8

Solid Elements: Overview

TABLE OF CONTENTS

§8.1. Introduction

Solid elements are three-dimensional finite elements that can model solid bodies and structures without any *a priori* geometric simplification.

Finite element models of this type have the advantage of *directness*. Geometric, constitutive and loading assumptions required to effect dimensionality reduction, for example to planar or axisymmetric behavior, are avoided. Boundary conditions on both forces and displacements can be more realistically treated. Another attractive feature is that the finite element mesh visually looks like the physical system. See Figure 8.1.

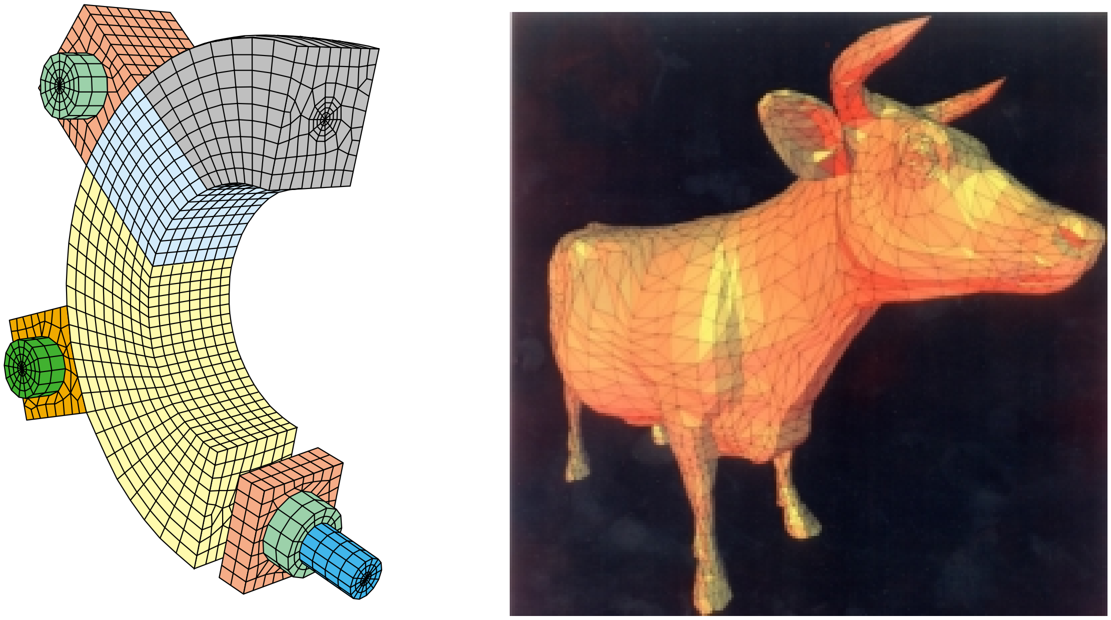

FIGURE 8.1. Use of solid elements simplifies visualization of the physical object being modeled.

This directness does not come for free. It is paid in terms of modeling, mesh preparation, computing and postprocessing effort. The rapid increase in computer time as the mesh is refined should be noted, as discussed in the "Curse of Dimensionality" section of a later Chapter. To keep these within reasonable limits it may be necessary to use coarser meshes than with two dimensional models. This in turn can degrade accuracy, negating the advantage of directness. Thus finite element users should not automatically look upon solid elements as snake oil. Several tradeoffs are at work.

In summary: use of solid elements should be restricted to problem and analysis stages, such as verification, where the generality and flexibility of full 3D models is warranted. They should be avoided during design stages. Furthermore they should also be avoided in thin-wall structures such as aerospace shells, since solid elements tend to perform poorly because of locking problems.

The present Chapter describes general attributes of solid elements for linear elastostatic problems.

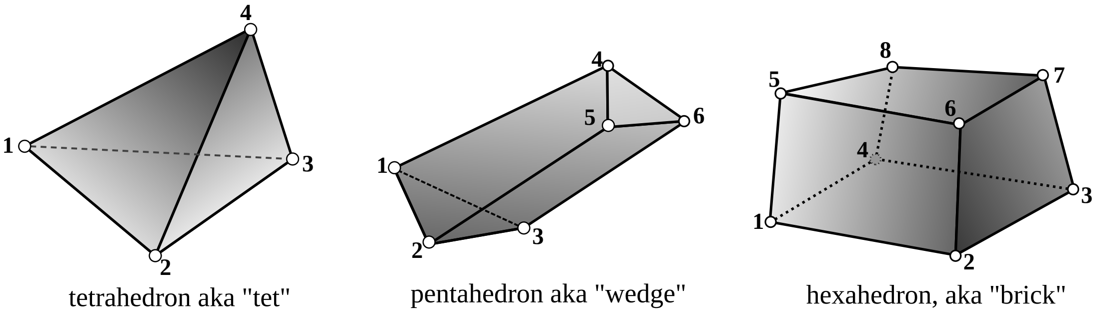

FIGURE 8.2. The three standard solid element geometries: tetrahedron (left), wedge (center) and brick (right). Only elements with corner nodes are shown.

