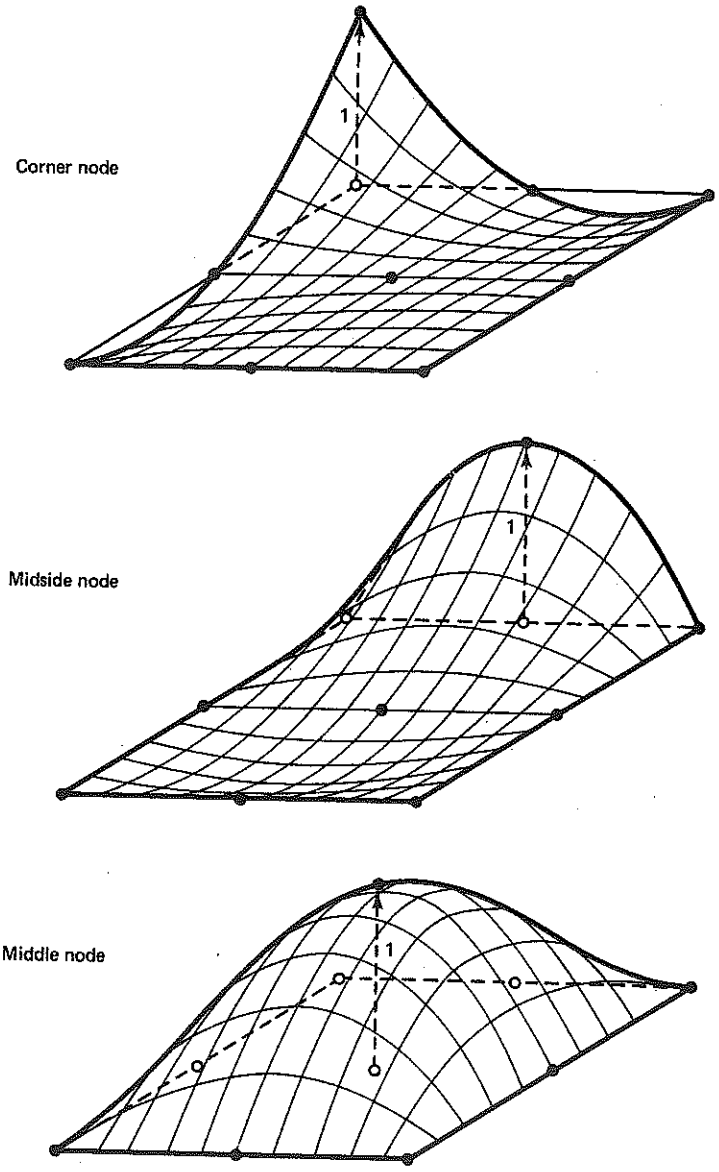


Figure 3.6.6 Typical Lagrange shape functions for the nine-node element.

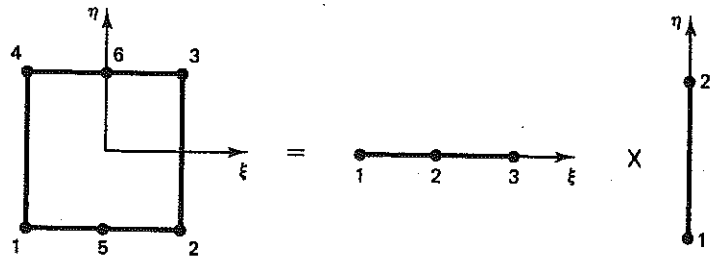


Figure 3.6.7

3.7 ELEMENTS WITH VARIABLE NUMBERS OF NODES

In this section we shall derive the shape functions of a two-dimensional "quadrilateral" element, which possesses four through nine nodes. When all nine nodes are present, the element is the Lagrange element considered in the previous section. When only four nodes are present, the element degenerates to the basic bilinear quadrilateral of Sec. 3.2. Any of nodes 5 through 9 (see Fig. 3.6.5) may be added or omitted. Thus either linear or quadratic behavior may be accommodated along any edge, which permits the "mixing" of basic linear and higher-order elements in one mesh.

We begin with the bilinear shape functions of the four-node quadrilateral element:

$$N_a(\xi, \eta) = \tfrac{1}{4}(1 + \xi_a\xi)(1 + \eta_a\eta), \qquad a = 1, 2, 3, 4 \tag{3.7.1}$$

Our first objective is to add the fifth node, permitting a curved edge and quadratic behavior along the edge connecting nodes 1 and 2 (see Fig. 3.7.1). Note that N_1 and N_2 do not vanish at node 5 (i.e., $N_a(\xi_5, \eta_5) = \frac{1}{2}$, $a = 1, 2$), an essential requirement of the shape functions of the five-node element. Shape functions 3 and 4 are satisfactory in this regard. Thus shape functions 1 and 2 will need to be modified. Let us first introduce a shape function for node 5. Let

$$N_5(\xi, \eta) = \underbrace{l_2^2(\xi)}_{\substack{\text{Middle-node}\\ \text{quadratic}\\ \text{Lagrange}\\ \text{polynomial}}} \underbrace{l_1^1(\eta)}_{\substack{\text{Left-node}\\ \text{linear}\\ \text{Lagrange}\\ \text{polynomial}}}$$

$$= \frac{1}{2}(1 - \xi^2)(1 - \eta) \tag{3.7.2}$$

Note that

$$N_5(\xi_a, \eta_a) = \delta_{a5}, \qquad a = 1, 2, \ldots, 5 \tag{3.7.3}$$

This shape function is illustrated in Fig. 3.7.2 and satisfies all requisite properties. In

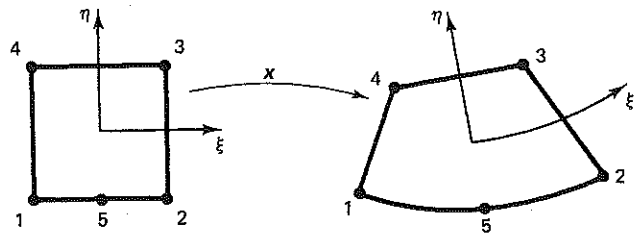


Figure 3.7.1 Five-node element.

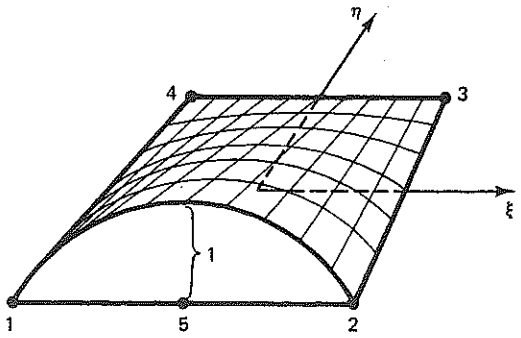


Figure 3.7.2

addition, we may employ it to correct the behavior of bilinear shape functions at node 5. To do this we subtract a multiple of N_5 from N_1 and N_2 so that the resulting functions vanish identically at node 5. Specifically, we define the new shape functions

$$N_a - \frac{1}{2}N_5, \qquad a = 1, 2 \tag{3.7.4}$$

It is easily seen that

$$N_a(\xi_b, \eta_b) - \frac{1}{2}N_5(\xi_b, \eta_b) = \delta_{ab}, \qquad a = 1, 2, \quad b = 1, 2, \ldots, 5 \tag{3.7.5}$$

and so (3.7.4) are appropriate shape functions for nodes 1 and 2 of the five-node element. These functions are illustrated in Fig. 3.7.3. In summary, the shape functions of the five-node element are given by (3.7.2), (3.7.4), and the original bilinear shape functions for nodes 3 and 4. Observe that in addition to introducing N_5, only the bilinear shape functions associated with the nodes along the edge containing node 5 needed to be modified. A similar situation occurs if we wish to add any of the other midside nodes. We may construct shape functions for midside nodes 6 through 8 in analogous fashion to the construction of N_5 (see (3.7.2)). These are

$$N_6(\xi, \eta) = \tfrac{1}{2}(1 - \eta^2)(1 + \xi) \tag{3.7.6}$$

$$N_7(\xi, \eta) = \tfrac{1}{2}(1 - \xi^2)(1 + \eta) \tag{3.7.7}$$

$$N_8(\xi, \eta) = \tfrac{1}{2}(1 - \eta^2)(1 - \xi) \tag{3.7.8}$$

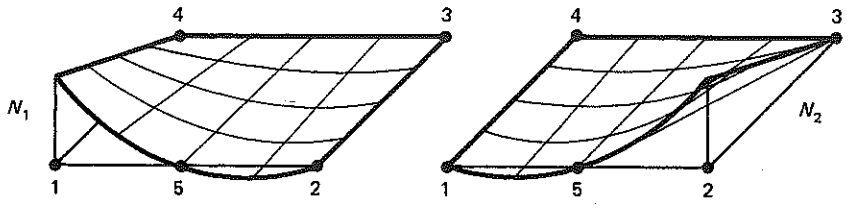


Figure 3.7.3

If any of nodes 5, 6, 7 and 8 are absent, we may formally define N_5, N_6, N_7, and N_8 to be identically zero, respectively. Then the modified shape functions for nodes 1 through 4 are

$$N_1 \leftarrow N_1 - \tfrac{1}{2}(N_5 + N_8) \tag{3.7.9}$$

$$N_2 \leftarrow N_2 - \tfrac{1}{2}(N_5 + N_6) \tag{3.7.10}$$

$$N_3 \leftarrow N_3 - \tfrac{1}{2}(N_6 + N_7) \tag{3.7.11}$$

$$N_4 \leftarrow N_4 - \tfrac{1}{2}(N_7 + N_8) \tag{3.7.12}$$

The reader may wish to verify that the modified shape functions, (3.7.9) through (3.7.12), satisfy

$$N_a(\xi_b, \eta_b) = \delta_{ab} \tag{3.7.13}$$

for all b corresponding to nodes present.

To include node 9 we introduce the ***bubble function:***

$$N_9(\xi, \eta) = (1 - \xi^2)(1 - \eta^2) \tag{3.7.14}$$

N_9 vanishes at nodes 1 through 8. However, $N_1, N_2, \ldots, N_8$ do not in general vanish at node 9, and therefore they must be modified. The procedure is carried out most expeditiously if the original bilinear shape functions and N_5, N_6, N_7, N_8 are first modified to account for the presence of node 9. This is accomplished as follows:

$a = 1, 2, 3, 4$:

$$N_a \leftarrow \underbrace{N_a}_{\substack{\text{Bilinear} \\ \text{shape functions; (3.7.1)}}} - \tfrac{1}{4}N_9 \tag{3.7.15}$$

$a = 5, 6, 7, 8$:

$$N_a = \begin{cases} \overbrace{N_a}^{\substack{\text{Defined by (3.7.2), (3.7.6)–(3.7.8)}}} - \tfrac{1}{2}N_9 & \text{(if node } a \text{ is present)} \\ 0 & \text{(if node } a \text{ is absent)} \end{cases} \tag{3.7.16}$$

Following this modification, the corrections indicated in (3.7.9) through (3.7.12) need be performed. The procedure is most succinctly summarized in a flowchart; see Fig. 3.7.4.

Care must also be taken that high-order elements are not too distorted. For example, in the case of the eight-node serendipity quadrilateral (see Fig. 3.7.5), all interior angles must be less than 180° (as for the four-node quadrilateral) and, in addition, the midside nodes should not migrate too far off the center of the sides. It is recommended in [12], p. 186, that the safe zone is confined to the middle third of the side. Although rules of thumb are useful for initiatory considerations, in general, we need rely on numerical checks of the sign of the Jacobian determinant to determine if the element geometry is "too distorted."

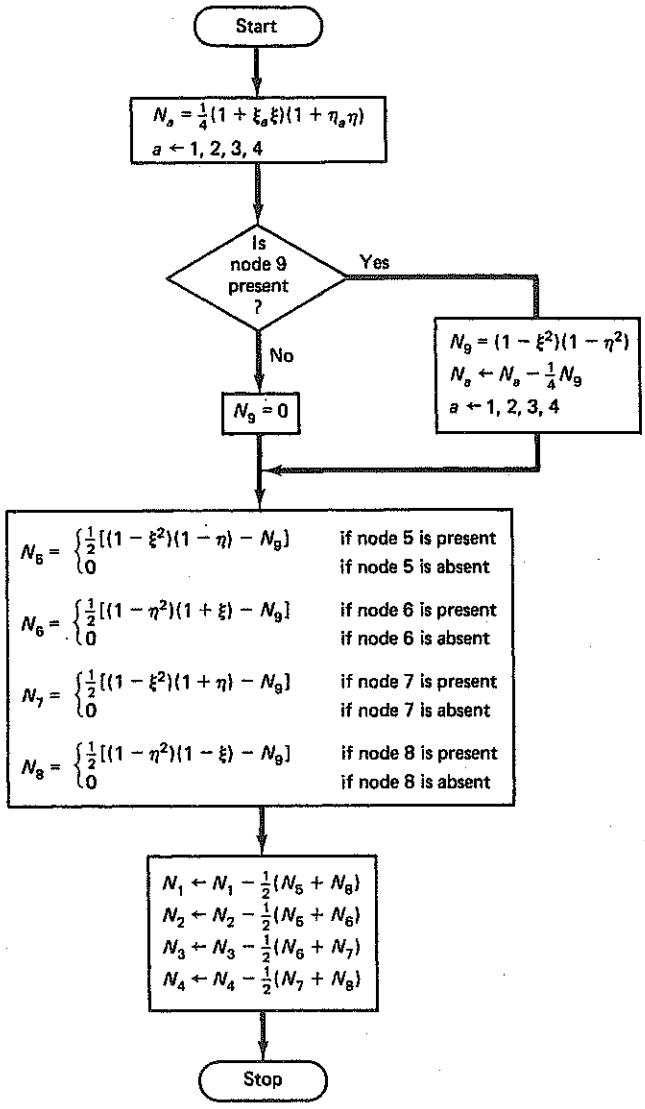


Figure 3.7.4 Calculation of a shape function for a four- through nine-node element.

Exercise 1. Use the flowchart in Fig. 3.7.4 to derive explicitly the shape functions of the eight-node ***serendipity quadrilateral*** for which the internal node (node 9) is omitted.

Exercise 2. Use the results of Exercise 1 and the degeneration technique to derive the shape functions of the six-node quadratic triangle; see Fig. 3.7.5.

Exercise 3. Generalize the flowchart of Fig. 3.7.4 so that the four- through nine-node element may be degenerated to a triangle.

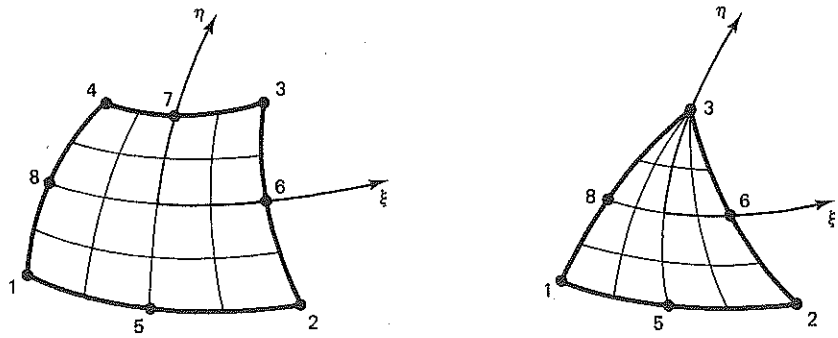


Figure 3.7.5 Degeneration of the eight-node serendipity element to the six-node triangle.

Exercise 4. Develop a flowchart analogous to the one in Fig. 3.7.4 for an 8- through 27-variable-number-of-nodes three-dimensional "brick" element. The node numbering is indicated in Fig. 3.7.6.

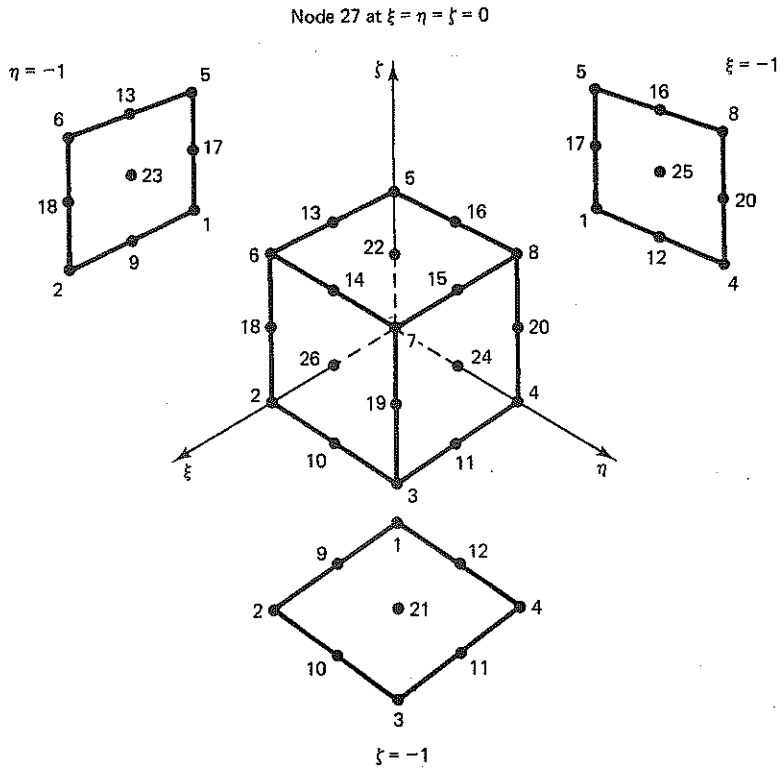
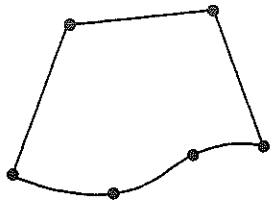


Figure 3.7.6 Nodal numbering for the 8- through 27-variable-number-of-nodes element.

