

17

Isoparametric Quadrilaterals

TABLE OF CONTENTS

	Page
§17.1 Introduction	17-3
§17.2 Partial Derivative Computation	17-3
§17.2.1 The Jacobian	17-3
§17.2.2 Shape Function Derivatives	17-4
§17.2.3 Computing the Jacobian Matrix	17-4
§17.2.4 The Strain-Displacement Matrix	17-5
§17.2.5 A Shape Function Implementation	17-5
§17.3 Numerical Integration	17-6
§17.3.1 One Dimensional Rules	17-7
§17.3.2 Two Dimensional Gauss Product Rules	17-9
§17.3.3 *Non-Product Gauss Rules	17-10
§17.4 The Stiffness Matrix	17-12
§17.5 Implementation of Quad4 Element	17-14
§17.5.1 Quad4 Element Stiffness	17-14
§17.5.2 Quad4 Element Body Force Vector	17-16
§17.5.3 Quad4 Element Stresses	17-16
§17.6 *Quad4 Shear Locking	17-18
§17.6.1 *Plane Beam In Pure Bending	17-18
§17.6.2 *The Quad4 Bending Energy Ratio	17-19
§17.6.3 *Shear Unlocking Methods	17-20
§17.7 *Modified Gauss Integration	17-20
§17.7.1 *Shear Unlocking By WI	17-21
§17.7.2 *Shear Unlocking By SRI: Isotropic Material	17-21
§17.7.3 *Shear Unlocking By SRI: Anisotropic Material	17-22
§17. Notes and Bibliography	17-22
§17. References	17-23
§17. Exercises	17-24

§17.1. Introduction

In this Chapter the isoparametric representation of element geometry and shape functions discussed in the previous Chapter is used to construct *quadrilateral* elements for the plane stress problem. Formulas given in Chapter 14 for the stiffness matrix and consistent load vector of general plane stress elements are of course applicable to these models. For a practical implementation, however, we must go through more specific steps:

1. Construction of shape functions.
2. Computations of shape function derivatives to form the strain-displacement matrix.
3. Numerical integration over the element by Gauss quadrature rules.

The first topic was dealt in the previous Chapter in recipe form, and is systematically covered in the next one. Assuming the shape functions have been constructed (or readily found in the FEM literature) the second and third items are combined in an algorithm suitable for programming any isoparametric quadrilateral. Implementation as element modules is explained with focus on the 4-node bilinear quadrilateral, and covered more generally in Chapter 23.

We shall not deal with isoparametric triangles here to keep the exposition focused. Triangular coordinates, being linked by a constraint, require “special handling” techniques that would complicate and confuse the exposition. Chapter 24 discusses isoparametric triangular elements in detail.

§17.2. Partial Derivative Computation

Partial derivatives of shape functions with respect to the Cartesian coordinates x and y are required for the strain and stress calculations. Because shape functions are not directly functions of x and y but of the natural coordinates ξ and η , the determination of Cartesian partial derivatives is not trivial. The derivative calculation procedure is presented below for the case of an arbitrary isoparametric quadrilateral element with n nodes.

§17.2.1. The Jacobian

In quadrilateral element derivations we will need the Jacobian of two-dimensional transformations that connect the differentials of $\{x, y\}$ to those of $\{\xi, \eta\}$ and vice-versa. Using the chain rule:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}, \quad \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = \mathbf{J}^{-T} \begin{bmatrix} dx \\ dy \end{bmatrix}. \quad (17.1)$$

Here \mathbf{J} denotes the Jacobian matrix of (x, y) with respect to (ξ, η) , whereas \mathbf{J}^{-1} is the Jacobian matrix of (ξ, η) with respect to (x, y) :

$$\mathbf{J} = \frac{\partial(x, y)}{\partial(\xi, \eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}, \quad \mathbf{J}^{-1} = \frac{\partial(\xi, \eta)}{\partial(x, y)} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \frac{1}{J} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix}, \quad (17.2)$$

where $J = |\mathbf{J}| = \det(\mathbf{J}) = J_{11}J_{22} - J_{12}J_{21}$. In FEM work \mathbf{J} and \mathbf{J}^{-1} are called simply the *Jacobian* and *inverse Jacobian*, respectively; the fact that it is a matrix being understood. The scalar symbol

J is reserved for the determinant of \mathbf{J} . In one dimension \mathbf{J} and J coalesce. Jacobians play a crucial role in differential geometry. For the general definition of Jacobian matrix of a differential transformation, see Appendix D.

Remark 17.1. Observe that the matrices relating the differentials in (17.1) are the *transposes* of what we call \mathbf{J} and \mathbf{J}^{-1} . The reason is that coordinate differentials transform as contravariant quantities: $dx = (\partial x/\partial \xi) d\xi + (\partial x/\partial \eta) d\eta$, etc. But Jacobians are arranged as in (17.2) because of earlier use in covariant transformations: $\partial \phi/\partial x = (\partial \xi/\partial x)(\partial \phi/\partial \xi) + (\partial \eta/\partial x)(\partial \phi/\partial \eta)$, as in (17.5) below.

The reader is cautioned that notations vary among application areas. As quoted in Appendix D, one author puts it this way: “When one does matrix calculus, one quickly finds that there are two kinds of people in this world: those who think the gradient is a row vector, and those who think it is a column vector.”

