Chapter 9

GENERAL INTERPOLATION

9.1 Introduction

The previous sections have illustrated the heavy dependence of finite element methods on both spatial interpolation and efficient integrations. In a one-dimensional problem it does not make a great deal of difference if one selects a local or global coordinate system for the interpolation equations, because the inter-element continuity requirements are relatively easy to satisfy. That is not true in higher dimensions. To obtain practical formulations it is almost essential to utilize local coordinate interpolations. Doing this does require a small amount of additional work in relating the derivatives in the two coordinate systems.

9.2 Unit Coordinate Interpolation

The use of unit coordinates have been previously mentioned in Chap. 4. Here some of the procedures for deriving the interpolation functions in unit coordinates will be presented. Consider the three-node triangular element shown in Fig. 9.2.1. The local coordinates of its three nodes are $(0,0)$, $(1,0)$, and $(0,1)$, respectively. Once again we wish to utilize polynomial functions for our interpolations. In two dimensions the simplest complete polynomial has three constants. Thus, this linear function can be related to the three nodal quantities of the element. Assume the polynomial for some quantity, u, is defined as:

$$u^e(r,s) \;=\; d_1^e + d_2^e r + d_3^e s \;=\; \mathbf{P}(r,s)\,\mathbf{d}^e\,. \tag{9.1}$$

If it is valid everywhere in the element then it is valid at its nodes. Substituting the local coordinates of a node into Eq. 9.1 gives an identity between the $\mathbf{d}^e$ and a nodal value of $\mathbf{u}$. Establishing these identities at all three nodes gives

$$\left\{\begin{array}{c} u_1^e \\ u_2^e \\ u_3^e \end{array}\right\} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{array}\right] \left\{\begin{array}{c} d_1^e \\ d_2^e \\ d_3^e \end{array}\right\}$$

or

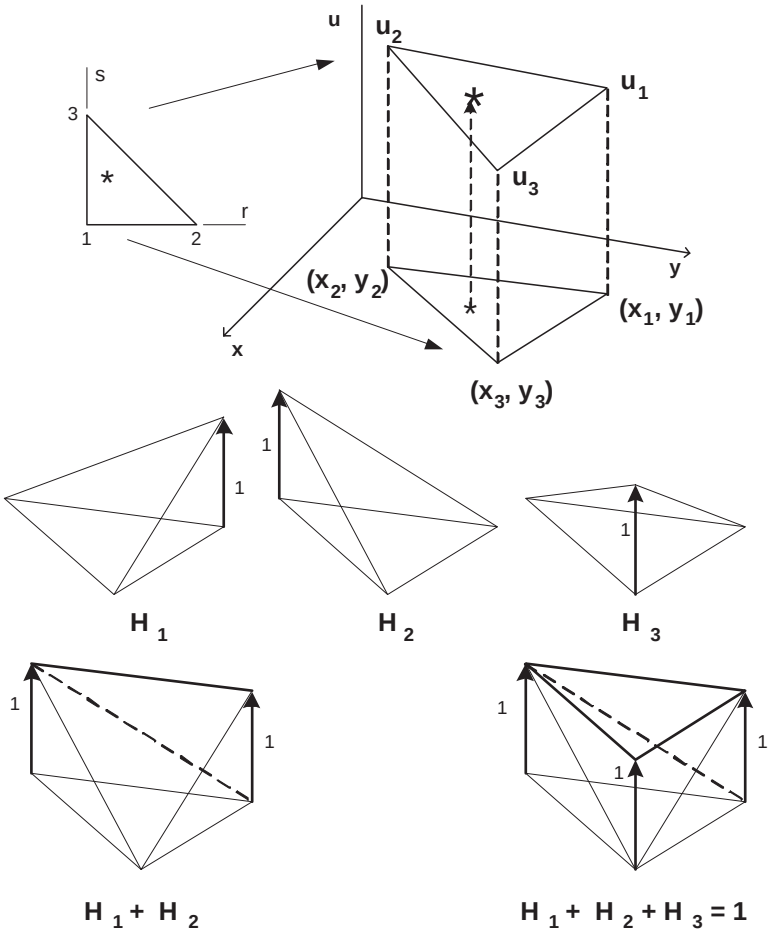


Figure 9.2.1 Isoparametric interpolation on a simplex triangle

$$\mathbf{u}^e = \mathbf{g}\,\mathbf{d}^e . \tag{9.2}$$

Iff the inverse exists, and it does here, this equation can be solved to yield

$$\mathbf{d}^e = \mathbf{g}^{-1}\mathbf{u}^e \tag{9.3}$$

and

$$u^e(r,s) = \mathbf{P}(r,s)\,\mathbf{g}^{-1}\,\mathbf{u}^e = \mathbf{H}(r,s)\mathbf{u}^e . \tag{9.4}$$

Here

$$\mathbf{g}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{9.5}$$

and

$$H_1(r,s) = 1 - r - s\ , \qquad H_2(r,s) = r\ , \qquad H_3(r,s) = s\ . \tag{9.6}$$

By inspection, one can see that the sum of these functions at all points in the local domain is unity. This is illustrated graphically at the bottom of Fig. 9.2.1. Typical coding for these relations and their local derivatives are shown as subroutines *SHAPE*_3_*T* and *DERIV*_3_*T* in Fig. 9.2.2. Similarly, for the unit coordinate bilinear quadrilateral

mapping from $0 < (r, s) < 1$ one could assume that

$$u^e(r,s) = d_1^e + d_2^e\, r + d_3^e\, s + d_4^e\, rs \tag{9.7}$$

so that

$$\mathbf{g} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{9.8}$$

and

```
SUBROUTINE SHAPE_3_T (S, T, H)                                    ! 1
! *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-*    ! 2
! SHAPE FUNCTIONS FOR A THREE NODE UNIT TRIANGLE                  ! 3
! *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-*    ! 4
Use Precision_Module                                              ! 5
 IMPLICIT NONE                                                    ! 6
 REAL(DP), INTENT(IN)  :: S, T                                    ! 7
 REAL(DP), INTENT(OUT) :: H (3)                                   ! 8
                                                                  ! 9
! S,T = LOCAL COORDINATES OF THE POINT    3     T                 !10
! H   = SHAPE FUNCTIONS                   . .   .                 !11
! NODAL COORDS 1-(0,0)  2-(1,0)  3-(0,1)  1..2  0..S              !12
                                                                  !13
  H (1) = 1.d0 - S - T                                            !14
  H (2) = S                                                       !15
  H (3) = T                                                       !16
END SUBROUTINE SHAPE_3_T                                          !17
                                                                  !18
SUBROUTINE DERIV_3_T (S, T, DH)                                   !19
! *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-*    !20
! LOCAL DERIVATIVES OF A THREE NODE UNIT TRIANGLE                 !21
!            SEE SUBROUTINE SHAPE_3_T                             !22
! *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-*    !23
Use Precision_Module                                              !24
 IMPLICIT NONE                                                    !25
 REAL(DP), INTENT(IN)  :: S, T                                    !26
 REAL(DP), INTENT(OUT) :: DH (2, 3)                               !27
                                                                  !28
! S,T     = LOCAL COORDINATES OF THE POINT                        !29
! DH(1,K) = DH(K)/DS                                              !30
! DH(2,K) = DH(K)/DT                                              !31
! NODAL COORDS ARE :  1-(0,0) 2-(1,0) 3-(0,1)                     !32
                                                                  !33
  DH (1, 1) = - 1.d0                                              !34
  DH (1, 2) = 1.d0                                                !35
  DH (1, 3) = 0.d0                                                !36
  DH (2, 1) = - 1.d0                                              !37
  DH (2, 2) = 0.d0                                                !38
  DH (2, 3) = 1.d0                                                !39
END SUBROUTINE DERIV_3_T                                          !40
```

Figure 9.2.2 Coding a linear unit coordinate triangle

$$\begin{aligned} H_1(r,s) &= 1 \quad -r \quad -s \quad +rs \\ H_2 &= \quad r \quad\quad -rs \\ H_3 &= \quad\quad\quad rs \\ H_4 &= \quad\quad s \quad -rs\,. \end{aligned} \tag{9.9}$$

However, for the quadrilateral it is more common to utilize the natural coordinates, as shown in Fig. 9.2.3. In that coordinate system $-1 \le a, b \le +1$ so that

$$\mathbf{g} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

and the alternate interpolation functions are

$$H_i\,(a,b) \;=\; (1 + aa_i\,)\,(1 + bb_i\,)/4\,, \qquad 1 \le i \le 4 \tag{9.10}$$

where $(a_i\,,\ b_i\,)$ are the local coordinates of node i. These four functions and their local derivatives can be coded as shown in Fig. 9.2.3.

Note that up to this point we have utilized the local element coordinates for interpolation. Doing so makes the geometry matrix, $\mathbf{g}$, depend only on element type instead of element number. If we use global coordinates then the geometric matrix, $\mathbf{g}^e$ is always dependent on the element number, e. For example, if Eq. 9.1 is written in physical coordinates then

$$u^e(x,y) \;=\; d_1^e + d_2^e\,x + d_3^e\,y \tag{9.11}$$

so when the identities are evaluated at each node the result is

$$\mathbf{g}^e \;=\; \begin{bmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{bmatrix}. \tag{9.12}$$

Inverting and simplifying the algebra gives the global coordinate equivalent of Eq. 9.6 for a specific element :

$$H_i^e(x,y) \;=\; (a_i^e + b_i^e\,x + c_i^e\,y)/2A^e\,, \qquad 1 \le i \le 3 \tag{9.13}$$

where the algebraic constants are

$$\begin{aligned} a_1^e &= x_2^e y_3^e - x_3^e y_2^e \qquad b_1^e = y_2^e - y_3^e \qquad c_1^e = x_3^e - x_2^e \\ a_2^e &= x_3^e y_1^e - x_1^e y_3^e \qquad b_2^e = y_3^e - y_1^e \qquad c_2^e = x_1^e - x_3^e \\ a_3^e &= x_1^e y_2^e - x_2^e y_1^e \qquad b_3^e = y_1^e - y_2^e \qquad c_3^e = x_2^e - x_1^e \end{aligned} \tag{9.14}$$

and A^e is the area of the element, that is, $A^e = (a_1^e + a_2^e + a_3^e)/2$, or

$$A^e \;=\; \left[\; x_1^e(y_2^e - y_3^e) + x_2^e(y_3^e - y_1^e) + x_3^e(y_1^e - y_2^e)\;\right] / 2\,.$$

These algebraic forms assume that the three local nodes are numbered counter-clockwise from an arbitrarily selected corner. If the topology is defined in a clockwise order then


```
SUBROUTINE SHAPE_4_Q (R, S, H)                                      ! 1
! *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* ! 2
! SHAPE FUNCTIONS OF A 4 NODE PARAMETRIC QUAD                       ! 3
!          IN NATURAL COORDINATES                                   ! 4
! *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* ! 5
Use Precision_Module                                                ! 6
 IMPLICIT NONE                                                      ! 7
 REAL(DP), INTENT(IN)  :: R, S                                      ! 8
 REAL(DP), INTENT(OUT) :: H (4)                                     ! 9
 REAL(DP)              :: R_P, R_M, S_P, S_M                        !10
                                                                    !11
! (R,S) = A POINT IN THE NATURAL COORDS     4---3                   !12
! H     = LOCAL INTERPOLATION FUNCTIONS     |   |                   !13
! H(I)  = 0.25d0*(1+R*R(I))*(1+S*S(I))      |   |                   !14
! R(I)  = LOCAL R-COORDINATE OF NODE I      1---2                   !15
! LOCAL COORDS, 1=(-1,-1)   3=(+1,+1)                               !16
                                                                    !17
  R_P = 1.d0 + R ; R_M = 1.d0 - R                                   !18
  S_P = 1.d0 + S ; S_M = 1.d0 - S                                   !19
  H (1) = 0.25d0*R_M*S_M                                            !20
  H (2) = 0.25d0*R_P*S_M                                            !21
  H (3) = 0.25d0*R_P*S_P                                            !22
  H (4) = 0.25d0*R_M*S_P                                            !23
END SUBROUTINE SHAPE_4_Q                                            !24
                                                                    !25
SUBROUTINE DERIV_4_Q (R, S, DELTA)                                  !26
! *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* !27
! LOCAL DERIVATIVES OF THE SHAPE FUNCTIONS FOR AN                   !28
! PARAMETRIC QUADRILATERAL WITH FOUR NODES                          !29
!            SEE SHAPE_4_Q                                        !30
! *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* *-* !31
Use Precision_Module                                                !32
 IMPLICIT NONE                                                      !33
 REAL(DP), INTENT(IN)  :: R, S                                      !34
 REAL(DP), INTENT(OUT) :: DELTA (2, 4)                              !35
 REAL(DP)              :: R_P, R_M, S_P, S_M                        !36
                                                                    !37
! DELTA(1,I) = DH/DR                                                !38
! DELTA(2,I) = DH/DS                                                !39
! H          = LOCAL INTERPOLATION FUNCTIONS                        !40
! (R,S)      = A POINT IN THE LOCAL COORDINATES                     !41
! HERE D(H(I))/DR = 0.25d0*R(I)*(1+S*S(I)), ETC.                    !42
                                                                    !43
  R_P = 1.d0 + R ; R_M = 1.d0 - R                                   !44
  S_P = 1.d0 + S ; S_M = 1.d0 - S                                   !45
  DELTA (1, 1) = -0.25d0 * S_M                                      !46
  DELTA (1, 2) =  0.25d0 * S_M                                      !47
  DELTA (1, 3) =  0.25d0 * S_P                                      !48
  DELTA (1, 4) = -0.25d0 * S_P                                      !49
  DELTA (2, 1) = -0.25d0 * R_M                                      !50
  DELTA (2, 2) = -0.25d0 * R_P                                      !51
  DELTA (2, 3) =  0.25d0 * R_P                                      !52
  DELTA (2, 4) =  0.25d0 * R_M                                      !53
END SUBROUTINE DERIV_4_Q                                            !54
```

Figure 9.2.3 Coding a bi-linear quadrilateral

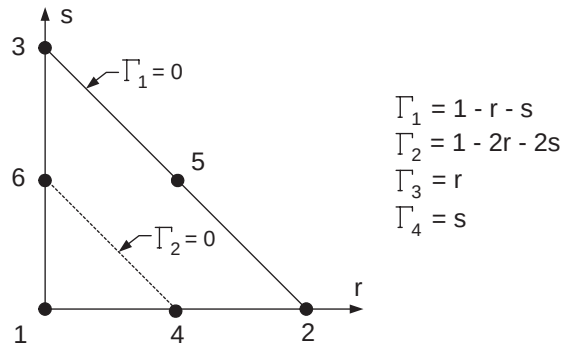


Figure 9.2.4 Boundary curves through element nodes

the area, A^e, becomes negative.

