# 11

# The 8-Node Hexahedron

# TABLE OF CONTENTS

## §11.1. Introduction

Triangles in two dimensions generalize to tetrahedra in three. Quadrilaterals generalize to *hexahedra*. A hexahedron is a polyhedron with six faces, eight corners and twelve edges or sides. It is topologically equivalent to a cube. It is informally known in the finite element literature as *brick*.

Finite elements with this geometry are extensively used in modeling three-dimensional solids. Hexahedra also have been the motivating factor for the development of "Ahmad-Pawsey" thick shell elements through the use of the "degenerated solid" concept.

The construction of hexahedral shape functions and the computation of the stiffness matrix was greatly facilitated by three advances in finite element technology: natural coordinates, isoparametric mapping and numerical integration. Together these revolutionized FEM in the mid-1960's, making possible the systematic construction of 2D and 3D finite element *families*.

Three members of the isoparametric hexahedral family are shown in Figure 11.1(a–c). They have 8, 20 and 27 nodes respecively. In this Chapter we will focus on the 8-node hexahedral element, often abbreviated to `Hex8`.[1] The other two are considered in the following Chapter.

## §11.2. Hex8 Geometrical Properties

This section studies the geometric properties of the 8-node hexahedron (Hex8). Terminology for some basic graphic objects follows.

*Sides*. Also known as *edges*. Each side is defined by its two end corner nodes. They are straight line segments. The center of the segment is the *side midpoint*.

*Faces*. Each face is defined by its four corners. If the corners are coplanar, the face is a plane quadrilateral; else it is a warped quadrilateral surface. The straight lines that join opposite side midpoints are the *face medians*. Even if the faces are not planar, the medians intersect at one point called the *face center*. The *median face* is the surface halfway between two opposite faces.

*Center medians*. The line segment that joins a face center with the center of the opposite face is called a *face median*. These lines are use to define the natural coordinates in §11.2.2

### §11.2.1. Corner Numbering Rules

The eight corners are locally numbered $1, 2 \ldots 8$. The corner numbering rule is similar to that stated for the 4-node tetrahedron in §9.1.1. Again the purpose is to guarantee a positive volume (or, more precisely, a positive Jacobian determinant at every point). The transcription of those rules to the 8-node hexahedron element is as follows:

1. Pick a *starting corner*, which is numbered 1, and an *initial face* pertaining to that corner (given a corner, there are three possible faces meeting there that may be selected).
2. Number the other 3 corners as 2,3,4 traversing the chosen face edges counterclockwise[2] when one looks at the initial face from the opposite one.
3. Number the corners of the opposite face directly opposite to 1,2,3,4 as 5,6,7,8, respectively.

The rule is illustrated in Figure 11.1(a).

[1] Also known as the *trilinear brick* in the FEM literature, on account of the form of its shape functions.

[2] "Anticlockwise" in British.

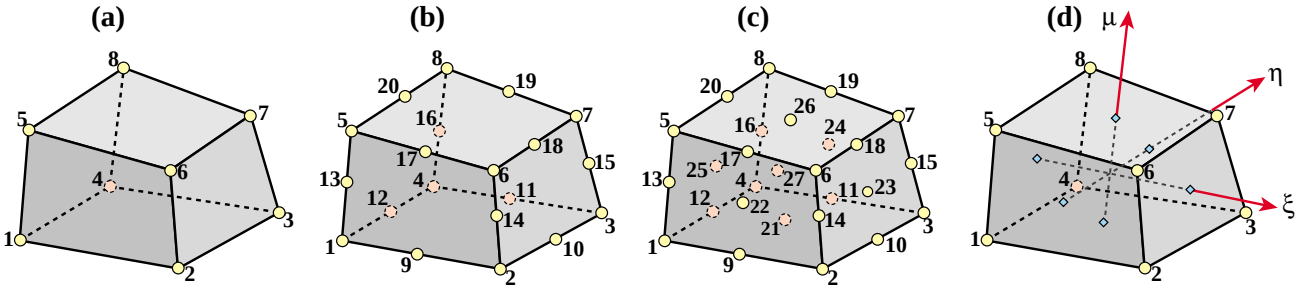


FIGURE 11.1. Sample hexahedral elements: (a) Hex8 (8 nodes, 24 DOF); (b) Hex20 (20 nodes, 60 DOF); (c) Hex27 (27 nodes, 81 DOF); (d) Hex8 showing natural coordinates $\xi$, $\eta$ and $\mu$ (also called isoparametric and hexahedral coordinates; these are unique once the corner nodes are numbered). Local node numbering shown. Faces are pictured as plane for visualization convenience. Blue squares in (d) mark face centers.

### §11.2.2. Natural Coordinates

The *natural coordinates* for the Hex8 element are called $\xi$, $\eta$ and $\mu$. Other names in the literature are *isoparametric coordinates* and *hexahedral coordinates*.

These coordinates are illustrated in Figure 11.1(d). They are similar to the quadrilateral coordinates $\xi$ and $\eta$ used in IFEM [273]. They vary from $-1$ on one face to $+1$ on the opposite face, taking the value zero on the "median" face. As in the case of quadrilaterals, this particular choice of limits was made to facilitate the use of the standard Gauss integration formulas.

Here are the precise definitions:

$\xi$ goes from $-1$ at center of face 1485 to $+1$ at center of face 2376

$\eta$ goes from $-1$ at center of face 1265 to $+1$ at center of face 3487

$\mu$ goes from $-1$ at center of face 1234 to $+1$ at center of face 5678

The natural coordinates of the corners, sides and faces are provided in the following tables.

| node | $\xi$ | $\eta$ | $\mu$ | node | $\xi$ | $\eta$ | $\mu$ |
|---|---|---|---|---|---|---|---|
| 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 | $-1$ | $+1$ | $+1$ |

(11.1)

| side | equation | side | equation |
|---|---|---|---|
| 1–2 | $\eta=-1,\ \mu=-1$ | 2–3 | $\xi=+1,\ \mu=-1$ |
| 3–4 | $\eta=+1,\ \mu=-1$ | 1–4 | $\xi=-1,\ \mu=-1$ |
| 1–5 | $\xi=-1,\ \eta=-1$ | 2–6 | $\xi=+1,\ \eta=-1$ |
| 3–7 | $\xi=+1,\ \eta=+1$ | 4–8 | $\xi=-1,\ \eta=+1$ |
| 5–6 | $\eta=+1,\ \mu=+1$ | 6–7 | $\xi=+1,\ \mu=+1$ |
| 7–8 | $\eta=+1,\ \mu=+1$ | 5–8 | $\xi=-1,\ \mu=+1$ |

(11.2)

| face | equation | face | equation |
|---|---|---|---|
| 1-4-8-5 | $\xi = -1$ | 2-3-7-6 | $\xi = +1$ |
| 1-2-6-5 | $\eta = -1$ | 3-7-8-4 | $\eta = +1$ |
| 1-2-3-4 | $\mu = -1$ | 5-6-7-8 | $\mu = +1$ |

(11.3)

Figure 11.2 collects miscellaneous pictures of a Hex8 element drwan by the graphic utilities provided in Appendix G.

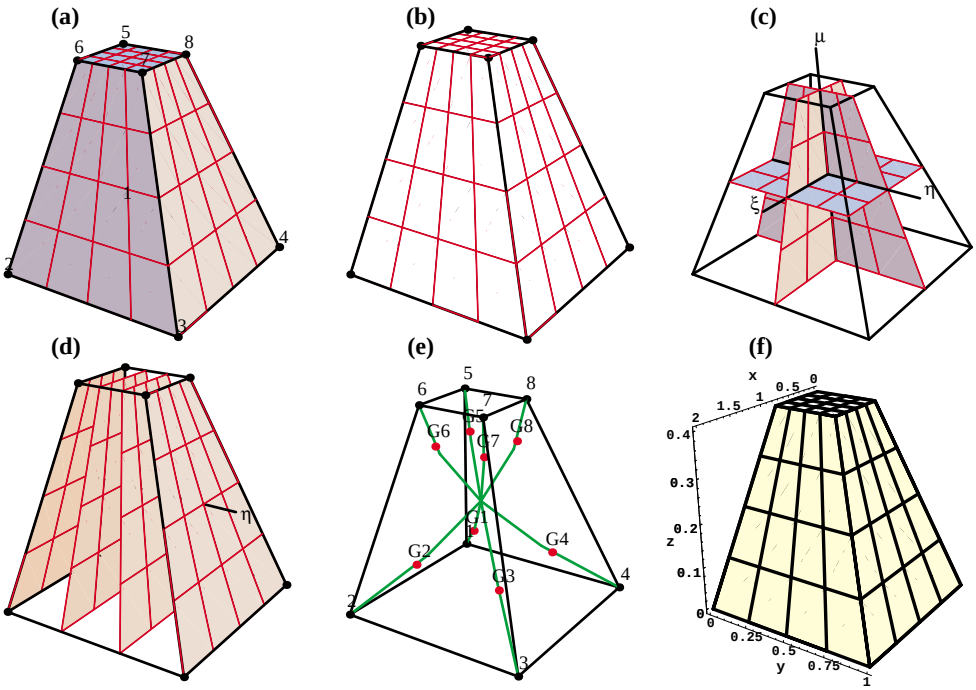


FIGURE 11.2. Sample plots of an individual Hex8 element, usineg the graphic utilities described in Appendix G: (a) shows faces, edges and labeled corners; (b) same as (a) except for white lighting set and corner node labels omitted; (c) shows median faces and labeled natural axes $\xi, \eta, \mu$ (axis labels slightly moved with Illustrator for better view); (d) shows four natural coordinate surfaces $\eta = c$, $c = -1, -1/3, 1/3, 1$ plus the labeled $\eta$ axis; (e) shows green-colored diagonals (curved because element has variable metric) and red-colored labeled sample points of the $2 \times 2 \times 2$ Gauss rule — note that Gauss point fall on the diagonals; (f) shows all lines (edges and face polygon edges) plotted in black with same ambient light set to yellow, and labeled $\{x, y, z\}$ axes drawn along plotting box edges.