§8.2. Geometrical Configurations

Two dimensional (2D) finite elements have two standard geometries: quadrilateral and triangle. All other geometric configurations, such as polygons with five or more sides, are classified as nonstandard or special.

Three dimensional (3D) finite elements offer more variety. There are three *standard geometries*: the tetrahedron, the wedge, and the hexahedron or “brick”. These have 4, 6 and 8 corners, respectively, with *three* faces meeting at each corner. See Figure 8.2. These elements can be used to build topologically regular meshes as illustrated in Figure 8.3.

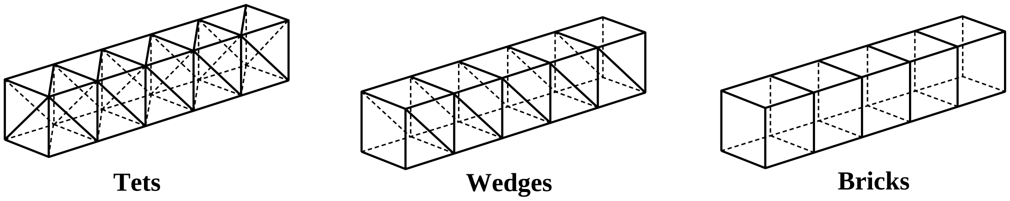

FIGURE 8.3. Regular 3D meshes can be built with cube-like repeating mesh units built with bricks, wedges or tetrahedra.

There are two *nonstandard geometries* that deserve consideration as they are ocassionally useful to complete generated 3D meshes: the pyramid and the wrick. (The latter term is a contraction of “wedge” and “brick”) These have 5 and 7 corners, respectively. See Figure 8.4. One of the corners (the last numbered one in the elements pictured in that figure) is special in that *four* faces meet, which leads to a singular metric there. This singularity disqualifies those elements for use in stress analysis in highly stressed regions. However they may be acceptable away from such regions, and in vibration analysis.

Both standard and nonstandard elements can be refined with additional nodes. For example, Figure 8.4 shows the elements of Figures 8.1 and 8.3 equipped with midside nodes. These refined elements are of interest for more accurate stress analysis. Of course, the midside nodes may be moved away from the midpoints to fit curved geometries better.

Tables 8.1 and 8.2 summarize various properties of solid elements. These are discussed at length in this and ensuing chapters.

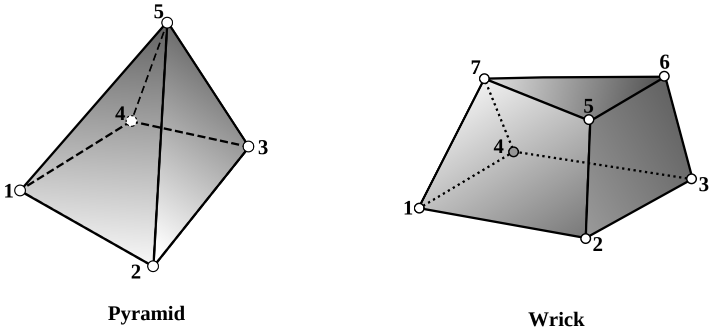

FIGURE 8.4. Two nonstandard solid element geometries: pyramid and wrick [w(edge)+(b)rick]. Four faces meet at corners 5 and 7, leading to a singular metric.