It would be natural at this point to attempt to utilize a similar procedure to define the four node quadrilateral in the same manner. For example, if Eq. 9.7 is written as

$$u^e(x, y) = d_1^e + d_2^e\, x + d_3^e\, y + d_4^e\, xy\,. \tag{9.15}$$

However, we now find that for a general quadrilateral the inverse of matrix $\mathbf{g}^e$ may not exist. This means that the global coordinate interpolation is in general very sensitive to the orientation of the element in global space. That is very undesirable. This important disadvantage vanishes only when the element is a rectangle. This global form of interpolation also yields an element that fails to satisfy the required interelement continuity requirements. These difficulties are typical of those that are encountered in two- and three-dimensions when global coordinate interpolation is utilized. Therefore, it is most common to employ the local coordinate mode of interpolation. Doing so also easily allows for the treatment of curvilinear elements. That is done with *isoparametric elements* that will be mentioned later.

It is useful to illustrate the lack of continuity that develops in the global coordinate form of the quadrilateral. First, consider the three-node triangular element and examine the interface or boundary where two elements connect. Along the interface between the two elements one has the geometric restriction that the edge is a straight line given by $y = m^b x + n^b$. The general form of the global coordinate interpolation functions for the triangle is $u(x, y) = d_1^e + d_2^e\, x + d_3^e\, y$ where the g_i are element constants. Along the typical interface this reduces to $u = d_1^e + d_2^e\, x + d_3^e\,(m^b x + n^b)$, or simply $u = f_1 + f_2 x$. Clearly, this shows that the boundary displacement is a linear function of x. The two constants, f_i, could be uniquely determined by noting that $u(x_1) = u_1$ and $u(x_2) = u_2$. Since those two quantities are common to both, elements the displacement, $u(x)$, will be continuous between the two elements. By way of comparison when the same substitution is made in Eq. 9.15 the resulting edge value for the quadrilateral element is $u = d_1^e + d_2^e\, x + d_3^e\,(m^b x + n^b) + d_4^e\, x\,(m^b x + n^b)$, or simply $u = f_1 + f_2\, x + f_3\, x^2$. This quadratic function cannot be uniquely defined by the two constants u_1 and u_2. Therefore, it is not possible to prove that the displacements will be continuous between elements. This is an undesirable feature of quadrilateral elements when formulated in global coordinates. If the quadrilateral interpolation is given in local coordinates such as Eq. 9.9

or Eq. 9.10, this problem does not occur. On the edge $s = 0$, Eq. 9.9 reduces to $u = f_1 + f_2 r$. A similar result occurs on the edge $s = 1$. Likewise, for the other two edges $u = f_1 + f_2 s$. Thus, in local coordinates the element degenerates to a linear function on any edge, and therefore will be uniquely defined by the two shared nodal displacements. In other words, the local coordinate four node quadrilateral will be compatible with elements of the same type and with the three-node triangle. The above observations suggest that global coordinates could be utilized for the four-node element only so long as it is a rectangle parallel to the global axes.

The extension of the unit coordinates to the three-dimensional tetrahedra illustrated in Fig. 3.2.2 is straightforward. In the result given below

$$\begin{aligned} H_1(r,s,t) &= 1 - r - s - t \qquad & H_2(r,s,t) &= r \\ H_3(r,s,t) &= s & H_4(r,s,t) &= t\,, \end{aligned} \tag{9.16}$$

and comparing this to Eqs. 9.6 and 4.11, we note that the 2-D and 1-D forms are contained in the three-dimensional form. This concept was suggested by the topology relations shown in Fig. 3.2.2. The unit coordinate interpolation is easily extended to quadratic, cubic, or higher interpolation. The procedure employed to generate Eq. 9.6 can be employed. An alternate geometric approach can be utilized. We want to generate an interpolation function, H_i, that vanishes at the j-th node when $i \neq j$. Such a function can be obtained by taking the products of the equations of selected curves through the nodes on the element. For example, let $H_1(r,s) = C_1 \Gamma_1 \Gamma_2$ where the Γ_i are the equations of the lines are shown in Fig. 9.2.4, and where C_1 is a constant chosen so that $H_1(r_1\,,\ s_1) = 1$. This yields

$$H_1 = (1 - 3r - 3s + 2r^2 + 4rs + 2s^2)\,.$$

Similarly, letting $H_4 = C_4 \Gamma_1 \Gamma_3$ gives $C_4 = 4$ and $H_4 = 4r(1 - r - s)$. This type of procedure is usually quite straightforward. However, there are times when there is not a unique choice of products, and then care must be employed to select the proper products. The resulting two-dimensional interpolation functions for the quadratic triangle are

$$\begin{aligned} H_1(r,s) &= 1 - 3r + 2r^2 - 3s + 4rs + 2s^2 \\ H_2(r,s) &= -r + 2r^2 \\ H_3(r,s) &= -s + 2s^2 \\ H_4(r,s) &= 4r - 4r^2 - 4rs \\ H_5(r,s) &= 4rs \\ H_6(r,s) &= 4s - 4rs - 4s^2\,. \end{aligned} \tag{9.17}$$

Once again, it is possible to obtain the one-dimensional quadratic interpolation on a typical edge by setting $s = 0$. Figure 9.2.5 shows the shape of the typical interpolation functions for a linear and quadratic triangular element.

Figure 9.2.6 illustrates the concept of Pascal's triangle for representing the complete polynomial terms in three dimensions. Beginning with the constant vertex (1), it can also be thought of as as showing the polynomials that occur in the tetrahedron of linear, quadratic, cubic, and quartic degree, respectively, and the relative location of the nodes on the edges, faces, and interior of the tetrahedron. If one sets $z = 1$ then it can also show

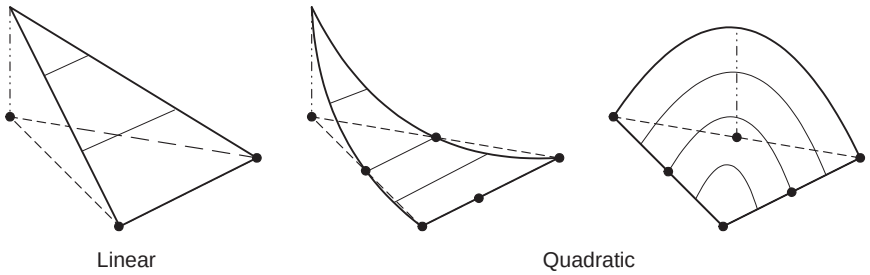


Figure 9.2.5 Linear and quadratic triangle interpolation

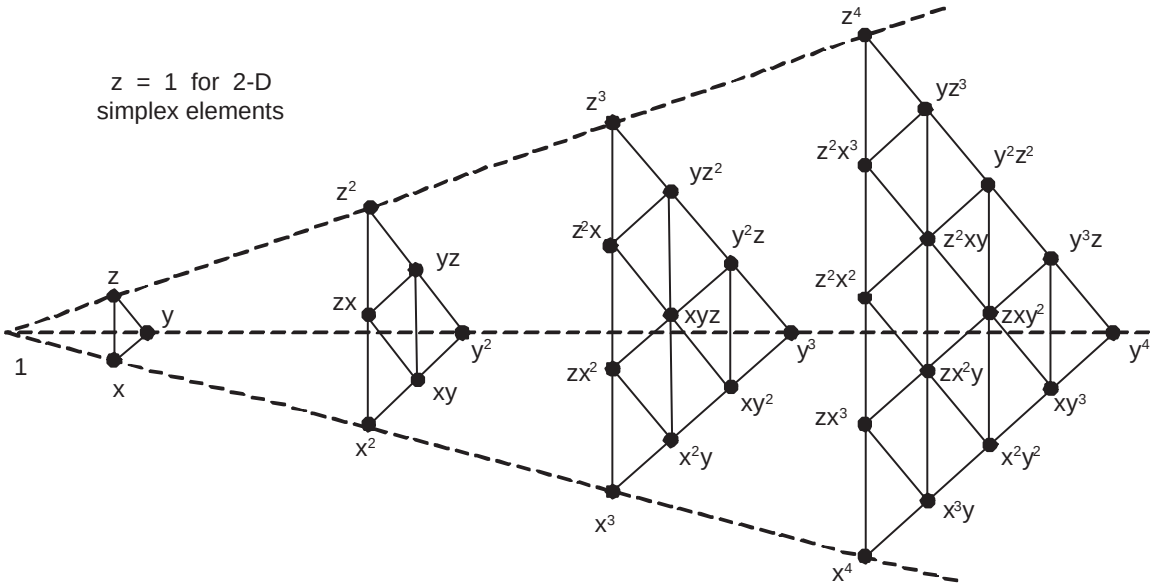


Figure 9.2.6 The 2-D Pascal triangles and the 3-D simplex family

the relative nodes and polynomials for the triangular elements of linear, quadratic, cubic, and quartic degree from the left-most to the right-most triangles, respectively.

9.3 Natural Coordinates

The natural coordinate formulations for the interpolation functions can be generated in a similar manner to that illustrated in Eq. 9.10. However, the inverse geometric matrix, $\mathbf{G}^{-1}$, may not exist. However, the most common functions have been known for several years and will be presented here in two groups. They are generally denoted as Lagrangian elements and as the Serendipity elements (see Tables 9.1 and 9.2). For the four-node quadrilateral element both forms yield Eq. 9.10. This is known as the bi-linear quadrilateral since it has linear interpolation on its edges and a bi-linear (incomplete quadratic) interpolation on its interior. This element is easily extended to the tri-linear hexahedra of Table 9.2. Its resulting interpolation functions are

$$H_i(a,b,c) = (1+aa_i)(1+bb_i)(1+cc_i)/8, \tag{9.18}$$

for $1 \le i \le 8$ where (a_i, b_i, c_i) are the local coordinates of node i. On a given face, e.g., $c = \pm 1$, these degenerate to the four functions in Eq. 9.10 and four zero terms. For quadratic (or higher) edge interpolation, the Lagrangian and Serendipity elements are different. The Serendipity interpolation functions for the corner quadratic nodes are

$$H_i(a,b) = (1+aa_i)(1+bb_i)(aa_i+bb_i-1)/4, \tag{9.19}$$

where $1 \le i \le 4$ and for the mid-side nodes

$$H_i(a,b) = a_i^2(1-b^2)(1+a_i\,a)/2 + b_i^2(1-a^2)(1+b_i\,b)/2, \quad 5 \le i \le 8. \tag{9.20}$$

Other members of this family are listed in Tables 9.1 and 9.2. The two-dimensional Lagrangian functions are obtained from the products of the one-dimensional equations. The resulting quadratic functions are

$$\begin{aligned}
H_1(a,b) &= (a^2-a)(b^2-b)/4 & H_6(a,b) &= (a^2+a)(1-b^2)/2\\
H_2(a,b) &= (a^2+a)(b^2-b)/4 & H_7(a,b) &= (1-a^2)(b^2+b)/2\\
H_3(a,b) &= (a^2+a)(b^2+b)/4 & H_8(a,b) &= (a^2-a)(1-b^2)/2\\
H_4(a,b) &= (a^2-a)(b^2+b)/4 & H_9(a,b) &= (1-a^2)(1-b^2)\\
H_5(a,b) &= (1-a^2)(b^2-b)/2.
\end{aligned}$$

The typical shapes of these functions are shown in Fig. 9.3.1. The function $H_9(a,b)$ is referred to as a *bubble function* because it is zero on the boundary of the element and looks like a soap bubble blown up over the element. Similar functions are commonly used in hierarchical elements to be considered later. It is possible to mix the order of interpolation on the edges of an element. Figure 9.3.2 illustrates the Serendipity interpolation functions for quadrilateral elements that can be either linear, quadratic, or cubic on any of its four sides. Such an element is often referred to as a *transition element*. They can also be employed as *p-adaptive elements*. Those types of elements are sketched in Fig. 9.3.3. From the previous figures one will note that the supplied routines in the interpolation library generally start with the names SHAPE_ and DERIV_ and have the number of nodes and shape codes (L-line, T-triangle, Q-quadrilateral, H-hexahedron, P-pyramid or tetrahedron, and W-wedge) appended to those names. The class of elements shown in Fig. 9.3.3 are appended with the name L_Q_H because they can be determined for any of the three shapes. For elements of degree four or higher one needs to also include interior nodes for elements in Fig. 9.3.3 to form complete polynomials, or the rate of convergence will be decreased.