Exercise 5. Generalize the flowchart of Exercise 4 so that the 8- through 27-node element may be degenerated to a wedge-shaped element.

Exercise 6. Develop a flowchart for a 4- through 16-variable-number-of-nodes quadrilateral element. (The 16-node element is the bicubic Lagrange element of Exercise 2, Sec. 3.6.)

Exercise 7. Develop shape functions for a six-node quadrilateral, which exhibits cubic behavior along one edge and linear behavior along the other three (see the accompanying figure).



Standard Element Families

In Fig. 3.7.7, some members of standard two-dimensional element families are shown. (Brick, wedge, and tetrahedral three-dimension analogs may easily be envisioned.) Isoparametric shape functions for all these elements may easily be derived by the techniques described in this section.

We may note from Fig. 3.7.7 that the serendipity elements have nodeless interiors. Beyond cubic level, the serendipity elements may be enhanced by the inclusion of some internal nodes to achieve higher degrees of polynomial completeness. This may be understood by considering the ***Pascal triangles*** shown in Fig. 3.7.8. As may be seen, for quartic and higher-order serendipity elements, the degree of polynomial completeness is fixed at cubic. The degree of completeness is generally considered to be the most important measure of accuracy of an element. Thus it is seen to be necessary to include functions associated with internal nodes in order to achieve higher-order accurate elements.

3.8 NUMERICAL INTEGRATION; GAUSSIAN QUADRATURE

Let $f: \Omega^e \subset \mathbb{R}^{n_{sd}} \rightarrow \mathbb{R}$ be a given function. We are interested in computing

$$\int_{\Omega^e} f(x)\, d\Omega \tag{3.8.1}$$

for purposes of constructing element arrays. (f may be thought of as any term in an integrand that we have encountered so far. For example, $f = \boldsymbol{B}_a^T \boldsymbol{D} \boldsymbol{B}_b$, the integrand of the stiffness matrix in heat conduction; see Sec. 2.5.) Throughout we shall assume f is smooth and integrable, so there is no ambiguity as to the meaning of (3.8.1). We

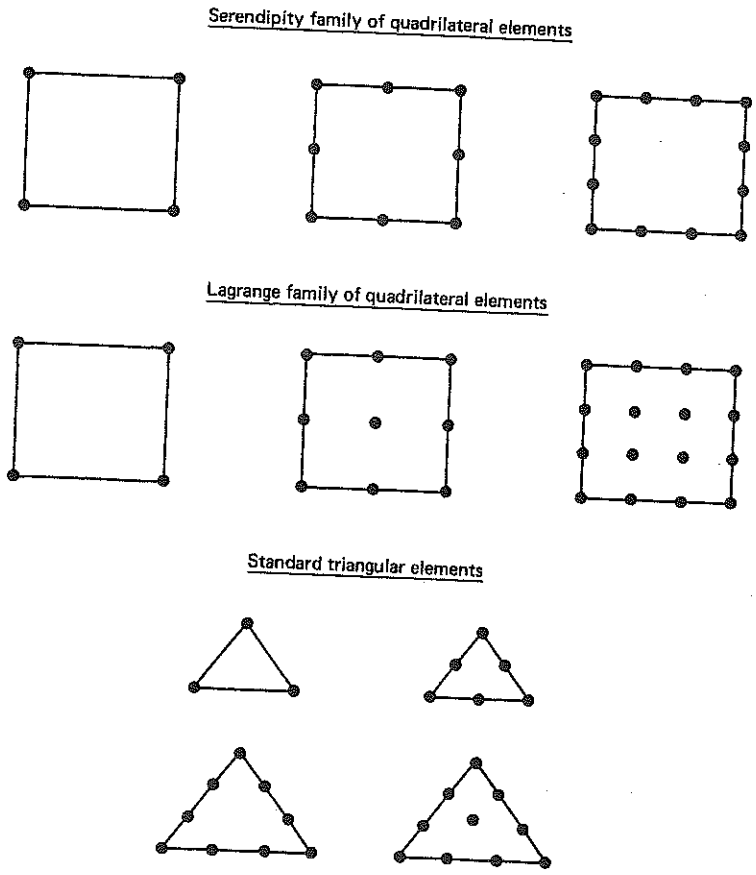


Figure 3.7.7 Standard two-dimensional element families. Note that the first members of the Lagrange and serendipity families are identical.

will need to make use of the change of variables formula. In each case we will pull back the integral over the element domain in x-space to one over the parent domain (i.e., the biunit n_{sd}-cube). For completeness, we begin by recalling the one-dimensional case (see Sec. 1.15):

$n_{sd} = 1$:

$$\int_{\Omega^e} f(x)\, dx = \int_{-1}^{1} f(x(\xi)) x_{,\xi}(\xi)\, d\xi \tag{3.8.2}$$

The two- and three-dimensional cases are given as follows:

$n_{sd} = 2$:

$$\int_{\Omega^e} f(x, y)\, d\Omega = \int_{-1}^{1} \int_{-1}^{1} f(x(\xi, \eta), y(\xi, \eta)) j(\xi, \eta)\, d\xi\, d\eta \tag{3.8.3}$$

Lagrange quadrilaterals

$$
\begin{array}{ccccccccccccccccccc}
 & & & & & & & & 1 & & & & & & & & & & \\
 & & & & & & & & \xi & & & & & & \eta & & & & \\
 & & & & & & \xi^2 & & & & \xi\eta & & & & \eta^2 & & & & \\
 & & & & \xi^3 & & & & \xi^2\eta & & & & \xi\eta^2 & & & & \eta^3 & & \\
 & & \xi^4 & & & & \xi^3\eta & & & & \xi^2\eta^2 & & & & \xi\eta^3 & & & & \eta^4 \\
 & \iddots & & & \xi^4\eta & & & & \xi^3\eta^2 & & & & \xi^2\eta^3 & & & & \xi\eta^4 & & & \ddots \\
\xi^m & & & & & & \xi^4\eta^2 & & & & \xi^3\eta^3 & & & & \xi^2\eta^4 & & & & & & \eta^m \\
 & \xi^m\eta & & & & & & & \xi^4\eta^3 & & & & \xi^3\eta^4 & & & & & & \xi\eta^m & & \\
 & & \xi^m\eta^2 & & & & & & & & \xi^4\eta^4 & & & & & & & \xi^2\eta^m & & & \\
 & & & & \xi^m\eta^3 & & & & & & & & & & & & \xi^3\eta^m & & & & \\
 & & & & & & \xi^m\eta^4 & & & & & & & & & & \xi^4\eta^m & & & & \\
 & & & & & & & \ddots & & & & & & \iddots & & & & & & & \\
 & & & & & & & & \xi^m\eta^m & & & & & & & & & & &
\end{array}
$$

Serendipity quadrilaterals

$$
\begin{array}{ccccccccccccccccccc}
 & & & & & & & & 1 & & & & & & & & & & \\
 & & & & & & & & \xi & & & & & & \eta & & & & \\
 & & & & & & \xi^2 & & & & \xi\eta & & & & \eta^2 & & & & \\
 & & & & \xi^3 & & & & \xi^2\eta & & & & \xi\eta^2 & & & & \eta^3 & & \\
 & & \xi^4 & & & & \xi^3\eta & & & & & & & & \xi\eta^3 & & & & \eta^4 \\
 & \iddots & & & \xi^4\eta & & & & & & & & & & & & \xi\eta^4 & & & \ddots \\
\xi^m & & \iddots & & & & & & & & & & & & & & & & & & \eta^m \\
 & \xi^m\eta & & & & & & & & & & & & & & & & & \xi\eta^m & & \\
 & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & &
\end{array}
$$

Triangles (internal nodal functions included)

$$
\begin{array}{ccccccccccccccccccc}
 & & & & & & & & 1 & & & & & & & & & & \\
 & & & & & & & & \xi & & & & & & \eta & & & & \\
 & & & & & & \xi^2 & & & & \xi\eta & & & & \eta^2 & & & & \\
 & & & & \xi^3 & & & & \xi^2\eta & & & & \xi\eta^2 & & & & \eta^3 & & \\
 & & \xi^4 & & & & \xi^3\eta & & & & \xi^2\eta^2 & & & & \xi\eta^3 & & & & \eta^4 \\
 & \iddots & & \iddots & & \iddots & & \iddots & & \iddots & & \iddots & & \iddots & & \iddots & & \iddots & \\
\xi^m & & \xi^{m-1}\eta & & \xi^{m-2}\eta^2 & & & \cdots & & & \xi^2\eta^{m-2} & & & & \xi\eta^{m-1} & & & & \eta^m
\end{array}
$$

Figure 3.7.8. Pascal triangles for standard two-dimensional element families.

$n_{sd} = 3$:

$$\int_{\Omega^e} f(x, y, z)\, d\Omega = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), z(\xi, \eta, \zeta))j(\xi, \eta, \zeta)\, d\xi\, d\eta\, d\zeta \tag{3.8.4}$$

Recall that $j = \det(\partial \boldsymbol{x}/\partial \boldsymbol{\xi})$, the Jacobian determinant.

In each case we have to evaluate an integral of the form:

$$\underbrace{\int_{-1}^{1} \cdots \int_{-1}^{1}}_{n_{sd} \text{ times}} \; \underbrace{g(\xi, \cdots)}_{n_{sd} \text{ arguments}} \; \underbrace{d\xi \cdots}_{n_{sd} \text{ differentials}} \tag{3.8.5}$$

Consider the one-dimensional case, in which we wish to evaluate

$$\int_{-1}^{1} g(\xi)\, d\xi \tag{3.8.6}$$

This can be approximately computed by way of a numerical integration (quadrature) formula as follows:

$$\int_{-1}^{1} g(\xi)\, d\xi = \sum_{l=1}^{n_{\text{int}}} g(\tilde{\xi}_l) W_l + R \cong \sum_{l=1}^{n_{\text{int}}} g(\tilde{\xi}_l) W_l \tag{3.8.7}$$

where n_{int} is the number of integration (quadrature) points, $\tilde{\xi}_l$ is the coordinate of the lth integration point, W_l is the "weight" of the lth integration point, and R is the remainder. The reader is probably familiar with some of the classical numerical integration formulas, such as the trapezoidal and Simpson's rules. We review these now as examples.

Example 1 (Trapezoidal rule)

$$n_{\text{int}} = 2$$
$$\tilde{\xi}_1 = -1$$
$$\tilde{\xi}_2 = 1$$
$$W_l = 1, \qquad l = 1, 2$$
$$R = -\tfrac{2}{3} g_{,\xi\xi}(\bar{\xi})$$

$\bar{\xi}$ denotes some point in the interval $[-1, 1]$. The trapezoidal rule is exact for constants

and linear polynomials but only approximate for quadratic and higher-order polynomials. It is said to be "second-order accurate."

Example 2 (Simpson's rule)

$$n_{\text{int}} = 3$$
$$\tilde{\xi}_1 = -1$$
$$\tilde{\xi}_2 = 0$$
$$\tilde{\xi}_3 = 1$$
$$W_1 = W_3 = \frac{1}{3}$$
$$W_2 = \frac{4}{3}$$
$$R = \frac{-g^{(4)}(\bar{\xi})}{90}$$

where $g^{(4)} = g_{,\xi\xi\xi\xi}$. Simpson's rule integrates general cubic polynomials exactly and is thus said to be "fourth-order accurate."

Although the above rules are widely used they are inefficient in the sense that there exist integration rules that are just as accurate but involve fewer integration points. *This issue is of great importance in practice since the fewer the integration points, the less the cost.* Considerable savings can be engendered by choosing an appropriate numerical integration rule.

Gaussian Quadrature

In one dimension the Gauss quadrature formulas are optimal. Accuracy of order $2n_{\text{int}}$ is achieved by n_{int} integration points. The locations of the quadrature points and values of associated weights are determined to attain maximum accuracy. The theory is thoroughly discussed in [13]. We give the first three Gaussian rules in the following:

Gaussian quadrature rules.

1. $n_{\text{int}} = 1$

$$\tilde{\xi}_1 = 0$$
$$W_1 = 2$$
$$R = \frac{g_{,\xi\xi}(\bar{\xi})}{3}$$

2. $n_{\text{int}} = 2$

$$\tilde{\xi}_1 = \frac{-1}{\sqrt{3}}$$

$$\tilde{\xi}_2 = \frac{1}{\sqrt{3}}$$

$$W_1 = W_2 = 1$$

$$R = \frac{g^{(4)}(\bar{\xi})}{135}$$

3. $n_{\text{int}} = 3$

$$\tilde{\xi}_1 = -\sqrt{\frac{3}{5}}$$

$$\tilde{\xi}_2 = 0$$

$$\tilde{\xi}_3 = \sqrt{\frac{3}{5}}$$

$$W_1 = W_3 = \frac{5}{9}$$

$$W_2 = \frac{8}{9}$$

$$R = \frac{g^{(6)}(\bar{\xi})}{15{,}750}$$

Exercise 1 Verify the accuracy of the preceding Gaussian rules by integrating the monomials $1, \xi, \xi^2, \ldots$ in turn.

The derivation of Gaussian quadrature formulas is illustrated by the following example.

Example 3 (Derivation of the Gaussian quadrature formula for $n_{\text{int}} = 2$)

We seek a rule involving two integration points that is exact for a general cubic polynomial. Let $g(\xi) = \alpha_0 + \alpha_1\xi + \alpha_2\xi^2 + \alpha_3\xi^3$, where the α's are arbitrary constants, and assume the integration points are equally weighted and symmetrically spaced (i.e., $W_1 = W_2$ and $\tilde{\xi}_1 = -\tilde{\xi}_2$, respectively). The exact integral of

$$\int_{-1}^{1} g(\xi)\, d\xi = 2\alpha_0 + \tfrac{2}{3}\alpha_2 \tag{3.8.8}$$

and this is to be equal to

$$\sum_{i=1}^{2} g(\tilde{\xi}_i)W_i = 2W_2(\alpha_0 + \alpha_2\tilde{\xi}_2^2) \tag{3.8.9}$$

This is to hold for arbitrary values of α_0 and α_2, and it thus follows that $W_2 = 1$ and $\tilde{\xi}_2 = 1/\sqrt{3}$.

Exercise 2. Derive the Gauss quadrature rule for $n_{\text{int}} = 3$. *Hint:* From considerations of symmetry, assume $\tilde{\xi}_1 = -\tilde{\xi}_3$, $\tilde{\xi}_2 = 0$ and $W_1 = W_3$.

General Gaussian Quadrature Rule

The general Gaussian quadrature rule is defined by the following:

$$W_l = \frac{2}{[(1 - \tilde{\xi}_l^2)(P'_{n_{\text{int}}}(\tilde{\xi}_l)^2]}, \qquad 1 \le l \le n_{\text{int}} \tag{3.8.10}$$

$$R = \frac{2^{2n_{\text{int}}+1}(n_{\text{int}}!)^4}{(2n_{\text{int}} + 1)[(2n_{\text{int}})!]^3} g_{,\underbrace{\xi\ldots\xi}_{2n_{\text{int}}\text{ times}}}(\bar{\xi}) \tag{3.8.11}$$

where $\tilde{\xi}_l$ is the lth zero of the Legendre polynomial $P_{n_{\text{int}}}(\xi)$, and $P'_{n_{\text{int}}}$ denotes the derivative of $P_{n_{\text{int}}}$. The Legendre polynomials are defined by

$$P_{n_{\text{int}}}(\xi) = \frac{1}{2^{n_{\text{int}}} n_{\text{int}}!} \frac{d^{n_{\text{int}}}}{d\xi^{n_{\text{int}}}}(\xi^2 - 1)^{n_{\text{int}}} \tag{3.8.12}$$

See [13] for tabulated values of $\tilde{\xi}_l$ and W_l.