Remark 17.2. To show that \mathbf{J} and \mathbf{J}^{-1} are in fact inverses of each other we form their product:

$$\mathbf{J}^{-1}\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial x}{\partial \eta} \frac{\partial \eta}{\partial x} & \frac{\partial y}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial y}{\partial \eta} \frac{\partial \eta}{\partial x} \\ \frac{\partial x}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial x}{\partial \eta} \frac{\partial \eta}{\partial y} & \frac{\partial y}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial y}{\partial \eta} \frac{\partial \eta}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial x} & \frac{\partial y}{\partial x} \\ \frac{\partial x}{\partial y} & \frac{\partial y}{\partial y} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (17.3)$$

in which we have taken into account that $x = x(\xi, \eta)$, $y = y(\xi, \eta)$ and the fact that x and y are independent coordinates. This proof would collapse, however, if instead of $\{\xi, \eta\}$ we had the triangular coordinates $\{\zeta_1, \zeta_2, \zeta_3\}$ because rectangular matrices have no conventional inverses. This case requires special handling and is covered in Chapter 24.

§17.2.2. Shape Function Derivatives

The shape functions of a quadrilateral element are expressed in terms of the quadrilateral coordinates ξ and η introduced in §16.5. The derivatives with respect to x and y are given by the chain rule:

$$\frac{\partial N_i^e}{\partial x} = \frac{\partial N_i^e}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial N_i^e}{\partial \eta} \frac{\partial \eta}{\partial x}, \quad \frac{\partial N_i^e}{\partial y} = \frac{\partial N_i^e}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial N_i^e}{\partial \eta} \frac{\partial \eta}{\partial y}. \quad (17.4)$$

This can be put in matrix form as

$$\begin{bmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \end{bmatrix} = \frac{\partial(\xi, \eta)}{\partial(x, y)} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \end{bmatrix}. \quad (17.5)$$

in which \mathbf{J}^{-1} is defined in (17.2). The computation of \mathbf{J} is addressed next.

§17.2.3. Computing the Jacobian Matrix

To compute the entries of \mathbf{J} at any quadrilateral location we make use of the last two geometric relations in (16.4), which are repeated here for convenience:

$$x = \sum_{i=1}^n x_i N_i^e, \quad y = \sum_{i=1}^n y_i N_i^e. \quad (17.6)$$

Differentiating with respect to the quadrilateral coordinates we get

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^n x_i \frac{\partial N_i^e}{\partial \xi}, \quad \frac{\partial y}{\partial \xi} = \sum_{i=1}^n y_i \frac{\partial N_i^e}{\partial \xi}, \quad \frac{\partial x}{\partial \eta} = \sum_{i=1}^n x_i \frac{\partial N_i^e}{\partial \eta}, \quad \frac{\partial y}{\partial \eta} = \sum_{i=1}^n y_i \frac{\partial N_i^e}{\partial \eta}. \quad (17.7)$$

because the x_i and y_i do not depend on ξ and η . In matrix form:

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \mathbf{P}\mathbf{X} = \begin{bmatrix} \frac{\partial N_1^e}{\partial \xi} & \frac{\partial N_2^e}{\partial \xi} & \cdots & \frac{\partial N_n^e}{\partial \xi} \\ \frac{\partial N_1^e}{\partial \eta} & \frac{\partial N_2^e}{\partial \eta} & \cdots & \frac{\partial N_n^e}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}. \quad (17.8)$$

Given a quadrilateral point of coordinates ξ, η we calculate the entries of \mathbf{J} using (17.8). The inverse Jacobian \mathbf{J}^{-1} is then obtained by numerically inverting this 2×2 matrix.

A quadrilateral element is said to have *constant metric* (CM) if $J = \det(\mathbf{J})$ is constant over its domain. Examples are rectangles and parallelograms. Otherwise it has *variable metric* (VM). In the latter case J is a rational function of the natural coordinates.

Remark 17.3. The symbolic inversion of \mathbf{J} for arbitrary ξ, η generally leads to complicated expressions unless the element has constant metric (for example rectangles in the first three Exercises). This was one of the difficulties that motivated the use of Gaussian numerical quadrature, which is discussed in §17.3.

§17.2.4. The Strain-Displacement Matrix

The strain-displacement matrix \mathbf{B} that appears in the computation of the element stiffness matrix is given by the general expression (14.18), which is reproduced here for convenience:

$$\mathbf{e} = \begin{bmatrix} e_{xx} \\ e_{yy} \\ 2e_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & 0 & \frac{\partial N_2^e}{\partial x} & 0 & \cdots & \frac{\partial N_n^e}{\partial x} & 0 \\ 0 & \frac{\partial N_1^e}{\partial y} & 0 & \frac{\partial N_2^e}{\partial y} & \cdots & 0 & \frac{\partial N_n^e}{\partial y} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} & \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} & \cdots & \frac{\partial N_n^e}{\partial y} & \frac{\partial N_n^e}{\partial x} \end{bmatrix} \mathbf{u}^e = \mathbf{B}^e \mathbf{u}^e. \quad (17.9)$$

The nonzero entries of \mathbf{B} are shape functions partials with respect to x and y . Their calculation is done by computing \mathbf{J} via (17.8), inverting and using the chain rule (17.5).