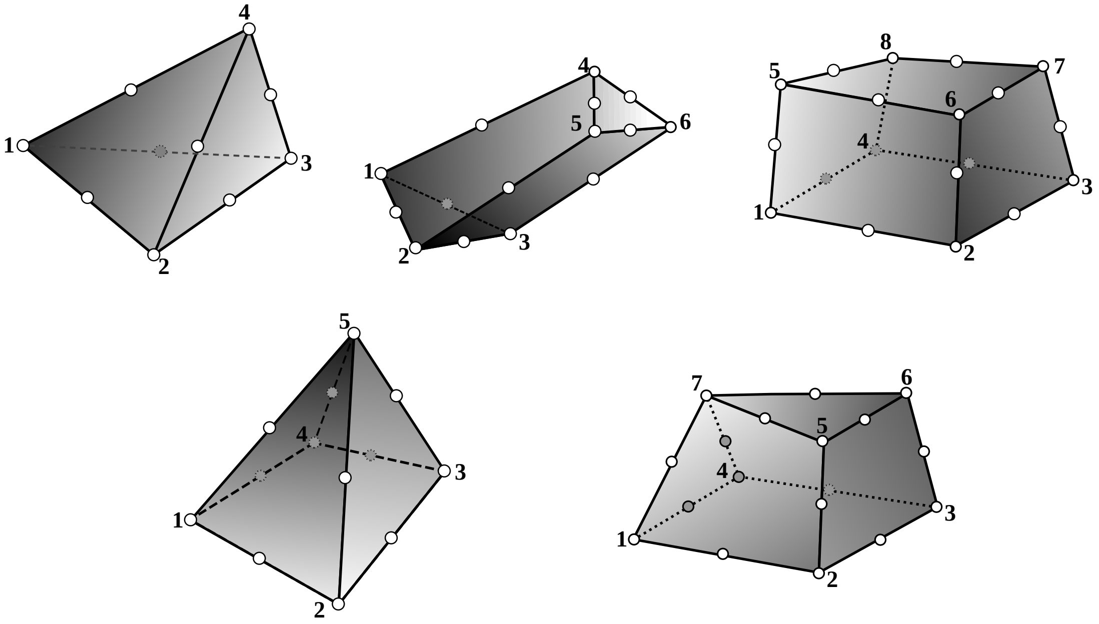

FIGURE 8.5. Solid elements refined with midside nodes. Although shown at midpoints for simplicity, in practice they can be placed away from such locations to fit curved geometries better.

§8.3. Isoparametric Definition

In this Chapter we will restrict consideration here to *isoparametric* solid elements with three translational degrees of freedom (DOF) per node. Much of the development of such elements can be carried out assuming an *arbitrary number of nodes n*. In fact a general "template module" can be written to form the element stiffness matrix and mass matrix.

Nodal quantities will be identified by the node subscript. Thus $\{x_i, y_i, z_i\}$ denote the node coordinates of the i^{th} node, while $\{u_{xi}, u_{yi}, u_{zi}\}$ are the nodal displacement DOFs. The shape function for the i^{th} node is denoted by N_i. These are expressed in term of natural coordinates which vary according to the type of element, as listed in table 8.2. Throughout this Chapter, however, the shape functions are left in generic form.

Table 8.1. Summary of Properties of Solid Elements

Region	Acronym *	C+E+F †	Global coords	Restrictions
Tetrahedron	`Tetr`	4+6+4	x, y, z	None
Wedge	`Wedg`	6+9+5	x, y, z	None
Pyramid	`Pyra`	5+8+5	x, y, z	None
Wrick	`Wric`	7+11+6	x, y, z	None
Hexahedron	`Hexa`	8+12+6	x, y, z	None

* Acronym may be followed by a node count, e.g.`Tetr10` means a 10 node tetrahedron.
† C: corners, E: edges, F: faces.

Table 8.2. Natural Coordinates and Isoparametric Geometry Definition

Region	Natural coordinates*	Range of natural coordinates	Iso-P geometry definition in terms of n geometric nodes and shape functions N_i
Tetrahedron	$\zeta_1, \zeta_2, \zeta_3, \zeta_4$	[0, 1]	$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ y_1 & y_2 & \ldots & y_n \\ z_1 & z_2 & \ldots & z_n \end{bmatrix} \begin{bmatrix} N_1 \\ \vdots \\ N_n \end{bmatrix}$
Wedge	$\zeta_1, \zeta_2, \zeta_3, \xi$	ζ_i:[0, 1], ξ:[−1, 1]	same as tetrahedron
Pyramid	ξ, η, μ	[−1, 1]	same as hexahedron
Wrick	ξ, η, μ	[−1, 1]	same as hexahedron
Hexahedron	ξ, η, μ	[−1, 1]	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \\ y_1 & y_2 & \ldots & y_n \\ z_1 & z_2 & \ldots & z_n \end{bmatrix} \begin{bmatrix} N_1 \\ \vdots \\ N_n \end{bmatrix}$

*NC constraints: $\zeta_1 + \zeta_2 + \zeta_3 = 1$ for wedges, $\zeta_1 + \zeta_2 + \zeta_3 + \zeta_4 = 1$ for tetrahedra.