Gaussian Quadrature in Several Dimensions

Gaussian rules for integrals in several dimensions are constructed by employing one-dimensional Gaussian rules on each coordinate separately. For example, in two dimensions

$$\begin{aligned}\int_{-1}^{1}\int_{-1}^{1} g(\xi, \eta)\, d\xi\, d\eta &\cong \int_{-1}^{1}\left\{\sum_{l^{(1)}=1}^{n_{\text{int}}^{(1)}} g(\tilde{\xi}^{(1)}_{l^{(1)}}, \eta) W^{(1)}_{l^{(1)}}\right\} d\eta \qquad \text{(Gaussian quadrature rule 1 applied to } \xi\text{-coordinate)} \\ &\cong \sum_{l^{(1)}=1}^{n_{\text{int}}^{(1)}} \sum_{l^{(2)}=1}^{n_{\text{int}}^{(2)}} g(\tilde{\xi}^{(1)}_{l^{(1)}}, \tilde{\eta}^{(2)}_{l^{(2)}}) W^{(1)}_{l^{(1)}} W^{(2)}_{l^{(2)}} \qquad \text{(Gaussian quadrature rule 2 applied to } \eta\text{-coordinate)}\end{aligned} \tag{3.8.13}$$

Example 4

If the one-point rule is used in each direction (3.8.13) specializes to

$$\int_{-1}^{1}\int_{-1}^{1} g(\xi, \eta)\, d\xi\, d\eta = 4g(0, 0) \tag{3.8.14}$$

Example 5

If the two-point rule is used in each direction, (3.8.13) specializes to

$$\int_{-1}^{1}\int_{-1}^{1} g(\xi, \eta)\, d\xi\, d\eta = g\left(\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) + g\left(\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) \tag{3.8.15}$$

It is frequently preferable to tabulate multidimensional rules in terms of a single index so that in place of (3.8.13) we would have

$$\int_{-1}^{1}\int_{-1}^{1} g(\xi, \eta)\, d\xi\, d\eta \cong \sum_{l=1}^{n_{\text{int}}} g(\tilde{\xi}_l, \tilde{\eta}_l) W_l \tag{3.8.16}$$

Comparison with (3.8.13) indicates that

$$n_{\text{int}} = n_{\text{int}}^{(1)} n_{\text{int}}^{(2)}, \qquad 1 \le l \le n_{\text{int}} \tag{3.8.17}$$

$$\tilde{\xi}_l = \tilde{\xi}_{l^{(1)}}^{(1)} \tag{3.8.18}$$

$$\tilde{\eta}_l = \tilde{\eta}_{l^{(2)}}^{(2)} \tag{3.8.19}$$

$$W_l = W_{l^{(1)}}^{(1)} W_{l^{(2)}}^{(2)} \tag{3.8.20}$$

where a tabular relationship among the indices must be established (e.g., see Table 3.8.1)

Example 6

If the one-point rule is used in each direction, corresponding to (3.8.17) through (3.8.20), we have

$$n_{\text{int}} = 1, \qquad \tilde{\xi}_1 = \tilde{\eta}_1 = 0, \qquad W_1 = 2 \cdot 2 = 4 \tag{3.8.21}$$

Thus (3.8.14), (3.8.16), and (3.8.21) are consistent.

Example 7

Consider use of the two-point rule in each direction. We assume the relationship among the indices is as given in Table 3.8.1.

TABLE 3.8.1

l	$l^{(1)}$	$l^{(2)}$
1	1	1
2	2	1
3	2	2
4	1	2

In this case (3.8.17) through (3.8.20) yield

$$n_{\text{int}} = 2 \cdot 2 = 4 \tag{3.8.22}$$

$$\tilde{\xi}_1 = \frac{-1}{\sqrt{3}}, \quad \tilde{\xi}_2 = \frac{1}{\sqrt{3}}, \quad \tilde{\xi}_3 = \frac{1}{\sqrt{3}}, \quad \tilde{\xi}_4 = \frac{-1}{\sqrt{3}} \tag{3.8.23}$$

$$\tilde{\eta}_1 = \frac{-1}{\sqrt{3}}, \quad \tilde{\eta}_2 = \frac{-1}{\sqrt{3}}, \quad \tilde{\eta}_3 = \frac{1}{\sqrt{3}}, \quad \tilde{\eta}_4 = \frac{1}{\sqrt{3}} \tag{3.8.24}$$

$$W_1 = W_2 = W_3 = W_4 = 1 \tag{3.8.25}$$

Clearly (3.8.15), (3.8.16), and (3.8.22) through (3.8.25) are consistent.

Exercise 3. Consider use of the three-point Gauss rule in each direction. Define a relationship between the indices l, $l^{(1)}$, and $l^{(2)}$ by constructing a table similar to Table 3.8.1. Determine $\tilde{\xi}_l$, $\tilde{\eta}_l$, and W_l, $1 \le l \le 9$.

Locations of the quadrature points for the one-point, 2×2, and 3×3 Gaussian rules are illustrated in Fig. 3.8.1.

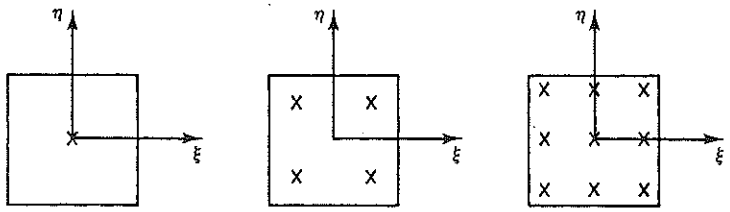


Figure 3.8.1 Locations of Gauss quadrature points for one-point, 2×2, and 3×3 rules.

Analogous arguments lead to the form of rules in three dimensions, viz.,

$$\begin{aligned}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} g(\xi, \eta, \zeta)\, d\xi\, d\eta\, d\zeta &\cong \sum_{l^{(1)}=1}^{n_{\text{int}}^{(1)}} \sum_{l^{(2)}=1}^{n_{\text{int}}^{(2)}} \sum_{l^{(3)}=1}^{n_{\text{int}}^{(3)}} g(\tilde{\xi}_{l^{(1)}}^{(1)}, \tilde{\eta}_{l^{(2)}}^{(2)}, \tilde{\zeta}_{l^{(3)}}^{(3)}) W_{l^{(1)}}^{(1)} W_{l^{(2)}}^{(2)} W_{l^{(3)}}^{(3)} \\ &= \sum_{l=1}^{n_{\text{int}}} g(\tilde{\xi}_l, \tilde{\eta}_l, \tilde{\zeta}_l) W_l\end{aligned} \tag{3.8.26}$$

Exercise 4. Determine $\tilde{\xi}_l$, $\tilde{\eta}_l$, $\tilde{\zeta}_l$, and W_l for the one-point and $2 \times 2 \times 2$ Gaussian rules in three dimensions.

In multidimensional cases the Gaussian quadrature rules are no longer necessarily optimal. For example, in three dimensions the six-point (non-Gaussian) rule given in Table 3.8.2 is fourth-order accurate [14]. The quadrature points are located at the centers of the six faces of the cube. (This means that all monomials of the form

$\xi^i\eta^j\zeta^k$, $0 \le i, j, k \le 3$, $i + j + k \le 3$, are exactly integrated.) The fourth-order Gaussian rule requires eight points ($2 \times 2 \times 2$ rule) and is thus more expensive to use.

TABLE 3.8.2 Fourth-Order Accurate, Six-Point Quadrature Rule in Three Dimensions

l	$\tilde{\xi}_l$	$\tilde{\eta}_l$	$\tilde{\zeta}_l$	W_l
1	1	0	0	4/3
2	−1	0	0	4/3
3	0	1	0	4/3
4	0	−1	0	4/3
5	0	0	1	4/3
6	0	0	−1	4/3

A 14-point rule for three dimensions that is sixth-order accurate is described in [15, 16]. This rule represents a considerable savings when compared with the 27-point ($3 \times 3 \times 3$) Gaussian rule [17].

Unfortunately, a general theory of optimal integration formulas for even such simple shapes as squares and cubes does not yet seem to be known. Despite some measure of inefficiency in multidimensional applications, Gaussian quadrature formulas are still widely used and will be sufficient for our present needs. Recent progress toward the development of improved rules is reported in [18].

Exercise 5. Consider the following four-point rule in two dimensions:

l	$\tilde{\xi}_l$	$\tilde{\eta}_l$	W_l
1	0	$+a$	1
2	0	$-a$	1
3	$+a$	0	1
4	$-a$	0	1

Determine a such that fourth-order accuracy is attained.

3.9 DERIVATIVES OF SHAPE FUNCTIONS AND SHAPE FUNCTION SUBROUTINES

We need to calculate explicitly the derivatives of the shape functions to construct the element stiffness matrices. For simplicity let us take the case $n_{sd} = 2$. The derivatives

of N_a with respect to x and y may be evaluated with the aid of the chain rule:

$$N_{a,x} = N_{a,\xi}\xi_{,x} + N_{a,\eta}\eta_{,x} \tag{3.9.1}$$

$$N_{a,y} = N_{a,\xi}\xi_{,y} + N_{a,\eta}\eta_{,y} \tag{3.9.2}$$

It is worthwhile to recast these relations in the following matrix form:

$$\langle N_{a,x} N_{a,y} \rangle = \langle N_{a,\xi} N_{a,\eta} \rangle \begin{bmatrix} \xi_{,x} & \xi_{,y} \\ \eta_{,x} & \eta_{,y} \end{bmatrix} \tag{3.9.3}$$

The derivatives $N_{a,\xi}$ and $N_{a,\eta}$ may be explicitly computed. However, the terms in the matrix cannot be directly computed since we do not have explicit expressions $\xi = \xi(x, y)$ and $\eta = \eta(x, y)$. On the other hand, we do have the inverse relations

$$x(\xi, \eta) = \sum_{a=1}^{n_{en}} N_a(\xi, \eta) x_a^e \tag{3.9.4}$$

$$y(\xi, \eta) = \sum_{a=1}^{n_{en}} N_a(\xi, \eta) y_a^e \tag{3.9.5}$$

which enables us to calculate the matrix

$$\boldsymbol{x}_{,\xi} = \begin{bmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{bmatrix} \tag{3.9.6}$$

The components are:

$$x_{,\xi} = \sum_{a=1}^{n_{en}} N_{a,\xi} x_a^e, \qquad x_{,\eta} = \sum_{a=1}^{n_{en}} N_{a,\eta} x_a^e \tag{3.9.7}$$

$$y_{,\xi} = \sum_{a=1}^{n_{en}} N_{a,\xi} y_a^e, \qquad y_{,\eta} = \sum_{a=1}^{n_{en}} N_{a,\eta} y_a^e \tag{3.9.8}$$

The matrix (3.9.6) is the inverse of the matrix in (3.9.3), i.e.,

$$\begin{bmatrix} \xi_{,x} & \xi_{,y} \\ \eta_{,x} & \eta_{,y} \end{bmatrix} = (\boldsymbol{x}_{,\xi})^{-1} = \frac{1}{j} \begin{bmatrix} y_{,\eta} & -x_{,\eta} \\ -y_{,\xi} & x_{,\xi} \end{bmatrix} \tag{3.9.9}$$

where

$$j = \det(\boldsymbol{x}_{,\xi}) = x_{,\xi} y_{,\eta} - x_{,\eta} y_{,\xi} \tag{3.9.10}$$

All the preceding relations are typically evaluated in ***shape function subroutines.*** It is convenient to segregate the calculations into those which can be performed once for

all elements of the same type (i.e., an ***element group***) and those which need to be repeated for each element.

For the element group, calculate integration-rule weights, shape functions, and local derivatives:

For $l = 1, \ldots, n_{\text{int}}$

Determine: W_l, $\bar{\xi}_l$, $\bar{\eta}_l$

For $a = 1, \ldots, n_{en}$

Calculate: $N_a(\bar{\xi}_l, \bar{\eta}_l)$, $N_{a,\xi}(\bar{\xi}_l, \bar{\eta}_l)$, $N_{a,\eta}(\bar{\xi}_l, \bar{\eta}_l)$

Subroutine QDCSHL performs these calculations for the four-node bilinear quadrilateral element in program DLEARN (see Chapter 11).

Given the eth element's coordinates $(x_a^e, y_a^e$ *where* $1 \le a \le n_{en})$, *calculate global derivatives of the shape functions and Jacobian determinants:*

For $l = 1, \ldots, n_{\text{int}}$

Calculate:

$$x_{,\xi}(\bar{\xi}_l, \bar{\eta}_l) = \sum_{a=1}^{n_{en}} N_{a,\xi}(\bar{\xi}_l, \bar{\eta}_l) x_a^e$$

$$x_{,\eta}(\bar{\xi}_l, \bar{\eta}_l) = \sum_{a=1}^{n_{en}} N_{a,\eta}(\bar{\xi}_l, \bar{\eta}_l) x_a^e$$

$$y_{,\xi}(\bar{\xi}_l, \bar{\eta}_l) = \sum_{a=1}^{n_{en}} N_{a,\xi}(\bar{\xi}_l, \bar{\eta}_l) y_a^e$$

$$y_{,\eta}(\bar{\xi}_l, \bar{\eta}_l) = \sum_{a=1}^{n_{en}} N_{a,\eta}(\bar{\xi}_l, \bar{\eta}_l) y_a^e$$

$$j(\bar{\xi}_l, \bar{\eta}_l) = x_{,\xi}(\bar{\xi}_l, \bar{\eta}_l) y_{,\eta}(\bar{\xi}_l, \bar{\eta}_l) - x_{,\eta}(\bar{\xi}_l, \bar{\eta}_l) y_{,\xi}(\bar{\xi}_l, \bar{\eta}_l)$$

For $a = 1, \ldots, n_{en}$

Calculate:

$$N_{a,x}(\bar{\xi}_l, \bar{\eta}_l) = \frac{N_{a,\xi}(\bar{\xi}_l, \bar{\eta}_l) y_{,\eta}(\bar{\xi}_l, \bar{\eta}_l) - N_{a,\eta}(\bar{\xi}_l, \bar{\eta}_l) y_{,\xi}(\bar{\xi}_l, \bar{\eta}_l)}{j(\bar{\xi}_l, \bar{\eta}_l)}$$

$$N_{a,y}(\bar{\xi}_l, \bar{\eta}_l) = \frac{-[N_{a,\xi}(\bar{\xi}_l, \bar{\eta}_l) x_{,\eta}(\bar{\xi}_l, \bar{\eta}_l) - N_{a,\eta}(\bar{\xi}_l, \bar{\eta}_l) x_{,\xi}(\bar{\xi}_l, \bar{\eta}_l)]}{j(\bar{\xi}_l, \bar{\eta}_l)}$$

In obtaining the formulas for $N_{a,x}$ and $N_{a,y}$, we have combined (3.9.3) and (3.9.9). Subroutine QDCSHG performs these calculations in DLEARN for both the four-node bilinear quadrilateral and three-node linear triangle (using the "degeneration" technique of Sec. 3.4). The logical variable QUAD is used to indicate whether the element under consideration is a quadrilateral or triangle:

QUAD = .TRUE. (quadrilateral)

QUAD = .FALSE. (triangle)

Table 3.9.1 translates the text notation into the FORTRAN names used in QDCSHL and QDCSHG. The reader should study these routines carefully.

TABLE 3.9.1 Notation for Shape Function Subroutines QDCSHL and QDCSHG

Notation	FORTRAN name	Array dimensions
n_{int}	NINT (= 4)	
n_{sd}	NSD (= 2)	
n_{en}	NEN (= 4)	
W_l	W(L)	NINT
$\tilde{\xi}_l$	R	
$\tilde{\eta}_l$	S	
$N_a(\tilde{\xi}_l, \tilde{\eta}_l)$	SHL(3,I,L)	NROWSH × NEN × NINT (= 3 × 4 × 4)
$N_{a,\xi}(\tilde{\xi}_l, \tilde{\eta}_l)$	SHL(1,I,L)	
$N_{a,\eta}(\tilde{\xi}_l, \tilde{\eta}_l)$	SHL(2,I,L)	
x_a^e	XL(1,I)	NSD × NEN
y_a^e	XL(2,I)	
$N_a(\tilde{\xi}_l, \tilde{\eta}_l)$	SHG(3,I,L)	NROWSH × NEN × NINT
$N_{a,x}(\tilde{\xi}_l, \tilde{\eta}_l)$	SHG(1,I,L)	
$N_{a,y}(\tilde{\xi}_l, \tilde{\eta}_l)$	SHG(2,I,L)	
$\frac{\xi_a}{2}$	RA(I)	NEN
$\frac{\eta_a}{2}$	SA(I)	NEN
$\begin{bmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{bmatrix}$	XS(I,J)	NSD × NSD
$j(\tilde{\xi}_l, \tilde{\eta}_l)$	DET(L)	NINT

The case $n_{sd} = 3$ proceeds analogously. The main results are summarized as follows:

$$\text{cof}_{11} = y_{,\eta}z_{,\zeta} - y_{,\zeta}z_{,\eta} \tag{3.9.11}$$

$$\text{cof}_{12} = y_{,\zeta}z_{,\xi} - y_{,\xi}z_{,\zeta} \tag{3.9.12}$$

$$\text{cof}_{13} = y_{,\xi}z_{,\eta} - y_{,\eta}z_{,\xi} \tag{3.9.13}$$

$$\text{cof}_{21} = z_{,\eta}x_{,\zeta} - z_{,\zeta}x_{,\eta} \tag{3.9.14}$$

$$\text{cof}_{22} = z_{,\xi}x_{,\xi} - z_{,\xi}x_{,\zeta} \tag{3.9.15}$$

$$\text{cof}_{23} = z_{,\xi}x_{,\eta} - z_{,\eta}x_{,\xi} \tag{3.9.16}$$

$$\text{cof}_{31} = x_{,\eta}y_{,\zeta} - x_{,\zeta}y_{,\eta} \tag{3.9.17}$$

$$\text{cof}_{32} = x_{,\zeta}y_{,\xi} - x_{,\xi}y_{,\zeta} \tag{3.9.18}$$

$$\text{cof}_{33} = x_{,\xi}y_{,\eta} - x_{,\eta}y_{,\xi} \tag{3.9.19}$$

$$j = x_{,\xi}\text{cof}_{11} + x_{,\eta}\text{cof}_{12} + x_{,\zeta}\text{cof}_{13} \tag{3.9.20}$$

$$N_{a,x} = \frac{N_{a,\xi}\text{cof}_{11} + N_{a,\eta}\text{cof}_{12} + N_{a,\zeta}\text{cof}_{13}}{j} \tag{3.9.21}$$

$$N_{a,y} = \frac{N_{a,\xi}\text{cof}_{21} + N_{a,\eta}\text{cof}_{22} + N_{a,\zeta}\text{cof}_{23}}{j} \tag{3.9.22}$$

$$N_{a,z} = \frac{N_{a,\xi}\text{cof}_{31} + N_{a,\eta}\text{cof}_{32} + N_{a,\zeta}\text{cof}_{33}}{j} \tag{3.9.23}$$

The cof_{ij}'s are the cofactors of the matrix $\boldsymbol{x}_{,\xi}$. (Recall the definition of matrix inverse: $(\boldsymbol{x}_{,\xi})^{-1} = (\mathbf{cof})^T/j$.)