9.4 Isoparametric and Subparametric Elements

By introducing local coordinates to formulate the element interpolation functions we were able to satisfy certain continuity requirements that could not be satisfied by global coordinate interpolation. We will soon see that a useful by-product of this approach is the ability to treat elements with curved edges. At this point there may be some concern about how one relates the local coordinates to the global coordinates. This

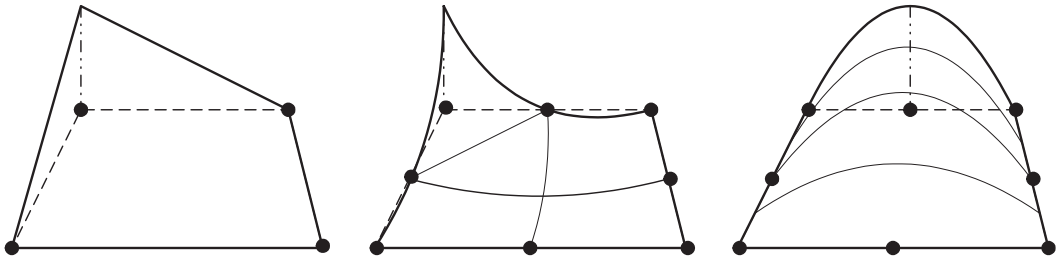


Figure 9.3.1 Quadratic Serendipity quadrilateral interpolation

Topology:

```
4 − 11 − 7 − 3
|           |
8     S     10
|     *R    |
12          6
|           |
1 − 5 − 9 − 2
```

If Cubic Side : $i = 5, 9,$ or $6, 10$ or $7, 11$ or $8, 12$

$$H_i(r,s) = (1-s^2)(1+9ss_i)(1+rr_i)\,9/32$$
$$H_i(r,s) = (1-r^2)(1+9rr_i)(1+ss_i)\,9/32$$

If Quadratic Side : $i = 5,\ 6,\ 7,$ or 8

$$H_i(r,s) = (1+rr_i)(1-s^2)/2$$
$$H_i(r,s) = (1+ss_i)(1-r^2)/2$$
$$H_j = 0, \quad j = i+4$$

If Linear Side :

$$H_j = H_k = 0, \quad j = i+4, \quad k = i+8, \quad i = 1, 2, 3, \text{ or } 4$$

If Corners : $i = 1, 2, 3, 4$ $\quad H_i(r,s) = (P_r + P_s)(1+ss_i)/4$

See subroutine SHAPE_4_12_Q

Order of Side	$P_r,\ s_i = \pm 1$	$P_s,\ r_i = \pm 1$
Linear	1/2	1/2
Quadratic	$rr_i - 1/2$	$ss_i - 1/2$
Cubic	$(9r^2 - 5)/8$	$(9s^2 - 5)/8$

Figure 9.3.2 Linear to cubic transition quadrilateral

Table 9.1. Serendipity quadrilaterals in natural coordinates

Node Location		Interpolation Functions	Name
a_i	b_i	$H_i(a,b)$	
± 1	± 1	$(1+aa_i)(1+bb_i)/4$	Q4
± 1	± 1	$(1+aa_i)(1+bb_i)(aa_i+bb_i-1)/4$	Q8
± 1	0	$(1+aa_i)(1-b^2)/2$	
0	± 1	$(1+bb_i)(1-a^2)/2$	
± 1	± 1	$(1+aa_i)(1+bb_i)[9(a^2+b^2)-10]/32$	Q12
± 1	$\pm 1/3$	$9(1+aa_i)(1-b^2)(1+9bb_i)/32$	
$\pm 1/3$	± 1	$9(1+bb_i)(1-a^2)(1+9aa_i)/32$	
± 1	± 1	$(1+aa_i)(1+bb_i)[4(a^2-1)aa_i + 4(b^2-1)bb_i + 3abaa_i b_i]/12$	Q16
± 1	0	$2(1+aa_i)(b^2-1)(b^2-aa_i)/4$	
0	± 1	$2(1+bb_i)(a^2-1)(a^2-bb_i)/4$	
± 1	$\pm 1/2$	$4(1+aa_i)(1-b^2)(b^2+bb_i)/3$	
$\pm 1/2$	± 1	$4(1+bb_i)(1-a^2)(a^2+aa_i)/3$	
0	0	$(a^2-1)(b^2-1)$	

Table 9.2. Serendipity hexahedra in natural coordinates

Node Location			Interpolation Functions	Name
a_i	b_i	c_i	$H_i(a,b,c)$	
± 1	± 1	± 1	$(1+aa_i)(1+bb_i)(1+cc_i)/8$	H8
± 1	± 1	± 1	$(1+aa_i)(1+bb_i)(1+cc_i)(aa_i+bb_i+cc_i-2)/8$	H20
0	± 1	± 1	$(1-a^2)(1+bb_i)(1+cc_i)/4$	
± 1	0	± 1	$(1-b^2)(1+aa_i)(1+cc_i)/4$	
± 1	± 1	0	$(1-c^2)(1+aa_i)(1+bb_i)/4$	
± 1	± 1	± 1	$(1+aa_i)(1+bb_i)(1+cc_i)[9(a^2+b^2+c^2)-19]/64$	H32
$\pm 1/3$	± 1	± 1	$9(1-a^2)(1+9aa_i)(1+bb_i)(1+cc_i)/64$	
± 1	$\pm 1/3$	± 1	$9(1-b^2)(1+9bb_i)(1+aa_i)(1+cc_i)/64$	
± 1	± 1	$\pm 1/3$	$9(1-c^2)(1+9cc_i)(1+bb_i)(1+aa_i)/64$	

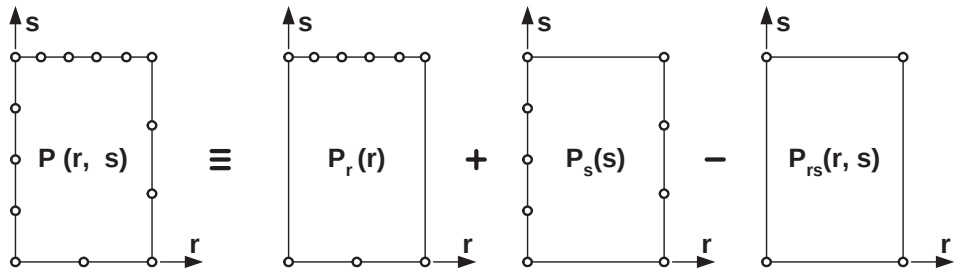


Figure 9.3.3 Blended quadrilaterals of different edge degrees

must be done since the governing integral is presented in global (physical) coordinates and it involves derivatives with respect to the global coordinates. This can be accomplished with the popular *isoparametric elements*, and *subparametric* elements.

Isoparametric elements utilize a local coordinate system to formulate the element matrices. The local coordinates, say r, s, and t, are usually dimensionless and range from 0 to 1, or from -1 to 1. The latter range is usually preferred since it is directly compatible with the usual definition of abscissa utilized in numerical integration by Gaussian quadratures. The elements are called isoparametric since the same (iso) local coordinate parametric equations (interpolation functions) used to define any quantity of interest within the elements are also utilized to define the global coordinates of any point within the element in terms of the global spatial coordinates of the nodal points. If a lower order polynomial is used to describe the geometry then it is called a *subparametric element*. These are quite common when used with the newer hierarchical elements. Let the global spatial coordinates again be denoted by x, y, and z,. let the number of nodes per element be n_n. For simplicity, consider a single scalar quantity of interest, say $V(r, s, t)$. The value of this variable at any local point (r, s, t) within the element is assumed to be defined by the values at the n_n nodal points of the element (V_i^e, $1 \le i \le n_n$), and a set of interpolation functions ($H_i(r, s, t)$, $1 \le i \le n_n$). That is,

$$V(r, s, t) \;=\; \sum_{i=1}^{n_n} H_i(r, s, t)\, V_i^e \;=\; \mathbf{H}(\mathbf{r})\, \mathbf{V}^e\,, \tag{9.21}$$

where $\mathbf{H}$ is a row vector. Generalizing this concept, the global coordinates are defined with a geometric interpolation, or blending, function, $\mathbf{G}$. If it interpolates between n_x geometric data points then it is subparametric if $n_x < n_n$, isoparametric if $n_x = n_n$ so $\mathbf{G} = \mathbf{H}$, and superparametric if $n_x > n_n$. Blending functions typically use geometric data everywhere on the edge of the geometric element. The geometric interpolation, or blending, is denoted as: $x(r, s, t) = \mathbf{G}\,\mathbf{x}^e$, $y = \mathbf{G}\,\mathbf{y}^e$, and $z = \mathbf{G}\,\mathbf{z}^e$. Programming considerations make it desirable to write the last three relations as a position row matrix, $\mathbf{R}$, written in a partitioned form

$$\mathbf{R}(r, s, t) \;=\; \mathbf{G}(r, s, t)\, \mathbf{R}^e \;=\; \mathbf{G}\,[\,\mathbf{x}^e\ \mathbf{y}^e\ \mathbf{z}^e\,] \tag{9.22}$$

where the last matrix simply contains the spatial coordinates of the n_n nodal points incident with the element. If $\mathbf{G} = \mathbf{H}$, it is an isoparametric element. To illustrate a typical two-dimensional isoparametric element, consider a quadrilateral element with nodes at

the four corners, as shown in Fig. 9.2.3. The global coordinates and local coordinates of a typical corner, i, are $(x_i\,, y_i\,)$, and $(r_i\,, s_i\,)$, respectively. The following local coordinate interpolation functions have been developed earlier for this element:

$$H_i\,(r,s) \;=\; \frac{1}{4}\,(1+rr_i\,)\,(1+ss_i\,)\,, \qquad 1 \le i \le 4\,.$$

We interpolate any variable, V, as

$$V(r,s) \;=\; \mathbf{H}(r,s)\,\mathbf{V}^e \;=\; \lfloor\, H_1 \quad H_2 \quad H_3 \quad H_4 \,\rfloor \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{Bmatrix}^e .$$

Note that along an edge of the element ($r = \pm 1$ or $s = \pm 1$), these interpolation functions become linear and thus any of these three quantities can be uniquely defined by the two corresponding nodal values on that edge. If the adjacent element is of the same type (linear on the boundary), then these quantities will be continuous between elements since their values are uniquely defined by the shared nodal values on that edge. Since the variable of interest, V, varies linearly on the edge of the element, it is called the linear isoparametric quadrilateral although the interpolation functions are bilinear inside the element. If the (x, y) coordinates are also varying linearly with r or s on a side it means this element has straight sides.

For future reference, note that if one can define the interpolation functions in terms of the local coordinates then one can also define their partial derivatives with respect to the local coordinate system. For example, the local derivatives of the interpolation functions of the above element are

$$\partial H_i\,(r,s)/\partial r \;=\; r_i\,(1+ss_i\,)/4\,, \qquad \partial H_i\,(r,s)/\partial s \;=\; s_i\,(1+rr_i\,)/4\,.$$

In three dimensions ($n_s = 3$), let the array containing the local derivatives of the interpolation functions be denoted by **DL_H**, a $3 \times n_n$ matrix, where

$$\mathbf{DL\_H}\,(r,s,t) \;=\; \begin{bmatrix} \dfrac{\partial}{\partial r}\,\mathbf{H} \\ \dfrac{\partial}{\partial s}\,\mathbf{H} \\ \dfrac{\partial}{\partial t}\,\mathbf{H} \end{bmatrix} \;=\; \partial_L \mathbf{H}\,. \tag{9.23}$$

Although x, y, and z can be defined in an isoparametric element in terms of the local coordinates, r, s, and t, a unique inverse transformation is not needed. Thus, one usually does not define r, s, and t in terms of x, y, and z. What one must have, however, are the relations between derivatives in the two coordinate systems. From calculus, it is known that the derivatives are related by the *Jacobian*. From the chain rule of calculus one can write, in general,

$$\frac{\partial}{\partial r} \;=\; \frac{\partial}{\partial x}\,\frac{\partial x}{\partial r} + \frac{\partial}{\partial y}\,\frac{\partial y}{\partial r} + \frac{\partial}{\partial z}\,\frac{\partial z}{\partial r}$$

with similar expressions for $\partial/\partial s$ and $\partial/\partial t$. In matrix form these identities become

$$\begin{Bmatrix} \dfrac{\partial}{\partial r} \\ \dfrac{\partial}{\partial s} \\ \dfrac{\partial}{\partial t} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} & \dfrac{\partial z}{\partial r} \\ \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} & \dfrac{\partial z}{\partial s} \\ \dfrac{\partial x}{\partial t} & \dfrac{\partial y}{\partial t} & \dfrac{\partial z}{\partial t} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial z} \end{Bmatrix} \tag{9.24}$$

where the square matrix is called the *Jacobian.* Symbolically, one can write the derivatives of a quantity, such as $V(r, s, t)$, which for convenience is written as $V(x, y, z)$ in the global coordinate system, in the following manner: $\boldsymbol{\partial}_L V = \mathbf{J}(r, s, t)\, \boldsymbol{\partial}_g V$, where $\mathbf{J}$ is the Jacobian matrix, and where the subscripts L and g have been introduced to denote local and global derivatives, respectively. Similarly, the inverse relation is

$$\boldsymbol{\partial}_g V = \mathbf{J}^{-1}\, \boldsymbol{\partial}_L V\,. \tag{9.25}$$