§17.2.5. A Shape Function Implementation

To make the foregoing material more specific, Figure 17.1 shows two versions of a *shape function module* for the 4-node bilinear quadrilateral, henceforth abbreviated to Quad4. Both versions return the value of the shape functions and their $\{x, y\}$ derivatives at a given point of quadrilateral coordinates $\{\xi, \eta\}$, and are invoked by the same sequence:

$$\begin{aligned} \{Nf, Nfx, Nfy, Jdet\} &= \text{Quad4IsoPShapeFunDer}[\text{encoor}, \text{qcoor}] \\ \{Nf, Nfx, Nfy, Jdet\} &= \text{Quad4IsoPShapeFunDerFast}[\text{encoor}, \text{qcoor}] \end{aligned} \quad (17.10)$$

The arguments are

- encoor Element node coordinates arranged in two-dimensional list form:
 $\{\{x_1, y_1\}, \{x_2, y_2\}, \{x_3, y_3\}, \{x_4, y_4\}\}$.
- qcoor Quadrilateral coordinates $\{\xi, \eta\}$ of the point.

```

Quad4IsoPShapeFunDer[enccoor_,qccoor_]:= Module[
{Nf,dNx,dNy,dNξ,dNη,i,J11,J12,J21,J22,Jdet,ξ,η,x,y,
x1,x2,x3,x4,y1,y2,y3,y4},
{ξ,η}=qccoor; {{x1,y1},{x2,y2},{x3,y3},{x4,y4}}=enccoor;
Nf={{(1-ξ)*(1-η),(1+ξ)*(1-η),(1+ξ)*(1+η),(1-ξ)*(1+η)}/4;
dNξ ={- (1-η), (1-η),(1+η),-(1+η)}/4;
dNη= {- (1-ξ),-(1+ξ),(1+ξ), (1-ξ)}/4;
x={x1,x2,x3,x4}; y={y1,y2,y3,y4};
J11=dNξ.x; J12=dNξ.y; J21=dNη.x; J22=dNη.y;
Jdet=Simplify[J11*J22-J12*J21];
dNx= ( J22*dNξ-J12*dNη)/Jdet; dNx=Simplify[dNx];
dNy= (-J21*dNξ+J11*dNη)/Jdet; dNy=Simplify[dNy];
Return[{Nf,dNx,dNy,Jdet}]];

Quad4IsoPShapeFunDerFast[enccoor_,qccoor_]:= Module[
{Nf,dNx,dNy,Jdet,Jdet8,ξ,η,x1,x2,x3,x4,y1,y2,y3,y4,
x21,x31,x41,x32,x42,x43,y21,y31,y41,y32,y42,y43},
{ξ,η}=qccoor; {{x1,y1},{x2,y2},{x3,y3},{x4,y4}}=enccoor;
Nf={{(1-ξ)*(1-η),(1+ξ)*(1-η),(1+ξ)*(1+η),(1-ξ)*(1+η)}/4;
x21=x2-x1; x31=x3-x1; x41=x4-x1; x32=x3-x2; x42=x4-x2; x43=x4-x3;
y21=y2-y1; y31=y3-y1; y41=y4-y1; y32=y3-y2; y42=y4-y2; y43=y4-y3;
Jdet8=-y31*x42+x31*y42+(y21*x43-x21*y43)*ξ+(x32*y41-x41*y32)*η;
Jdet=Jdet8/8;
dNx={-y42+y43*ξ+y32*η, y31-y43*ξ-y41*η, y42-y21*ξ+y41*η,
-y31+y21*ξ-y32*η}/Jdet8;
dNy={ x42-x43*ξ-x32*η,-x31+x43*ξ+x41*η,-x42+x21*ξ-x41*η,
x31-x21*ξ+x32*η}/Jdet8;
Return[{Nf,dNx,dNy,Jdet}]];

```

FIGURE 17.1. Shape function modules for the 4-node bilinear quadrilateral Quad4. Top is standard code; bottom is a faster inlined version.

The module returns:

Nf	Value of shape functions, arranged as list { Nf1,Nf2,Nf3,Nf4 }.
Nfx	Value of x -derivatives of shape functions, arranged as list { Nfx1,Nfx2,Nfx3,Nfx4 }.
Nfy	Value of y -derivatives of shape functions, arranged as list { Nfy1,Nfy2,Nfy3,Nfy4 }.
Jdet	Jacobian determinant.

Remark 17.4. The second version, identified as Quad4IsoPShapeFunDerFast, is an inlined version of the output of the first one, which avoids any array operations. It computes its outputs in 41 additions/subtractions, 27 multiplications, and 11 divisions. If implemented in C or C++ on a RISC machine, most of the internal variables should be kept in registers to reduce memory traffic. The difference between the first and second versions on an interpreted language such as *Mathematica*, *Matlab* or *Python* would be unnoticeable. But a modified “Fast” version may be useful for symbolic downstream inlining of the stiffness matrix code to avoid all loops, treating x_{21}, y_{21}, \dots as primitive variables.