Exercise 1. Program a shape function subroutine for the eight-node trilinear brick. (To do this it will be helpful first to generalize explicitly the notation table (Table 3.9.1) to the three-dimensional case. Mimic the style of routines QDCSHL and QDCSHG.)

Exercise 2. Program shape function subroutines for the four- through nine-node element described in Fig. 3.7.4. Assume the IEN array is made available to the routines through the argument list and that if IEN(a) = 0, then node a is absent for the element under consideration. The IEN array may be taken to have dimension 9 within the routines. (Note that all operations performed to calculate the shape functions need also be performed to calculate the derivatives.)

Exercise 3. Generalize the subroutine of Exercise 1 to allow for the possibility of degeneration to a wedge-shaped element. Assume that the logical variable BRICK is used where

BRICK = .TRUE. (brick)

BRICK = .FALSE. (wedge)

The reader may set many additional shape-function-subroutine exercises by considering the different possibilities discussed in previous sections of this chapter.

3.10 ELEMENT STIFFNESS FORMULATION

Programming an element stiffness matrix is an essential step in learning the finite element method. There are several ways to go about this. In this section we describe three of the most important implementational styles of stiffness coding. Each has attributes and it is recommended that the reader fully understand each of these procedures.

Implementation 1. The first implementation employs the following form of the stiffness:

$$\boldsymbol{k}^e = \int_{\Omega^e} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B}\, d\Omega = \int_{\square} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} j\, d\square \tag{3.10.1}$$

Applying numerical quadrature to (3.10.1) enables us to write

$$\boldsymbol{k}^e \cong \sum_{l=1}^{n_{\text{int}}} (\boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} j)\Big|_{\widetilde{\xi}_l} W_l \tag{3.10.2}$$

It simplifies the coding to combine j and W_l with $\boldsymbol{D}$. In this case we write

$$\widetilde{\boldsymbol{D}} = j(\widetilde{\xi}_l) W_l \boldsymbol{D} \tag{3.10.3}$$

and thus

$$\boxed{\boldsymbol{k}^e \cong \sum_{l=1}^{n_{\text{int}}} (\boldsymbol{B}^T \widetilde{\boldsymbol{D}} \boldsymbol{B})_l} \tag{3.10.4}$$

In the actual FORTRAN programming $\widetilde{\boldsymbol{D}}$ is written over $\boldsymbol{D}$. Thus no additional storage for $\widetilde{\boldsymbol{D}}$ is required. This implementation is used in subroutines QDCK (quadrilateral continuum $\boldsymbol{k}^e$) and TRUSK (truss $\boldsymbol{k}^e$) in DLEARN (see Chapter 11). The main steps are summarized next.

For $l = 1, \ldots, n_{\text{int}}$

Set up the strain-displacement matrix $\boldsymbol{B}$.
Set up the constitutive matrix $\widetilde{\boldsymbol{D}}$.
Multiply $\widetilde{\boldsymbol{D}} * \boldsymbol{B}$.
Multiply $\boldsymbol{B}^T * (\widetilde{\boldsymbol{D}}\boldsymbol{B})$, taking account of symmetry, and accumulate in $\boldsymbol{k}^e$.

Table 3.10.1 translates the text notation into the FORTRAN names used in QDCK and TRUSK (see also Table 3.9.1 for names of shape function variables).

TABLE 3.10.1 Notation for Element Stiffness Routines QDCK and TRUSK

Notation	FORTRAN name	Array dimensions
n_{ee}	NEE	
$\boldsymbol{B}$	B	NROWB[a] × NEE
$\boldsymbol{D}$	C	NROWB × NROWB
$\widetilde{\boldsymbol{D}}$	DMAT	NROWB × NROWB
$\widetilde{\boldsymbol{D}}\boldsymbol{B}$	DB	NROWB × NEE
$\boldsymbol{k}^e$	ELSTIF	NEE × NEE

[a] NROWB is a variable used to dimension the arrays. In subroutine QDCK, NROWB = 4. This dimension is needed, for example, in the axisymmetric case. In plane stress, however, only the first three rows are needed. To save calculations, another variable is introduced, namely, NSTR (3 in plane stress; 4 in the axisymmetric case), which is used to define the limits of the matrix multiplication loops. To appreciate this point, the reader should examine the calling statement to subroutine MULTAB in QDCK and subroutine MULTAB itself.

Exercise 1. Determine formulas for the numbers of multiplications and additions required to form an element stiffness by the above procedure.

Implementation 2. In the opinion of the author the method of coding just described is the most important because it is completely general and does not assume any special structure of $\boldsymbol{B}$ or $\boldsymbol{D}$. However, if special structure is present, then greater efficiency can be gained. For example, if there are zero entries in $\boldsymbol{B}$ and/or $\boldsymbol{D}$, then operations can be saved in the multiplications $\widetilde{\boldsymbol{D}} * \boldsymbol{B}$ and $\boldsymbol{B}^T * (\widetilde{\boldsymbol{D}}\boldsymbol{B})$. This requires programming special subroutines for these calculations.

Example 1

Consider two-dimensional, isotropic, plane-stress elasticity theory. The matrices $\boldsymbol{B}$ and $\boldsymbol{D}$ take the form:

$$\boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{n_{en}}] \tag{3.10.5}$$

$$\boldsymbol{B}_a = \begin{bmatrix} N_{a,x} & 0 \\ 0 & N_{a,y} \\ N_{a,y} & N_{a,x} \end{bmatrix}, \qquad 1 \leq a \leq n_{en} \tag{3.10.6}$$

$$\boldsymbol{D} = \begin{bmatrix} D_{11} & D_{12} & 0 \\ & D_{22} & 0 \\ \text{symmetric} & & D_{33} \end{bmatrix} \tag{3.10.7}$$

Recall that $\boldsymbol{k}^e$ can be written in terms of $n_{en} \times n_{en}$ nodal submatrices, $\boldsymbol{k}^e_{ab}$, where (see (2.9.6) through (2.9.9))

$$\underbrace{k^e_{ab}}_{n_{ed} \times n_{ed}} = \int_{\Omega^e} \boldsymbol{B}_a^T \boldsymbol{D} \boldsymbol{B}_b \, d\Omega \tag{3.10.8}$$

We can write (3.10.8) in numerical-integration form as follows:

$$\boxed{k^e_{ab} \cong \sum_{l=1}^{n_{\text{int}}} (\boldsymbol{B}_a^T \widetilde{\boldsymbol{D}} \boldsymbol{B}_b)_l} \tag{3.10.9}$$

The nodal submatrix k^e_{ab} has dimension 2×2 in two dimensions, viz.,

$$k^e_{ab} = \begin{bmatrix} k^e_{2a-1,\,2b-1} & k^e_{2a-1,\,2b} \\ k^e_{2a,\,2b-1} & k^e_{2a,\,2b} \end{bmatrix} \tag{3.10.10}$$

In the computer implementation, k^e may be computed by calculating the nodal submatrices, namely, (3.10.10), for each a and b such that $1 \le a, b \le n_{en}$. Only the nodal submatrices on or above the diagonal need be computed because the lower part is determined by symmetry. The procedure is given by the algorithm in the box.

For $l = 1, \ldots, n_{\text{int}}$
  Set up $\widetilde{\boldsymbol{D}}$.
  For $b = 1, \ldots, n_{en}$
    Let $B_1 = N_{b,x}$ and $B_2 = N_{b,y}$.
    Multiply $\widetilde{\boldsymbol{D}} * \boldsymbol{B}_b$, taking account of zeros in $\widetilde{\boldsymbol{D}}$ and $\boldsymbol{B}_b$:

$$\widetilde{\boldsymbol{D}}\boldsymbol{B}_b = \begin{bmatrix} \widetilde{D}B_{11} & \widetilde{D}B_{12} \\ \widetilde{D}B_{21} & \widetilde{D}B_{22} \\ \widetilde{D}B_{31} & \widetilde{D}B_{32} \end{bmatrix} = \begin{bmatrix} \widetilde{D}_{11}B_1 & \widetilde{D}_{12}B_2 \\ \widetilde{D}_{12}B_1 & \widetilde{D}_{22}B_2 \\ \widetilde{D}_{33}B_2 & \widetilde{D}_{33}B_1 \end{bmatrix}$$

    For $a = 1, \ldots, b$
      Let $B_1 = N_{a,x}$ and $B_2 = N_{a,y}$.
      Multiply $\boldsymbol{B}_a^T * (\widetilde{\boldsymbol{D}}\boldsymbol{B}_b)$, taking account of zeros in $\boldsymbol{B}_a$, and accumulate in k^e:

$$\begin{bmatrix} k^e_{2a-1,\,2b-1} & k^e_{2a-1,\,2b} \\ k^e_{2a,\,2b-1} & k^e_{2a,\,2b} \end{bmatrix} \leftarrow \begin{bmatrix} k^e_{2a-1,\,2b-1} & k^e_{2a-1,\,2b} \\ k^e_{2a,\,2b-1} & k^e_{2a,\,2b} \end{bmatrix} + \begin{bmatrix} (B_1\widetilde{D}B_{11} + B_2\widetilde{D}B_{31}) & (B_1\widetilde{D}B_{12} + B_2\widetilde{D}B_{32}) \\ (B_2\widetilde{D}B_{21} + B_1\widetilde{D}B_{31}) & (B_2\widetilde{D}B_{22} + B_1\widetilde{D}B_{32}) \end{bmatrix}$$

Exercise 2. Determine the numbers of multiplications and additions required to form a four-node element stiffness by the boxed procedure. Specialize the results of Exercise 1

to the present case and compare. (The savings in three dimensions are even more impressive!)

Implementation 3. This implementation was first proposed by Gupta and Mohraz [19]. It emanates from the following development. Consider two- and three-dimensional elasticity theory (see Secs. 2.7 through 2.11). Note that for these cases $n_{ed} = n_{sd}$ (i.e., the number of element degrees of freedom equals the number of space dimensions). Let

$$w_i^h = \sum_{a=1}^{n_{en}} c_{ia} N_a, \qquad u_i^h = \sum_{a=1}^{n_{en}} d_{ia} N_a \tag{3.10.11}$$

Recall the equivalent single-index representations of the coefficients,

$$c_p = c_{ia}, \qquad d_q = d_{jb} \tag{3.10.12}$$

where

$$p = n_{ed}(a - 1) + i, \qquad q = n_{ed}(b - 1) + j \tag{3.10.13}$$

With these we may write[2]

$$\begin{aligned}
\int_{\Omega^e} w^h_{(i,k)} c_{ikjl} u^h_{(j,l)}\, d\Omega &= \int_{\Omega^e} w^h_{i,k} c_{ikjl} u^h_{j,l}\, d\Omega \quad \text{(by the minor symmetries of the } c_{ijkl}\text{'s and Lemma 2 of Sec. 2.7)} \\
&= \sum_{a,b=1}^{n_{en}} c_{ia} \left(\int_{\Omega^e} N_{a,k} c_{ikjl} N_{b,l}\, d\Omega \right) d_{jb} \\
&= \sum_{a,b=1}^{n_{en}} c_{ia} k^e_{iajb} d_{jb} = \sum_{p,q=1}^{n_{ee}} c_p k^e_{pq} d_q
\end{aligned} \tag{3.10.14}$$

from which it follows that

$$k^e_{pq} = k^e_{iajb} = \int_{\Omega^e} N_{a,k} c_{ikjl} N_{b,l}\, d\Omega \tag{3.10.15}$$

It is convenient here to work with the four-indexed version of the element stiffness, namely, k^e_{iajb}. In FORTRAN, the two-indexed version and four-indexed version are stored in identical fashion in consecutive addresses in memory if the indices are arranged in the order given. This enables us to deal with the array ELSTIF (i.e., k^e) as a two-dimensional array in one routine and as a four-dimensional array in another.

Two-dimensional version	Four-dimensional version
DIMENSION ELSTIF(NEE, NEE)	DIMENSION ELSTIF(NED, NEN, NED, NEN)
ELSTIF(IP, JQ)	ELSTIF(I, IA, J, JB)
where	
IP = NED*(IA − 1) + I	
JQ = NED*(JB − 1) + J	

(3.10.16)

[2]Recall that the summation over repeated spatial indices (i.e., i, j, k, l) is in force.

Assume that the material occupying Ω^e is ***isotropic,*** that is,

$$c_{ikjl} = \mu(\delta_{ij}\delta_{kl} + \delta_{il}\delta_{kj}) + \lambda\delta_{ik}\delta_{jl} \tag{3.10.17}$$

and ***homogeneous,*** that is, λ and μ are constants. With these assumptions, (3.10.15) can be written as:

$$\begin{aligned} k^e_{iajb} &= (\mu(\delta_{ij}\delta_{kl} + \delta_{il}\delta_{kj}) + \lambda\delta_{ik}\delta_{jl}) \int_{\Omega^e} N_{a,k}N_{b,l}\, d\Omega \\ &= \mu\left(\delta_{ij}\int_{\Omega^e} N_{a,k}N_{b,k}\, d\Omega + \int_{\Omega^e} N_{a,j}N_{b,i}\, d\Omega\right) \\ &\quad + \lambda\int_{\Omega^e} N_{a,i}N_{b,j}\, d\Omega \end{aligned} \tag{3.10.18}$$

Thus to calculate the element stiffness, the expression

$$\int_{\Omega^e} N_{a,i}N_{b,j}\, d\Omega \cong \sum_{l=1}^{n_{\text{int}}} (N_{a,i}N_{b,j}j)\Big|_{\tilde{\xi}_l} W_l \tag{3.10.19}$$

must be evaluated first. Observe that (3.10.19) is symmetric under the interchange of indices $(i, a) \leftrightarrow (j, b)$. This fact reduces the number of calculations involved. The algorithm in the following box calculates the element stiffness based upon (3.10.18) and (3.10.19). Note that the integrals, (3.10.19), are first stored in k^e_{iajb}, which is then overwritten by the stiffness.

$c_1 := \lambda + \mu$
$c_2 := \mu$
$c_3 := \lambda$
For $l = 1, \ldots, n_{\text{int}}$
  const $= j(\tilde{\xi}_l)W_l$
  For $b = 1, \ldots, n_{en}$
    For $j = 1, \ldots, n_{ed}$
      temp $=$ const $\cdot N_{b,j}(\tilde{\xi}_l)$
      For $a = 1, \ldots, b$
        For $i = 1, \ldots, j$
        $k^e_{iajb} \leftarrow k^e_{iajb} +$ temp $\cdot N_{a,i}(\tilde{\xi}_l)$
For $b = 1, \ldots, n_{en}$
  For $a = 1, \ldots, b$
    temp $= \sum_{k=1}^{n_{ed}} k^e_{kakb}$

```
For j = 1, . . . , n_ed
    For i = 1, . . . , j
        if i = j, then
            k^e_iaib ← c1 k^e_iaib + c2 temp
        else
            if a = b, then
                k^e_iaja ← c1 k^e_iaja
            else
                k^e_iajb ← c3 k^e_iajb + c2 k^e_jaib
            endif
        endif
```

Exercise 3. Determine the number of multiplications and additions for a four-node element stiffness and compare with results of Exercise 2.

Exercise 4. Determine the numbers of multiplications and additions for an eight-node brick element for each of the three implementations. Compare.

Discussion

The first implementation is the most general and easiest to program. The second is somewhat more specialized but registers a significant gain in efficiency. The third is the most specialized and most efficient. In anticipation of extending the present ideas to nonlinear situations, we may note that the matrix $\boldsymbol{D}$ is typically full. Furthermore, a popular method for improving the behavior of elements in both linear and nonlinear analysis engenders a full $\boldsymbol{B}$. (This method is described in Sec. 4.5.) Under these circumstances the first implementation is still applicable, whereas the others are not. However, the latter two implementations are preferable for the cases to which they pertain. The "best" implementation clearly depends upon what our goals are. For additional information on finite element programming see [20–24].

3.11 ADDITIONAL EXERCISES

Exercise 1. Consider a boundary-value problem in which

$$\Omega = \{(x, y) \mid x \in]a, b[\,, y \in]a, b[\}$$

A mesh of isoparametric elements is proposed for obtaining an approximate solution to the boundary-value problem (see Figure 3.11.1), in which elements 1 and 2 are eight-node quadrilaterals and elements 3 and 4 are four-node quadrilaterals. What fundamental requirement of the finite element spaces is not met by this mesh? Without changing the number of nodes, how would you fix the mesh?