Thus, to evaluate global and local derivatives, one must be able to establish the Jacobian, $\mathbf{J}$, of the geometric mapping and its inverse, $\mathbf{J}^{-1}$. In practical application, these two quantities usually are evaluated numerically. Consider the first term in $\mathbf{J}$ that relates the geometric mapping: $\partial x/\partial r = \partial\,(\mathbf{G}\,\mathbf{x}^e)/\partial r = \partial \mathbf{G}/\partial r\ \mathbf{x}^e$. Similarly, for any component in Eq. 9.22 $\partial \mathbf{R}/\partial r = \partial(\mathbf{G}\,\mathbf{R}^e)/\partial r$. Repeating for all local directions, and noting that the $\mathbf{R}^e$ values are constant input coordinate data for the element, we find the identity that

$$\begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} & \dfrac{\partial z}{\partial r} \\ \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} & \dfrac{\partial z}{\partial s} \\ \dfrac{\partial x}{\partial t} & \dfrac{\partial y}{\partial t} & \dfrac{\partial z}{\partial t} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial r}\,\mathbf{G} \\ \dfrac{\partial}{\partial s}\,\mathbf{G} \\ \dfrac{\partial}{\partial t}\,\mathbf{G} \end{bmatrix} \mathbf{R}^e$$

or, in symbolic form, the evaluation of the definition of the Jacobian within a specific element takes the form

$$\mathbf{J}^e(r, s, t) = \mathbf{DL\_G}(r, s, t)\, \mathbf{R}^e\,. \tag{9.26}$$

This numerically defines the Jacobian matrix, $\mathbf{J}$, at a local point inside a typical element in terms of the spatial coordinates of the element's nodes, $\mathbf{R}^e$, which is referenced by the name *COORD* in the subroutines, and the local derivatives, **DL_G**, of the geometric interpolation functions, $\mathbf{G}$. Thus, at any point (r, s, t) of interest, such as a numerical integration point, it is possible to define the values of $\mathbf{J}$, $\mathbf{J}^{-1}$, and the determinant of the Jacobian, $|\mathbf{J}|$. In practice, evaluation of the Jacobian is simply a matrix product, such as *AJ = MATMUL(DL_G, COORD)*. We usually will consider two-dimensional problems. Then the Jacobian matrix is

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} \\ \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} \end{bmatrix}.$$

In general, the inverse Jacobian in two dimensions is

$$\mathbf{J}^{-1} = \frac{1}{|J|}\begin{bmatrix} \dfrac{\partial y}{\partial s} & -\dfrac{\partial y}{\partial r} \\ -\dfrac{\partial x}{\partial s} & \dfrac{\partial x}{\partial r} \end{bmatrix}, \qquad \text{where} \quad |J| = x_{,r}\, y_{,s} - y_{,r}\, x_{,s}\,.$$

For future reference, note that by denoting $(\ )_{,r} = \partial(\ )/\partial r$, etc. the determinant and inverse of the three-dimensional Jacobian are

$$|\mathbf{J}| = x_{,r}\,(y_{,s}\,z_{,t} - y_{,t}\,z_{,s}) + x_{,s}\,(y_{,t}\,z_{,r} - y_{,r}\,z_{,t}) + x_{,t}\,(y_{,r}\,z_{,s} - y_{,s}\,z_{,r})$$

and

$$\mathbf{J}^{-1} = \begin{bmatrix} (y_{,s}\,z_{,t} - y_{,t}\,z_{,s}) & (y_{,t}\,z_{,r} - y_{,r}\,z_{,t}) & (y_{,r}\,z_{,s} - y_{,s}\,z_{,r}) \\ (x_{,t}\,z_{,s} - x_{,s}\,z_{,t}) & (x_{,r}\,z_{,t} - x_{,t}\,z_{,r}) & (x_{,s}\,z_{,r} - x_{,r}\,z_{,s}) \\ (x_{,s}\,y_{,t} - x_{,t}\,y_{,s}) & (x_{,t}\,y_{,r} - x_{,r}\,y_{,t}) & (x_{,r}\,y_{,s} - x_{,s}\,y_{,r}) \end{bmatrix} \Big/ |\mathbf{J}|.$$

Of course, one can in theory also establish the algebraic form of **J**. For simplicity consider the three-node isoparametric triangle in two dimensions. From Eq. 9.6 we note that the local derivatives of **G** are

$$\mathbf{DL\_G} = \begin{bmatrix} \partial \mathbf{G}/\partial r \\ \partial \mathbf{G}/\partial s \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}. \tag{9.27}$$

Thus, the element has constant local derivatives since no functions of the local coordinates remain. Usually the local derivatives are also polynomial functions of the local coordinates. Employing Eq. 9.26 for a specific T3 element:

$$\mathbf{J}^e = \mathbf{DL\_G}\,\mathbf{R}^e = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}^e$$

or simply

$$\mathbf{J}^e = \begin{bmatrix} (x_2 - x_1) & (y_2 - y_1) \\ (x_3 - x_1) & (y_3 - y_1) \end{bmatrix}^e \tag{9.28}$$

which is also constant. The determinant of this 2×2 Jacobian matrix is

$$|\,\mathbf{J}^e\,| = (x_2 - x_1)^e\,(y_3 - y_1)^e - (x_3 - x_1)^e\,(y_2 - y_1)^e = 2A^e\,,$$

which is twice the physical area of the element physical domain, Ω^e. For the above three-node triangle, the inverse relation is simply

$$\mathbf{J}^{e^{-1}} = \frac{1}{2A^e}\begin{bmatrix} (y_3 - y_1) & -(y_2 - y_1) \\ -(x_3 - x_1) & (x_2 - x_1) \end{bmatrix}^e = \frac{1}{2A^e}\begin{bmatrix} b_2 & b_3 \\ c_2 & c_3 \end{bmatrix}^e. \tag{9.29}$$

For most other elements it is common to form these quantities numerically by utilizing the numerical values of $\mathbf{R}^e$ given in the data. The use of the local coordinates in effect represents a change of variables. In this sense the Jacobian has another important function. The determinant of the Jacobian, $|\mathbf{J}|$, relates differential changes in the two coordinate systems, that is,

$$\begin{aligned} dL &= dx = |\mathbf{J}|\, dr \\ da &= dx\, dy = |\mathbf{J}|\, dr\, ds \\ dv &= dx\, dy\, dz = |\mathbf{J}|\, dr\, ds\, dt \end{aligned}$$

in one-, two-, and three-dimensional problems. When the local and physical spaces have the same number of dimensions we can write this symbolically as $d\,\Omega^e = |\mathbf{J}|\, d\,\square^e$.

The integral definitions of the element matrices usually involve the global derivatives of the quantity of interest. From Eq. 9.21 it is seen that the local derivatives of V are related to the nodal parameters by

$$\begin{Bmatrix} \dfrac{\partial V}{\partial r} \\ \dfrac{\partial V}{\partial s} \\ \dfrac{\partial V}{\partial t} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial r}\mathbf{H} \\ \dfrac{\partial}{\partial s}\mathbf{H} \\ \dfrac{\partial}{\partial t}\mathbf{H} \end{bmatrix} \mathbf{V}^e ,$$

or symbolically,

$$\boldsymbol{\partial}_L V(r,s,t) = \mathbf{DL\_H}(r,s,t)\, \mathbf{V}^e . \tag{9.30}$$

To relate the global derivatives of V to the nodal parameters, $\mathbf{V}^e$, one substitutes the above expression, and the geometry mapping Jacobian into Eq. 9.25 to obtain

$$\boldsymbol{\partial}_g V = \mathbf{J}^{-1}\, \mathbf{DL\_H}\, \mathbf{V}^e \equiv \mathbf{d}(r,s,t)\, \mathbf{V}^e ,$$

where

$$\mathbf{d}(r,s,t) = \mathbf{J}(r,s,t)^{-1} \mathbf{DL\_H}(r,s,t) . \tag{9.31}$$

The matrix $\mathbf{d}$ is very important since it relates the global derivatives of the quantity of interest to the quantity's nodal values. Note that it depends on both the Jacobian of the geometric mapping and the local derivatives of the solution interpolation functions. For the sake of completeness, note that $\mathbf{d}$ can be partitioned as

$$\mathbf{d}(r,s,t) = \begin{bmatrix} \mathbf{d}_x \\ \text{-----} \\ \mathbf{d}_y \\ \text{-----} \\ \mathbf{d}_z \end{bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial x}\mathbf{H} \\ \text{-------} \\ \dfrac{\partial}{\partial y}\mathbf{H} \\ \text{-------} \\ \dfrac{\partial}{\partial z}\mathbf{H} \end{bmatrix} = \boldsymbol{\partial}_g \mathbf{H} \tag{9.32}$$

so that each row represents a derivative of the solution interpolation functions with respect to a global coordinate direction. Sometimes it is desirable to compute and store the rows of $\mathbf{d}$ independently. In practice the $\mathbf{d}$ matrix usually exists only in numerical

form at selected points. Once again, it is simply a matrix product such as GLOBAL = MATMUL (AJ_INV, DL_H), where GLOBAL represents the physical derivatives of the parametric functions **H**. For the linear triangle **J**, **DL_G**, and **d** are all constant. Substituting the results from Eqs. 9.27 and 9.29 into 9.31 yields

$$\mathbf{d}^e = \frac{1}{2A^e}\begin{bmatrix} (y_2 - y_3) & (y_3 - y_1) & (y_1 - y_2) \\ (x_3 - x_2) & (x_1 - x_3) & (x_2 - x_1) \end{bmatrix}^e = \frac{1}{2A^e}\begin{bmatrix} b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}^e . \quad (9.33)$$

As expected for a linear triangle, all the terms are constant. This element is usually referred to as the Constant Strain Triangle (CST). For Poisson problems $\mathbf{B}^e = \mathbf{d}^e$.

Any finite element analysis ultimately leads to the evaluation of the integrals that define the element and/or boundary segment matrices. The element matrices, $\mathbf{S}^e$ or $\mathbf{C}^e$, are usually defined by integrals of the symbolic form

$$\mathbf{I}^e = \int \int_{\Omega^e} \int \mathbf{F}^e(x, y, z)\, dx\, dy\, dz \;=\; \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \tilde{\mathbf{F}}^e(r, s, t) \mid \mathbf{J}^e(r, s, t) \mid dr\, ds\, dt\,, \quad (9.34)$$

where $\mathbf{F}^e$ is usually the sum of products of other matrices involving the element interpolation functions, **H**, their derivatives, **d**, and problem properties. In practice, on would usually use numerical integration to obtain

$$\mathbf{I}^e = \sum_{i=1}^{n_q} W_i\, \tilde{\mathbf{F}}^e(r_i\,, s_i\,, t_i\,) \mid \mathbf{J}^e(r_i\,, s_i\,, t_i\,) \mid \quad (9.35)$$

where $\tilde{\mathbf{F}}^e$ and $|\,\mathbf{J}\,|$ are evaluated at each of the n_q integration points, and where $(r_i\,, s_i\,, t_i\,)$ and W_i denote the tabulated abscissae and weights, respectively. It should be noted that this type of coding makes repeated calls to the interpolation functions to evaluate them at the quadrature points. If the element type is constant, then the quadrature locations would not change. Thus, these computations are repetitious. Since machines have larger memories today, it would be more efficient to evaluate the interpolation functions and their local derivatives once at each quadrature point and store those data for later use. This is done by adding an additional subscript to those arrays that correspond to the quadrature point number.

9.5 Hierarchical Interpolation

In Sec. 4.6 we introduced the typical hierarchical functions on line elements and let the mid-point tangential derivatives from order m to order n be denoted by $m \rightarrow n$. The exact same functions can be utilized on each edge of a two-dimensional or three-dimensional hierarchical element. We will begin by considering quadrilateral elements, or the quadrilateral faces of a solid element. To apply the previous one-dimensional element to each edge of the element requires an arbitrary choice of which way(s) we consider to the positive tangential direction. Our choice is to use the "right hand rule" so that the tangential derivatives are taken counterclockwise around the element. In other words, if we circle the fingers of our right hand in the direction of the tangential circuit, our thumb points in the direction of the outward normal vector perpendicular to that face.

Usually a (sub-parametric) four node element will be used to describe the geometry of the element. The element starts with the standard isoparametric form of four nodal

values to begin the hierarchical approximation of the function. As needed, tangential derivatives of the unknown solution are added as additional degrees of freedom. It is well known that it is desirable to have complete polynomials included in the interpolation polynomials. Thus, at some point it becomes necessary to add internal (bubble) functions at the centroid of the element. There is more than one way to go about doing this. The main question is does one want to use the function value at the centroid as a dof or just its higher derivatives? The latter is simpler to automate if we use the Q4 element.