Example 17.1. Consider a 4-node bilinear quadrilateral shaped as an axis-aligned 2:1 rectangle, with $2a$ and a as the x and y dimensions, respectively. The node coordinate array is $\text{enccoor}=\{\{0,0\},\{2*a,0\},\{2*a,a\},\{0,a\}\}$. The shape functions and their $\{x,y\}$ derivatives are to be evaluated at the rectangle center $\xi = \eta = 0$. The appropriate call is

$$\{Nf,Nfx,Nfy,Jdet\}=\text{Quad4IsoPShapeFunDer}[\text{enccoor},\{0,0\}]$$

This returns $Nf=\{1/8,1/8,3/8,3/8\}$, $Nfx=\{-1/(8*a),1/(8*a),3/(8*a),-3/(8*a)\}$, $Nfy=\{-1/(2*a),-1/(2*a),1/(2*a),1/(2*a)\}$ and $Jdet=a^2/2$.

Table 17.1. One-Dimensional Gauss Rules with 1 through 5 Sample Points

Points	Rule
1	$\int_{-1}^1 F(\xi) d\xi \approx 2F(0)$
2	$\int_{-1}^1 F(\xi) d\xi \approx F(-1/\sqrt{3}) + F(1/\sqrt{3})$
3	$\int_{-1}^1 F(\xi) d\xi \approx \frac{5}{9}F(-\sqrt{3/5}) + \frac{8}{9}F(0) + \frac{5}{9}F(\sqrt{3/5})$
4	$\int_{-1}^1 F(\xi) d\xi \approx w_{14}F(\xi_{14}) + w_{24}F(\xi_{24}) + w_{34}F(\xi_{34}) + w_{44}F(\xi_{44})$
5	$\int_{-1}^1 F(\xi) d\xi \approx w_{15}F(\xi_{15}) + w_{25}F(\xi_{25}) + w_{35}F(\xi_{35}) + w_{45}F(\xi_{45}) + w_{55}F(\xi_{55})$

For the 4-point rule, $\xi_{34} = -\xi_{24} = \sqrt{(3 - 2\sqrt{6/5})/7}$, $\xi_{44} = -\xi_{14} = \sqrt{(3 + 2\sqrt{6/5})/7}$, $w_{14} = w_{44} = \frac{1}{2} - \frac{1}{6}\sqrt{5/6}$, and $w_{24} = w_{34} = \frac{1}{2} + \frac{1}{6}\sqrt{5/6}$.

For the 5-point rule, $\xi_{55} = -\xi_{15} = \frac{1}{3}\sqrt{5 + 2\sqrt{10/7}}$, $\xi_{45} = -\xi_{35} = \frac{1}{3}\sqrt{5 - 2\sqrt{10/7}}$, $\xi_{35} = 0$, $w_{15} = w_{55} = (322 - 13\sqrt{70})/900$, $w_{25} = w_{45} = (322 + 13\sqrt{70})/900$ and $w_{35} = 512/900$.

§17.3. Numerical Integration

Numerical integration is an essential tool for practical evaluation of integrals over isoparametric element domains of variable metric (VM). Since the mid-1960s the standard practice has been to use *Gauss integration* because such rules use a *minimal number of sample points to achieve a desired level of accuracy*. This is important for efficient element calculations, because a double matrix product, namely $\mathbf{B}^T \mathbf{E} \mathbf{B}$, is evaluated at each Gauss sample point. The fact that the location of those points is usually given by non-rational numbers is of no concern in digital computation.

Gauss rules for quadrilateral regions can be divided into product and non-product rules. The former, which are the standard ones for FEM use, are constructed as tensor products of the classical one-dimensional (1D) rules. Non-product rules are covered in §17.3.3 as advanced material.

§17.3.1. One Dimensional Rules

The classical Gauss integration rules are defined by

$$\int_{-1}^1 F(\xi) d\xi \approx \sum_{i=1}^p w_i F(\xi_i). \quad (17.11)$$

Here $p \geq 1$ is the number of Gauss integration points (also known as sample points), w_i are the integration weights, and ξ_i are sample-point abscissas inside $\in [-1, 1]$. The use of the canonical interval $[-1, 1]$ is no restriction, because an integral over another range, say from a to b , can be transformed to $[-1, +1]$ via a simple linear transformation of the independent variable, as shown in Remark 17.5 below.

The first five unidimensional (1D) Gauss rules are listed in Table 17.1, and pictured in Figure 17.2. These integrate exactly polynomials in ξ of orders up to 1, 3, 5, 7 and 9, respectively. In general a 1D Gauss rule with p points integrates exactly polynomials of order up to, and including, $2p - 1$. This is called the *degree* of the formula.

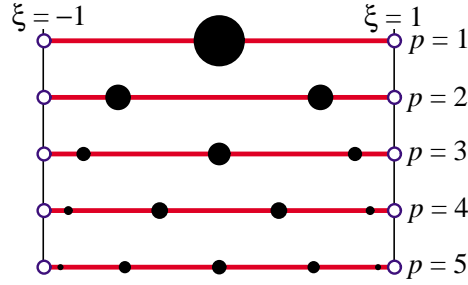


FIGURE 17.2. The first five 1D Gauss rules pictured over the segment $\xi \in [-1, +1]$. Sample point locations are marked with black circles. Circle radii are proportional to the integration weights.

```

LineGaussRuleInfo[{rule_, number_}, point_] := Module[
  {g2 = {-1, 1}/Sqrt[3],
   g3 = {-Sqrt[3/5], 0, Sqrt[3/5]}, w3 = {5/9, 8/9, 5/9},
   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 [number, Return[N[info]], Return[Simplify[info]]];
];

```

FIGURE 17.3. *Mathematica* module that returns information on the first five 1D Gauss rules.