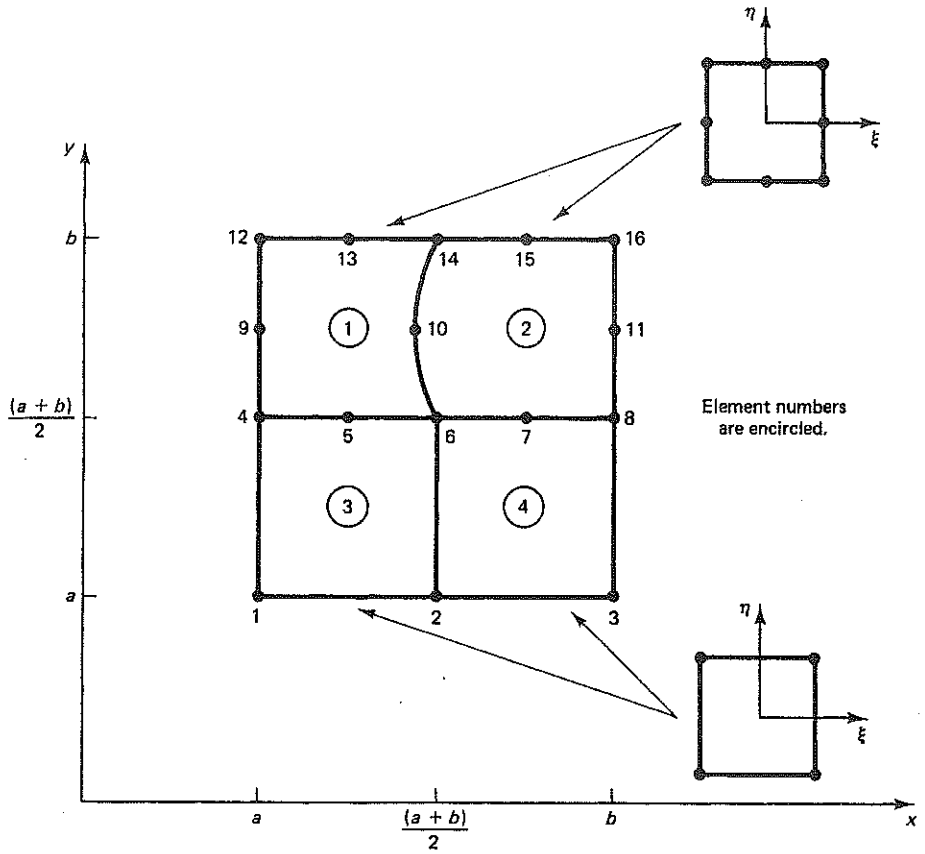


Figure 3.11.1

Exercise 2. Consider a one-dimensional, quadratic, three-node element shown in Fig. 3.11.2. The following relations hold:

$$x(\xi) = \sum_{a=1}^{3} N_a(\xi) x_a^e \qquad h^e = x_3^e - x_1^e$$

$$u^h(\xi) = \sum_{a=1}^{3} N_a(\xi) d_a^e \qquad f^e = \{f_a^e\}$$

$$N_1(\xi) = \tfrac{1}{2}\xi(\xi - 1) \qquad f_a^e = \int_{x_1^e}^{x_3^e} N_a \ell \, dx$$

$$N_2(\xi) = 1 - \xi^2 \qquad k^e = [k_{ab}^e]$$

$$N_3(\xi) = \tfrac{1}{2}\xi(\xi + 1) \qquad k_{ab}^e = \int_{x_1^e}^{x_3^e} N_{a,x} N_{b,x} \, dx, \quad 1 \le a, b \le 3$$

Given a "loading" ℓ, the elements of the consistently derived "force" vector f^e are sometimes surprising, especially for higher-order elements. This problem establishes

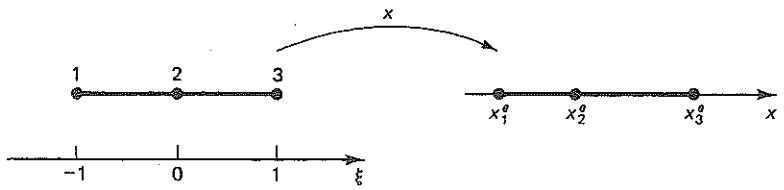


Figure 3.11.2

some basic results concerning the calculation of f^e, appropriate quadrature formulas, and Barlow stress points for the three-node element.

a. Assume ℓ = constant and let $x_2^e = (x_1^e + x_3^e)/2$. Determine exact expressions for f_a^e, $a = 1, 2, 3$.

b.[3] Assume $\ell = \delta(x - \bar{x})$, $x_1^e \le \bar{x} \le x_3^e$, (the delta function) and let $x_2^e = (x_1^e + x_3^e)/2$. Obtain an exact expression for f_a^e, $a = 1, 2, 3$, and specialize for the cases $\bar{x} = x_1^e$ and $\bar{x} = x_2^e$.

c. Assume ℓ = constant, but make no assumption on the location of x_2^e other than $x_1^e < x_2^e < x_3^e$. Determine the lowest-order Gaussian quadrature formula ($n_{\text{int}} = ?$) which exactly integrates f^e. Justify your answer.

d. Assume $x_2^e = (x_1^e + x_3^e)/2$. Determine the lowest-order Gaussian quadrature formula ($n_{\text{int}} = ?$) which exactly integrates k^e. Justify your answer.

e. An analysis of the quadratic element is being performed to locate the Barlow stress points (i.e., the points at which $e_{,x} = u^h_{,x} - u_{,x}$ optimally converges). It is assumed in the analysis that $x_2^e = (x_1^e + x_3^e)/2$ and the nodal values are exact, that is, $d_a^e = u(x_a)$. Determine the Barlow stress points.

Solution of part (e) We present a simple solution to illustrate that we do not have to perform a lot of algebraic manipulations to solve problems of this type.

Note that the position of node 2 implies that

[3]We need to be careful in changing variables when delta functions are present. Consider the multidimensional case in which

$$d\Omega = j\, d\square$$

The delta function transforms by multiplication with the *inverse* Jacobian determinant:

$$\delta_{\bar{x}} = \delta(x - \bar{x}) = j^{-1}\delta(\xi - \bar{\xi}) = j^{-1}\delta_{\bar{\xi}}$$

where

$$\bar{x} = x(\bar{\xi})$$

In one dimension, this specializes as follows:

$$dx = x_{,\xi}(\xi)\, d\xi$$

$$\delta_{\bar{x}} = \delta(x - \bar{x}) = (x_{,\xi})^{-1}\delta(\xi - \bar{\xi}) = (x_{,\xi})^{-1}\delta_{\bar{\xi}}$$

$$\bar{x} = x(\bar{\xi})$$

Further details concerning transformation rules for generalized functions may be found in Chapter 5 of I. Stakgold, *Boundary Value Problems of Mathematical Physics*, vol. II. New York: Macmillan, 1968.

$$e_{,x} = e_{,\xi}\xi_{,x} = \frac{2e_{,\xi}}{h^e}$$

Observe that if $u(\xi)$ is a quadratic polynomial, then $u^h(\xi) = u(\xi)$ by the assumption that the nodal values are exact. Therefore, take $u(\xi) = c\xi^3$, where c is an arbitrary constant. Compute as follows:

$$\begin{aligned} e_{,\xi} &= u^h_{,\xi} - u_{,\xi} \\ &= c(-\xi + \tfrac{1}{2} + \xi + \tfrac{1}{2} - 3\xi^2) \end{aligned}$$

Thus $e_{,\xi} = 0$ at $\xi = \pm 1\sqrt{3}$, the Gauss points of the two-point rule. If a formal Taylor-expansion approach is adopted, the coefficient of the $u_{,\xi\xi\xi}$ term leads to the same result after considerably more work!

f. Let

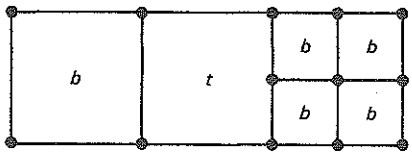
$$x_2^e = x_1^e + \frac{h^e}{4} \qquad \text{(quarter point)}$$

$$r = \frac{x - x_1^e}{h^e}$$

$$h^e = x_3^e - x_1^e$$

Determine an expression for $u^h_{,r}(r)$ and indicate the order of the singularity at $r = 0$ [i.e., determine α, where $u^h_{,r} = O(r^{-\alpha})$]. (This result is useful for the construction of ***crack elements***.)

Exercise 3. It is desired to develop a ***transition element,*** which will facilitate abrupt changes in mesh refinement between domains consisting of bilinear quadrilateral elements. The idea is illustrated in Fig. 3.11.3. The transition element is to be designed to maintain continuity across all interfaces. A shape function associated with node 5 which achieves this end is illustrated in Fig. 3.11.4. Starting with the usual bilinear shape functions, determine the appropriate modifications. Sketch the modified shape function associated with node 1.



Key: b = bilinear quadrilateral
t = transition element

Figure 3.11.3

Exercise 4. Let

$$\begin{Bmatrix} r(\xi, \eta) \\ \theta(\xi, \eta) \end{Bmatrix} = \sum_{a=1}^{4} N_a(\xi, \eta) \begin{Bmatrix} r_a^e \\ \theta_a^e \end{Bmatrix}$$

Determine the N_a's for a ***sector element*** (see Fig. 3.11.5), so that the following conditions are met:

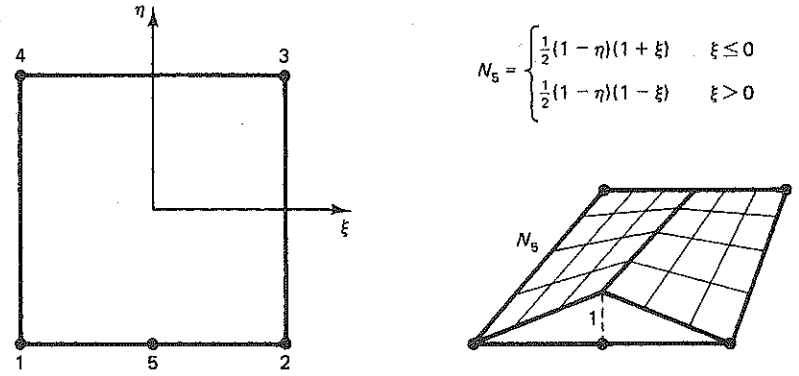


Figure 3.11.4

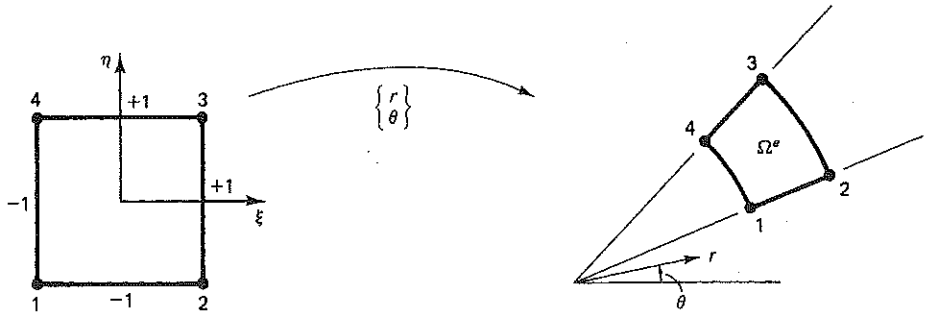


Figure 3.11.5

i. If $r_4 = r_1$, then $r(-1, \eta) = r_1$.
ii. If $r_3 = r_2$, then $r(+1, \eta) = r_2$.
iii. If $\theta_2 = \theta_1$, then $\theta(\xi, -1) = \theta_1$.
iv. If $\theta_4 = \theta_3$, then $\theta(\xi, +1) = \theta_3$.

Exercise 5. (See the solution of Exercise 2, part (e).) Determine the Barlow stress points for the four-node element in one dimension. Assume that

$$x_2^e = x_1^e + \frac{h^e}{3}$$

$$x_3^e = x_1^e + \frac{2h^e}{3}$$

$$h^e = x_4^e - x_1^e$$

$$d_a^e = u(x_a^e)$$

The regular positioning of the internal nodes implies $x_{,\xi} = h^e/2$. (*Ans.*: 0, $\pm (\frac{5}{9})^{1/2}$. The three Barlow points in this example do *not* coincide with the Gauss points of the three-point rule!)

Exercise 6. A boundary-value problem for the ***convection-diffusion equation*** is as follows: Given $\ell : \Omega \to \mathbb{R}$, $g : \Gamma_g \to \mathbb{R}$, and $h : \Gamma_h \to \mathbb{R}$, find $u : \overline{\Omega} \to \mathbb{R}$ such that

$$b_i u_{,i} = (\kappa_{ij} u_{,j})_{,i} + \ell \qquad \text{in } \Omega$$
$$u = g \qquad \text{on } \Gamma_g$$
$$\kappa_{ij} u_{,j} n_i = h \qquad \text{on } \Gamma_h$$

a. Establish a weak formulation for this problem in which the boundary condition on Γ_g is essential and the boundary condition on Γ_h is natural.
b. Establish the Galerkin formulation and obtain expressions for K_{PQ} and F_P.
c. Show that if $b_{j,j} = 0$ on Ω and $b_j n_j = 0$ on Γ_h, then the b-term contribution to K_{PQ} is skew-symmetric (i.e., $K^b_{PQ} = -K^b_{QP}$).

Exercise 7.

a. Consider the four-node bilinear element. Prove that the one-point Gauss quadrature formula in two dimensions is sufficient to exactly integrate the area

$$\int_{\Omega^e} d\Omega$$

b. Let

$$u_i^h(\xi, \eta) = \sum_{a=1}^{4} N_a(\xi, \eta) d_{ia}^e, \qquad i = 1, 2$$

where the N_a's are the bilinear shape functions. Show that if

$$u^h_{i,i}(0, 0) = 0$$

then

$$\int_{\Omega^e} u^h_{i,i}\, d\Omega = 0 \qquad \text{("mean incompressibility")}$$

Exercise 8. Consider a bilinear quadrilateral element. Assume that the prescribed surface traction (i.e., h_i) along one edge is constant. What is the lowest-order, one-dimensional Gauss quadrature formula which will exactly integrate the prescribed traction contribution to f^e? Justify your answer.

Solution The setup is shown in Fig. 3.11.6. Note that

$$s = \overbrace{N_a(\xi)}^{\text{Piecewise linear shape functions}} s_a + N_b(\xi) s_b$$

Calculate as follows:

$$\int_{\Gamma_{h_i}} N_a h_i \, d\Gamma = \int_0^L N_a h_i \, ds \qquad \text{(continued)}$$

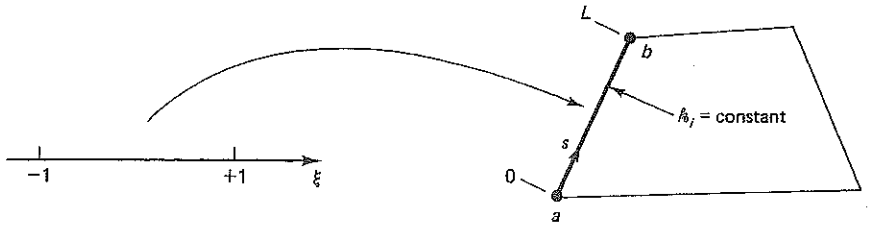


Figure 3.11.6

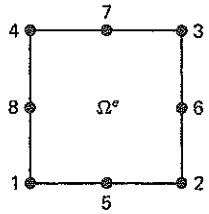
$$= h_i \int_{-1}^{+1} N_a(\xi) \frac{L}{2}\, d\xi$$

$$= \frac{L}{2} h_i \int_{-1}^{+1} \underbrace{N_a(\xi)}_{\text{Linear}}\, d\xi$$

$$= \frac{L}{2} h_i$$

Because the integral involves a linear polynomial in ξ, one-point Gauss quadrature suffices.

Exercise 9. Consider the eight-node isoparametric element shown:



Assume the nodal coordinates are as follows:

a	x_a	y_a
1	$-\frac{h}{2}$	$-\frac{h}{2}$
2	$\frac{h}{2}$	$-\frac{h}{2}$
3	$\frac{h}{2}$	$\frac{h}{2}$
4	$-\frac{h}{2}$	$\frac{h}{2}$
5	0	$-\frac{h}{2}$
6	$\frac{h}{2}$	0
7	0	$\frac{h}{2}$
8	$-\frac{h}{2}$	0

Consider the following body force:

$$\ell = \begin{Bmatrix} 0 \\ -g \end{Bmatrix}; \qquad g \text{ constant}$$

Compute (by hand) the nodal forces for nodes 1 and 5, i.e.,

$$f_{21}^e = \int_{\Omega^e} N_1 \ell_2 \, d\Omega$$

$$f_{25}^e = \int_{\Omega^e} N_5 \ell_2 \, d\Omega$$

(Note that $f_{1a} = 0$, $1 \leq a \leq 8$, and, by symmetry, $f_{21}^e = f_{22}^e = f_{23}^e = f_{24}^e$ and $f_{25}^e = f_{26}^e = f_{27}^e = f_{28}^e$.)

The results of this simple computation should be somewhat surprising. Comment.

Exercise 10. Generalize Exercise 8 to the case in which the quadrilateral is of biquadratic type. Assume that the three nodes along an edge define a curved edge (i.e., they do not lie along a straight line).

Exercise 11. Generalize Exercise 10 to the case in which h_i represents normal pressure. That is,

$$h_i = -pn_i$$

where p, the applied pressure, is assumed constant and n_i is the unit outward normal to the boundary.

Exercise 12. Consider a trilinear brick element subjected to normal pressure along one surface. What is the lowest-order, two-dimensional Gauss quadrature formula which will exactly integrate the prescribed traction contribution to f^e? Justify your answer. *Hint:* The following change-of-variables formula is useful in situations like these:

$$\int_{\Gamma_{h_i}} N_a h_i \, d\Gamma = \int_{-1}^{+1} \int_{-1}^{+1} N_a h_i \, \| \boldsymbol{x}_{,\xi} \times \boldsymbol{x}_{,\eta} \| \, d\xi \, d\eta$$

where $\times$ denotes cross product and $\| \cdot \|$ is the Euclidean length.

Exercise 13. Generalize Exercise 12 to the case of the triquadratic brick element.

Appendix 3.I

Triangular and Tetrahedral Elements

It is often stated that triangular and tetrahedral elements are responsible for the geometric flexibility of the finite element method. This is perhaps somewhat of an exaggeration as triangular and tetrahedral shapes are often not needed in practice. Most regions are conveniently discretized by arbitrary quadrilateral and brick-shaped elements, as described earlier.