Since the hierarchical derivative interpolation functions are all zero at both ends of their edge they will also be zero on their two adjoining edges of the quadrilateral. Thus, to use these functions on the interior of the Q4 element we must multiply them by a function that is unity on the edge where the hierarchical functions are defined and zero on the opposite parallel edge. From the discussion of isoparametric elements it should be clear that on each of the four sides (see Table 9.1) the necessary functions (in natural coordinates a, b) are

$$\begin{aligned} N^{(1)}(b) &= (1-b)/2\,, \qquad N^{(3)}(b) = (1+b)/2 \\ N^{(2)}(a) &= (1+a)/2\,, \qquad N^{(4)}(a) = (1-a)/2 \end{aligned} \tag{9.36}$$

respectively, where $N^{(i)}$ denotes the interpolation normal to side i. If T_{ij} denotes the hierarchical tangential interpolations on side i and node j, then their net interior contributions are $H_{ij}(a,b) = N^{(i)}\,T_{ij}$. That is, the p-th degree edge interpolation enrichments of the Q4 element are

$$\begin{aligned} &\text{Side 1} \quad (b=-1) \qquad H_p^{(1)}(a,b) = \tfrac{1}{2}(1-b)\,\Psi_p(a) \\ &\text{Side 2} \quad (a=1) \qquad H_p^{(2)}(a,b) = \tfrac{1}{2}(1+a)\,\Psi_p(b) \\ &\text{Side 3} \quad (b=1) \qquad H_p^{(3)}(a,b) = \tfrac{1}{2}(1+b)\,\Psi_p(-a) \\ &\text{Side 4} \quad (a=-1) \qquad H_p^{(4)}(a,b) = \tfrac{1}{2}(1-a)\,\Psi_p(-b) \end{aligned} \tag{9.37}$$

where the $\Psi_p(a) = [\,P_p(a) - P_{p-2}(a)\,]\,2p-1\,, \quad p \ge 2$. They are normalized such that their p-th tangential derivative is unity. Note that there are $4(p-1)$ such enrichments. Likewise, there are $(p-2)(p-3)/2$ internal enrichments for $p \ge 4$. They occur at the center $(0,0)$ of the element. Their degrees of freedom are the cross-partial derivatives $\partial^{\,p-2}/\partial a^j\,\partial b^k$, for $j+k = p-2$, and $1 \le j,k \le p-3$. The general form of the internal (centroid) enrichments are a product of "bubble functions" and other functions

$$H_p^{(0)}(a,b) \;=\; (1-a^2)(1-b^2)\,P_{p-4-j}(a)\,P_j(b)\;, \qquad j = 0,\,1,\ldots,\,p-4\;, \tag{9.38}$$

where $P_j(a)$ is the Legendre polynomial of degree j given in Eq. 3.25. The number of internal degrees of freedom, n, are

p	4	5	6	7	8	9	10
n	1	2	3	4	5	6	7
Total	1	3	6	10	15	21	28

so that we see the number of internal terms corresponds to the number of coefficients in a complete polynomial of degree $(p-3)$. The n terms for degree 4 to 10 are given in Table 9.3. It can be shown that the above combinations are equivalent to a complete polynomial of degree p, plus the two monomial terms $a^p b$, $a b^p$ for $p \geq 2$. This boundary and interior enrichment of the Q4 element is shown in Fig. 9.5.2. There p denotes the order of the edge polynomial, n is the total number of degrees of freedom (interpolation functions), and c is the number of dof needed for a complete polynomial form. For a quadrilateral we note that the total number of shape functions on any side is $n = p + 1$ for $p \geq 1$, and the number of interior nodes is $n_i = (p-2)(p-3)/2$ for $p \geq 4$, and the total for the element is $n_t = (p-2)(p-3)/2 + 4p$, or simply $n_t = (p^2 + 3p + 6)/2$ for $p \geq 4$. Note that the number of dof grows rapidly and by the time $p = 9$ is reached the element has almost 15 times as many dof as it did originally.

At this point the reader should see that there is a very large number of alternate forms of this same element. Consider the case where an error estimator has predicted the need for a different polynomial order on each edge. This is called *anisotropic hierarchical p-enrichment*. For maximum value of $p = 8$ there are a total of 32 possible interpolation combinations, including the six uniform ones shown in Fig. 9.5.2. It is likely that future codes will take advantage of anisotropic enrichment, although very few do so today. If one is going to use a nine node quadrilateral (Q9) to describe the geometry then the same types of enrichments can be added to it. However, the Q4 form would have better orthogonality behavior, that is, it would produce square matrices that are more diagonally dominant. For triangular and tetrahedral elements one could generate different interpolation orders on each edge, and in the interior, by utilizing the enhancement procedures for Lagrangian elements to be described later. This is probably easier to do in baracentric coordinates.

Since these elements have so much potential power they tend to be relatively large in size, and/or distorted in shape, and small in number. That trend might begin to conflict with the major appeal of finite elements: the ability to match complicated shapes. Thus, the choice of describing the geometry (and it's Jacobian) by isoparametric, or sub-parametric methods might be dropped in favor of other geometric modeling methods. That is, the user may want to exactly match an ellipse or circle rather than approximate it with a parametric curve. One way to do that is to employ *blending functions* such as Coon's functions to describe the geometry. To do this we use local analytical functions to describe each physical coordinate on the edge of the element rather than 2, 3, or 4 discrete point values as we did with isoparametric elements in the previous sections. Let (a, b) denote the quadrilateral's natural coordinates, $-1 \leq (a, b) \leq 1$. Consider only the x physical coordinate of any point in the element. Let the four corner values of x be denoted by X_i. Number the sides in a CCW manner also starting from the first (LLH)

Table 9.3. Quadrilateral hierarchical internal functions

$$\Psi_p(a,b) = (1-a^2)(1-b^2)\, P_m(a)\, P_n(b), \quad p \geq 3$$

p	m	n	j	k
4	0	0	1	1
5	1	0	1	2
	0	1	2	1
6	2	0	1	3
	1	1	2	2
	0	2	3	1
7	3	0	1	4
	2	1	2	3
	1	2	3	2
	0	3	4	1
8	4	0	1	5
	3	1	2	4
	2	2	3	3
	1	3	4	2
	0	4	5	1
9	5	0	1	6
	4	1	2	5
	3	2	3	4
	2	3	4	3
	1	4	5	2
	0	5	6	1
10	6	0	1	7
	5	1	2	6
	4	2	3	5
	3	3	4	4
	2	4	5	3
	1	5	6	2
	0	6	7	1

P_i = Legendre polynomial of degree i; $dof = \partial^{j+k} / \partial a^j \, \partial b^k$

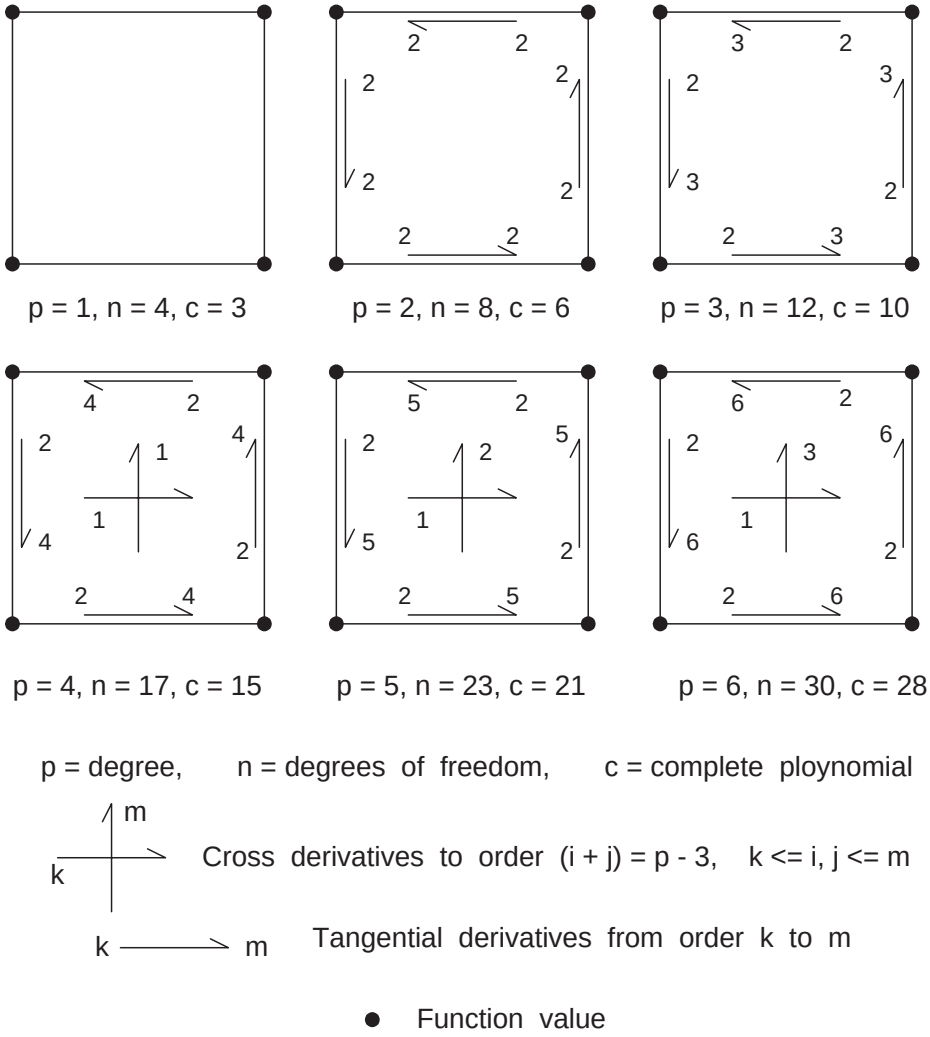


Figure 9.5.2 Hierarchical enrichments of the Q4 element

corner node. Let x_j be a function of the tangential coordinate describing x on side j. Then the *Coon's blending function* for the x-component of the geometry is:

$$x(a,b) = [\, x_1(a)(1-b) + x_2(b)(1+a) + x_3(a)(1+b) + x_4(a)(1-a)\,]/2 - \sum_{i=1}^{4} x_i\,(1+aa_i)(1+bb_i)/4 \tag{9.39}$$

where $(a_i\,, b_i\,)$ denote the local coordinates of the i-th corner. Since the term in brackets includes each corner twice (e.g., $x_1(1) = x_2(-1) = X_2$), the last summation simply subtracts off one full set of corner contributions.

The computational aspects of implementing the use of the tangential derivatives are not trivial. That is due to the fact that when mutiple elements share an edge one must decide which one is moving in "the" positive direction for that edge. One must establish some heuristic rule on how to handle the sign conflicts that can develop among different elements, or faces, on a common edge. The above suggested right hand rule means that

edges share degrees of freedom, but view them as having opposite signs. These sign conflicts must be accounted for during the element assembly process, or by invoking a different rule when assigning equation numbers so that shared dof are always viewed as having the same sign when viewed from any face or element on that edge. One could, for example, take the tangential derivative to be acting from the end with the lowest node number toward the end with the higher node number. One must plan for these difficulties before developing a hierarchical program. However, the returns on such an investment of effort is clearly worth it many times over.

9.6 Differential Geometry *

When the physical space is a higher dimension than the parametric space defining the geometry then the geometric mapping is no longer one-to-one and it is necessary to utilize the subject of *differential geometry*. This is covered in texts on vector analysis or calculus. It is also an introductory topic in most books on the mechanics of thin shell structures. Here we cover most of the basic topics except for the detailed calculation of surface curvatures. Every surface in a three-dimensional Cartesian coordinate system (x, y, z) may also be expressed by a pair of independent parametric coordinates (r, s) that lie on the surface. In our geometric parametric form, we have defined the x-coordinate as

$$x(r,s) \;=\; \mathbf{G}(r,s)\,\mathbf{x}^e\,. \tag{9.40}$$

The y- and z-coordinates are defined similarly. The components of the *position vector* to a point on the surface

$$\vec{R}(r,s) \;=\; x(r,s)\,\hat{\mathbf{i}} + y(r,s)\,\hat{\mathbf{j}} + z(r,s)\,\hat{\mathbf{k}}\,, \tag{9.41}$$

where $\hat{\mathbf{i}},\ \hat{\mathbf{j}},\ \hat{\mathbf{k}}$ are the constant unit base vectors, could be written in array form as

$$\mathbf{R}^T \;=\; [\,x \quad y \quad z\,] \;=\; \mathbf{G}(r,s)\,[\,\mathbf{x}^e\ \mathbf{y}^e\ \mathbf{z}^e\,]\,. \tag{9.42}$$

The local parameters (r, s) constitute a system of curvilinear coordinates for points on the physical surface. Equation 9.41 is called the *parametric equation* of a surface. If we eliminate the parameters (r, s) from Eq. 9.41, we obtain the familiar implicit form of the equation of a surface, $f(x, y, z) = 0$. Likewise, any relation between r and s, say $g(r, s) = 0$, represents a curve on the physical surface. In particular, if only one parameter varies while the other is constant, then the curve on the surface is called a *parametric curve*. Thus, the surface can be completely defined by a doubly infinite set of parametric curves, as shown in Fig. 9.6.1. We will often need the differential lengths, differential areas, tangent vectors, etc. We begin with differential changes in position on the surface. Since $\vec{R} = \vec{R}(r, s)$, we have

$$d\vec{R} \;=\; \frac{\partial \vec{R}}{\partial r}\,dr + \frac{\partial \vec{R}}{\partial s}\,ds \tag{9.43}$$

where $\partial\vec{R}/\partial r$ and $\partial\vec{R}/\partial s$ are the *tangent vectors* along the parametric curves. The physical distance, dl, associated with such a change in position on the surface is found from

$$(dl)^2 = dx^2 + dy^2 + dz^2 = d\vec{R}\cdot d\vec{R}\,. \tag{9.44}$$

This gives three contributions:

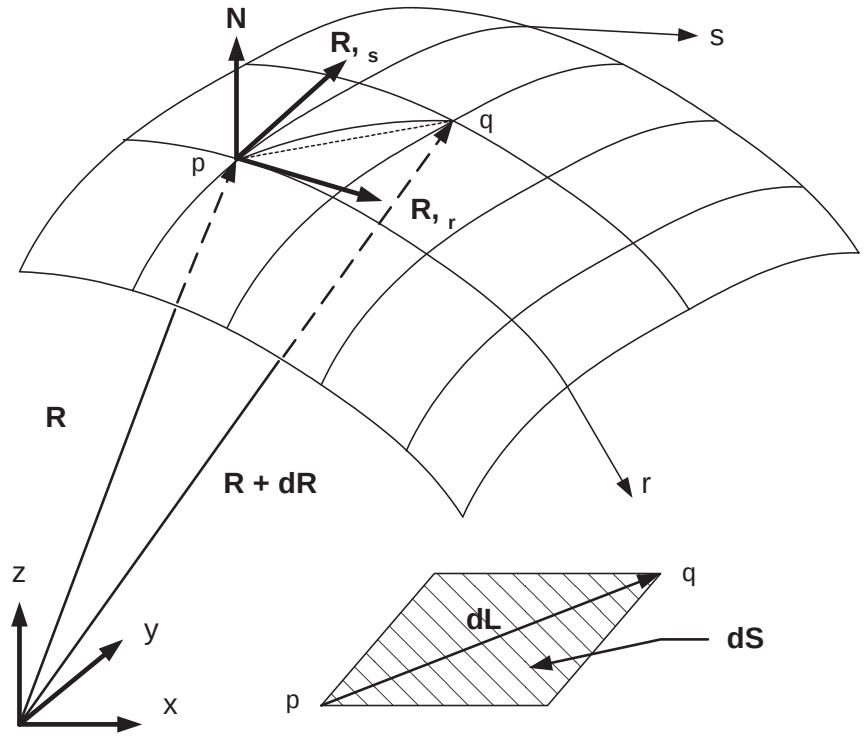


Figure 9.6.1 Parametric surface coordinates

$$(dl)^2 = \left(\frac{\partial \vec{R}}{\partial r} \cdot \frac{\partial \vec{R}}{\partial r}\right) dr^2 + 2\left(\frac{\partial \vec{R}}{\partial r} \cdot \frac{\partial \vec{R}}{\partial s}\right) dr\, ds + \left(\frac{\partial \vec{R}}{\partial s} \cdot \frac{\partial \vec{R}}{\partial s}\right) ds^2 .$$

In the common notation of differential geometry this is called the *first fundamental form* of a surface, and is usually written as

$$(dl)^2 = E\, dr^2 + 2F\, dr\, ds + G\, ds^2 \tag{9.45}$$

where

$$E = \frac{\partial \vec{R}}{\partial r} \cdot \frac{\partial \vec{R}}{\partial r}, \qquad F = \frac{\partial \vec{R}}{\partial r} \cdot \frac{\partial \vec{R}}{\partial s}, \qquad G = \frac{\partial \vec{R}}{\partial s} \cdot \frac{\partial \vec{R}}{\partial s} \tag{9.46}$$

are called the first *fundamental magnitudes* (or metric tensor) of the surface. For future reference we will use this notation to note that the magnitudes of the surface tangent vectors are

$$\left| \frac{\partial \vec{R}}{\partial r} \right| = \sqrt{E}, \qquad \left| \frac{\partial \vec{R}}{\partial s} \right| = \sqrt{G} .$$

Of course, these magnitudes can be expressed in terms of the parametric derivatives of the surface coordinates, (x, y, z). For example, from Eq. 9.46,

$$F = \frac{\partial x}{\partial r}\frac{\partial x}{\partial s} + \frac{\partial y}{\partial r}\frac{\partial y}{\partial s} + \frac{\partial z}{\partial r}\frac{\partial z}{\partial s} \tag{9.47}$$

can be evaluated for an isoparametric surface by utilizing Eq. 9.42. Define a parametric

surface gradient array given by

$$\mathbf{g} = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} & \dfrac{\partial z}{\partial r} \\ \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} & \dfrac{\partial z}{\partial s} \end{bmatrix}. \tag{9.48}$$

The rows contain the components of the tangent vectors along the parametric r and s curves, respectively. In the notation of Eq. 9.26, this becomes

$$\mathbf{g}(r,s) = [\partial_l \mathbf{R}] = \mathbf{DL\_G}\,\mathbf{R}^e = \Big[\ \partial_l \mathbf{G}(r,s)\ \Big]\Big[\ \mathbf{x}^e\ \mathbf{y}^e\ \mathbf{z}^e\ \Big]. \tag{9.49}$$

In other words, the surface gradient array at any point is the product of the parametric function derivatives evaluated at that point and the array of nodal data for the element of interest. The *metric array*, $\mathbf{m}$, is the product of the surface gradient and its transpose

$$\mathbf{m} \equiv \mathbf{g}\,\mathbf{g}^T = \begin{bmatrix} (x_{,r}^2 + y_{,r}^2 + z_{,r}^2) & (x_{,r}x_{,s} + y_{,r}y_{,s} + z_{,r}z_{,s}) \\ (x_{,r}x_{,s} + y_{,r}y_{,s} + z_{,r}z_{,s}) & (x_{,s}^2 + y_{,s}^2 + z_{,s}^2) \end{bmatrix} \tag{9.50}$$

where the subscripts denote partial derivatives with respect to the parametric coordinates. Comparing this relation with Eq. 9.46 we note that

$$\mathbf{m} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} \tag{9.51}$$

contains the fundamental magnitudes of the surface. This surface metric has a determinant that is always positive. It is denoted in differential geometry as

$$|\mathbf{m}| \equiv H^2 = EG - F^2 > 0\,. \tag{9.52}$$

We can degenerate the differential length measure in Eq. 9.44 to the common special case where we are moving along a parametric curve, that is, $dr = 0$ or $ds = 0$. In the first case of $r = constant$, we have $(dl)^2 = G\,ds^2$ where dl is a physical differential length on the surface and ds is a differential change in the parametric surface. Then $dl = \sqrt{G}\,ds$ and likewise, for the parametric curve $s = constant$, $dl = \sqrt{E}\,dr$. The quantities $\sqrt{G}$ and $\sqrt{E}$ are known as the *Lame parameters*. They convert differential changes in the parametric coordinates to differential lengths on the surface when moving on a parametric curve. From Fig. 9.6.1 we note that the vector tangent to the parametric curves r and s are $\partial\vec{R}/\partial r$ and $\partial\vec{R}/\partial s$, respectively. While the isoparametric coordinates may be orthogonal, they generally will be non-orthogonal when displayed as parametric curves on the physical surface. The angle θ between the parametric curves on the surface can be found by using these tangent vectors and the definition of the dot product. Thus, $F \equiv \partial\vec{R}/\partial r \cdot \partial\vec{R}/\partial s = \sqrt{E}\sqrt{G}\,\text{Cos}\,\theta$ and the angle at any point comes from

$$\text{Cos}\,\theta = \frac{F}{\sqrt{E}\sqrt{G}}\,. \tag{9.53}$$

Therefore, we see that the parametric curves form an orthogonal curvilinear coordinate system on the physical surface only when $F = 0$. Only in that case does Eq. 9.44 reduce to the orthogonal form $(dl)^2 = E\,dr^2 + G\,ds^2$. The calculations of the most general relations between local parametric derivatives and global derivatives are shown in

Fig. 9.6.1. Later we will utilize the function PARM_GEOM_METRIC when computing fluxes or pressures on curved surfaces or edges.

Denote the parametric curve tangent vectors as $\vec{t}_r = \partial\vec{R}/\partial r$ and $\vec{t}_s = \partial\vec{R}/\partial s$. We have seen that the differential lengths in these two directions on the surface are $\sqrt{E}\,dr$ and $\sqrt{G}\,ds$. In a vector form, those lengths are $\vec{t}_r\,dr$ and $\vec{t}_s\,ds$, and they are separated by the angle θ. The corresponding differential surface area of the surface parallelogram is

```
FUNCTION PARM_GEOM_METRIC (DL_G, GEOMETRY) RESULT (FFM_ROOT)    ! 1
! * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   ! 2
!  FUNDAMENTAL MAGNITUDE FROM PARAMETRIC TO GEOMETRIC SPACE     ! 3
! * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   ! 4
 USE Elem_Type_Data   ! for LT_GEOM, LT_PARM                    ! 5
 USE System_Constants ! for DP, N_SPACE                        ! 6
 IMPLICIT NONE                                                 ! 7
 REAL(DP), INTENT(IN) :: DL_G     (LT_PARM, LT_GEOM)           ! 8
 REAL(DP), INTENT(IN) :: GEOMETRY (LT_GEOM, N_SPACE)           ! 9
 REAL(DP) :: FFM, FFM_ROOT  ! first fundamental form data      !10
                                                               !11
!            Automatic arrays                                 !12
 REAL(DP) :: METRIC (LT_PARM, LT_PARM)                         !13
 REAL(DP) :: P_GRAD (LT_PARM, N_SPACE)  ! Tangent vectors      !14
                                                               !15
! GEOMETRY = COORDINATES OF THE ELEMENT'S GEOMETRIC NODES      !16
! DL_G     = LOCAL DERIVATIVES OF THE GEOMETRIC SHAPE FUNCTIONS !17
! FFM      = DET(A), D_PHYSICAL = FFM * D_PARAMETRIC           !18
! LT_GEOM  = NUMBER OF NODES DEFINING THE GEOMETRY             !19
! LT_PARM  = DIMENSION OF PARAMETRIC SPACE FOR ELEMENT TYPE    !20
! METRIC   = 1-ST FUNDAMENTAL MAGNITUDE (METRIC MATRIX)       !21
! P_GRAD   = PARAMETRIC DERIVATIVES OF PHYSICAL SPACE          !22
                                                               !23
!  ESTABLISH PARAMETRIC GRADIENTS                              !24
  P_GRAD = MATMUL (DL_G, GEOMETRY)  ! Tangent vectors         !25
                                                               !26
!  FORM METRIC MATRIX                                          !27
  METRIC = MATMUL (P_GRAD, TRANSPOSE (P_GRAD))                !28
                                                               !29
!  COMPUTE DETERMINANT OF METRIC MATRIX                        !30
  SELECT CASE (LT_PARM) ! size of parametric space            !31
    CASE (1) ; FFM = METRIC (1, 1)                            !32
    CASE (2) ; FFM = METRIC (1, 1) * METRIC (2, 2)          & !33
                   - METRIC (1, 2) * METRIC (2, 1)            !34
    CASE (3) ; FFM = METRIC(1,1)*( METRIC(2,2)*METRIC(3,3)  & !35
                                 - METRIC(3,2)*METRIC(2,3)) & !36
                   + METRIC(1,2)*(-METRIC(2,1)*METRIC(3,3)  & !37
                                 + METRIC(3,1)*METRIC(2,3)) & !38
                   + METRIC(1,3)*( METRIC(2,1)*METRIC(3,2)  & !39
                                 - METRIC(3,1)*METRIC(2,2))   !40
    CASE DEFAULT ; STOP 'INVALID LT_PARM, P_GRAD_METRIC'      !41
  END SELECT ! LT_PARM                                        !42
  FFM_ROOT = SQRT (FFM)     ! CONVERT TO METRIC MEASURE       !43
END FUNCTION PARM_GEOM_METRIC                                 !44
```

Figure 9.7.1 Computing the general metric tensor

$$dS \;=\; (\sqrt{E}\,dr)\,(\sqrt{G}\,ds\,\text{Sin}\,\theta) \;=\; \sqrt{E}\,\sqrt{G}\,\text{Sin}\,\theta\,dr\,ds\,.$$

By substituting the relation between Cos θ and the surface metric, this simplifies to

$$dS^2 \;=\; EG\,\text{Sin}^{\,2}\,\theta\;dr^2\,ds^2 \;=\; EG\,(1 - \text{Cos}^{\,2}\,\theta)\;dr^2\,ds^2$$

$$dS^2 \;=\; (EG - F^2)\;dr^2\,ds^2\;,$$

or simply

$$dS \;=\; \sqrt{H}\,dr\,ds\,. \tag{9.54}$$

We also note that this calculation can be expressed as a vector cross product of the tangent vectors :

$$dS\,\vec{N} \;=\; \vec{t}_r \times \vec{t}_s\,dr\,ds$$

where $\vec{N}$ is a vector normal to the surface. We also note that the *normal vector* has a magnitude of

$$|\vec{N}| \;=\; |\,\vec{t}_r \times \vec{t}_s| \;=\; H\,. \tag{9.55}$$

Sometimes it is useful to note that the components of $\vec{N}$ are

$$\vec{N} \;=\; (y_{,r}\;z_{,s} - y_{,s}\;z_{,r})\;\hat{i} + (x_{,r}\;z_{,s} - x_{,s}\;z_{,r})\;\hat{j} + (x_{,r}\;y_{,s} - x_{,s}\;y_{,r})\;\hat{k}\,.$$

We often want the unit vector, $\vec{n}$, normal to the surface. It is

$$\vec{n} \;=\; \frac{\vec{N}}{H} \;=\; \frac{\vec{t}_r \times \vec{t}_s}{|\,\vec{t}_r \times \vec{t}_s|}\,. \tag{9.56}$$

9.7 Mass Properties *

Mass properties and geometric properties are often needed in a design process. These computations provide a useful check on the model, and may also lead to reducing more complicated calculations by identifying geometrically equivalent elements. To illustrate the concept consider the following area, centroid, and inertia terms for a two-dimensional general curvilinear isoparametric element:

$$A \;=\; \int_A 1^2\;da\,, \quad A\bar{x} \;=\; \int_A x1\;da\,, \quad A\bar{y} \;=\; \int_A y1\;da \tag{9.60}$$

$$I_{xx} \;=\; \int_A y^2\;da\,, \quad -\,I_{xy} \;=\; \int_A xy\;da\,, \quad I_{yy} \;=\; \int_A x^2\;da\,, \quad I_{zz} \;=\; I_{xx} + I_{yy}\,.$$

From the parallel axis theorem we know that

$$\bar{I}_{xx} \;=\; I_{xx} - \bar{y}^2\,A\,, \quad \bar{I}_{xy} \;=\; I_{xy} + \bar{x}\;\bar{y}\,A\,, \quad \bar{I}_{yy} \;=\; I_{yy} - \bar{x}^2\,A\,, \quad \bar{I}_{zz} \;=\; \bar{I}_{xx} + \bar{I}_{yy}\,.$$

The corresponding two general inertia tensor definitions are

$$I_{ij} \;=\; \int_V (\,x_k\;x_k\;\delta_{ij} - x_i\;x_j\,)\;dV\,, \quad \bar{I}_{ij} \;=\; I_{ij} - (\,\bar{x}_k\;\bar{x}_k\;\delta_{ij} - \bar{x}_i\;\bar{x}_j\,)\;V \tag{9.61}$$

where x_i are the components of the position vector of a point in volume, V and δ_{ij} is the Kronecker delta. Typically, elements that have the same area, and inertia tensor, relative to the element centroid will have the same square matrix integral if the properties do not depend of physical coordinates $(x\,,\,y)$.