## §11.3. Isoparametric 8-Node Hexahedron Formulation

The simplest solid element of the isoparametric hexahedral family is the 8-node hexahedron, often abbreviated to Hex8. It is often referred to as "brick" in the FEM literature. The following developments assume linear elasticity.

### §11.3.1. Isoparametric Hex8 Definition

As usual with isoparametric elements, Hex8 can be defined in matrix form [222] by nodal values

postmultiplied by shape functions:

$$\begin{bmatrix} 1 \\ x \\ y \\ z \\ u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ u_{x1} & u_{x2} & u_{x3} & u_{x4} & u_{x5} & u_{x6} & u_{x7} & u_{x8} \\ u_{y1} & u_{y2} & u_{y3} & u_{y4} & u_{y5} & u_{y6} & u_{y7} & u_{y8} \\ u_{z1} & u_{z2} & u_{z3} & u_{z4} & u_{z5} & u_{z6} & u_{z7} & u_{z8} \end{bmatrix} \begin{bmatrix} N_1^e \\ N_2^e \\ \\ \vdots \\ \\ N_8^e \end{bmatrix} \tag{11.4}$$

The shape functions are

$$\begin{array}{ll} N_1^e = \frac{1}{8}(1-\xi)(1-\eta)(1-\mu), & N_2^e = \frac{1}{8}(1+\xi)(1-\eta)(1-\mu), \\ N_3^e = \frac{1}{8}(1+\xi)(1+\eta)(1-\mu), & N_4^e = \frac{1}{8}(1-\xi)(1+\eta)(1-\mu), \\ N_5^e = \frac{1}{8}(1-\xi)(1-\eta)(1+\mu), & N_6^e = \frac{1}{8}(1+\xi)(1-\eta)(1+\mu), \\ N_7^e = \frac{1}{8}(1+\xi)(1+\eta)(1+\mu), & N_8^e = \frac{1}{8}(1-\xi)(1+\eta)(1+\mu) \end{array} \tag{11.5}$$

The eight formulas in (11.5) can be summarized in a single expression:

$$N_i^e = \tfrac{1}{8}\,(1+\xi\xi_i)\,(1+\eta\eta_i)\,(1+\mu\mu_i), \tag{11.6}$$

in which $\xi_i$, $\eta_i$ and $\mu_i$ are the natural coordinates of the $i^{th}$ node.

The shape functions can be easily constructed by inspection through the product technique explained in Chapter 18 of IFEM [273]. The main difference is that lines (or curves) are replaced by planes (or surfaces). For example, to find $N_1^e$ take the product of the three faces $\xi = 1$, $\eta = 1$ and $\mu = 1$ that cross all nodes except 1: $N_1^e(\xi,\eta,\mu) \stackrel{\text{guess}}{=} c_1(\xi-1)(\eta-1)(\mu-1)$. Replace the coordinates $\xi = \eta = \mu = -1$ of node 1 to get $c_1(-2)(-2)(-2) = -8\,c_1 = 1$, whence $c_1 = -1/8$, and finally reverse factor signs as convenient.

It is easily verified that the sum of the 8 shape functions is identically one, thus satisfying the completeness requirements.

### §11.3.2. Shape Function Partial Derivatives

The calculation of partial derivatives of hexahedron shape functions with respect to Cartesian coordinates follows techniques similar to that discussed for two-dimensional quadrilateral elements in IFEM. Only the size of the matrices changes changes because of the appearance of the third dimension. The chain rule provides the Cartesian derivatives as

$$\begin{aligned} \frac{\partial N_i^e}{\partial x} &= \frac{\partial N_i^e}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial N_i^e}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial N_i^e}{\partial \mu}\frac{\partial \mu}{\partial x}, \\ \frac{\partial N_i^e}{\partial y} &= \frac{\partial N_i^e}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial N_i^e}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial N_i^e}{\partial \mu}\frac{\partial \mu}{\partial y}, , \\ \frac{\partial N_i^e}{\partial z} &= \frac{\partial N_i^e}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial N_i^e}{\partial \eta}\frac{\partial \eta}{\partial z} + \frac{\partial N_i^e}{\partial \mu}\frac{\partial \mu}{\partial z}. \end{aligned} \tag{11.7}$$

Rewriting in matrix form:

$$\begin{bmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \\ \frac{\partial N_i^e}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \mu}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \mu}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \mu}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \\ \frac{\partial N_i^e}{\partial \mu} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \\ \frac{\partial N_i^e}{\partial \mu} \end{bmatrix}. \tag{11.8}$$

The $3 \times 3$ matrix that appears in (11.8) is the inverse of

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \mu} & \frac{\partial y}{\partial \mu} & \frac{\partial z}{\partial \mu} \end{bmatrix}. \tag{11.9}$$

Matrix $\mathbf{J}$ is the *Jacobian matrix* of $(x, y, z)$ with respect to $(\xi, \eta, \mu)$.

**Remark 11.1**. In the FEM literature, matrices $\mathbf{J}$ and $\mathbf{J}^{-1}$ are called simply the *Jacobian* and *inverse Jacobian*, respectively, although such a short name is sometimes ambiguous. The notation

$$\mathbf{J} = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \mu)}, \qquad \mathbf{J}^{-1} = \frac{\partial(\xi, \eta, \mu)}{\partial(x, y, z)}. \tag{11.10}$$

is standard in multivariable calculus and suggests that the Jacobian may be viewed as a generalization of the ordinary derivative, to which it reduces for a scalar function $\mathbf{x} = x(\xi)$.

### §11.3.3. Computing the Jacobian Matrix

Rows 2 through 4 of the isoparametric definition (11.4) provide the connection between Cartesian and natural coordinates: $x = x_i\, N_i^e \quad y = y_i\, N_i^e$, and $z = z_i\, N_i^e$, in which the summation convention is understood to apply over $i = 1, 2, \ldots 8$. Differentiating these relations with respect to the hexahedron coordinates we construct the matrix $\mathbf{J}$ as follows:

$$\mathbf{J} = \begin{bmatrix} x_i \frac{\partial N_i^e}{\partial \xi} & y_i \frac{\partial N_i^e}{\partial \xi} & z_i \frac{\partial N_i^e}{\partial \xi} \\ x_i \frac{\partial N_i^e}{\partial \eta} & y_i \frac{\partial N_i^e}{\partial \eta} & z_i \frac{\partial N_i^e}{\partial \eta} \\ x_i \frac{\partial N_i^e}{\partial \mu} & y_i \frac{\partial N_i^e}{\partial \mu} & z_i \frac{\partial N_i^e}{\partial \mu} \end{bmatrix}. \tag{11.11}$$

in which the summartion convention over $i = 1, \ldots 8$ is assumed. The partials that appear in (11.11) can be easily obtained since the $N_i$ are trilinear functions of $\{\xi, \eta, \mu\}$. To get its inverse efficiently, the closed form inversion formulas given in the following remark can be used.

**Remark 11.2**. The inversion formula for a matrix of order 3 is

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \qquad \mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \tag{11.12}$$

```
Hex8ShapeFunCarDer[encoor_,hcoor_,numer_]:= Module[{xn,yn,zn,
   ξ,η,μ,Nf,dNξ,dNη,dNμ,J,J11,J21,J31,J12,
   J22,J32,J13,J23,J33,Jinv,Jdet,Bx,By,Bz},
   xn=Table[encoor[[i,1]],{i,8}]; yn=Table[encoor[[i,2]],{i,8}];
   zn=Table[encoor[[i,3]],{i,8}]; {ξ,η,μ}=hcoor;
   Nf=(1/8)*{(1-ξ)*(1-η)*(1-μ),(1+ξ)*(1-η)*(1-μ),(1+ξ)*(1+η)*(1-μ),
             (1-ξ)*(1+η)*(1-μ),(1-ξ)*(1-η)*(1+μ),(1+ξ)*(1-η)*(1+μ),
             (1+ξ)*(1+η)*(1+μ),(1-ξ)*(1+η)*(1+μ)};
   {dNξ,dNη,dNμ}=(1/8)*{
     {-(1-η)*(1-μ), (1-η)*(1-μ), (1+η)*(1-μ),-(1+η)*(1-μ),
      -(1-η)*(1+μ), (1-η)*(1+μ), (1+η)*(1+μ),-(1+η)*(1+μ)},
     {-(1-ξ)*(1-μ),-(1+ξ)*(1-μ), (1+ξ)*(1-μ), (1-ξ)*(1-μ),
      -(1-ξ)*(1+μ),-(1+ξ)*(1+μ), (1+ξ)*(1+μ), (1-ξ)*(1+μ)},
     {-(1-ξ)*(1-η),-(1+ξ)*(1-η),-(1+ξ)*(1+η),-(1-ξ)*(1+η),
       (1-ξ)*(1-η), (1+ξ)*(1-η), (1+ξ)*(1+η), (1-ξ)*(1+η)}};
   J11=dNξ.xn; J21=dNη.xn; J31=dNμ.xn;
   J12=dNξ.yn; J22=dNη.yn; J32=dNμ.yn;
   J13=dNξ.zn; J23=dNη.zn; J33=dNμ.zn;
   J={{J11,J12,J13},{J21,J22,J23},{J31,J32,J33}};
   If [!numer,J=Simplify[J]];
   {{J11,J12,J13},{J21,J22,J23},{J31,J32,J33}}=J;
   Jdet= J11*J22*J33+J21*J32*J13+J31*J12*J23-
         J31*J22*J13-J11*J32*J23-J21*J12*J33;
   If [!numer,Jdet=Simplify[Jdet]];
   Jinv={{J22*J33-J32*J23,J32*J13-J12*J33,J12*J23-J22*J13},
         {J31*J23-J21*J33,J11*J33-J31*J13,J21*J13-J11*J23},
         {J21*J32-J31*J22,J31*J12-J11*J32,J11*J22-J21*J12}};
   If [!numer,Jinv=Simplify[Jinv]];
   {Bx,By,Bz}=Jinv.{dNξ,dNη,dNμ}/Jdet;
   If [!numer,{Bx,By,Bz}=Simplify[{Bx,By,Bz}]];
   Return[{Nf,Bx,By,Bz,Jdet}]];
```