The *Mathematica* module listed in Figure 17.3 returns either exact or floating-point information for the first five unidimensional Gauss rules. To get information for the i^{th} point of the p^{th} rule, in which $1 \leq i \leq p$ and $p = 1, 2, 3, 4, 5$, the module should be invoked as

$$\{\xi_i, w_i\} = \text{LineGaussRuleInfo}[\{p, \text{number}\}, i] \quad (17.12)$$

Logical flag *number* is *True* to get numerical (floating-point) information, or *False* to get exact information. The module returns the sample point abscissa ξ_i in ξ_i and the weight w_i in w_i . If p is not in the range 1 through 5, the module returns $\{\text{Null}, 0\}$.

Example 17.2. The call $\{\xi_i, w_i\} = \text{LineGaussRuleInfo}[\{3, \text{False}\}, 2]$ returns $\xi_i = 0$ and $w_i = 8/9$, whereas $\{\xi_i, w_i\} = \text{LineGaussRuleInfo}[\{3, \text{True}\}, 2]$ returns (to 16 places) $\xi_i = 0.$ and $w_i = 0.8888888888888889$.

Remark 17.5. A more general 1D integral, such as $F(x)$ over $[a, b]$, in which $\ell = b - a > 0$, is transformed to the canonical interval $[-1, 1]$ through the mapping $x = \frac{1}{2}a(1 - \xi) + \frac{1}{2}b(1 + \xi) = \frac{1}{2}(a + b) + \frac{1}{2}\ell\xi$, or $\xi = (2/\ell)(x - \frac{1}{2}(a + b))$. The Jacobian of this mapping is $J = dx/d\xi = \frac{1}{2}\ell$. Thus

$$\int_a^b F(x) dx = \int_{-1}^1 F(\xi) J d\xi = \int_{-1}^1 F(\xi) \frac{1}{2}\ell d\xi. \quad (17.13)$$

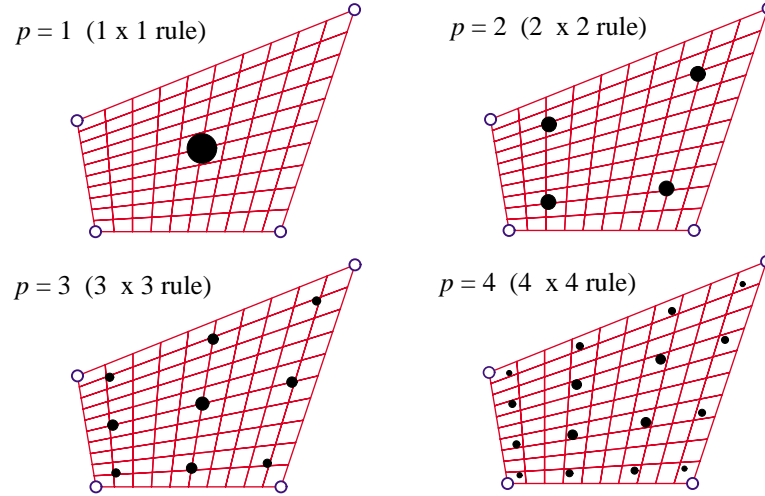


FIGURE 17.4. The first four two-dimensional Gauss product rules $p = 1, 2, 3, 4$ depicted over a straight-sided quadrilateral region. Sample points are marked with black circles. Circle areas are proportional to integration weights.

Remark 17.6. Higher order Gauss rules are tabulated in manuals for numerical computation. For example, the widely used Handbook of Mathematical Functions [2] lists rules with up to 96 points in Table 25.4. For $p > 6$ the abscissas and weights of sample points are not expressible as rational numbers or radicals, and can only be given as floating-point numbers.

§17.3.2. Two Dimensional Gauss Product Rules

The simplest bidimensional (2D) Gauss integration formulas are called *product rules*. They are obtained by applying the unidimensional rules of Table 17.1 along each natural coordinate in turn. To use them we must first reduce the integrand to the canonical form:

$$\int_{-1}^1 \int_{-1}^1 F(\xi, \eta) d\xi d\eta = \int_{-1}^1 d\eta \int_{-1}^1 F(\xi, \eta) d\xi. \quad (17.14)$$

Once this is done we can process numerically each integral in turn:

$$\int_{-1}^1 \int_{-1}^1 F(\xi, \eta) d\xi d\eta \approx \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} w_i w_j F(\xi_i, \eta_j) = \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} w_{ij} F(\xi_i, \eta_j). \quad (17.15)$$

Here p_1 and p_2 denote the number of Gauss points in the ξ and η directions, respectively. Usually the same number $p = p_1 = p_2$ is chosen if the shape functions are taken to be the same in the ξ and η directions. This is in fact the case for all quadrilateral elements presented here. The first four 2D Gauss product rules with $p = p_1 = p_2$ are illustrated in Figure 17.4.