It is also often stated that triangles and tetrahedra enable modeling of particularly intricate geometries and that these shapes facilitate transition from coarsely meshed zones of a grid to finely meshed zones. This is, of course, true, but quadrilaterals and bricks are capable of doing the same thing at least to some degree. To illustrate this point, consider Fig. 3.I.1 in which a triangular zone is discretized into three quadrilaterals. Thus we see that any triangular element could be replaced by quadrilaterals. (Mesh generation for triangular regions via quadrilaterals may be handled similarly; see Fig. 3.I.2.) Quadrilaterals may also be used to perform mesh transition as illustrated in Fig. 3.I.3.

Nevertheless, if a few triangles are desired (or tetrahedra, or wedges) in any particular situation, they may be easily obtained from quadrilaterals (respectively, bricks) by way of the element degeneration technique described previously. Consequently, no special finite element subroutines are required.

However, because many individuals are fond of particular triangular and tetrahedral elements and do not perceive them as being degenerated from quadrilaterals and bricks and because there are a number of useful triangular and tetrahedral elements for which there are no quadrilateral and brick counterparts, we shall describe the machinery necessary for the direct development of triangular and tetrahedral shape functions. The following techniques should also be employed if extensive use of triangles or tetrahedra is envisioned, because, in this case, the degeneration procedure proves inefficient.

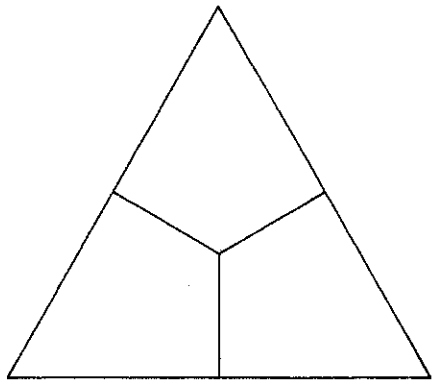


Figure 3.I.1

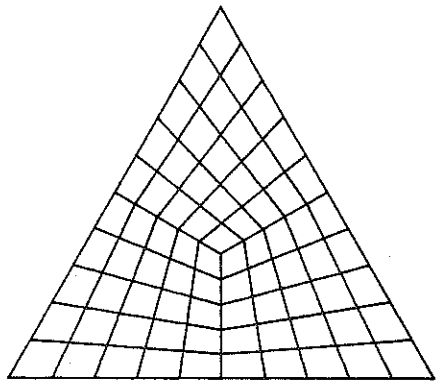


Figure 3.I.2

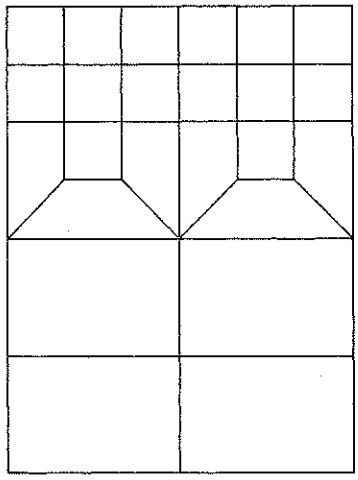


Figure 3.I.3

Shape Functions for Triangles

Consider the ***parent triangle*** illustrated in Fig. 3.I.4. This serves as the counterpart of the biunit square in ξ, η-space which was described in Sec. 3.2. We take r and s as the independent natural coordinates. Note that the equation of the inclined edge is

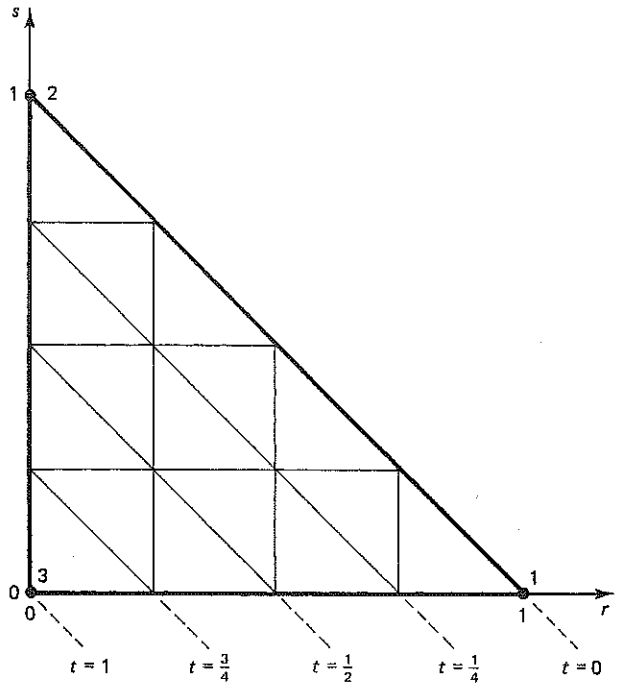


Figure 3.I.4

$1 - r - s = 0$. If we define

$$t = t(r, s) = 1 - r - s \tag{3.I.1}$$

then the family of t = constant lines are parallel to the inclined edge. The triple r, s, t defines ***triangular coordinates***. Each is zero along one edge and takes on the value 1 at the opposite vertex. The shape functions of the piecewise linear triangle may be concisely expressed with the aid of the triangular coordinates, viz.,

$$N_1(r, s) = r \tag{3.I.2}$$

$$N_2(r, s) = s \tag{3.I.3}$$

$$N_3(r, s) = t(r, s) = 1 - r - s \tag{3.I.4}$$

These functions are equivalent to those computed by the degeneration technique in Sec. 3.4. Higher-order shape functions for triangles may be systematically defined through the use of triangular coordinates. The general formula for ***Lagrange-type interpolation*** over triangles is given by

$$N_a(r, s, t) = T_I(r)T_J(s)T_K(t) \tag{3.I.5}$$

where

$$T_I(r) = \begin{cases} l_I^{I-1}\left(\dfrac{2r}{r_I} - 1\right) & I \neq 1 \\ 1, & I = 1 \end{cases} \tag{3.I.6}$$

and $a = a(I, J, K)$ is the formula defining the single nodal index and l in (3.I.6) stands for the Lagrange interpolation formula (see Sec. 3.6). Illustrations of the use of (3.I.5) are given in the following examples.

Example 1

Consider the three-node triangle depicted in Fig. 3.I.5. The shape functions are:

$$\begin{aligned} N_1(r, s, t) &= T_2(r)T_1(s)T_1(t) \\ &= l_2^1\left(\frac{2r}{r_2} - 1\right) \\ &= r \end{aligned} \tag{3.I.7}$$

$$\begin{aligned} N_2(r, s, t) &= T_1(r)T_2(s)T_1(t) \\ &= l_2^1\left(\frac{2s}{s_2} - 1\right) \\ &= s \end{aligned} \tag{3.I.8}$$

$$\begin{aligned} N_3(r, s, t) &= T_1(r)T_1(s)T_2(t) \\ &= l_2^1\left(\frac{2t}{t_2} - 1\right) \\ &= t \end{aligned} \tag{3.I.9}$$

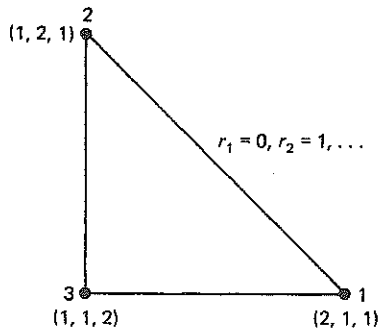


Figure 3.I.5

Example 2

Consider the six-node triangle shown in Fig. 3.I.6.

$$\begin{aligned} N_1(r, s, t) &= T_3(r)T_1(s)T_1(t) \\ &= l_3^2\left(\frac{2r}{r_3} - 1\right) \\ &= r(2r - 1) \end{aligned} \tag{3.I.10}$$

$$N_2(r, s, t) = T_1(r)T_3(s)T_1(t)$$

$$= l_3^2\left(\frac{2s}{s_3} - 1\right)$$

(continued)

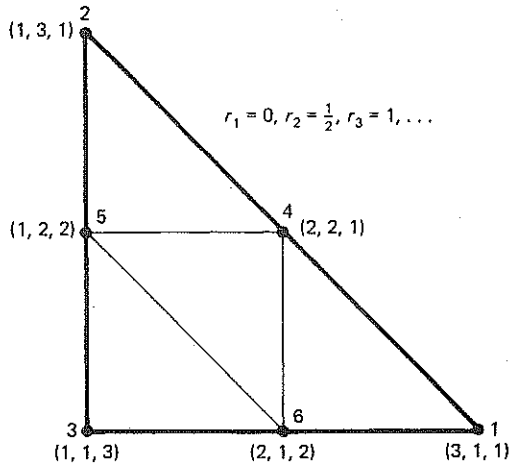


Figure 3.I.6

$$= s(2s - 1) \tag{3.I.11}$$

$$N_3(r, s, t) = T_1(r)T_1(s)T_3(t)$$
$$= l_3^2\left(\frac{2t}{t_3} - 1\right)$$
$$= t(2t - 1) \tag{3.I.12}$$

$$N_4(r, s, t) = T_2(r)T_2(s)T_1(t)$$
$$= l_2^1\left(\frac{2r}{r_2} - 1\right)l_2^1\left(\frac{2s}{s_2} - 1\right)$$
$$= 4rs \tag{3.I.13}$$

$$N_5(r, s, t) = T_1(r)T_2(s)T_2(t)$$
$$= l_2^1\left(\frac{2s}{s_2} - 1\right)l_2^1\left(\frac{2t}{t_2} - 1\right)$$
$$= 4st \tag{3.I.14}$$

$$N_6(r, s, t) = T_2(r)T_1(s)T_2(t)$$
$$= l_2^1\left(\frac{2r}{r_2} - 1\right)l_2^1\left(\frac{2t}{t_2} - 1\right)$$
$$= 4rt \tag{3.I.15}$$

Example 3

Consider the 10-node triangle depicted in Fig. 3.I.7.

$$N_1(r, s, t) = T_4(r)T_1(s)T_1(t)$$
$$= l_4^3\left(\frac{2r}{r_4} - 1\right) \qquad \text{(continued)}$$

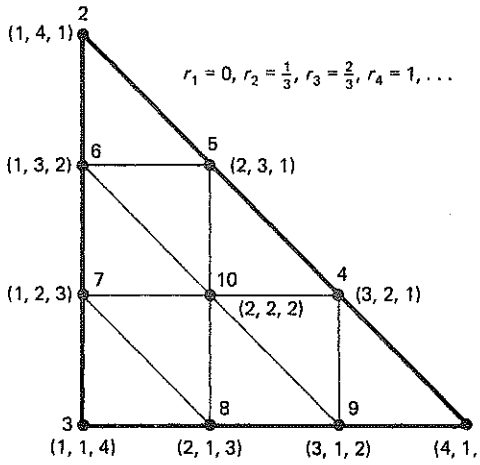


Figure 3.I.7

$$= \frac{9}{2}r\left(r^2 - r + \frac{2}{9}\right) \tag{3.I.16}$$

$$N_4(r, s, t) = T_3(r)T_2(s)T_1(t)$$

$$= l_3^2\left(\frac{2r}{r_3} - 1\right)l_2^1\left(\frac{2s}{s_2} - 1\right)$$

$$= \frac{9}{2}sr(3r - 1) \tag{3.I.17}$$

$$N_{10}(r, s, t) = T_2(r)T_2(s)T_2(t)$$

$$= l_2^1\left(\frac{2r}{r_2} - 1\right)l_2^1\left(\frac{2s}{s_2} - 1\right)l_2^1\left(\frac{2t}{t_2} - 1\right)$$

$$= 27rst \tag{3.I.18}$$

The reader may wish to calculate the remaining shape functions.

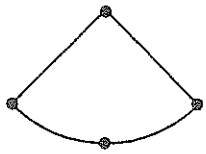
Exercise 1. Determine the shape functions for the 15-node quartic Lagrange-type triangle.

Remark

Note that the Lagrange-type shape functions for triangles result in complete polynomials without any extraneous monomials (see the Pascal triangle, Fig. 3.7.8).

Serendipity-type triangular elements may also be derived with the aid of triangular coordinates. However, there does not seem to be much practical interest in elements of this type.

Exercise 2. Use triangular coordinates to develop shape functions for a four-node triangle which exhibits quadratic behavior along one edge and linear behavior along the other two (see the accompanying figure on page 170).



Shape Functions for Tetrahedra

The ***parent tetrahedron*** is illustrated in Fig. 3.I.8. The ***tetrahedral coordinates*** are r, s, t, u. Each is zero on one surface of the tetrahedron and takes on the value 1 at the opposite vertex. The shape functions of the linear tetrahedron are simply

$$N_1(r, s, t) = r \tag{3.I.19}$$

$$N_2(r, s, t) = s \tag{3.I.20}$$

$$N_3(r, s, t) = t \tag{3.I.21}$$

$$N_4(r, s. t) = u(r, s, t) = 1 - r - s - t \tag{3.I.22}$$

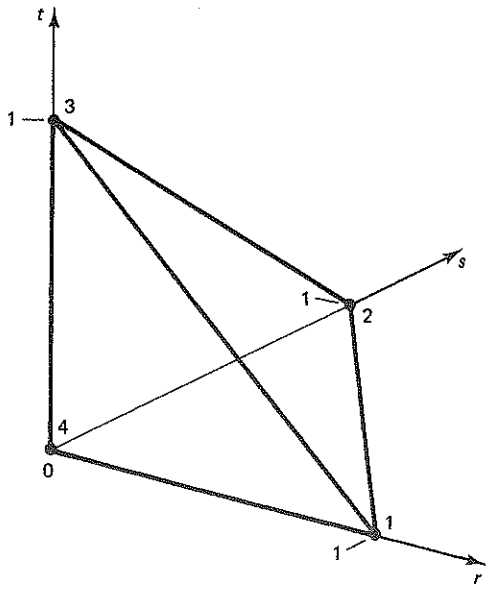


Figure 3.I.8

As in the case of triangles, shape functions for Lagrange-type tetrahedra may be derived with the aid of a simple formula:

$$N_a(r, s, t, u) = T_I(r)T_J(s)T_K(t)T_L(u) \tag{3.I.23}$$

where the T's are again defined by (3.I.6) and $a = a(I, J, K, L)$.

Exercise 3. Consider the 10-node quadratic tetrahedron shown in Fig. 3.I.9. Using (3.I.23) derive the shape functions and verify that they satisfy the interpolation property (i.e., $N_a(r_b, s_b, t_b, u_b) = \delta_{ab}$).

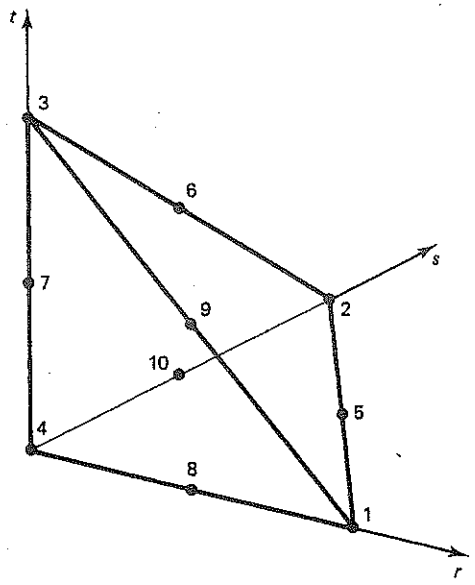


Figure 3.I.9

Shape Functions for Wedge-Shaped Elements

Shape functions of Lagrange-type for wedge-shaped elements may be derived by taking products of one-dimensional and triangular shape functions.

Example 4

Consider the basic six-node wedge element illustrated in Fig. 3.I.10. Shape functions for this element derive from the linear one-dimensional element and linear triangle, viz.,

$$N_1(\xi, r, s, t) = \tfrac{1}{2}(1 - \xi)r \tag{3.I.24}$$

$$N_2(\xi, r, s, t) = \tfrac{1}{2}(1 - \xi)s \tag{3.I.25}$$

$$N_3(\xi, r, s, t) = \tfrac{1}{2}(1 - \xi)t \tag{3.I.26}$$

$$N_4(\xi, r, s, t) = \tfrac{1}{2}(1 + \xi)r \tag{3.I.27}$$

$$N_5(\xi, r, s, t) = \tfrac{1}{2}(1 + \xi)s \tag{3.I.28}$$

$$N_6(\xi, r, s, t) = \tfrac{1}{2}(1 + \xi)t \tag{3.I.29}$$

Higher-order wedges may be similarly derived.

Integration

The integration of monomials over straight-edged triangles and flat-surfaced tetrahedra may be performed with the aid of the following exact expressions:

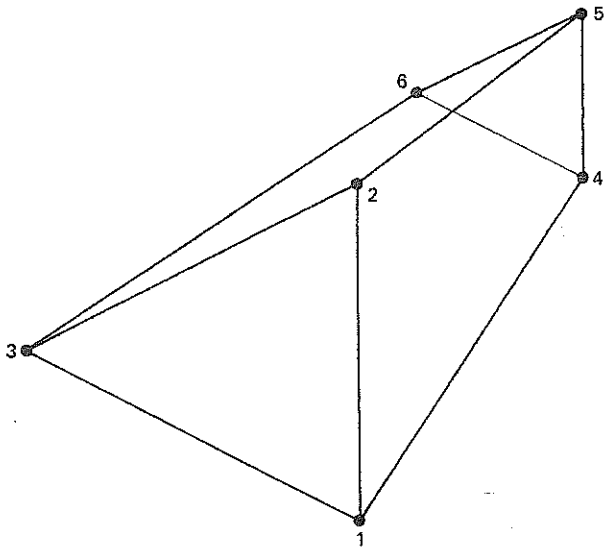


Figure 3.I.10

Triangles

$$\int_\Omega r^\alpha s^\beta t^\gamma \, d\Omega = \frac{\alpha!\ \beta!\ \gamma!}{(\alpha + \beta + \gamma + 2)!} 2A \tag{3.I.30}$$

where A, the area of Ω, is given by

$$2A = \det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \tag{3.I.31}$$

in which x_a and y_a are the coordinates of vertex a.