FIGURE 11.3. Shape function module for Hex8.

in which

$$
\begin{aligned}
A_{11} &= a_{22}a_{33} - a_{23}a_{32}, \qquad & A_{22} &= a_{33}a_{11} - a_{31}a_{13},\\
A_{33} &= a_{11}a_{22} - a_{12}a_{21}, \qquad & A_{12} &= a_{23}a_{31} - a_{21}a_{33},\\
A_{23} &= a_{31}a_{12} - a_{32}a_{11}, \qquad & A_{31} &= a_{12}a_{23} - a_{13}a_{22},\\
A_{21} &= a_{32}a_{13} - a_{12}a_{33}, \qquad & A_{32} &= a_{13}a_{21} - a_{23}a_{11},\\
A_{13} &= a_{21}a_{22} - a_{31}a_{22}, \qquad & |\mathbf{A}| &= a_{11}A_{11} + a_{12}A_{21} + a_{13}A_{31}.
\end{aligned}
\tag{11.13}
$$

(The determinant can in fact be computed in 9 different ways.)

The foregoing computations are implemented in the modules listed below.

### §11.3.4. Shape Function Modules

The calculation of Hex8 shape functions and their Cartesian derivatines is implemented in the module listed in Figure 11.3. It is invoked as

```
{Nf,Bx,By,Bz,Jdet = HexShapeFunCarDer[encoor,hcoor,numer]
```
(11.14)

The arguments are

`encoor` Element node coordinates stored in two-dimensional list form: `{{x1,y1,z1},{x2,y2,z2}, ... {x8,y8,z8}}`.

hcoor List of natural coordinates: $\{\xi, \eta, \mu\}$ of the point at which the shape functions and derivatives are to be evaluated. May be numeric or symbolic.

numer A logical flag. Set to `True` for numeric (floating-point) computation and to `False` for exact or symbolic computation.

The function returns

Nf List of 8 shape function valuesn $N_i$

Bx List of 8 shape function $x$ derivatives $\partial N_i/\partial x$.

By List of 8 shape function $y$ derivatives $\partial N_i/\partial y$.

Bz List of 8 shape function $z$ derivatives $\partial N_i/\partial z$.

Jdet Jacobian determinant.

### §11.3.5. The Strain Displacement Matrix

Having obtained the shape function derivatives, the starin-displacement matrix **B** for Hex8 displays the usual structure for solid elements:

$$\mathbf{e} = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & 0 & 0 & \frac{\partial N_2^e}{\partial x} & 0 & 0 & \dots & \frac{\partial N_8^e}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1^e}{\partial y} & 0 & 0 & \frac{\partial N_2^e}{\partial y} & 0 & \dots & 0 & \frac{\partial N_8^e}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1^e}{\partial z} & 0 & 0 & \frac{\partial N_2^e}{\partial z} & \dots & 0 & 0 & \frac{\partial N_8^e}{\partial z} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} & 0 & \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} & 0 & \dots & \frac{\partial N_n^e}{\partial y} & \frac{\partial N_n^e}{\partial x} & 0 \\ 0 & \frac{\partial N_1^e}{\partial z} & \frac{\partial N_1^e}{\partial y} & 0 & \frac{\partial N_2^e}{\partial z} & \frac{\partial N_2^e}{\partial y} & \dots & 0 & \frac{\partial N_n^e}{\partial z} & \frac{\partial N_n^e}{\partial y} \\ \frac{\partial N_1^e}{\partial z} & 0 & \frac{\partial N_1^e}{\partial x} & \frac{\partial N_2^e}{\partial z} & 0 & \frac{\partial N_2^e}{\partial x} & \dots & \frac{\partial N_n^e}{\partial z} & 0 & \frac{\partial N_n^e}{\partial x} \end{bmatrix} \mathbf{u}^e = \mathbf{B}^e\,\mathbf{u}^e, \tag{11.15}$$

in which the strain 6-vector has the configuration

$$\mathbf{e} = [\, e_{xx} \quad e_{yy} \quad e_{zz} \quad 2e_{xy} \quad 2e_{yz} \quad 2e_{zx} \,]^T \,. \tag{11.16}$$

The rows of $\mathbf{B}^e$ can be built from the output of the shape function module as shown in §11.5.1.

## §11.4. Numerical Integration Over Hexahedra

Except for simple constant-metric (CM) geometries, such as rectangular parallelepipeds,[3] numerical integration is required for hexahedral elements of arbitrary geometry, which display variable metric (VM). All practical integration rules are of Gaussian type. These can be divided into two classes: product and non-product rules. Both are treated in more detail below.

[3] A hexahedron with rectangular faces that meet at right angles; also called rectangular cuboid, right cuboid, rectangular box, rectangular hexahedron, and right rectangular prism. Informally: a box.

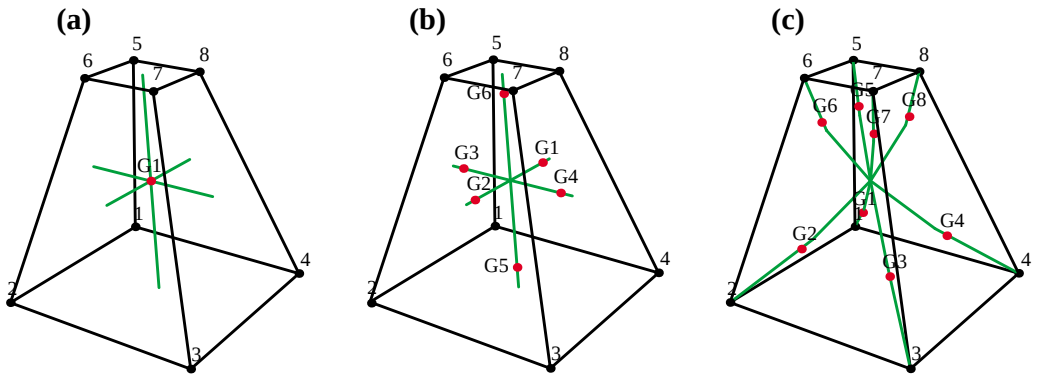


FIGURE 11.4. Graphical representation of three hehahedral Gauss rules: (a) $1 \times 1 \times 1$ product rule (sample point at element center); (b) 6-point non-product rule (sample points on face-to-face medians) ; and (c) $2 \times 2 \times 2$ product rule (sample points on diagonals).

What is the motivation behind Gauss rules? They use a *minimal number of sample points to achieve a desired level of accuracy*. This economy is important for efficient stiffness matrix calculations, since a *double matrix product*, namely $\mathbf{B}^T\,\mathbf{E}\,\mathbf{B}$, is evaluated at each point. The fact that their location is often given by non-rational numbers is of no concern in digital computation.

In what follows the terms *sample points* (used by mathematicians) and *Gauss points* (used by FEM practitioners) are used interchangeably. Figure 11.4 depicts the Gauss point location of two product rules and one non-product rule.

### §11.4.1. Hexahedral Product Gauss Rules

Product rules are the easiest to construct and implement. They are obtained as tensor product of conventional 1D Gauss rules applied in the natural coordinate directions. This is an obvious generalization of the two-dimensional case discussed in [273, Chapter 17].

The number of integration points along the $\xi$, $\eta$ and $\mu$ directions is denoted by $p_1$, $p_2$ and $p_3$, respectively. The rule is identified as $p_1 \times p_2 \times p_3$ and has $p_1\,p_2\,p_3$ points. For conventional hexahedral elements, the same rule is chosen for the three directions: $p = p_1 = p_2 = p_3$, in which case the number of Gauss points is $p^3$. This is known as an *isotropic* product rule.

The *Mathematica* module in Figure 11.5 implements hexahedral product Gauss rules with 1 through 5 points in each direction. The number of points in each direction may be the same or different. If the rule has the same number $p$ in all directions the module is called in either of two ways:

```
{{xii,etaj,muk},wijk}=HexGaussRuleInfo[{p,numer}, {i,j,k}]
{{xii,etaj},wij}=HexGaussRuleInfo[{p,numer},m]
```
(11.17)

The first form is used to get information for point $\{i,\ j,\ k\}$ of the $p \times p \times p$ rule, in which $1 \le i \le p$, $1 \le j \le p$, and $1 \le k \le p$. The second form specifies that point by a "visiting counter" $m$ that runs from 1 through $p^3$; if so $\{i,\ j,\ k\}$ are internally extracted as shown in the code.