The *Mathematica* module listed in Figure 17.5 implements 2D product Gauss rules having 1 through 5 points in each direction.¹ The number of points in each direction may be the same or different.

¹ Written originally in Fortran IV for the 1966 thesis [221].

```

QuadGaussRuleInfo[{rule_,number_},point_]:= Module[
  {ξ,η,p1,p2,i,j,w1,w2,m,info={{Null,Null},0}},
  If [Length[rule]==2, {p1,p2}=rule, p1=p2=rule];
  If [p1<0, Return[QuadNonProductGaussRuleInfo[
    {-p1,number},point]];
  If [Length[point]==2, {i,j}=point, m=point;
    j=Floor[(m-1)/p1]+1; i=m-p1*(j-1)];
  {ξ,w1}= LineGaussRuleInfo[{p1,number},i];
  {η,w2}= LineGaussRuleInfo[{p2,number},j];
  info={{ξ,η},w1*w2};
  If [number, Return[N[info]], Return[Simplify[info]]];
];

```

FIGURE 17.5. *Mathematica* module that returns information on two-dimensional product Gauss rules.

If the rule has the same number of points p in both directions the module is called in either of two ways:

$$\begin{aligned}
 \{\{\xi_i, \eta_j\}, w_{ij}\} &= \text{QuadGaussRuleInfo}[\{p, \text{number}\}, \{i, j\}] \\
 \{\{\xi_i, \eta_j\}, w_{ij}\} &= \text{QuadGaussRuleInfo}[\{p, \text{number}\}, k]
 \end{aligned}
 \tag{17.16}$$

The first form is used to get information for point $\{i, j\}$ of the $p \times p$ rule, in which $1 \leq i \leq p$ and $1 \leq j \leq p$. The second form specifies that point by a “visiting counter” k that runs from 1 through p^2 ; if so $\{i, j\}$ are internally extracted as $j = \text{Floor}[(k-1)/p] + 1$; $i = k - p*(j-1)$.

If the integration rule has p_1 points in the ξ direction and p_2 points in the η direction, the module may be called also in two ways:

$$\begin{aligned}
 \{\{\xi_i, \eta_j\}, w_{ij}\} &= \text{QuadGaussRuleInfo}[\{\{p_1, p_2\}, \text{number}\}, \{i, j\}] \\
 \{\{\xi_i, \eta_j\}, w_{ij}\} &= \text{QuadGaussRuleInfo}[\{\{p_1, p_2\}, \text{number}\}, k]
 \end{aligned}
 \tag{17.17}$$

The meaning of the second argument is as follows. In the first form i runs from 1 to p_1 and j from 1 to p_2 . In the second form k runs from 1 to $p_1 p_2$; if so i and j are extracted by $j = \text{Floor}[(k-1)/p_1] + 1$; $i = k - p_1*(j-1)$. In all four forms, logical flag *number* is set to *True* if numerical (floating-point) information is desired and to *False* if exact information is desired.

The module returns ξ_i and η_j in ξ_i and η_j , respectively, and the weight product $w_{ij} = w_i w_j$ in w_{ij} . This code is used in the Exercises at the end of the chapter. If the inputs are not in valid range, the module returns $\{\{\text{Null}, \text{Null}\}, 0\}$.

Example 17.3. $\{\{\xi_i, \eta_j\}, w\} = \text{QuadGaussRuleInfo}[\{3, \text{False}\}, \{2, 3\}]$ returns $\xi_i=0$, $\eta_j=\text{Sqrt}[3/5]$ and $w=40/81$.

Example 17.4. $\{\{\xi_i, \eta_j\}, w\} = \text{QuadGaussRuleInfo}[\{3, \text{True}\}, \{2, 3\}]$ returns (to 16-place precision) $\xi_i=0.$, $\eta_j=0.7745966692414834$ and $w=0.49382716049382713$.