We want to illustrate these calculations in a finite element context for a two-dimensional geometry. For the parametric form in local coordinates $(r,\,s)$

$$x(r,s) = \mathbf{G}(r,s)\,\mathbf{x}^e, \qquad y(r,s) = \mathbf{G}(r,s)\,\mathbf{y}^e$$
$$1 = \mathbf{G}(r,s)\,\mathbf{1} = \sum_i H_i(r,s)$$

where **1** is a vector of unity terms. Then the above measures become

$$A^e = \mathbf{1}^T \int_A^e \mathbf{G}^T\,\mathbf{G}\,d\,A\,\mathbf{1} = \mathbf{1}^T\,\mathbf{M}^e\,\mathbf{1}$$

where $\mathbf{M}^e$ is thought of as the element measure (or mass) matrix

$$A^e\,\bar{x}^e = \mathbf{1}^T\,\mathbf{M}^e\,\mathbf{x}^e, \qquad A^e\,\bar{y}^e = \mathbf{1}^T\,\mathbf{M}^e\,\mathbf{y}^e \tag{9.62}$$
$$I^e_{xx} = \mathbf{x}^{e^T}\,\mathbf{M}^e\,\mathbf{x}^e, \quad -I^e_{xy} = \mathbf{x}^{e^T}\,\mathbf{M}^e\,\mathbf{y}^e, \quad I^e_{yy} = \mathbf{y}^{e^T}\,\mathbf{M}^e\,\mathbf{y}^e.$$

The measure matrix is defined as:

$$\mathbf{M}^e = \int_A^e \mathbf{G}^T\,\mathbf{G}\,da = \int_\square \mathbf{G}^T\,\mathbf{G}\,|\mathbf{J}^e|\,d\,\square \tag{9.63}$$

where $\square$ denotes any non-dimensional parent domain (triangular or square) and $|\mathbf{J}^e|$ is the Jacobian of the transformation from $\square$ to A^e. For any straight sided triangular element it has a constant value of $|\mathbf{J}^e| = 2A^e$. Likewise, for a straight rectangular element or parallelogram element $|\mathbf{J}^e|$ is again constant. For a one-to-one geometric mapping, we always have the relation that

$$\mathbf{A}^e = \int_{A^e} d\,a = \int_\square |\mathbf{J}^e|\,d\,\square$$

so that when $\mathbf{J}^e$ is constant $A^e = |\mathbf{J}^e|\,\square_m$, and where here $\square_m$ is the measure (volume) of the non-dimensional parent domain. For example, for the unit coordinate triangle we have $\square_m = \frac{1}{2}$ so that we get $A^e = (2A^e)(\frac{1}{2})$, as expected. The calculation of the mass properties of each element and the total analysis domain is a data checking feature.

9.8 Interpolation Error

Here we will briefly outline some elementary error concepts in two-dimensions. From the Taylor expansion of a function, u, at a point (x, y) in two-dimensions:

$$u(x+h,\ y+k) = u(x,y) + \left[\ h\,\frac{\partial u}{\partial x}(x,y) + k\,\frac{\partial u}{\partial y}(x,y)\ \right]$$
$$+\ \frac{1}{2!}\left[\ h^2\,\frac{\partial^2 u}{\partial x^2} + 2hk\,\frac{\partial^2 u}{\partial x\,\partial y} + k^2\,\frac{\partial^2 u}{\partial y^2}\ \right] + \cdots \tag{9.64}$$

The objective here is to show that if the third term is neglected, then the relations for a linear interpolation triangle are obtained. That is, we will find that the third term is proportional to the error between the true solution and the interpolated solutions. Consider a linear triangle whose maximum length in the $x-$ and $y-$directions are h and k, respectively. Let the three node numbers, given in CCW order, be i, j, and m. Employ Eq. 9.64 to estimate the nodal values u_j and u_m in terms of u_i :

$$u_j = u_i + \left[x_j \frac{\partial u}{\partial x}(x_i, y_i) + y_m \frac{\partial u}{\partial y}(x_i, y_i) \right].$$

The value of $\partial u(x_i, y_i)/\partial x$ can be found by multiplying the first relation by y_m, and subtracting the product of y_i and the second relation. The result is

$$\frac{\partial u}{\partial x}(x_i, y_i) = \frac{1}{2A}\left[u_i(y_j - y_m) + u_m(y_i - y_j) + u_j(y_m - y_i) \right]$$

where A is the area of the triangle. In a similar manner, if we compute this derivative at the other two nodes, we obtain

$$\frac{\partial u}{\partial x}(x_j, y_j) = \frac{\partial u}{\partial x}(x_m, y_m) = \frac{\partial u}{\partial x}(x_i, y_i).$$

That is, $\partial u/\partial x$ is a constant in the triangle. Likewise, $\partial u/\partial y$ is a constant. We will see later that a linear interpolation triangle has constant derivatives. Thus, these common elements will represent the first two terms in Eq. 9.64. Thus, the element error is proportional to the third term:

$$E \propto \left(h^2 \frac{\partial^2 u}{\partial x^2} + 2hk \frac{\partial^2 u}{\partial x\, \partial y} + k^2 \frac{\partial^2 u}{\partial y^2} \right) \tag{9.65}$$

where u is the exact solution, and h and k measure the element size in the x and y directions.

Once again, we would find that these second derivatives are related to the strain and stress gradients. If the strains (e.g., $\varepsilon_x = \partial u/\partial x$) are constant, then the error is small or zero. Before leaving these error comments, note that Eq. 9.65 could also be expressed in terms of the ratio (k/h). This is a measure of the relative shape of the element, and it is often called the *aspect ratio.* For an equilateral element, this ratio would be near unity. However, for a long narrow triangle, it could be quite large. Generally, it is best to keep the aspect ratio near unity (say < 5).

9.9 Element Distortion *

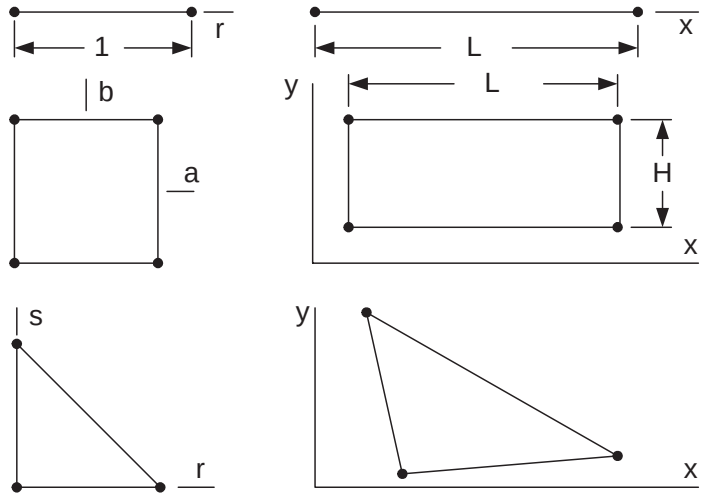
The effects of distorting various types of elements can be serious, and most codes do not adequately validate data in this respect. As an example, consider a quadratic isoparametric line element. As shown in Fig. 9.9.1, let the three nodes be located in physical (x) space at points 0, ah, and h, where h is the element length, and $0 \le a \le 1$ is a location constant. The element is defined in a local unit space where $0 \le s \le 1$. The relation between x and s is easily shown to be

$$x(s) = h(4a - 1)\, s + h(2 - 4a)\, s^2$$

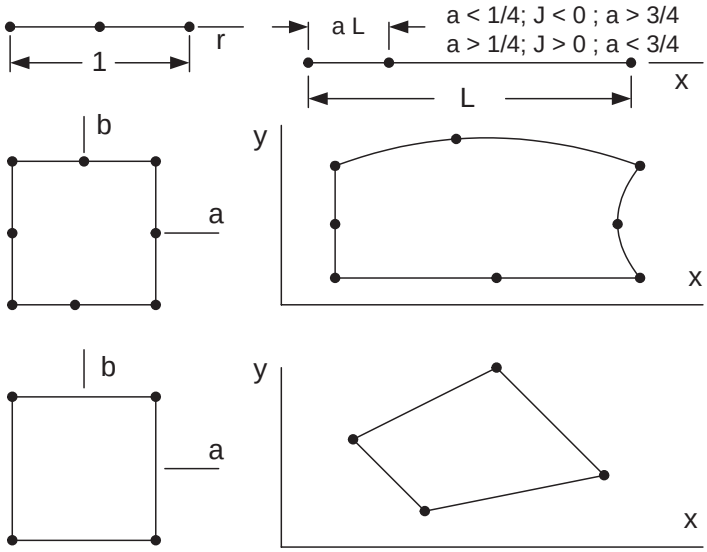
and the two coordinates have derivatives related by

$$\partial x/\partial s = h(4a - 1) + 4h(1 - 2a)\, s.$$

The Jacobian of the transformation, J, is the inverse relation; that is, $J = \partial s/\partial x$. The integrals required to evaluate the element matrices utilize this Jacobian. The mathematical principles require that J be positive definite. Distortion of the elements can cause J to go to zero or become negative. This possibility is easily seen in the present



a) Constant positive Jacobian maps



b) Variable Jacobian maps

Figure 9.9.1 Constant and variable Jacobian elements

1-D example. If one locates the interior ($s = 1/2$) node at the standard midpoint position, then $a = 1/2$ so that $\partial x / \partial s = h$ and J is constant throughout the element. Such an element is generally well formulated. However, if the interior node is distorted to any other position, the Jacobian will not be constant and the accuracy of the element may suffer. Generally, there will be points where $\partial x / \partial s$ goes to zero, so that the stiffness becomes singular due to division by zero. For slightly distorted elements, say $0.4 < a < 0.6$, the singular points lie outside the element domain. As the distortion increases, the singularities move to the element boundary, e.g., $a = 1/4$ or $a = 3/4$. Eventually, the distortions cause singularities of J inside the element. Such situations can

cause poor stiffness matrices and very bad stress estimates, unless the true solution has the same singularity, as they do in linear fracture mechanics. In that special case these distorted elements are known as the quarter point element.

The effects of distortions of two- or three-dimensional elements are similar. For example, the edge of a quadratic element may have the non-corner node displaced in a similar way, or it may be moved normal to the line between the corners. Similar analytic singularities can be developed for such elements. However, the presence of singularities due to element distortions can easily be checked by numerical experiments. Several such analytic and numerical studies have led to useful criteria for checking the element geometry for undesirable effects. For example, envision a typical two- or three-dimensional quadratic element with a curved edge. Let L be the cord length of that edge, D the normal displacement of the mid-side node on that edge, and α the angle between the corner tangent and the cord line. Suggested ranges for linear elliptical problems are:

$$\begin{aligned} &\text{warning range :} \quad 1/7 < D/L < 1/3, \qquad \alpha \le 30° \\ &\text{error range :} \quad\ 1/3 < D/L, \qquad\qquad\ \alpha \ge 53°. \end{aligned}$$

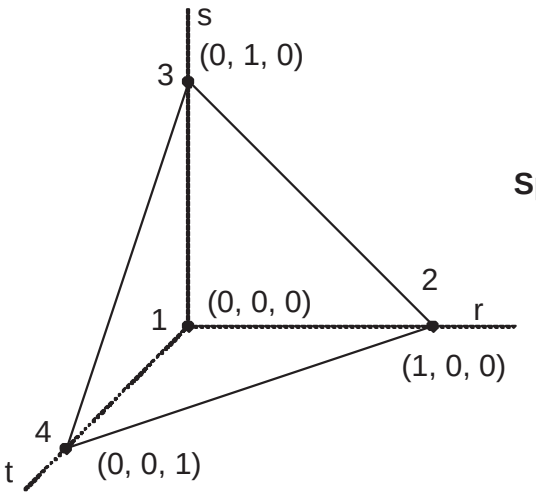
These values are obtained when only one edge is considered. If more than one edge of a single element causes a warning state, then the warnings should be considered more serious. Other parameters influence the seriousness of element distortion. Let R be a measure of the aspect ratio, that is, R is the ratio of the longest side to the shortest side. Let the minimum and maximum angles between corner cord lines be denoted by θ and γ, respectively. Define H, the lack of flatness, to be the perpendicular distance of a fourth node from the plane of the first three divided by the maximum side length. Then the following guidelines in Table 9.4 should be considered when validating geometric data for membrane or solid elements.