§8.3.1. Geometry Description

Following the isoparametric notation introduced by Felippa and Clough[1] the element geometry is described by

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ y_1 & y_2 & \ldots & y_n \\ z_1 & z_2 & \ldots & z_n \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{bmatrix}. \tag{8.1}$$

[1] C. A. Felippa and R. W. Clough, The Finite Element Method in Solid Mechanics, in *Numerical Solution of Field Problems in Continuum Physics*, ed. by G. Birkhoff and R. S. Varga, SIAM–AMS Proceedings II, American Mathematical Society, Providence, R.I., 1969, pp. 210–252

The four rows of this matrix relation express the *geometric* conditions

$$\boxed{1 = \sum_{i=1}^{n} N_i, \qquad x = \sum_{i=1}^{n} x_i N_i, \qquad y = \sum_{i=1}^{n} y_i N_i, \qquad z = \sum_{i=1}^{n} z_i N_i,} \tag{8.2}$$

which the shape functions must identically satisfy. The first one: sum of shape functions must be unity, is an essential part of the verification of *completeness*.

§8.3.2. Displacement Interpolation

The displacement interpolation is

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} u_{x1} & u_{x2} & \dots & u_{xn} \\ u_{y1} & u_{y2} & \dots & u_{yn} \\ u_{z1} & u_{z2} & \dots & u_{zn} \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{bmatrix}. \tag{8.3}$$

The three rows of this matrix relation express the *interpolation* conditions

$$\boxed{u_x = \sum_{i=1}^{n} u_{xi} N_i, \qquad u_y = \sum_{i=1}^{n} u_{yi} N_i, \qquad u_z = \sum_{i=1}^{n} u_{zi} N_i.} \tag{8.4}$$

The identical structure of the geometry definition (8.1) and displacement interpolation (8.3) characterizes an isoparametric element (iso=same).

For future development the element node displacements of (8.3) are collected in the 3*n* column vector **u** which is configured with the 3 components grouped node by node:

$$\mathbf{u}^T = [\, u_{x1} \quad u_{y1} \quad u_{z1} \quad u_{x2} \quad \dots \quad u_{xn} \quad u_{yn} \quad u_{zn} \,] \tag{8.5}$$

§8.4. The Stiffness Matrix

For convenience in the stiffness matrix formulation the six components of the strains and stresses are grouped as follows to form into 6-component vectors:

$$\mathbf{e} = \begin{bmatrix} e_{xx} \\ e_{yy} \\ e_{zz} \\ 2e_{xy} \\ 2e_{yz} \\ 2e_{zx} \end{bmatrix}, \qquad \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix}. \tag{8.6}$$

§8.4.1. The Strain-Displacement Equations

The strains are related to the element node displacements by the $6 \times 3n$ strain-displacement matrix:

$$\mathbf{e} = \begin{bmatrix} \dfrac{\partial u_x}{\partial x} \\ \dfrac{\partial u_y}{\partial y} \\ \dfrac{\partial u_z}{\partial z} \\ \dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \\ \vdots \end{bmatrix} = \begin{bmatrix} N_{x1} & 0 & 0 & \dots & N_{xn} & 0 & 0 \\ 0 & N_{y1} & 0 & \dots & 0 & N_{yn} & 0 \\ 0 & 0 & N_{z1} & \dots & 0 & 0 & N_{zn} \\ N_{y1} & N_{x1} & 0 & \dots & N_{yn} & N_{xn} & 0 \\ 0 & N_{z1} & N_{y1} & \dots & 0 & N_{yn} & N_{xn} \\ N_{z1} & 0 & N_{x1} & \dots & N_{zn} & 0 & N_{xn} \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{z1} \\ \vdots \\ u_{xn} \\ u_{yn} \\ u_{zn} \end{bmatrix} = \mathbf{Bu}, \tag{8.7}$$

in which N_{xi} N_{yi} and N_{zi} denote the derivatives of shape function N_i with respect to x, y and z, respectively.