§17.3.3. *Non-Product Gauss Rules

Non-product (NP) Gauss rules for quadrilaterals include those that are not necessarily composed as the tensor product of two unidimensional rules. This generalization is occasionally convenient in three scenarios: (1) developing quadrilateral elements that are not in the isoparametric class, (2) support of the modified Gauss integration techniques covered in §17.7, and (3) reducing the number of sample points to speed up formation

```

QuadNPGaussRuleInfo[{rule_, numer_}, point_, s_, w_] := Module[{ $\alpha$ ,  $\beta$ ,
w1=1, w2, w3, i=point, ns=Length[s], nw=Length[w], s1=Sqrt[1/3],
s2=Sqrt[2/3], s3=Sqrt[3/5], s4=Sqrt[(7-Sqrt[14])/15],
s5=Sqrt[(7+Sqrt[14])/15], info={{Null, Null}, 0}},
If [!MemberQ[{1, 4, -4, 5, -5, 8, -8, 9, -9}, rule], Return[info]];
If [i<1||i>Abs[rule], Return[info]];
If [rule==1, info={{0, 0}, 4}];
If [rule==4,  $\alpha$ =s1; If [ns==1, { $\alpha$ }=s];
info=
{{{- $\alpha$ , - $\alpha$ }, w1}, {{ $\alpha$ , - $\alpha$ }, w1}, {{ $\alpha$ ,  $\alpha$ }, w1}, {{- $\alpha$ ,  $\alpha$ }, w1}}[[i]];
If [rule==4,  $\alpha$ =s2; If [ns==1, { $\alpha$ }=s];
info=
{{{- $\alpha$ , 0}, w1}, {{ $\alpha$ , 0}, w1}, {{0, - $\alpha$ }, w1}, {{0,  $\alpha$ }, w1}}[[i]];
If [rule==5,  $\alpha$ =s1; If [ns==1, { $\alpha$ }=s]; If [nw==1, {w1}=w]; w2=4-4*w1;
info=
{{{- $\alpha$ , - $\alpha$ }, w1}, {{ $\alpha$ , - $\alpha$ }, w1}, {{ $\alpha$ ,  $\alpha$ }, w1}, {{- $\alpha$ ,  $\alpha$ }, w1}, {{0, 0}, w2}}[[i]];
If [rule==5,  $\alpha$ =s2; If [ns==1, { $\alpha$ }=s]; If [nw==1, {w1}=w]; w2=4-4*w1;
info=
{{{- $\alpha$ , 0}, w1}, {{ $\alpha$ , 0}, w1}, {{0, - $\alpha$ }, w1}, {{0,  $\alpha$ }, w1}, {{0, 0}, w2}}[[i]];
If [rule==8,  $\alpha$ = $\beta$ =s1; If [ns==2, { $\alpha$ ,  $\beta$ }=s]; w1=1/2; If [nw==1, {w1}=w];
info=
{{{- $\alpha$ , - $\beta$ }, w1}, {{ $\alpha$ , - $\beta$ }, w1}, {{ $\alpha$ ,  $\beta$ }, w1}, {{- $\alpha$ ,  $\beta$ }, w1},
{{- $\beta$ , - $\alpha$ }, w1}, {{- $\beta$ ,  $\alpha$ }, w1}, {{ $\beta$ ,  $\alpha$ }, w1}, {{- $\beta$ ,  $\alpha$ }, w1}}[[i]];
If [rule==8,  $\alpha$ =s1;  $\beta$ =s2; If [ns==2, { $\alpha$ ,  $\beta$ }=s]; w1=w2=1/2;
If [nw==1, {w1}=w]; w2=4-4*w1;
info=
{{{- $\alpha$ , - $\alpha$ }, w1}, {{ $\alpha$ , - $\alpha$ }, w1}, {{ $\alpha$ ,  $\alpha$ }, w1}, {{- $\alpha$ ,  $\alpha$ }, w1},
{{- $\beta$ , 0}, w2}, {{ $\beta$ , 0}, w2}, {{0, - $\beta$ }, w2}, {{0,  $\beta$ }, w2}}[[i]];
If [rule==9,  $\alpha$ = $\beta$ =s3; If [ns==2, { $\alpha$ ,  $\beta$ }=s]; w1=1/(9* $\alpha$ ^4);
w2=8/(45* $\beta$ ^4); If [nw==2, {w1, w2}=w]; w3=4-4*w1-4*w2;
info=
{{{- $\alpha$ , - $\alpha$ }, w1}, {{ $\alpha$ , - $\alpha$ }, w1}, {{ $\alpha$ ,  $\alpha$ }, w1}, {{- $\alpha$ ,  $\alpha$ }, w1},
{{- $\beta$ , 0}, w2}, {{ $\beta$ , 0}, w2}, {{0, - $\beta$ }, w2}, {{0,  $\beta$ }, w2}, {{0, 0}, w3}}[[i]];
If [rule==9,  $\alpha$ =s4;  $\beta$ =s5; If [ns==2, { $\alpha$ ,  $\beta$ }=s]; w1=5/14;
If [nw==1, {w1}=w]; w2=4-8*w1;
info=
{{{- $\alpha$ , - $\beta$ }, w1}, {{ $\alpha$ , - $\beta$ }, w1}, {{ $\alpha$ ,  $\beta$ }, w1}, {{- $\alpha$ ,  $\beta$ }, w1},
{{- $\beta$ , - $\alpha$ }, w1}, {{- $\beta$ ,  $\alpha$ }, w1}, {{ $\beta$ ,  $\alpha$ }, w1}, {{- $\beta$ ,  $\alpha$ }, w1}, {{0, 0}, w2}}[[i]];
If [numer, Return[N[info]], Return[Simplify[info]]];

```

FIGURE 17.6. *Mathematica* module that returns information on two-dimensional non-product Gauss rules.

of higher order quadrilaterals.² For convenience, rules with up to 9 points are presented here along with a *Mathematica* implementation. That module is listed in Figure 17.6. 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.³
- 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 can be usually tolerated.

Product rules automatically satisfy all three requirements.

² The last scenario does not apply to the four-node iso-P quadrilateral, since there are no FS rules with 2 or 3 points.

³ Stated mathematically: the numerically evaluated integral must remain *invariant* under all affine transformations of the element domain onto itself.