Tetrahedra

$$\int_\Omega r^\alpha s^\beta t^\gamma u^\delta \, d\Omega = \frac{\alpha!\ \beta!\ \gamma!\ \delta!}{(\alpha + \beta + \gamma + \delta + 3)!} 6V \tag{3.I.32}$$

where V, the volume of Ω, is given by

$$6V = \det \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \tag{3.I.33}$$

in which x_a, y_a, and z_a are the coordinates of vertex a.

In the case of distorted elements, integrands will not consist simply of monomials. Consequently, numerical integration needs to be used. Formulas that are symmetrical with respect to the tetrahedral coordinates have been derived. References [25–27] may be consulted for results of this kind. A sampling of results is presented in Tables 3.I.1 and 3.I.2. Higher-order rules for tetrahedra are presented in [28].

TABLE 3.I.1 Integration Rules for Triangles [26, 27]

W_i	$\tilde{r}$	$\tilde{s}$	$\tilde{t}$	Multiplicity[1]
	3-point formula	*degree of precision 2*[2]		
0.33333 33333 33333	0.66666 66666 66667	0.16666 66666 66667	0.16666 66666 66667	3
	3-point formula	*degree of precision 2*		
0.33333 33333 33333	0.50000 00000 00000	0.50000 00000 00000	0.00000 00000 00000	3
	4-point formula	*degree of precision 3*		
−0.56250 00000 00000	0.33333 33333 33333	0.33333 33333 33333	0.33333 33333 33333	1
0.52083 33333 33333	0.60000 00000 00000	0.20000 00000 00000	0.20000 00000 00000	3
	6-point formula	*degree of precision 3*		
0.16666 66666 66667	0.65902 76223 74092	0.23193 33685 53031	0.10903 90090 72877	6
	6-point formula	*degree of precision 4*		
0.10995 17436 55322	0.81684 75729 80459	0.09157 62135 09771	0.09157 62135 09771	3
0.22338 15896 78011	0.10810 30181 68070	0.44594 84909 15965	0.44594 84909 15965	3
	7-point formula	*degree of precision 4*		
0.37500 00000 00000	0.33333 33333 33333	0.33333 33333 33333	0.33333 33333 33333	1
0.10416 66666 66667	0.73671 24989 68435	0.23793 23664 72434	0.02535 51345 51932	6
	7-point formula	*degree of precision 5*		
0.22500 00000 00000	0.33333 33333 33333	0.33333 33333 33333	0.33333 33333 33333	1
0.12593 91805 44827	0.79742 69853 53087	0.10128 65073 23456	0.10128 65073 23456	3
0.13239 41527 88506	0.47014 20641 05115	0.47014 20641 05115	0.05971 58717 89770	3
	9-point formula	*degree of precision 5*		
0.20595 05047 60887	0.12494 95032 33232	0.43752 52483 83384	0.43752 52483 83384	3
0.06369 14142 86223	0.79711 26518 60071	0.16540 99273 89841	0.03747 74207 50088	6
	12-point formula	*degree of precision 6*		
0.05084 49063 70207	0.87382 19710 16996	0.06308 90144 91502	0.06308 90144 91502	3
0.11678 62757 26379	0.50142 65096 58179	0.24928 67451 70910	0.24928 67451 70911	3
0.08285 10756 18374	0.63650 24991 21399	0.31035 24510 33785	0.05314 50498 44816	6

[1] The formulas are symmetric in the triangular coordinates. All permutations of the quadrature points need to be accounted for.

[2] The degree of precision refers to the order of the complete polynomial which is exactly integrated by the rule. (One-point, centroidal quadrature achieves a degree of precision equal to one.)

TABLE 3.I.1 Integration Rules for Triangles [26, 27] (*cont.*)

W_l	$\tilde{r}$	$\tilde{s}$	$\tilde{t}$	Multiplicity[(1)]
	13-point formula	*degree of precision 7*		
−0.14957 00444 67670	0.33333 33333 33333	0.33333 33333 33333	0.33333 33333 33333	1
0.17561 52574 33204	0.47930 80678 41923	0.26034 59660 79038	0.26034 59660 79038	3
0.05334 72356 08839	0.86973 97941 95568	0.06513 01029 02216	0.06513 01029 02216	3
0.07711 37608 90257	0.63844 41885 69809	0.31286 54960 04875	0.08690 31542 53160	6

TABLE 3.I.2 Integration Rules for Tetrahedra [25]

W_l	$\tilde{r}$	$\tilde{s}$	$\tilde{t}$	$\tilde{u}$	Multiplicity
	1-point formula		*degree of precision 1*		
1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	1
	4-point formula		*degree of precision 2*		
$\frac{1}{4}$	0.58541020	0.13819660	0.13819660	0.13819660	4
	5-point formula		*degree of precision 3*		
$-\frac{4}{5}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	1
$\frac{9}{20}$	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	4

It is traditional to tabulate quadrature rules for triangles and tetrahedra such that the weights sum to 1. Because the area of the parent triangle equals $\frac{1}{2}$ and the volume of the parent tetrahedron equals $\frac{1}{6}$, the proper form for a quadrature rule in these cases is

$$\int_{\Omega} f \, d\Omega \cong c \sum_{l=1}^{n_{\text{int}}} f_l j_l W_l \tag{3.I.34}$$

where $c = \frac{1}{2}$ for triangles and $c = \frac{1}{6}$ for tetrahedra.

Derivatives of Shape Functions

To calculate derivatives of shape functions with respect to global coordinates we may employ the techniques described in Sec. 3.9. All formulas of Sec. 3.9 may be employed with the following substitutions.

Triangles

Replace ξ and η by r and s, respectively; set $t = 1 - r - s$ throughout.

Tetrahedra

Replace ξ, η, and ζ by r, s, and t, respectively; set $u = 1 - r - s - t$ throughout.

Appendix 3.II

Methodology for Developing Special Shape Functions with Application to Singularities

In many areas of finite element analysis, elements with special properties are required to achieve maximal accuracy. As examples, we may mention infinite elements for the representation of spatial domains that extend to infinity [29, 30] and singular elements for modeling point and line singularities engendered by geometric features such as reentrant corners and cracks [31–33]. In this appendix we are concerned with techniques for developing shape functions for special elements in general but with particular emphasis on singularities.

Geometric features of the spatial domain may give rise to highly singular response in the solution of a field problem [34]. Eigenfunction analyses are frequently available for purposes of characterizing the asymptotic strength of the singularity [35]. However, the actual solution depends upon the nature of the loading and there are situations when the singularity does not contribute significantly [36]. For this reason it has been suggested that, ideally, singular finite elements should strike some balance between being able to represent smooth behavior and potential singular behavior [36]. In this way the element will remain effective under all loading conditions. Translated into analytical terms, the finite element shape functions should be designed to represent the standard low-order polynomials as well as the singular functions emanating from asymptotic analysis. To construct shape functions of this type gives rise to an interpolation problem, which involves a system of linear simultaneous equations. In special circumstances explicit representations are available for the solution of the interpolation problem. The most well known example is the Lagrange interpolation formula, which is the solution of the interpolation problem for polynomials passing through distinct points. It appears that no such representation exists when different classes of functions are merged, as seems appropriate for singular elements. Although, in principle, the interpolation problem could be solved numerically for each singular element during formation, this would be highly inefficient due to the increased number

of computations required. Furthermore, numerical sensitivity may manifest itself in certain configurations. Consequently, it is of practical interest to develop techniques for systematically defining shape functions for singularity modeling (and for developing special elements in general), which circumvent the interpolation problem. In what follows, a highly concise algorithm for generating shape functions from an arbitrary starting set of independent functions is presented [37]. A novel feature of the procedure is that *no* coefficient matrix is involved (i.e., no system of simultaneous equations need be solved). Some useful one-dimensional interpolatory schemes for modeling singularities are developed by employing the interpolation algorithm. These schemes are the basis for constructiong two- and three-dimensional elements for modeling point and line singularities.

Let $r = r(\xi) = (1 + \xi)/2$ and $r_a = r(\xi_a)$. Note that $r(\xi) \in [0, 1]$ for all $\xi \in [-1, 1]$. The r-coordinate description is convenient for modeling singularities. In multidimensional applications, we use boldfaced notations analogous to the preceding ones. For example, in two dimensions $\mathbf{r} = (r, s)$, where $s = (1 + \eta)/2$, $\eta \in [-1, 1]$.

The ***Lagrange polynomials*** are defined by

$$l_a(f) = \frac{\prod_{\substack{b=1 \\ b\neq a}}^{m} (f - f_b)}{\prod_{\substack{b=1 \\ b\neq a}}^{m} (f_a - f_b)}, \qquad a = 1, 2, \ldots, m \tag{3.II.1}$$

where the f_a's are distinct points in $\mathbb{R}$. It may be seen from (3.II.1) that $l_a(f)$ is an $(m - 1)$st order polynomial in f that satisfies the "interpolation property"(i.e., $l_a(f_b) = \delta_{ab}$, the Kronecker delta). *Note that f may be taken to be ξ, r, r^α, or any other convenient function of ξ.*

Algorithm for Constructing Interpolation Functions

Consider an n-node finite element. We assume we are given a set of "preliminary" shape functions, N_a, $a = 1, 2, \ldots, n$, which satisfy the interpolation property on the first m nodes only; viz., $N_a(\mathbf{r}_b) = \delta_{ab}$, where $a, b = 1, 2, \ldots, m < n$. The idea is to modify systematically the N_a's such that the interpolation property is satisfied on all n nodes. The procedure is given as follows:

Step 1
$$N_{m+1}(\mathbf{r}) \leftarrow \frac{N_{m+1}(\mathbf{r}) - \sum_{a=1}^{m} N_{m+1}(\mathbf{r}_a) N_a(\mathbf{r})}{N_{m+1}(\mathbf{r}_{m+1}) - \sum_{a=1}^{m} N_{m+1}(\mathbf{r}_a) N_a(\mathbf{r}_{m+1})} \tag{3.II.2}$$

Step 2
$$N_a(\mathbf{r}) \leftarrow N_a(\mathbf{r}) - N_a(\mathbf{r}_{m+1}) N_{m+1}(\mathbf{r}), \qquad a = 1, 2, \ldots, m \tag{3.II.3}$$

Step 3 If $m+1 < n$, replace m by $m + 1$ and repeat Steps 1 to 3.
If $m+1 = n$, stop.

Remarks

1. After completing Steps 1 and 2, the shape functions satisfy the interpolation property on the first $m + 1$ nodes. Clearly, after completing the entire procedure, the desired properties are achieved.

2. In many cases, the algorithm enables the explicit hand calculation of shape functions. *However, there is no need to do this in practice, as the algorithm itself may be programmed as part of a shape function subroutine.*

3. The derivatives of shape functions may be constructed in similar fashion. The analogs of Steps 1 and 2 are respectively

$$\partial N_{m+1}(r) \leftarrow \frac{\partial N_{m+1}(r) - \sum_{a=1}^{m} N_{m+1}(r_a)\partial N_a(r)}{N_{m+1}(r_{m+1}) - \sum_{a=1}^{m} N_{m+1}(r_a)N_a(r_{m+1})} \tag{3.II.4}$$

and

$$\partial N_a(r) \leftarrow \partial N_a(r) - N_a(r_{m+1})\partial N_{m+1}(r) \tag{3.II.5}$$

In (3.II.4) and (3.II.5), ∂ represents the partial differentiation operator with respect to any of the coordinates (e.g., in two dimensions, ∂ may represent $\partial/\partial r$ or $\partial/\partial s$).

4. Interpolation problems are generally formulated in terms of a system of simultaneous linear equations. It may be noted that, if $m = 1$, the algorithm encompassed by Steps 1–3 solves the interpolation problem *without* engendering any coefficient matrix. Thus no "storage" is required (beyond that for the N_a's) in programming the procedure. Another nice feature of the algorithm is that the interpolation property possessed by the preliminary shape functions is fully exploited in that only $n - m$ passes through Steps 1–3 need be performed.

In practice, the Lagrange polynomial formula may be employed to generate a partial set of preliminary shape functions, which satisfies the interpolation property on the first m nodes. This is illustrated in the examples to follow.

Example 1

Consider a one-dimensional three-node element with nodes given by $r_1 = 0$, $r_2 = \frac{1}{2}$, and $r_3 = 1$. We wish to construct shape functions capable of exactly representing 1, r, and r^α. (If $\alpha = 2$, we have the usual three-node quadratic element. Other values of α enable the modeling of singularities.) We begin with the following preliminary shape functions:

$$N_1(r) = 1 - 2r, \qquad N_2(r) = 2r, \qquad N_3(r) = r^\alpha \tag{3.II.6}$$

Observe that $N_a(r_b) = \delta_{ab}$, $1 \le a, b \le 2$. In terms of our algorithm, $m = 2$ and $n = 3$, so only one pass through Steps 1 and 2 is required:

$$N_3(r) \leftarrow \frac{N_3(r) - N_3(\frac{1}{2})N_2(r)}{N_3(1) - N_3(\frac{1}{2})N_2(1)} = \frac{r^\alpha - 2(\frac{1}{2})^\alpha r}{1 - 2(\frac{1}{2})^\alpha} \tag{3.II.7}$$

$$N_1(r) \leftarrow N_1(r) - N_1(1)N_3(r) = 1 - 2r + \left[\frac{r^\alpha - 2(\frac{1}{2})^\alpha r}{1 - 2(\frac{1}{2})^\alpha}\right] \tag{3.II.8}$$

$$N_2(r) \leftarrow N_2(r) - N_2(1)N_3(r) = 2r - 2\left[\frac{r^\alpha - 2(\frac{1}{2})^\alpha r}{1 - 2(\frac{1}{2})^\alpha}\right] \tag{3.II.9}$$

Likewise, the derivatives are given by:

$$\partial N_3(r) \leftarrow \frac{\partial N_3(r) - N_3(\frac{1}{2})\partial N_2(r)}{N_3(1) - N_3(\frac{1}{2})N_2(1)} = \frac{\alpha r^{\alpha-1} - 2(\frac{1}{2})^\alpha}{1 - 2(\frac{1}{2})^\alpha} \tag{3.II.10}$$

$$\partial N_1(r) \leftarrow \partial N_1(r) - N_1(1)\partial N_3(r) = -2 + \left[\frac{\alpha r^{\alpha-1} - 2(\frac{1}{2})^\alpha}{1 - 2(\frac{1}{2})^\alpha}\right] \tag{3.II.11}$$

$$\partial N_2(r) \leftarrow \partial N_2(r) - N_2(1)\partial N_3(r) = 2 - 2\left[\frac{\alpha r^{\alpha-1} - 2(\frac{1}{2})^\alpha}{1 - 2(\frac{1}{2})^\alpha}\right] \tag{3.II.12}$$

(In this case ∂N is taken to mean $\partial N/\partial r$)

Example 2

Consider a one-dimensional four-node element for which the nodes are given by $r_1 = 0$, $r_2 = \frac{1}{3}$, $r_3 = \frac{2}{3}$, and $r_4 = 1$. We wish to construct shape functions capable of representing 1, r^α, $r^{2\alpha}$, and r^β. The following special cases are of practical importance:

$\alpha = 1$, $\beta = 3$:	$1, r, r^2, r^3$	cubic polynomial in r
$\alpha = 1$, β arbitrary:	$1, r, r^2, r^\beta$	quadratic polynomial in r plus linear polynomial in r^β
α arbitrary, $\beta = 1$:	$1, r, r^\alpha, r^{2\alpha}$	linear polynomial in r plus quadratic polynomial in r^α
α arbitrary, $\beta = 3\alpha$:	$1, r^\alpha, r^{2\alpha}, r^{3\alpha}$	cubic polynomial in r^α

To obtain a starting set of shape functions, we may employ the Lagrange interpolation formula with $f = r^\alpha$ and $m = 3$, viz.,

$$N_1(r) = l_1(r^\alpha) = \frac{(r^\alpha - r_2^\alpha)(r^\alpha - r_3^\alpha)}{(r_1^\alpha - r_2^\alpha)(r_1^\alpha - r_3^\alpha)} = \frac{(3^\alpha r^\alpha - 1)(3^\alpha r^\alpha - 2^\alpha)}{2^\alpha} \tag{3.II.13}$$

$$N_2(r) = l_2(r^\alpha) = \frac{(r^\alpha - r_1^\alpha)(r^\alpha - r_3^\alpha)}{(r_2^\alpha - r_1^\alpha)(r_2^\alpha - r_3^\alpha)} = \frac{3^\alpha r^\alpha(3^\alpha r^\alpha - 2^\alpha)}{(1 - 2^\alpha)} \tag{3.II.14}$$

$$N_3(r) = l_3(r^\alpha) = \frac{(r^\alpha - r_1^\alpha)(r^\alpha - r_2^\alpha)}{(r_3^\alpha - r_1^\alpha)(r_3^\alpha - r_2^\alpha)} = \frac{3^\alpha r^\alpha(3^\alpha r^\alpha - 1)}{2^\alpha(2^\alpha - 1)} \tag{3.II.15}$$

It may be verified that $N_a(r_b) = \delta_{ab}$ for all $a, b = 1, 2, 3$. The last preliminary shape function is taken to be

$$N_4(r) = r^\beta \tag{3.II.16}$$

With regard to the algorithm, $m = 3$ and $n = 4$, so only one pass through Steps 1 and 2 is needed, viz.,

$$N_4(r) \leftarrow \frac{N_4(r) - \sum_{a=2}^{3} N_4(r_a)N_a(r)}{1 - \sum_{a=2}^{3} N_4(r_a)N_a(1)} \tag{3.II.17}$$

$$N_a(r) \leftarrow N_a(r) - N_a(1)N_4(r), \qquad a = 1, 2, 3 \tag{3.II.18}$$

Equations (3.II.17) and (3.II.18) may be programmed in a shape function subroutine along with corresponding expressions for derivatives.