If the integration rule has $p_1$, $p_2$, and $p_3$ points in the $\xi$, $\eta$, and $\mu$ directions, respectively, the module may be called also in two ways:

```
{{xii,etaj},wij}=HexGaussRuleInfo[{{p1,p2,p3},numer},{i,j,k}]
{{xii,etaj},wij}=HexGaussRuleInfo[{{p1,p2,p3},numer},m ]
```
(11.18)

```
HexGaussRuleInfo[{rule_,numer_},point_]:= Module[
 {ξ,η,μ,p1,p2,p3,p12,i,j,jj,k,m,w1,w2,w3,
  info={{Null,Null,Null},0}}, i=j=k=0;
  If [Length[rule]==3,  {p1,p2,p3}=rule,  p1=p2=p3=rule];
  If [Length[point]==3, {i,j,k}=point,    m=point;
     p12=p1*p2; k=Floor[(m-1)/p12]+1; jj=m-p12*(k-1);
     j=Floor[(jj-1)/p1]+1;  i=jj-p1*(j-1)];
  If [i<1||i>5||j<1||j>5||k<1||k>5, Return[info]];
  {ξ,w1}=  LineGaussRuleInfo[{p1,numer},i];
  {η,w2}=  LineGaussRuleInfo[{p2,numer},j];
  {μ,w3}=  LineGaussRuleInfo[{p3,numer},k];
  info={{ξ,η,μ},w1*w2*w3};
  If [numer, Return[N[info]], Return[Simplify[info]]]];

LineGaussRuleInfo[{rule_,numer_},point_]:= Module[
  {g2={-1,1}/Sqrt[3],w3={5/9,8/9,5/9},
   g3={-Sqrt[3/5],0,Sqrt[3/5]},
   w4={(1/2)-Sqrt[5/6]/6, (1/2)+Sqrt[5/6]/6,
       (1/2)+Sqrt[5/6]/6, (1/2)-Sqrt[5/6]/6},
   g4={-Sqrt[(3+2*Sqrt[6/5])/7],-Sqrt[(3-2*Sqrt[6/5])/7],
        Sqrt[(3-2*Sqrt[6/5])/7], Sqrt[(3+2*Sqrt[6/5])/7]},
   g5={-Sqrt[5+2*Sqrt[10/7]],-Sqrt[5-2*Sqrt[10/7]],0,
        Sqrt[5-2*Sqrt[10/7]], Sqrt[5+2*Sqrt[10/7]]}/3,
   w5={322-13*Sqrt[70],322+13*Sqrt[70],512,
       322+13*Sqrt[70],322-13*Sqrt[70]}/900,
   i=point,p=rule,info={Null,0}},
  If [p==1, info={0,2}];
  If [p==2, info={g2[[i]],1}];
  If [p==3, info={g3[[i]],w3[[i]]}];
  If [p==4, info={g4[[i]],w4[[i]]}];
  If [p==5, info={g5[[i]],w5[[i]]}];
  If [numer, Return[N[info]], Return[Simplify[info]]]];
```

FIGURE 11.5. Module `HexGaussRuleInfo` that returns information on Gauss product rules for hexahedral elements. It calls thrice the subordinate module `LineGaussRuleInfo`, which is also listed above. The number of Gauss points along the $\xi$, $\eta$ and $\mu$ directions can be 1 through 5; any combination being possible.

The meaning of the second argument is as follows. In the first form $i$ runs from 1 to $p_1$, $j$ from 1 to $p_2$ and $k$ from 1 to $p_3$. In the second form $m$ runs from 1 to $p_1\ p_2\ p_3$; if so $i$, $j$, and $k$ are extracted as shown in the code.

In all calling forms, logical flag `numer` is set to `True` if numerical information is desired and to `False` if exact information is desired.

The module returns a list configured as {{ξi,ηj,μk}, wijk}. The first sublist items are the natural coordinates of the sample point. This is followed by the weight product $w_{ijk} = w_i w_j w_k$. If the inputs are not in valid range (e.g, `rule=0`) the module returns {{`Null,Null,Null`},0}.

**Example 11.1**. The call {{xi,eta,mu},w}=HexGaussRuleInfo[{3,False},{2,3,2}] returns `xi=0`, `eta=Sqrt[3/5]`, `mu=0`, and `w=`$(5/9)\times(8/9)\times(5/9) = 200/727$.

**Example 11.2**. The variant call {{xi,eta,mu},w}=HexGaussRuleInfo[{3,True},{2,3,2}] returns (to 16-place precision) `xi=0.`, `eta=0.7745966692414834`, `mu=0.`, and `w=0.2751031636863824`.

**Remark 11.3**. Different number of integration points in each direction are used in certain hexahedral elements called Ahmad-Pawsey elements after [8,634] or, coloquially,"degenerated brick" elements. These are intended to model thick shell structures by making special behavioral assumptions along a "thickness" direction.

### §11.4.2. Selecting the Product Rule

Usually the number of integration points is taken the same in all directions: $p = p_1 = p_2 = p_3$. Exceptions arise in special circumstances such as those noted in Remark 11.3. The total number of Gauss points is then $p^3$. Each point adds at most $n_E = 6$ to the stiffness matrix rank, in which $n_E$ denotes the rank of the elasticity matrix **E**. For the 8-node hexahedron this rule gives $p \geq 2$ because $2^3 \times 6 = 48 > 24 - 6 = 18$ whereas $p = 1$ would incur a rank deficiency of $18 - 6 = 12$. For other configurations see Exercise 11.3.

### §11.4.3. *Hexahedral Non-Product Gauss Rules

Non-product Gauss rules for hexahedra include those that are not necessarily composed as the tensor product of three unidimensional rules. This generalization is occasionally convenient in three scenarios: (1) developing hexahedral elements that are not in the isoparametric class; (2) usage in modified Gauss integration techniques such as SRI, and (3) reducing the number of sample points to speed up formation of higher order hexahedra. (The last scenario is more important in 3D than 2D, since the reduction in the number of sample points is more dramatic.) For convenience, rules with up to 14 points are presented here along with a *Mathematica* implementation listed in Figure 11.6. To steer through the code embroidery it is appropriate to recall three properties that a Gauss integration rule must have to be useful in finite element work:

FS *Full Symmetry*. The same result must be obtained if the local element numbers are cyclically renumbered, which modifies the natural coordinates.[4]

POS *Positivity*. All integration weights must be positive.

INT *Interiority*. No sample point located outside the element. Preferably all points should be inside, but boundary points (on faces or at corners) can be tolerated.

Product rules automatically satisfy all three requirements.

The FS requirement leads to the notion of *stars*. Those are sample point groupings obtained by permutations and sign changes, and which have the same weight. The module of of Figure 11.6 uses four FS stars:

$S_1(w)$ 1 sample point: $\{0, 0, 0\}$, with weight $w$.

$S_6(\alpha, w)$ 6 sample points at $\{-\alpha, 0, 0\}$, $\{\alpha, 0, 0\}$, $\{0, -\alpha, 0\}$, $\{0, \alpha, 0\}$, $\{0, 0, -\alpha\}$,$\{0, 0, \alpha\}$, with weight $w$.

$S_8(\alpha, w)$ 8 sample points: $\{-\alpha, -\alpha, -\alpha\}$, $\{\alpha, -\alpha, -\alpha\}$, $\{\alpha, \alpha, -\alpha\}$, $\{-\alpha, \alpha, -\alpha\}$, $\{-\alpha, -\alpha, \alpha\}$,$\{\alpha, -\alpha, \alpha\}$, $\{\alpha, \alpha, \alpha\}$, $\{-\alpha, \alpha, \alpha\}$, with weight $w$.

$S_{12}(\alpha, w)$ 12 sample points: $\{-\alpha, -\alpha, 0\}$,$\{\alpha, -\alpha, 0\}$, $\{\alpha, \alpha, 0\}$, $\{-\alpha, \alpha, 0\}$, $\{0, -\alpha, -\alpha\}$,$\{0, \alpha, -\alpha\}$, $\{0, \alpha, \alpha\}$, $\{0, -\alpha, \alpha\}$, $\{-\alpha, 0, -\alpha\}$,$\{\alpha, 0, -\alpha\}$, $\{\alpha, 0, \alpha\}$, $\{-\alpha, 0, \alpha\}$, with weight $w$.

If a rule is made up of $n_1$, $n_6$, $n_8$ and $n_{12}$ stars of the foregoing types, respectively, the total number of Gauss sample points will be $n_G = n_1 + 6\,n_6 + 8\,n_8 + 12\,n_{12}$, in which $n_1$ can only be 0 or 1. Thus there are no FS rules with 2, 3, 4, 5, 10, or 11 points if $n_G \leq 14$. The module listed in Figure 11.6 implements 8 rules with up to 14 points. See Table 11.1 for details on star composition of implemented rules.

The module is invoked as

$$\{\{\xi\texttt{j},\eta\texttt{i},\mu\texttt{k}\},\texttt{wi}\}=\texttt{HexNPGaussRuleInfo[\{rule,numer\},point,s,w]} \tag{11.19}$$

The arguments are

rule Rule identifier; see first column of Table 11.1. Same as the number of sample points (no duplicates).