9.10 Space-Time Interpolation *

In Section 3.7 we addressed some of the aspects of space-time interpolation methods. In solving time dependent problems in three-dimensional space the main

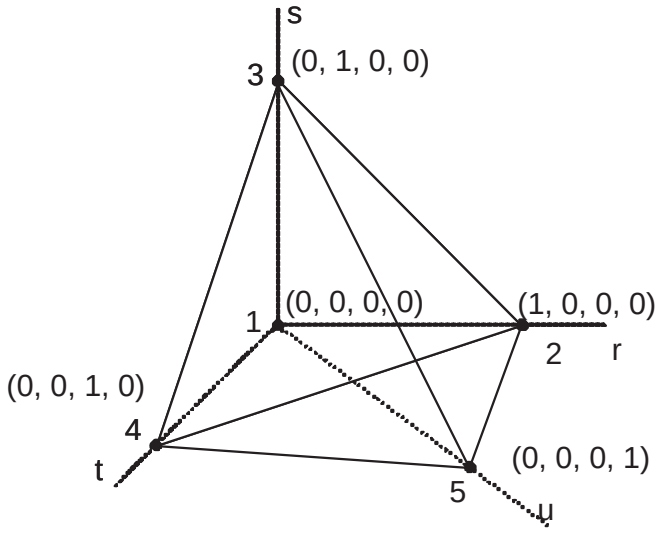
Table 9.4 Geometric criteria for two- and three-dimensional elements

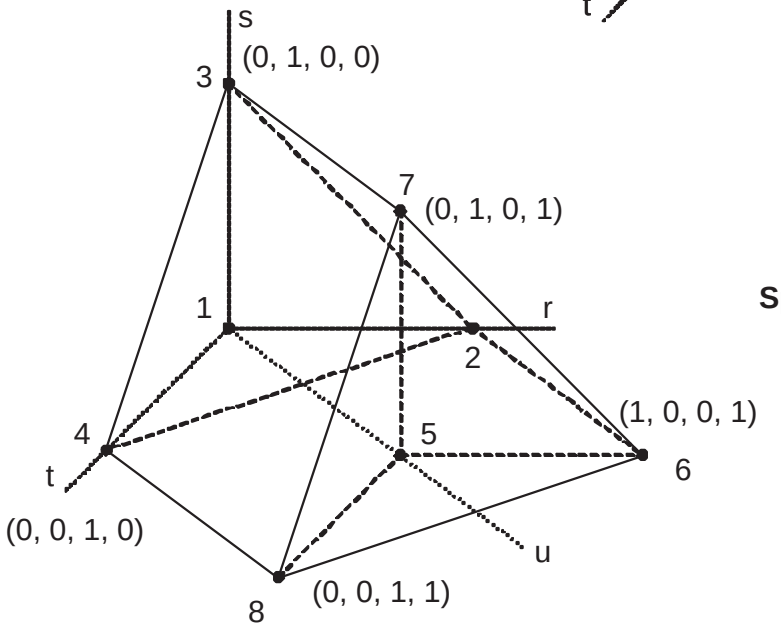
Shape	*Warning state*	*Error state*
Triangle	$5 < R < 15$	$R > 15$
	$15° < \theta < 30°$	$\theta \le 15°$
	$150° < \gamma < 165°$	$\gamma \ge 165°$
Quadrilateral	$5 < R < 15$	$R \ge 15$
	$25° < \theta < 45°$	$\theta \le 25°$
	$135° < \gamma < 155°$	$\gamma \ge 155°$
	$10^{-5} < H < 10^{-2}$	$H \ge 10^{-2}$
Solids: The above limits on R, θ and γ are checked on each face.		



Space-Time Simplex

$$H_1 = 1 - r - s - t - u$$
$$H_2 = r$$
$$H_3 = s$$
$$H_4 = t$$
$$H_5 = u$$





Space-Linear-Time Slab

Figure 9.10.1 Space-time forms for the solid simplex element

difficult is in visualization of the mesh and results. This is illustrated in Fig. 9.10.1 where the parametric solid tetraheda element has been expanded into 4-D by adding a fourth parametric coordinate of u. If one wants a fully unstructured formulation in space-time then the element becomes a 5 noded simplex. However, if we want to view it as being structured so as to simply translate through a time slab we double its number of nodes from 4 in 3-D to 8 in a 4-D time slab. Of course, we would only have to generate the 3-D mesh and define the connectivity of the first four nodes.

One important advantage of space-time interpolation is that it automatically allows for elements that must significantly change shape or spatial position with time (that is, moving meshes). There are many published results where 2-D and 3-D space elements have been extended to space-time formulations. See for example the applications by Aziz and Monk [1] Behr [4] Bonnerot and Jamet [5] Dettmer and Peric [9] Gardner, et. al [11]. Hansbo [12] Idesman, et. al [15]. and Tezduyar, et. al. [19−21] to cite a few.

One thing different about the space-time elements is in the calculation of their Jacobian matrix, which is now one dimension larger than in the pure space formulation. That is, it is a square matrix of size $(n_s + 1)$. Unlike Eq. 9.24 where we would allow the last column to compute how the physical space coordinate z varies with respect to all the element parametric coordinates, we know time will not depend on a spatial parametric coordinate. It will only depend on the non-dimensional time parametric coordinate (denoted by u in Fig. 9.10.1) and all but the last row of the right-most column of the Jacobian must be zero. Here, let τ denote time corresponding to the fourth parametric coordinate u. The generalization of the 3-D spatial Jacobian to 4-D space-time is

$$\begin{Bmatrix} \dfrac{\partial}{\partial r} \\ \dfrac{\partial}{\partial s} \\ \dfrac{\partial}{\partial t} \\ \dfrac{\partial}{\partial u} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} & \dfrac{\partial z}{\partial r} & 0 \\ \dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial s} & \dfrac{\partial z}{\partial s} & 0 \\ \dfrac{\partial x}{\partial t} & \dfrac{\partial y}{\partial t} & \dfrac{\partial z}{\partial t} & 0 \\ \dfrac{\partial x}{\partial u} & \dfrac{\partial y}{\partial u} & \dfrac{\partial z}{\partial u} & \dfrac{\partial \tau}{\partial u} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial z} \\ \dfrac{\partial}{\partial \tau} \end{Bmatrix} \tag{9.66}$$

If the spatial nodes of the domain do not change with respect to time then the non-diagonal terms on the last row of the space-time Jacobian will also be zero. Otherwise it automatically includes a moving domain formulation. For linear interpolation in time the term $\partial\tau/\partial u$ in the space-time Jacobian will be $\Delta t/1$ for the unit coordinates of Fig. 9.10.1, or $\Delta t/2$ of the natural parametric coordinate from −1 to +1 is used for u.

9.11 Exercises

1. Use the subroutines in Fig. 3.5.2 to form similar functions for a C^1 rectangular element by taking a tensor product of the one-dimensional Hermite interpolation relations. This will be a 16 degree of freedom element since each node will have $u, \partial u/\partial x, \partial u/\partial y,$ and $\partial^2 u/\partial x \partial y$ as nodal unknowns. This element will not be C^1 if

mapped to a quadrilateral shape. (Why not?)

2. Verify that for the H8 brick element in Table 9.2 that limiting its local coordinates to any one face, say $c = 1$, results in the interpolation functions not on that face becoming zero, and that the four non-zero interpolation functions on that face degenerate to those give for the Q4 quadrilateral in Table 9.1.

3. Create the local parametric derivatives ($\partial / \partial a$, etc.) of the interpolation functions for the: a) Q4 quadrilateral element of Table 9.1, b) the H8 hexahedra element of Table 9.2, c) the T6 triangular element of Eq. 9.17.

4. For a one-to-one geometric map the Jacobian matrix (of Eq. 9.26) is $\mathbf{J}^e = [\boldsymbol{\partial}_L \mathbf{H}] [\mathbf{x}^e \ \mathbf{y}^e \ \mathbf{z}^e]$. For a 2-D quadrilateral (Q4) verify that in natural coordinates this simplifies to

$$\mathbf{J}^e(a,b) = \begin{bmatrix} H_{1,a} & H_{2,a} & H_{3,a} & H_{4,a} \\ H_{1,b} & H_{2,b} & H_{3,b} & H_{4,b} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}^e$$

so that the Jacobian usually varies over the element with

$$[\boldsymbol{\partial}_L \mathbf{H}] = \frac{1}{4} \begin{bmatrix} (b-1) & (1-b) & (1+b) & (-1-b) \\ (-1-a) & (a-1) & (1+a) & (1-a) \end{bmatrix}.$$

5. Verify that if the above Q4 element maps onto a rectangle, with is sides parallel to the global axes, of length L_x and height L_y then the Jacobian is constant at all points in the element.

6. If a Q4 element is mapped to a trapezoid having the four nodal coordinates of $\mathbf{x}^{eT} = [0 \ \ 2 \ \ 2 \ \ 0]$, and $\mathbf{y}^{eT} = [0 \ \ 0 \ \ 2 \ \ 1]$ verify that its Jacobian matrix is

$$\mathbf{J}^e(a,b) = \frac{1}{4} \begin{bmatrix} 4 & (1+b) \\ 0 & (3+a) \end{bmatrix}.$$

6. Sketch how you thing an 8 noded parametric cube in 3-D parametric space would appear when extended to a time slab (with 16 nodes).

9.12 Bibliography

[1] Aziz, A.K. and Monk, P., "Continuous Finite Elements in Space and Time for the Heat Equation," *Math. Comp.*, **52**, pp. 255−274 (1989).

[2] Babuska, I., Griebel, M., and Pitkaranta, J., "The Problem of Selecting Shape Functions for a *p*-Type Finite Element," *Int. J. Num. Meth. Eng.*, **28**, pp. 1891−1908 (1989).

[3] Becker, E.B., Carey, G.F., and Oden, J.T., *Finite Elements − An Introduction*, Englewood Cliffs: Prentice-Hall (1981).

[4] Behr, M., "Stablized Space-Time Finite Element Formulations for Free-Surface Flows," *Comm. for Num. Meth. in Engr.*, **11**, pp. 813−819 (2001).

[5] Bonnerot, R. and Jamet, P., "Numerical Computation of the Free Boundary for the Two-Dimensional Stefan Problem by Space-Time Finite Elements," *J. Computational Physics*, **25**, pp. 163−181 (1977).
[6] Ciarlet, P.G., *The Finite Element Method for Elliptical Problems*, Philadelphia, PA: SIAM (2002).
[7] Connor, J.C. and Brebbia, C.A., *Finite Element Techniques for Fluid Flow*, London: Butterworth (1976).
[8] Cook, R.D., Malkus, D.S., Plesha, N.E., and Witt, R.J., *Concepts and Applications of Finite Element Analysis*, New York: John Wiley (2002).
[9] Dettmer, W. and Peric, D., "An Analysis of the Time Integration Algorithms for the Finite Element Solutions of Incompressible Navier-Stokes Equations Based on a Stabilised Formulation," *Comp. Meth. Appl. Mech. Eng.*, **192**, pp. 1177−1226 (2003).
[10] El-Zafrany, A. and Cookson, R.A., "Derivation of Lagrangian and Hermite Shape Functions for Quadrilateral Elements," *Int. J. Num. Meth. Eng.*, **23**, pp. 1939−1958 (1986).
[11] Gardner, G.A., Gardner, L.R.T., and Cunningham, J., "Simulations of a Fox-Rabies Epidemic on an Island Using Space-Time Finite Elements," *Z. Naturforsch*, **45c**, pp. 1230−1240 (1989).
[12] Hansbo, P., "A Crank-Nicolson Type Space-Time Finite Element Method for Computing on Moving Meshes," *J. Comp. Physics*, **159**, pp. 274−289 (2000).
[13] Hu, K-K., Swartz, S.E., and Kirmser, P.G., "One Formula Generates N−th Order Shape Functions," *J. Eng. Mech.*, **110**(4), pp. 640−647 (1984).
[14] Hughes, T.J.R., *The Finite Element Method*, Englewood Cliffs: Prentice-Hall (1987).
[15] Idesman, A., Niekamp, R., and Stein, E., "Finite Elements in Space and Time for Generalized Viscoelastic Maxwell Model," *Comp. Mech.*, (2001).
[16] Segerlind, L.J., *Applied Finite Element Analysis*, New York: John Wiley (1984).
[17] Silvester, P.P. and Ferrari, R.L., *Finite Elements for Electrical Engineers*, Cambridge: Cambridge University Press (1983).
[18] Szabo, B. and Babuska, I., *Finite Element Analysis*, New York: John Wiley (1991).
[19] Tezduyar, T.E. and Ganjoo, D.K., "Petrov-Galerkin Formulations with Weighting Functions Dependent Upon Spatial and Temporal Discretization," *Comp. Meth. Appl. Mech. Eng.*, **59**, pp. 47−71 (1986).
[20] Tezduyar, T.E., "Stabilized Finite Element Formulations for Incompressible Flow Computations," *Advances in Applied Mechanics*, **28**, pp. 1−44 (1991).
[21] Tezduyar, T.E., "Finite Element Methods for Flow Problems with Moving Boundaries and Interfaces," *Archives of Computational Methods in Engineering*, **8**, pp. 83−130 (2001).
[22] Zienkiewicz, O.C. and Morgan, K., *Finite Elements and Approximation*, Chichester: John Wiley (1983).