§8.4.2. The Constitutive Equations

We restrict attention to linear elastostatic analysis without initial stresses. For a general anisotropic material the stress-strain equations can be presented as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{16} \\ & E_{22} & E_{23} & E_{24} & E_{25} & E_{26} \\ & & E_{33} & E_{34} & E_{35} & E_{36} \\ & & & E_{44} & E_{45} & E_{46} \\ & & & & E_{55} & E_{56} \\ symm & & & & & E_{66} \end{bmatrix} \begin{bmatrix} e_{xx} \\ e_{yy} \\ e_{zz} \\ 2e_{xy} \\ 2e_{yz} \\ 2e_{zx} \end{bmatrix} = \mathbf{Ee}. \tag{8.8}$$

The stress-strain matrix **E** will be assumed to be constant over the element. This matrix can be significantly simplified in the case of isotropic or orthotropic material. However for the actual element implementation those simplifications are not considered here.

§8.4.3. The Stiffness Matrix

The element stiffness matrix is formally given by the volume integral

$$\mathbf{K}^e = \int_{V^e} \mathbf{B}^T \mathbf{E} \mathbf{B} \, dV \tag{8.9}$$

where the integral is taken over the element volume.

The usual treatment of (8.9) consists of evaluating the integral by numerical Gauss quadrature with n_G points:

$$\mathbf{K}^e = \sum_{k=1}^{n_G} w_k \, J_k \, \mathbf{B}_k^T \, \mathbf{E} \, \mathbf{B}_k \tag{8.10}$$

Here k is the integration point index; w_k is the integration weight, $\mathbf{B}_k$ is the stress-displacement matrix and J_k is the Jacobian determinant introduced later, respectively, evaluated at the integration point. Three dimensional integration rules are discussed in a later chapter.

Remark 8.1. The fast computation of the matrix product $\mathbf{B}^T\mathbf{E}\mathbf{B}$ is important in the efficient implementation of elements, because this evaluation is repeated at each integration point. For each node point pair i, j define

$$\mathbf{B}_i = \begin{bmatrix} N_{xi} & 0 & 0 \\ 0 & N_{yi} & 0 \\ 0 & 0 & N_{zi} \\ N_{yi} & N_{xi} & 0 \\ 0 & N_{zi} & N_{yi} \\ N_{zi} & 0 & N_{xi} \end{bmatrix}, \qquad \mathbf{B}_j = \begin{bmatrix} N_{xj} & 0 & 0 \\ 0 & N_{yj} & 0 \\ 0 & 0 & N_{zj} \\ N_{yj} & N_{xj} & 0 \\ 0 & N_{zj} & N_{yi} \\ N_{zj} & 0 & N_{xj} \end{bmatrix}, \qquad i,j = 1,\ldots n. \tag{8.11}$$

Then the 3×3 block $\mathbf{Q}_{ji} = \mathbf{B}_j^T\mathbf{E}\mathbf{B}_i = \mathbf{Q}_{ij}^T$ is efficiently computed as

$$\mathbf{C} = \begin{bmatrix} B_{xi}E_{11} + B_{yi}E_{14} + B_{zi}E_{16} & B_{yi}E_{12} + B_{xi}E_{14} + B_{zi}E_{15} & B_{zi}E_{13} + B_{yi}E_{15} + B_{xi}E_{16} \\ B_{xi}E_{12} + B_{yi}E_{24} + B_{zi}E_{26} & B_{yi}E_{22} + B_{xi}E_{24} + B_{zi}E_{25} & B_{zi}E_{23} + B_{yi}E_{25} + B_{xi}E_{26} \\ B_{xi}E_{13} + B_{yi}E_{34} + B_{zi}E_{36} & B_{yi}E_{23} + B_{xi}E_{34} + B_{zi}E_{35} & B_{zi}E_{33} + B_{yi}E_{35} + B_{xi}E_{36} \\ B_{xi}E_{14} + B_{yi}E_{44} + B_{zi}E_{46} & B_{yi}E_{24} + B_{xi}E_{44} + B_{zi}E_{45} & B_{zi}E_{34} + B_{yi}E_{45} + B_{xi}E_{46} \\ B_{xi}E_{15} + B_{yi}E_{45} + B_{zi}E_{56} & B_{yi}E_{25} + B_{xi}E_{45} + B_{zi}E_{55} & B_{zi}E_{35} + B_{yi}E_{55} + B_{xi}E_{56} \\ B_{xi}E_{16} + B_{yi}E_{46} + B_{zi}E_{66} & B_{yi}E_{26} + B_{xi}E_{46} + B_{zi}E_{56} & B_{zi}E_{36} + B_{yi}E_{56} + B_{xi}E_{66} \end{bmatrix}$$
$$\mathbf{Q}_{ij} = \begin{bmatrix} B_{xj}C_{11} + B_{yj}C_{41} + B_{zj}C_{61} & B_{xj}C_{12} + B_{yj}C_{42} + B_{zj}C_{62} & B_{xj}C_{13} + B_{yj}C_{43} + B_{zj}C_{63} \\ B_{yj}C_{21} + B_{xi}C_{41} + B_{zj}C_{51} & B_{yj}C_{22} + B_{xi}C_{42} + B_{zj}C_{52} & B_{yj}C_{23} + B_{xi}C_{43} + B_{zj}C_{53} \\ B_{zj}C_{31} + B_{yj}C_{51} + B_{xj}C_{61} & B_{zj}C_{32} + B_{yj}C_{52} + B_{xj}C_{62} & B_{zj}C_{33} + B_{yj}C_{53} + B_{xj}C_{63} \end{bmatrix} \tag{8.12}$$