[4] Stated mathematically: the numerically evaluated integral must remain *invariant* under all affine transformations of the element domain onto itself.

```
HexNPGaussRuleInfo[{rule_,numer_},point_,s_,w_]:= Module[{i=point,
 α,β,w1,w2,ns=Length[s],nw=Length[w],info={{Null,Null,Null},0}},
 If [!MemberQ[{1,6,7,8,9,12,13,14},rule], Return[info]];
 If [i<1||i>Abs[rule], Return[info]];
 If [rule==1, info={{0,0,0},8}];
 If [rule==6, α=1; If [ns==1,{α}=s]; w1=4/3;
     info=
    {{{-α,0,0},w1},{{α,0,0}, w1},{{0,-α,0},w1},
     {{0,α,0}, w1},{{0,0,-α},w1},{{0,0,α}, w1}}[[i]]];
 If [rule==7, α=1; If [ns==1,{α}=s]; w1=4/(3*α^2); w2=8-6*w1;
     info=
    {{{-α,0,0},w1},{{α,0,0}, w1},{{0,-α,0},w1},
     {{0,α,0}, w1},{{0,0,-α},w1},{{0,0,α}, w1},{{0,0,0},w2}}[[i]]];
 If [rule==8, α=Sqrt[1/3]; If [ns==1,{α}=s]; w1=1;
     info=
    {{{-α,-α,-α},w1},{{α,-α,-α},w1},{{α,α,-α},w1},
     {{-α,α,-α}, w1},{{-α,-α,α},w1},{{α,-α,α},w1},
     {{α,α,α},   w1},   {{-α,α,α}, w1}}[[i]]];
 If [rule==9, α=Sqrt[3/5]; If [ns==1,{α}=s]; w1=1/(3*α^2);
     If [nw==1,{w1}=w]; w2=8-8*w1;
     info=
    {{{-α,-α,-α},w1},{{α,-α,-α},w1},{{α,α,-α},w1},
     {{-α,α,-α}, w1},{{-α,-α,α},w1},{{α,-α,α},w1},
     {{α,α,α},   w1},{{-α,α,α}, w1},{{0,0,0}, w2}}[[i]]];
 If [rule==12, α=Sqrt[1/2]; If [ns==1,{α}=s]; w1=2/3;
     info=
    {{{-α,-α,0},w1},{{α,-α,0},w1},{{α,α,0},w1},{{-α,α,0},w1},
     {{-α,0,-α},w1},{{α,0,-α},w1},{{α,0,α},w1},{{-α,0,α},w1},
     {{0,-α,-α},w1},{{0,α,-α},w1},{{0,α,α},w1},{{0,-α,α},w1}}[[i]]];
 If [rule==13,  α=Sqrt[3/5]; If [ns==1,{α}=s]; w1=1/(3*α^2);
     If [nw==1,{w1}=w]; w2=8-12*w1;
     info=
    {{{-α,-α,0},w1},{{α,-α,0},w1},{{α,α,0},w1},{{-α,α,0},w1},
     {{-α,0,-α},w1},{{α,0,-α},w1},{{α,0,α},w1},{{-α,0,α},w1},
     {{0,-α,-α},w1},{{0,α,-α},w1},{{0,α,α},w1},{{0,-α,α},w1},
     {{0,0,0},w2}}[[i]]];
 If [rule==14, α=Sqrt[19/33]; β=Sqrt[19/30]; If [ns==2,{α,β}=s];
     w1=(1-β^2)/(3*α^2-β^2); If [nw==1,{w1}=w]; w2=(4-4*w1)/3;
     info=
    {{{-α,-α,-α},w1},{{α,-α,-α},w1},{{α,α,-α},w1},
     {{-α,α,-α}, w1},{{-α,-α,α},w1},{{α,-α,α},w1},
     {{α,α,α},   w1},   {{-α,α,α}, w1},
     {{-β,0,0},w2},{{β,0,0}, w2},{{0,-β,0},w2},
     {{0,β,0}, w2},{{0,0,-β},w2},{{0,0,β}, w2}}[[i]]];
 If [numer, Return[N[info]], Return[Simplify[info]]]];
```

FIGURE 11.6. A *Mathematica* module that returns information on hexahedral non-product Gauss rules having 1 through 14 sample points.

numer Same as in `HexGaussRuleInfo`; see §11.4.1.

point Sample point index, in the range 1 through |`rule`|. The alternative three-index specification {`i,j,k`} of `HexGaussRuleInfo` is not available.

s A list containing abcissa{ $\alpha$ } for rules with 6 or more points. If specified as an empty list: { }, the defaults listed in Table 11.1 are used.

w A list containing { $w_1$ } for rules that use two weights: $w_1, w_2$ (rules 7, 9, 13 and 14). The nonspecified weight is internally adjusted so the weight sum for all sample points is 8. If specified as an empty list: { }, the defaults listed in Table 11.1 are used.

The module returns the natural coordinates {$\xi$`i`,$\eta$`i`} and weight `wi` for the specified sample point index.

**Table 11.1. Non-Product Gauss Rules for Hexahedra**

| Rule id | Stars | #Points | Defaults |
|---|---|---|---|
| 1 | $S_1(w_1)$ | 1 | $w_1 = 4$ |
| 6 | $S_6(\alpha, w_1)$ | 6 | $\alpha = 1$, $w_1 = 1$ |
| 7 | $S_6(\alpha, w_1) \cup S_1(w_2)$ | 7 | $\alpha = \sqrt{3/5}$, $w_1 = 20/9$, $w_2 = -16/3$ |
| 8 | $S_8(\alpha, w_1)$ | 8 | $\alpha = \sqrt{1/3}$, $w_1 = 1$ |
| 9 | $S_8(\alpha, w_1) \cup S_1(w_2)$ | 9 | $\alpha = \sqrt{1/3}$, $w_1 = 5/9$, $w_2 = 32/9$ |
| 12 | $S_{12}(\alpha, w_1)$ | 12 | $\alpha = \sqrt{1/2}$, $w_1 = 2/3$ |
| 13 | $S_{12}(\alpha, w_1) \cup S_1(w_2)$ | 13 | $\alpha = \sqrt{3/5}$, $w_1 = 5/9$, $w_2 = 4/3$ |
| 14 | $S_8(\alpha, w_1) \cup S_6(\beta, w_2)$ | 14 | $\alpha = \sqrt{19/33}$, $\beta = \sqrt{19/30}$, $w_1 = 121/361$, $w_2 = 320/361$ |

For multiple-star rules, default weights change if $\alpha$ is specified; see code in Figure 11.6.
For rule 7, the weight at $\xi = \eta = \mu = 0$ is negative if defaults are used.
Rules 1 and 8 agree with 1x1x1 and 2x2x2 product rules, respectively, if defaults are used.

## §11.5. Stiffness Matrix Evaluation

The element stiffness matrix is given by the standard formula

$$\boxed{\mathbf{K}^{(e)} = \int_{\Omega^e} \mathbf{B}^T\, \mathbf{E}\, \mathbf{B}\, d\Omega.} \tag{11.20}$$

in which $\Omega^e$ denotes the integration domain. The analytical integral (11.20) is replaced by a Gauss numerical integration formula. There are two choices: the product rules covered in §11.4.1 and the non-product rules covered in §11.4.3. The latter are employed in the special scenarios listed there.

Assuming that a product Gauss formula is used and that the stress-strain matrix $\mathbf{E}$ is constant over the element, the numerically integrated stiffness is

$$\boxed{\mathbf{K}^{(e)} = \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \sum_{k=1}^{p_3} w_{ijk}\, \mathbf{B}_{ijk}^T\, \mathbf{E}\, \mathbf{B}_{ijk}\, J_{ijk}.} \tag{11.21}$$

Here $p_1$, $p_2$ and $p_3$ denote the number of Gauss points in the $\xi$, $\eta$ and $\mu$ direction, respectively, whereas $w_{ijk}$, $\mathbf{B}_{ijk}$, and $J_{ij\ell}$ are abbreviations for

$$w_{ijk} \equiv w_i\, w_j\, w_k, \quad \mathbf{B}_{ijk} \equiv \mathbf{B}(\xi_i, \eta_j, \mu_k), \quad J_{ijk} \equiv \det \mathbf{J}(\xi_i, \eta_j, \mu_k), \tag{11.22}$$

Usually the number of integration points is taken the same in all directions: $p = p_1 = p_2 = p_3$ (an isotropic rule). If so, the total number of Gauss points is $n_G = p^3$. Each point adds at most 6 to the stiffness matrix rank. For the 8-node hexahedron, that gives the condition

$$6n_G = 6p^3 \geq 24 - 6 = 18, \quad \Rightarrow \quad n_G \geq 3. \tag{11.23}$$

```
Hex8IsoPStiffMatrix[encoor_,Emat_,options_]:= Module[{i,n,m,
  p1,p2,p3,hcoor,w,Jdet,rule=2,numer=False,Bx,By,Bz,Be,
  nopt=Length[options],Ke=Table[0,{24},{24}]},
  If [nopt>=1, numer= options[[1]]];
  If [nopt>=2, rule=  options[[2]]]; p1=p2=p3=rule;
  If [Length[rule]==3, {p1,p2,p3}=rule]; m=0;
  For [m=1, m<=p1*p2*p3, m++,
      {hcoor,w}=HexGaussRuleInfo[{{p1,p2,p3},numer},m];
      {Bx,By,Bz,Jdet}=Hex8ShapeFunCarDer[encoor,hcoor,numer];
       Be={Flatten[Table[{Bx[[n]],0,0},{n,8}]],
           Flatten[Table[{0,By[[n]],0},{n,8}]],
           Flatten[Table[{0,0,Bz[[n]]},{n,8}]],
           Flatten[Table[{By[[n]],Bx[[n]],0},{n,8}]],
           Flatten[Table[{0,Bz[[n]],By[[n]]},{n,8}]],
           Flatten[Table[{Bz[[n]],0,Bx[[n]]},{n,8}]]};
       Ke+=w*Jdet*Transpose[Be].(Emat.Be);
      ]; If [!numer, Ke=Simplify[Ke], Ke=N[Ne]];
  Return[Ke]];
```

FIGURE 11.7. *Mathematica* module that returns the stiffness matrix of an isoparametric 8-node hexahedron using a product Gauss rule.