Remarks

5. The preceding one-dimensional examples cover most cases of practical interest. If the need arises, higher-order shape functions may be constructed along similar lines. The basic philosophy is to maximize m in order to minimize calculations in the shape function routines.

6. For completeness, we wish to mention the simplest one-dimensional shape functions that allow the modeling of singularities; namely, the two-node element with $r_1 = 0$, $r_2 = 1$, $N_1(r) = 1 - r^\alpha$, and $N_2(r) = r^\alpha$.

7. Let $u(r) = \sum_{a=1}^{n} N_a(r)d_a$, where (by construction) $d_a = u(r_a)$. The function u is capable of exactly representing the powers of r used in deriving the N_a's. (The d_a's need only be set accordingly.) In practice, it is usually desired that the same powers of the physical coordinate, $x = x(r)$, be representable. This will be attained if $x = x(r)$ is affine (i.e., of the form $x(r) = c_1 + c_2 r$). There are two ways of achieving this in practice.

The *first* depends upon 1 and r being present among the powers of r used in deriving the shape functions. In this case the isoparametric concept may be invoked; that is, we take $x(r) = \sum_{a=1}^{n} N_a(r)x_a$ and equally space the nodal coordinates (i.e., x_a's in x-space).

The *second* procedure does not require that 1 and r be present, but it involves an additional set of shape function calculations, so it is less efficient. In this case, we take $x(r) = \sum_{a=1}^{n} P_a(r)x_a$, where the P_a's are the standard polynomial shape functions, and the x_a's are, again, equally spaced.

To see the necessity of adopting the second procedure when 1 and r are absent from the N_a's, we need only consider the two-node element of Remark 6. If the isoparametric concept is adopted, then u varies linearly with x (rather than x^α). To see this we note that if $x_2 \neq x_1$, then we can write $d_a = c_1 + c_2 x_a$, where the c's are constants. Consequently

$$\begin{aligned} u &= \sum_{a=1}^{2} N_a d_a \\ &= \sum_{a=1}^{2} N_a(c_1 + c_2 x_a) \\ &= c_1\left(\sum_{a=1}^{2} N_a\right) + c_2\left(\sum_{a=1}^{2} N_a x_a\right) \\ &= c_1 + c_2 x \end{aligned} \tag{3.II.19}$$

We shall now employ the above results in deriving two-dimensional finite element shape functions for modeling line and point singularities.

Example 3

Consider the four-node quadrilateral illustrated in Fig. 3.II.1(a). The shape functions

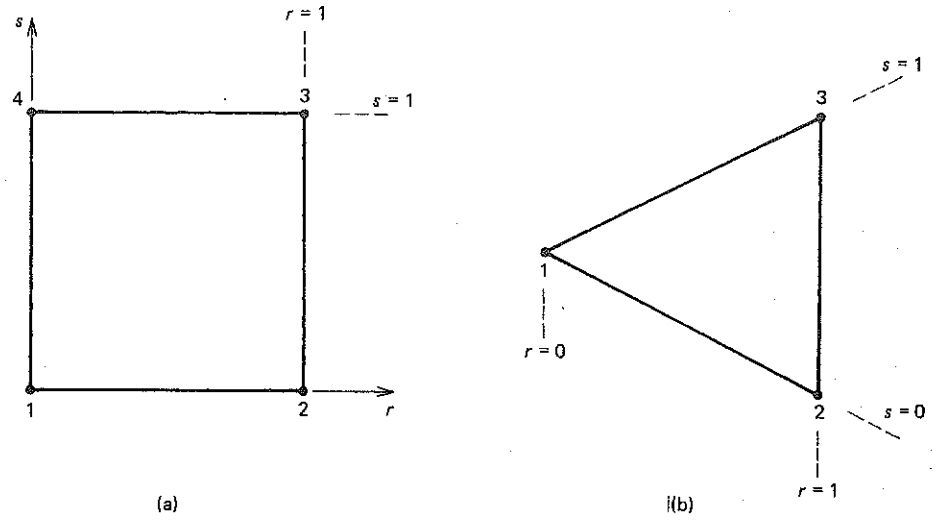


Figure 3.II.1 (a) Four-node quadrilateral element. (b) Three-node triangular element formed by degenerating the four-node quadrilateral.

may be constructed from products of linear polynomial shape functions in the s-direction and the shape functions of Remark 6 in the r-direction, viz.,

$$N_1(r, s) = (1 - r^\alpha)(1 - s) \tag{3.II.20}$$

$$N_2(r, s) = r^\alpha(1 - s) \tag{3.II.21}$$

$$N_3(r, s) = r^\alpha s \tag{3.II.22}$$

$$N_4(r, s) = (1 - r^\alpha)s \tag{3.II.23}$$

Thus $u(r, s) = \sum_{a=1}^{4} N_a(r, s)d_a$ is capable of exactly representing a line singularity of order r^α along edge $\widehat{14}$. By virtue of Remark 7, the standard polynomial shape functions must be employed to define the geometry. That is $x(r, s) = \sum_{a=1}^{4} P_a(r, s)x_a$, where

$$P_1(r, s) = (1 - r)(1 - s) \tag{3.II.24}$$

$$P_2(r, s) = r(1 - s) \tag{3.II.25}$$

$$P_3(r, s) = rs \tag{3.II.26}$$

$$P_4(r, s) = (1 - r)s \tag{3.II.27}$$

This element was first proposed by Akin [38].

Example 4

A three-node triangular element (see Fig. 3.II.1(b)) may be developed by combining the shape functions associated with nodes 1 and 4 in the previous example, viz.,

$$N_1(r, s) \leftarrow N_1(r, s) + N_4(r, s) = 1 - r^\alpha \tag{3.II.28}$$

$$P_1(r, s) \leftarrow P_1(r, s) + P_4(r, s) = 1 - r \tag{3.II.29}$$

The shape functions of nodes 2 and 3 remain the same. This element is capable of exactly

representing a point singularity of order r^α. Similar interpolations were used in [33] and [39].

Example 5

Consider the nine-node Lagrange-type quadrilateral illustrated in Fig. 3.II.2(a). The shape functions shall be constructed from products of the three-node one-dimensional shape functions of Example 1. That is, we define $N_a(r)$, $a = 1, 2, 3$, by (3.II.7)–(3.II.9) and $P_a(s)$ by the same expressions, with r replaced by s and $\alpha = 2$. The two-dimensional shape functions may then be defined as follows:

$$N_1(r, s) = N_1(r)P_1(s) \tag{3.II.30}$$

$$N_2(r, s) = N_3(r)P_1(s) \tag{3.II.31}$$

$$N_3(r, s) = N_3(r)P_3(s) \tag{3.II.32}$$

$$N_4(r, s) = N_1(r)P_3(s) \tag{3.II.33}$$

$$N_5(r, s) = N_2(r)P_1(s) \tag{3.II.34}$$

$$N_6(r, s) = N_3(r)P_2(s) \tag{3.II.35}$$

$$N_7(r, s) = N_2(r)P_3(s) \tag{3.II.36}$$

$$N_8(r, s) = N_1(r)P_2(s) \tag{3.II.37}$$

$$N_9(r, s) = N_2(r)P_2(s) \tag{3.II.38}$$

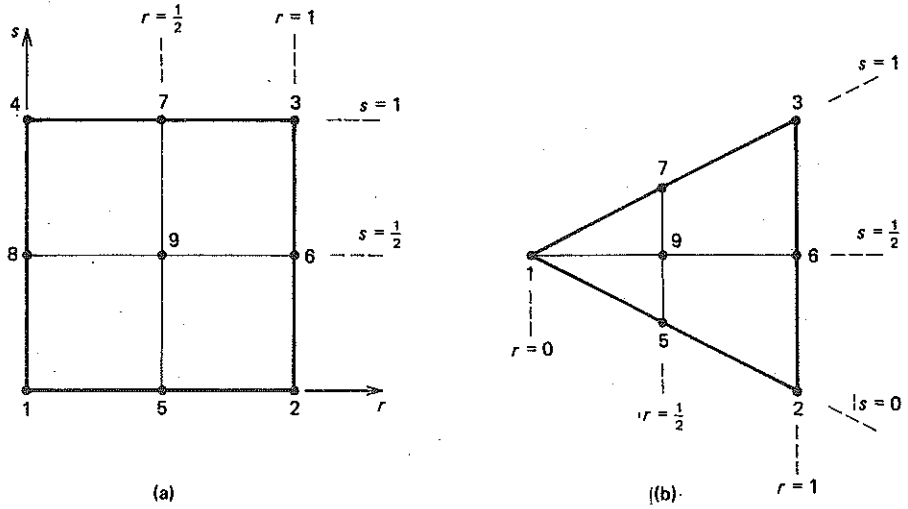


Figure 3.II.2 (a) Nine-node Lagrange quadrilateral element. (b) Seven-node triangular element formed by degenerating the nine-node Lagrange quadrilateral.

In this case $u(r, s) = \Sigma_{a=1}^{9} N_a(r, s)d_a$ is capable of exactly representing a line singularity of order r^α along edge $\widehat{14}$. Furthermore, the following monomials may be exactly represented: $1, r, s, r^\alpha, rs, s^2, r^\alpha s, s^2 r$, and $r^\alpha s^2$. By virtue of the presence of $1, r$ and s, either the isoparametric concept may be employed to define the geometry (i.e., $x(r, s) = \Sigma_{a=1}^{9} N_a(r, s)x_a$), or recourse may be made to the standard Lagrange poly-

nomials (i.e., $x(r, s) = \sum_{a=1}^{9} P_a(r, s)x_a$, where the $P_a(r, s)$'s may be obtained from (3.II.30) through (3.II.38) by replacing $N_a(r)$ by $P_a(r)$, $a = 1, 2, 3$).

Remark

More complicated two- and three-dimensional elements developed along the lines sketched here are presented in [37].

REFERENCES

Section 3.1

1. G.P. Bazeley, Y.K. Cheung, B.M. Irons, and O.C. Zienkiewicz, "Triangular Elements in Bending—Conforming and Non-conforming Solutions," *Proceedings of the Conference on Matrix Methods in Structural Mechanics*, Wright-Patterson Air Force Base, Ohio, 1965.
2. G. Strang and G.J. Fix, *An Analysis of the Finite Element Method*. Englewood Cliffs, N.J.: Prentice-Hall, 1973.
3. O. C. Zienkiewicz, *The Finite Element Method*. London: McGraw-Hill, 1977.

Section 3.2

4. I.C. Taig, *Structural Analysis by the Matrix Displacement Method*, English Electric Aviation Report No. S017, 1961.
5. J.H. Argyris and S. Kelsey, *Energy Theorems and Structural Analysis*. London: Butterworths, 1960 (originally published in a series of articles in *Aircraft Engineering*, 1954–55).

Section 3.3

6. I.C. Taig, *Structural Analysis by the Matrix Displacement Method*, English Electric Aviation Report No. S017, 1961.
7. B. M. Irons, "Engineering Application of Numerical Integration in Stiffness Method," *Journal of the American Institute of Aeronautics and Astronautics*, 14 (1966), 2035–2037.
8. S. Lang, *Real Analysis*. Reading Mass.: Addison-Wesley, 1969.

Section 3.4

9. R. Courant, "Variational Methods for the Solution of Problems of Equilibrium and Vibration," *Bulletin of the American Mathematical Society*, 49 (1943), 1–23.
10. M. J. Turner, R. W. Clough, H. C. Martin, and L. P. Topp, "Stiffness and Deflection Analysis of Complex Structures," *Journal of Aeronautical Sciences*, 23, no. 9 (1956), 805–823.

Section 3.5

11. R. H. Gallagher, J. Padlog, and P. P. Bijlaard, "Stress Analysis of Heated Complex Shapes," *American Rocket Society Journal*, 32, no. 5 (1962), 700–707.

Section 3.7

12. O. C. Zienkiewicz, *The Finite Element Method.* London: McGraw-Hill, 1977.

Section 3.8

13. A. H. Stroud and D. Secrest, *Gaussian Quadrature Formulas*. Englewood Cliffs, N.J.: Prentice-Hall, 1966.
14. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. Washington, D.C.: National Bureau of Standards, 1964.
15. P. C. Hammer and A. H. Stroud, "Numerical Evaluation of Multiple Integrals II," *Mathematical Tables and Aids of Computation*, 12, no. 64 (1958), 272–280.
16. B. M. Irons, "Quadrature Rules for Brick Based Finite Elements," *International Journal of Numerical Methods in Engineering*, 3, no. 2 (1971), 293–294.
17. T. K. Hellen, "Effective Quadrature Rules for Quadratic Solid Isoparametric Finite Elements," *International Journal for Numerical Methods in Engineering*, 4, no. 4 (1972), 597–599.
18. C. Maforano, S. Odorizzi, and R. Vitaliani, "Shortened Quadrature Rules for Finite Elements," *Advances in Engineering Software*, 4, no. 2 (1982).

Section 3.10

19. A. K. Gupta and B. Mohraz, "A Method of Computing Numerically Integrated Stiffness Matrices," *International Journal for Numerical Methods in Engineering*, 5 (1972), 83–89.
20. R. L. Taylor, "Computer Procedures for Finite Element Analysis," Chapter 24 in O. C. Zienkiewicz, *The Finite Element Method.* London: McGraw-Hill, 1977.
21. E. Hinton and D. R. J. Owen, *Finite Element Programming*. London: Academic Press, 1977.
22. K. J. Bathe and E. L. Wilson, *Numerical Methods in Finite Element Analysis*. Englewood Cliffs, N. J.: Prentice-Hall, 1976.
23. D. R. J. Owen and E. Hinton, *Finite Elements in Plasticity: Theory and Practice*. Swansea, U.K.: Pineridge Press, 1980.
24. J. E. Akin, *Application and Implementation of Finite Element Methods*. London: Academic Press, 1982.

Appendix 3.I

25. P. C. Hammer, O. P. Marlowe, and A. H. Stroud, "Numerical Integration over Simplexes and Cones," *Mathematical Tables and Aids to Computation*, 10 (1956), 130–137.
26. G. R. Cowper, "Gaussian Quadrature Formulas for Triangles," *International Journal for Numerical Methods in Engineering*, 7 (1973), 405–408.
27. G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*. Englewood Cliffs, N. J.: Prentice-Hall, 1973.
28. Y. Jinyun, "Symmetric Gaussian Quadrature Formulae for Tetrahedral Regions," *Computer Methods in Applied Mechanics and Engineering*, 43, no. 3 (1984), 349–353.

Appendix 3.II

29. D. Gartling and E. B. Becker, "Computationally Efficient Finite Element Analysis of

Viscous Flow Problems," *Computational Methods in Nonlinear Mechanics*, (ed. J. T. Oden). Austin: Texas Institute for Computational Mechanics, 1974, 603–614.

30. P. Bettess, "Infinite Elements," *International Journal for Numerical Methods in Engineering*, 11 (1977), 53–64.

31. W. S. Blackburn, "Calculation of Stress Intensity Factors at Crack Tips Using Special Finite Elements," *Mathematics of Finite Elements and Applications*, ed. J. R. Whiteman. London: Academic Press, (1973), 327–336.

32. R. D. Henshell and K. G. Shaw, "Crack Tip Elements are Unnecessary," *International Journal for Numerical Methods in Engineering*, 9 (1975), 495–507.

33. J. E. Akin, "Generation of Elements with Singularities," *International Journal for Numerical Methods in Engineering*, 10 (1976), 1249–1259.

34. R. E. Barnhill and J. R. Whiteman, "Error Analysis of Finite Element Methods with Triangles for Elliptic Boundary Value Problems," *Mathematics of Finite Element Methods and Applications*, ed. J. R. Whiteman. London: Academic Press, (1973), 83–112.

35. A. K. Rao, "Stress Concentrations and Singularities at Interface Corners," *ZAMM*, 51 (1971), 395–406.

36. R. S. Barsoum, "On the Use of Isoparametric Finite Elements in Linear Fracture Mechanics," *International Journal for Numerical Methods in Engineering*, 10 (1976), 25–37.

37. T. J. R. Hughes and J. E. Akin, "Techniques for Developing 'Special' Finite Element Shape Functions with Particular Reference to Singularities," *International Journal for Numerical Methods in Engineering*, 15 (1980), 733–751.

38. J. E. Akin, "Elements for Problems with Line Singularities," *Mathematics of Finite Elements and Applications III*, ed. J. R. Whiteman. London: Academic Press, 1978.

39. D. M. Tracey and T. S. Cook, "Analysis of Power Type Singularities Using Finite Elements," *International Journal for Numerical Methods in Engineering*, 10 (1976), 1249–1259.