The computation of each $\mathbf{Q}_{ij}$ block requires $54+27 = 81$ multiplications. (The slight savings for $i = j$ are not worth the coding complications.) For an element with n nodes and n_G Gauss integration points, $n_G\, n(n+1)/2$ multiplications are required since $\mathbf{Q}_{ji} = \mathbf{Q}_{ij}^T$, and the total effort associated with $\mathbf{B}^T\mathbf{E}\mathbf{E}$ is approximately $40 n_G\, n^2$ multiplications. This computation, plus associated indexing to access the $\mathbf{K}$ entries, dominates the total effort. As an example: an 8-node brick integrated with a $2 \times 2 \times 2$ rule would require roughly 20,000 multiplications whereas a 20-node brick integrated by a $3 \times 3 \times 3$ rule would consume 432,000; thus the formation time ratio would be roughly 22:1.

For an isotropic material two thirds of the stress-strain coefficients in $\mathbf{E}$ are zero. Explicit recognition of that fact would cut the computation of $\mathbf{Q}_{ij}$ to $15 + 21 = 36$ multiplications, roughly a 2:1 speedup. This would complicate the program logic, however, because branching to this special case would be needed, and it is not clear whether the complication is likely to be worth the special effort. In any event, the "unrolled" evaluation of the $\mathbf{Q}_{ij}$ should be preferred to general matrix multiplication.

§8.5. The Mass Matrix

For deriving the consistent mass matrix the iso-P displacement interpolation (8.4) is rearranged as

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & \ldots & N_n & 0 & 0 \\ 0 & N_1 & 0 & \ldots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \ldots & 0 & 0 & N_n \end{bmatrix} \mathbf{u} = \mathbf{N}\mathbf{u}. \tag{8.13}$$

Using either the Lagrange dynamical equations or Hamilton's variational principle, the consistent mass matrix is given by

$$\mathbf{M}_C^e = \int_V \rho\, \mathbf{N}\,\mathbf{N}^T\, dV. \tag{8.14}$$

where ρ is the mass density of the material. If ρ is constant over the element it may be taken out of the foregoing volume integral.

As in the case of the stiffness matrix, the expression (8.14) is usually evaluated by a Gauss quadrature rule. The sparsity of the shape function matrix $\mathbf{N}$ is easily accounted for when forming the 3×3 i, j blocks of $\mathbf{M}^e_C$.

Transformation of $\mathbf{M}^e_C$ to a lumped mass matrix $\mathbf{M}^e_L$ can be done through a variety of techniques. A good coverage is given in the book by Cook et al.[2].

[2] R. D. Cook, D. S. Malkus and M. E. Plesha, *Concepts and Application of Finite Element Methods*, 3rd ed., Wiley, New York, 1989

Homework Exercises for Chapter 8

Solid Elements: Overview

EXERCISE 8.1 [A/C:10]   Tabulate the rough number of multiplications needed to form $\mathbf{K}^e$ for elements with 4, 6, 8, 10, 15, 20 and 27 and 64 nodes, for integration rules containing 1, 4, 6, 8, 15 27 and 64 Gauss points. (Use the formula found in Remark 8.1.) If the computer can do 10^9 multiplications per second, tabulate how many elements can be formed in one second of CPU time.

EXERCISE 8.2 [A/C:20]   Find the approximate cost in multiplications to form the consistent mass matrix (8.14) of a solid element with n nodes if n_G Gauss points are used. Compare to the approximate count $40n_G n(n+1)$ to form the stiffness matrix $\mathbf{K}^e$ if the number of Gauss points n_G is the same.