The $1 \times 1 \times 1$ rule obviously does not meet (11.23), but the $2 \times 2 \times 2$ does. Computations indeed verify that the 8-point product rule makes $\mathbf{K}^e$ rank sufficient.

For non-product Gauss rules the three-natural-directions decomposition $\{i, j, k\}$ of product rules is no longer available, and (11.21) is replaced by a single loop:

$$\boxed{\mathbf{K}^{(e)} = \sum_{m=1}^{n_G} w_m \, \mathbf{B}_m^T \, \mathbf{E} \, \mathbf{B}_m \, J_m.} \tag{11.24}$$

in which $n_G$ is the total number of sample points, $m$ is the "visiting index", and

$$w_m \equiv w_m \quad \mathbf{B}_m \equiv \mathbf{B}(\xi_m, \eta_m, \mu_m), \quad J_m \equiv \det \mathbf{J}(\xi_m, \eta_m, \mu_m), \tag{11.25}$$

**Remark 11.4**. Because the mininum number of sample Gauss points for the Hex8 is 3, the $2 \times 2 \times 2$ product rule with 8 points seems overkill. Can a rule with less points be used?. Indeed Table 11.1 shows that there two FS rules of non-product type with 6 and 7 points. Unfortunately, both produce $\mathbf{K}^e$ matrices that are rank deficient by 3, because certain non-rigid displacement modes produce zero strains at all sample points. See Example 11.4. In this regard it should be noted that the condition (11.23) is necessary but not sufficient.

### §11.5.1. Hex8 Element Stiffness By Product Rule

The *Mathematica* module `Hex8IsoPStiff`, listed in Figure 11.7, computes and returns the element stiffness matrix of the Hex8 element using a product Gauss rule, which may have the same or different number of sample points along each natural direction. It is invoked by

```
Ke = Hex8IsoPStiffMatrix[encoor,Emat,options]
```  (11.26)

The arguments are:

| | |
|---|---|
| `encoor` | Element node coordinates stored in two-dimensional list form: `{{x1,y1,z1},{x2,y2,z2}, ... {x8,y8,z8}}`. |

`Emat` A two-dimensional list that provides the $6 \times 6$ matrix of elastic moduli shown in (8.8), stored as $\{\{E_{11}, E_{12}, \ldots E_{16}\}, \{E_{12}, E_{22}, \ldots E_{26}\}, \ldots \{E_{16}, E_{26}, \ldots E_{66}\}\}$. This matrix connects the stress 6-vector $\boldsymbol{\sigma}$ to the strain 6-vector $\mathbf{e}$ as $\boldsymbol{\sigma} = \mathbf{E}\,\mathbf{e}$, with $\boldsymbol{\sigma}$ and $\mathbf{e}$ cast as per (8.6). If the material is isotropic with elastic modulus $E$ and Poisson's ratio $\nu$, the elasticity matrix simplifies to (8.9).

`options` Processing options list. It may have the following formats: {`numer`}, {`numer,pG`}, {`numer,{p1,p2,p3}`}, or be the empty list: { }.

`numer` is a logical flag. Set to `True` to specify numerical (floating-point) computation, `False` for exact or symbolic computation. Default is `false`.

In {`numer,pG`}, `pG`: specifies an isotropic Gauss product rule having `pG` points in each natural direction. In {`numer,{p1,p2,p3}`}, list {`p1,p2,p3`} specifies an anisotropic Gauss product rule having `p1` points along $\xi$, `p2` points along $\eta$, and `p3` points along $\mu$. If the integration rule is not specified, an isotropic $2 \times 2 \times 2$ rule is assumed.

The module returns `Ke` as an $24 \times 24$ symmetric matrix pertaining to the node-by-node arrangement of nodal displacement freedoms:

$$\mathbf{u}^e = [\, u_{x1} \quad u_{y1} \quad u_{z1} \quad u_{x2} \quad \ldots u_{x8} \quad u_{y8} \quad u_{z8} \,]^T . \tag{11.27}$$

**Example 11.3**. We consider an element with cubic geometry with side lengths 2. The $\{x, y, z\}$ origin is placed at the cube center, as in Figure 11.9(a). The material is isotropic with $E = 32$ and $\nu = 1/3$. The $2 \times 2 \times 2$ product Gauss rule is used. The script of Figure 11.8 produces the element stiffness matrix

$$\begin{bmatrix}
16 & 6 & 6 & -8 & 2 & 2 & -6 & -6 & 1 & 4 & -2 & 3 & 4 & 3 & -2 & -6 & 1 & -6 & -4 & -3 & -3 & 0 & -1 & -1 \\
6 & 16 & 6 & -2 & 4 & 3 & -6 & -6 & 1 & 2 & -8 & 2 & 3 & 4 & -2 & -1 & 0 & -1 & -3 & -4 & -3 & 1 & -6 & -6 \\
6 & 6 & 16 & -2 & 3 & 4 & -1 & -1 & 0 & 3 & -2 & 4 & 2 & 2 & -8 & -6 & 1 & -6 & -3 & -3 & -4 & 1 & -6 & -6 \\
-8 & -2 & -2 & 16 & -6 & -6 & 4 & 2 & -3 & -6 & 6 & -1 & -6 & -1 & 6 & 4 & -3 & 2 & 0 & 1 & 1 & -4 & 3 & 3 \\
2 & 4 & 3 & -6 & 16 & 6 & -2 & -8 & 2 & 6 & -6 & 1 & 1 & 0 & -1 & -3 & 4 & -2 & -1 & -6 & -6 & 3 & -4 & -3 \\
2 & 3 & 4 & -6 & 6 & 16 & -3 & -2 & 4 & 1 & -1 & 0 & 6 & 1 & -6 & -2 & 2 & -8 & -1 & -6 & -6 & 3 & -3 & -4 \\
-6 & -6 & -1 & 4 & -2 & -3 & 16 & 6 & -6 & -8 & 2 & -2 & -4 & -3 & 3 & 0 & -1 & 1 & 4 & 3 & 2 & -6 & 1 & 6 \\
-6 & -6 & -1 & 2 & -8 & -2 & 6 & 16 & -6 & -2 & 4 & -3 & -3 & -4 & 3 & 1 & -6 & 6 & 3 & 4 & 2 & -1 & 0 & 1 \\
1 & 1 & 0 & -3 & 2 & 4 & -6 & -6 & 16 & 2 & -3 & 4 & 3 & 3 & -4 & -1 & 6 & -6 & -2 & -2 & -8 & 6 & -1 & -6 \\
4 & 2 & 3 & -6 & 6 & 1 & -8 & -2 & 2 & 16 & -6 & 6 & 0 & 1 & -1 & -4 & 3 & -3 & -6 & -1 & -6 & 4 & -3 & -2 \\
-2 & -8 & -2 & 6 & -6 & -1 & 2 & 4 & -3 & -6 & 16 & -6 & -1 & -6 & 6 & 3 & -4 & 3 & 1 & 0 & 1 & -3 & 4 & 2 \\
3 & 2 & 4 & -1 & 1 & 0 & -2 & -3 & 4 & 6 & -6 & 16 & 1 & 6 & -6 & -3 & 3 & -4 & -6 & -1 & -6 & 2 & -2 & -8 \\
4 & 3 & 2 & -6 & 1 & 6 & -4 & -3 & 3 & 0 & -1 & 1 & 16 & 6 & -6 & -8 & 2 & -2 & -6 & -6 & -1 & 4 & -2 & -3 \\
3 & 4 & 2 & -1 & 0 & 1 & -3 & -4 & 3 & 1 & -6 & 6 & 6 & 16 & -6 & -2 & 4 & -3 & -6 & -6 & -1 & 2 & -8 & -2 \\
-2 & -2 & -8 & 6 & -1 & -6 & 3 & 3 & -4 & -1 & 6 & -6 & -6 & -6 & 16 & 2 & -3 & 4 & 1 & 1 & 0 & -3 & 2 & 4 \\
-6 & -1 & -6 & 4 & -3 & -2 & 0 & 1 & -1 & -4 & 3 & -3 & -8 & -2 & 2 & 16 & -6 & 6 & 4 & 2 & 3 & -6 & 6 & 1 \\
1 & 0 & 1 & -3 & 4 & 2 & -1 & -6 & 6 & 3 & -4 & 3 & 2 & 4 & -3 & -6 & 16 & -6 & -2 & -8 & -2 & 6 & -6 & -1 \\
-6 & -1 & -6 & 2 & -2 & -8 & 1 & 6 & -6 & -3 & 3 & -4 & -2 & -3 & 4 & 6 & -6 & 16 & 3 & 2 & 4 & -1 & 1 & 0 \\
-4 & -3 & -3 & 0 & -1 & -1 & 4 & 3 & -2 & -6 & 1 & -6 & -6 & -6 & 1 & 4 & -2 & 3 & 16 & 6 & 6 & -8 & 2 & 2 \\
-3 & -4 & -3 & 1 & -6 & -6 & 3 & 4 & -2 & -1 & 0 & -1 & -6 & -6 & 1 & 2 & -8 & 2 & 6 & 16 & 6 & -2 & 4 & 3 \\
-3 & -3 & -4 & 1 & -6 & -6 & 2 & 2 & -8 & -6 & 1 & -6 & -1 & -1 & 0 & 3 & -2 & 4 & 6 & 6 & 16 & -2 & 3 & 4 \\
0 & 1 & 1 & -4 & 3 & 3 & -6 & -1 & 6 & 4 & -3 & 2 & 4 & 2 & -3 & -6 & 6 & -1 & -8 & -2 & -2 & 16 & -6 & -6 \\
-1 & -6 & -6 & 3 & -4 & -3 & 1 & 0 & -1 & -3 & 4 & -2 & -2 & -8 & 2 & 6 & -6 & 1 & 2 & 4 & 3 & -6 & 16 & 6 \\
-1 & -6 & -6 & 3 & -3 & -4 & 6 & 1 & -6 & -2 & 2 & -8 & -3 & -2 & 4 & 1 & -1 & 0 & 2 & 3 & 4 & -6 & 6 & 16
\end{bmatrix} \tag{11.28}$$

The eigenvalues are: 96, 28, 28, 28, 24, 24, 24, 24, 24, 16, 12, 12, 12, 8, 8, 8, 4, 4, 0, 0, 0, 0, 0, 0, which shows that $\mathbf{K}^e$ has the correct rank. Of interest for Example 11.4 are the 3 eigenvectors associated with the

```
ClearAll[Em,ν,a,b,c,pG,numer];
a=2; b=2; c=2; Em=32; ν=1/3; numer=False; pG=2;
encoor={{-a,-b,-c},{a,-b,-c},{a,b,-c},{-a,b,-c},
        {-a,-b, c},{a,-b, c},{a,b, c},{-a,b, c}}/2;
Emat=Em/((1+ν)*(1-2*ν))*{{1-ν,ν,ν,0,0,0},{ν,1-ν,ν,0,0,0},
        {ν,ν,1-ν,0,0,0},{0,0,0,1/2-ν,0,0},
        {0,0,0,0,1/2-ν,0},{0,0,0,0,0,1/2-ν}};
Ke=Hex8IsoPStiffMatrix[encoor,Emat,{numer,pG}];
Print["Ke=",Ke//MatrixForm];
Print["eigs of Ke=",Rationalize[Chop[Eigenvalues[N[Ke]]],0.0001]];
```

FIGURE 11.8. Script to do Example 11.3.

eigenvalues $\lambda_{14} = \lambda_{15} = \lambda_{16} = 8$. Their orthonormalized forms are collected in a $3 \times 24$ matrix:

$$\mathbf{V}_t = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}. \tag{11.29}$$

in which subscript $t$ stands for trapezoid. The associated displacement eigenmodes are

$$u_x = c\,\xi\eta\mu,\ u_y = u_z = 0, \quad u_y = c\,\xi\eta\mu,\ u_x = u_z = 0, \quad u_z = c\,\xi\eta\mu,\ u_x = u_y = 0. \tag{11.30}$$

in which $c = \pm 1/\sqrt{8}$. These modes are displayed in Figure 11.9(b–d). The pictures shows a square face that deforms into a trapezoid that linearly tapers toward the opposite-face reversed trapezoid.

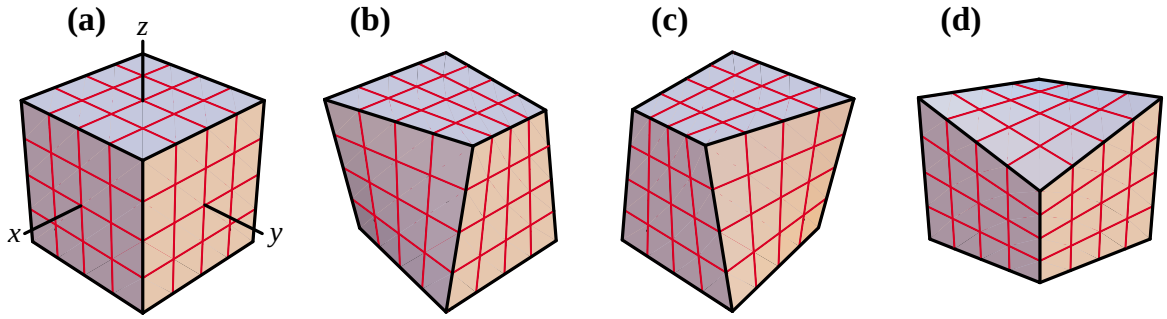


FIGURE 11.9. The "tapered trapezoid" displacement eigenmodes (11.30) associated with the eigenvectors (11.29)

### §11.5.2. *Hex8 Element Stiffness By Non-Product Rule

The module `Hex8IsoPNPStiffMatrix` listed in Figure 11.10 computes the Hex8 stiffness matrix using a non-product Gauss rule. It is invoked by

```
Ke = Hex8IsoPNPStiffMatrix[encoor,Emat,options]
```  (11.31)

Arguments `encoor` and `Emat` are identical to those for `Hex8IsoPNPStiffMatrix` described in §11.5.1, but the specification of the integration rule in the last argument is different.

options — It may have the following formats: {`numer`}, {`numer`,{`rule`,`s`,`w`}}, or be the empty list: { }. Here `numer` has the same meaning as before, but specification of the integration rule differs.

In the form {`numer`,{`rule`,`s`,`w`}}, `rule` specifies the rule identifier; see first column of Table 11.1. Item `s` is either an empty list, or a list of the form {$\alpha$} or {$\alpha$, $\beta$}, which specifies non-default sample

```
Hex8IsoPNPStiffMatrix[encoor_,Emat_,options_]:= Module[{i,n,m,
  Jdet,hcoor,wm,rule=6,numer=False,s={},w={},Bx,By,Bz,Be,
  nopt=Length[options],Ke=Table[0,{24},{24}]},
  If [nopt>=1, numer=    options[[1]]];
  If [nopt>=2,{rule,s,w}=options[[2]]];
  For [m=1, m<=Abs[rule], m++,
      {hcoor,wm}=HexNPGaussRuleInfo[{rule,numer},m,s,w];
      {Bx,By,Bz,Jdet}=Hex8ShapeFunCarDer[encoor,hcoor,numer];
       Be={Flatten[Table[{Bx[[n]],0,0},{n,8}]],
           Flatten[Table[{0,By[[n]],0},{n,8}]],
           Flatten[Table[{0,0,Bz[[n]]},{n,8}]],
           Flatten[Table[{By[[n]],Bx[[n]],0},{n,8}]],
           Flatten[Table[{0,Bz[[n]],By[[n]]},{n,8}]],
           Flatten[Table[{Bz[[n]],0,Bx[[n]]},{n,8}]]};
       Ke+=wm*Jdet*Transpose[Be].(Emat.Be);
      ]; If [!numer, Ke=Simplify[Ke], Ke=N[Ke]];
  Return[Ke]];
```

FIGURE 11.10. A *Mathematica* module that returns the stiffness matrix of an isoparametric 8-node hexahedron using a non-product rule.

point abcissa(s) as indicated in Table 11.1. To keep default abcissa(s), set `s` to the empty list. Item `w` specifies a non-default weight $w_1$ via { `w1` }, as indicated in Table 11.1. To keep default weight(s), set `w` to the empty list. The default non-product integration method is the 6-point rule.

**Example 11.4**. The foregoing example is repeated with the default 6-point non-product Gauss rule, using `Hex8IsoPNPStiffMatrix`. Two thirds of the stiffness entries remain the same while others vary by $\pm 1$. The computed stiffness matrix eigenvalues are 96, 28, 28, 28, 24, 24, 24, 24, 24, 16, 12, 12, 12, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, which shows a rank deficiency of 3. Comparing with those of the the full rank stiffness (11.28) it can be noticed that the only change is the three 8 eigenvalues have become zero, while all others are the same. It can be checked that the 6-point stiffness is related to the full rank one by

$$\mathbf{K}^e_6 = \mathbf{K}^e_{2x2x2} - 8\,\mathbf{V}_t\,\mathbf{V}_t^T. \tag{11.32}$$

where $\mathbf{V}_t$ is the 3-eigenvector matrix (11.29). The relation (11.32) shows that except for the three spurious modes, the spectrum is retained except that (11.30) become spurious (zero energy) eigenmodes. It is easily shown that all strains associated with these modes vanish on the three face-to-face medians, which is where the Gauss points of the 6-point rule are located; consequently their associated strain energy is missed.

### §11.5.3. Hex8 Element Body Force Vector

The *Mathematica* module `Hex8IsoPBodyForces` listed in Figure 11.11, computes and returns the consistent node force vector associated with a body force field given for a Hex8 element. The module is invoked as

```
fe=IsoHex8BodyForces[encoor,Emat,options,bfor]
```  (11.33)

Arguments `encoor`, `Emat`, and `options` are exactly the same as for the stiffness module described in §11.5.1. Actually `Emat` is not used (it is a dummy argument) but is kept to maintain the sequence order. The additional argument is

`bfor` Body force field. Two possibilities:

Field is constant over the element. Then `bfor` is specified as a 2-entry list: { `bx,by,bz` }, where `bx` and `by` are the body force values (force per unit of volume) along the $x$ and $y$ directions, respectively.

```
Hex8IsoPBodyForces[encoor_,Emat_,options_,bfor_]:= Module[
 {i,m,rule=2,p1,p2,p3,numer=False,hcoor,w,Nf,Bx,By,Bz,Jdet,
  nopt=Length[options],bx,by,bz,bxc=Table[0,{8}],
  byc=Table[0,{8}],bzc=Table[0,{8}],fe=Table[0,{24}]},
  If [nopt>=1, numer= options[[1]]];
  If [nopt>=2, rule=  options[[2]]]; p1=p2=p3=rule;
  If [Length[rule]==3, {p1,p2,p3}=rule]; m=0;
  If [Length[bfor]==3, bxc=Table[bfor[[1]],{8}];
      byc=Table[bfor[[2]],{8}]; bzc=Table[bfor[[3]],{8}]];
  If [Length[bfor]==8, bxc=Table[bfor[[i,1]],{i,8}];
      byc=Table[bfor[[i,2]],{i,8}]; bzc=Table[bfor[[i,3]],{i,8}]];
  For [m=1, m<=p1*p2*p3, m++,
      {hcoor,w}= HexGaussRuleInfo[{{p1,p2,p3},numer},m];
      {Nf,Bx,By,Bz,Jdet}=Hex8ShapeFunCarDer[encoor,hcoor,numer];
       bx=Nf.bxc; by=Nf.byc; bz=Nf.bzc;
       bk=Table[{Nf[[i]]*bx,Nf[[i]]*by,Nf[[i]]*bz},{i,8}];
       fe+=w*Jdet*Flatten[bk];
      ]; If [!numer, fe=Simplify[fe]];
  Return[fe]];
```

FIGURE 11.11. Module to compute node forces consistent with a given body force field for an isoparametric Hex8 element.

Field varies over the element and is specified in terms of node values. Then `bfor` is a two-dimensional list: `{{bx1,by1,bz1},{bx2,by2,bz2}, ... {bx8,by8,bz8}}`, in which `bxi`, `byi`, and `bzi` denote the body force values along the $x$, $y$ and $z$ directions, respectively, at the local node $i$. The field is interpolated to the Gauss points through the shape functions.

The module returns `fe` as an force 24-vector with components conjugate to the arrangement (11.27) of nodal displacements.

```
Hex8IsoPStresses[encoor_,Emat_,options_,udis_]:= Module[
 {i,k,numer=False,hcoor,hctab,Nf,Bx,By,Bz,Jdet,Be,
  nopt=Length[options],ue=udis,sige=Table[0,{8},{6}]},
  hctab={{-1,-1,-1},{1,-1,-1},{1,1,-1},{-1,1,-1},
         {-1,-1, 1},{1,-1, 1},{1,1, 1},{-1,1, 1}};
  If [nopt>=1, numer= options[[1]]];
  If [Length[udis]==8, ue=Flatten[udis]];
  For [k=1, k<=Length[sige], k++,
       hcoor=hctab[[k]]; If [numer, hcoor=N[hcoor]];
      {Nf,Bx,By,Bz,Jdet}=Hex8ShapeFunCarDer[encoor,hcoor,numer];
       Be={Flatten[Table[{Bx[[n]],0,0},{n,8}]],
           Flatten[Table[{0,By[[n]],0},{n,8}]],
           Flatten[Table[{0,0,Bz[[n]]},{n,8}]],
           Flatten[Table[{By[[n]],Bx[[n]],0},{n,8}]],
           Flatten[Table[{0,Bz[[n]],By[[n]]},{n,8}]],
           Flatten[Table[{Bz[[n]],0,Bx[[n]]},{n,8}]]};
       sige[[k]]=Emat.(Be.ue);
      ]; Return[sige]
   ];
```

FIGURE 11.12. Module to compute nodal stress values for an isoparametric Hex8 element, given its node displacements.

### §11.5.4. Hex8 Element Stresses

The *Mathematica* module `Hex8IsoPStresses`, listed in Figure **?**, recovers the nodal stresses for a Hex8 element in plane stress, given its nodal displacements. Initial stresses are assumed to vanish. The module is invoked as

```
Ke=Hex8IsoPStresses[encoor,Emat,options,udis]
```
(11.34)

Arguments `encoor`, `Emat` are exactly the same as for the stiffness module described in **?**.

Argumet `options` may be an empty list or contain just {`numer`}; an empty list, `numer=False` is assumed.

The additional argument is

`udis` Element node displacements. These may be provided in either of two formats:

A flat 24-vector as one-dimensional list: {`ux1,uy1,ux2, ... ux8,uy8,uz8`}

A node-by-node two-dimensional list: {{`ux1,uy1,uz1`}, ... {`ux8,uy8,uz8`}}

The module returns `sige`, which is a two-dimensional list storing computed node stresses in the node-by-node arrangement {{`sigxx1,sigyy1,sigzz1,sigxy1,sigyz1,sigzx1`}, ... {`sigxx8,sigyy8,sigzz8,sigxy8,sigyz8,sigzx8`}}.

In the module listed in Figure 11.12, the nodal stresses are evaluated directly at the node locations. The extrapolation-from-Gauss-points procedure described in Chapter 30 of IFEM is not implemented in this code.

#### Notes and Bibliography

Hexahedral elements of arbitrary geometry did not exist before the discovery of the isoparametric formulation. The 8-node hexahedron is the natural extension of the 4-node Taig quadrilateral [793] to three dimensions. While Taig can be credited with the original formulation of bilinear shape functions for quadrilaterals, the essential practical ingredient for future development was Gauss numerical integration [434]. The first systematic description of the isoparametric hexahedral family appeared in [904].

The Hex8 code presented here is the translation to *Mathematica* of a Fortran IV code writen by the author at UC Berkeley in 1967. The element was later incorporated in Boeing and Lockheed proprietary programs developed during the 1970s.

**Homework Exercises for Chapter 11**

**The 8-Node Hexahedron**

**EXERCISE 11.1** [A:20] Find the shape functions associated with the 16-node hexahedron depicted in Figure E11.1(a) for all nodes. (This kind of element is historically important as pitstop on the way to the "degenerated solid" thick-plate and thick-shell elements developed in the late 1960s; those are called Ahmad-Pawsey elements in the FEM literature, and are bread-and-butter in nonlinear commercial codes such as ABAQUS.) Verify that your shape functions satisfy two important conditions:

(1) Interelement compatibility over a typical 6-node face, say 1-2-6-5-9-13. (If used as a thick-plate or solid-shell element, those will be the faces connected to neighboring elements; $\mu$ is conventionally the plate or shell "thickness" direction.)

(2) Completeness in the sense that the sum of all shape functions must be identically one. (This must be verified algebraically, not numerically).

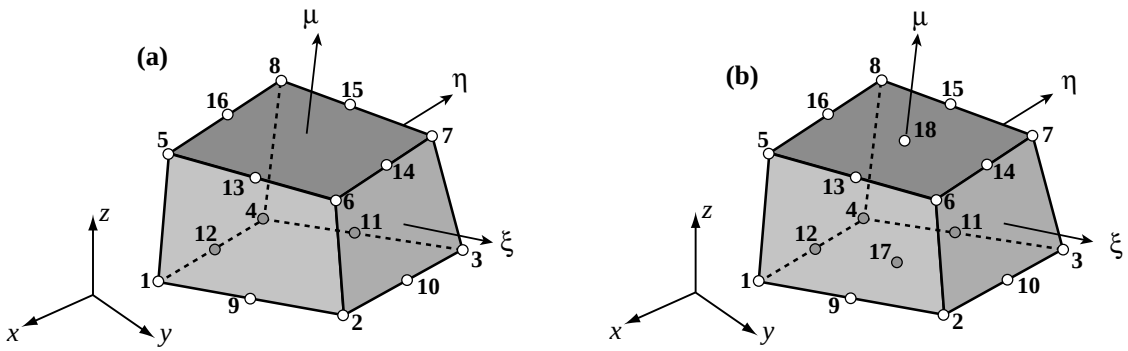


FIGURE E11.1. (a): 16-node hexahedron for Exercise 11.1; (b): 18-node hexahedron for Exercise 11.2.

**EXERCISE 11.2** [A:20] Find all shape functions associated with the 18-node hexahedron depicted in Figure E11.1(b). (As in the previous case, this configuration is used for some thick-plate and solid-shell elements discussed in the last part of the course.) Verify that your shape functions satisfy two important conditions:

(1) Interelement compatibility over a typical 6-node face, say 1-2-6-5-9-13.(If used as a thick-plate or solid-shell element, those will be the faces connected to neighboring elements; $\mu$ is conventionally the plate or shell "thickness" direction.)

(2) Completeness in the sense that the sum of all shape functions must be identically one. (This must be verified algebraically, not numerically).

**EXERCISE 11.3** [A:15] Which minimum integration rules of Gauss-product type gives a rank sufficient stiffness matrix for (a) the 20-node hexahedron, (b) the 27-node hexahedron, (c) the the 16-node hexahedron of Exercise 11.1 and (d) the 18-node hexahedron of Exercise 11.2. For the last two, would a formula containing less Gauss sample points in the $\mu$ direction (for example: $3 \times 3 \times 1$, work, at least on paper?