Table 17.2. Non-Product Gauss Rules for Quadrilaterals

Rule id	Stars	#Points	Defaults
1	$S_1(w_1)$	1	$w_1 = 4$
4	$S_4(\alpha, w_1)$	4	$\alpha = s_1, w_1 = 1$
-4	$\hat{S}_4(\alpha, w_1)$	4	$\alpha = s_2, \hat{w}_1 = 1$
5	$S_4(\alpha, w_1) \cup S_1(w_2)$	5	$\alpha = s_1, \hat{w}_1 = 1, w_2 = 0$
-5	$\hat{S}_4(\alpha, w_1) \cup S_1(w_2)$	5	$\alpha = s_2, \hat{w}_1 = 1, w_1 = 0$
8	$S_8(\alpha, \beta, w_1)$	8	$\alpha = \beta = s_1, w_1 = 1/2$
-8	$S_4(\alpha, w_1) \cup \hat{S}_4(\beta, w_2)$	8	$\alpha = s_1, \beta = s_2, w_1 = 1/2, w_2 = 1/2$
9	$S_4(\alpha, w_1) \cup \hat{S}_4(\beta, w_2) \cup S_1(w_3)$	9	$\alpha = \beta = s_3, w_4 = 25/81, \hat{w}_4 = 40/81, w_3 = 64/81$
-9	$S_8(\alpha, \beta, w_1) \cup S_1(w_2)$	9	$\alpha = s_4, \beta = s_5, w_1 = 5/14, w_2 = 18/7$

Abbreviations: $s_1 = \sqrt{1/3}, s_2 = \sqrt{2/3}, s_3 = \sqrt{3/5}, s_4 = (7 - \sqrt{14})/15, s_5 = (7 + \sqrt{14})/15$.
Rules 1, 4 and 9 agree with 1x1, 2x2 and 3x3 product rules, respectively, if defaults are used.

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. Four FS stars are used in the module of Figure 17.6:

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

$S_4(\alpha, w)$ 4 sample points at $\{-\alpha, -\alpha\}, \{\alpha, -\alpha\}, \{\alpha, \alpha\}$, and $\{-\alpha, \alpha\}$, with $\alpha \neq 0$, and positive weight w .

$\hat{S}_4(\alpha, w)$ 4 sample points at $\{-\alpha, 0\}, \{\alpha, 0\}, \{0, -\alpha\}$, and $\{0, \alpha\}$, with $\alpha \neq 0$, and positive weight w .

$S_8(\alpha, \beta, w)$ 8 sample points at $\{-\alpha, -\beta\}, \{\alpha, -\beta\}, \{\alpha, \beta\}, \{\alpha, -\beta\}, \{-\beta, -\alpha\}, \{\beta, -\alpha\}, \{\beta, \alpha\}$, and $\{\beta, -\alpha\}$, with $\alpha \neq 0, \beta \neq 0, \alpha \neq \beta$, and positive weight w .

If a rule is made up of n_1, n_4, \hat{n}_4 and n_8 stars of those types, respectively, the total number of Gauss points will be $n_G = n_1 + 4n_4 + 4\hat{n}_4 + 8n_8$, in which n_1 can only be 0 or 1. Thus there are no FS rules with 2, 3, 6 and 7 points if $n_G \leq 9$. The module listed in Figure 17.6 implements 8 rules with 1, 4, 4, 5, 5, 8, 9 and 9 points. To disambiguate two rules with same number of points, a sign is used. See Table 17.2 for details.

The module is invoked as

$$\{\{\xi_j, \eta_i\}, w_i\} = \text{QuadNPGaussRuleInfo}[\{\text{rule}, \text{numer}\}, \text{point}, s, w] \quad (17.18)$$

The arguments are

- rule Rule identifier; see first column of Table 17.2. Its absolute value is the number of sample points.
- numer Same as in `QuadGaussRuleInfo`; see §17.3.2.
- point Sample point index, in the range 1 through $|\text{rule}|$. The alternative two-index specification $\{i, j\}$ of `QuadGaussRuleInfo` is not available.
- s A list containing $\{\alpha\}$ for rules with 4 or 5 points, or $\{\alpha, \beta\}$ for rules with 8 or 9 points. If specified as an empty list: $\{\}$, the defaults listed in Table 17.2 are used.
- w A list containing $\{w_1\}$ for rules that use two weights: w_1, w_2 , or $\{w_1, w_2\}$ for rules that use three weights: w_1, w_2, w_3 . The nonspecified weight is internally adjusted so the weight sum for all sample points is 4. If specified as an empty list: $\{\}$, the defaults listed in Table 17.2 are used.

The module returns the natural coordinates $\{\xi_i, \eta_i\}$ and weight w_i for the specified sample point index.

§17.4. The Stiffness Matrix

The stiffness matrix of a *general* plane stress element is given by (14.23), which is reproduced here for convenience:

$$\mathbf{K}^e = \int_{\Omega^e} h \mathbf{B}^T \mathbf{E} \mathbf{B} d\Omega^e. \quad (17.19)$$

Of the integrand terms in (17.19) the strain-displacement matrix \mathbf{B} has been discussed in §17.2.4. A constant thickness h may be pulled out of the integral, whereas a variable h may be interpolated from node values with the element shape functions. The elasticity matrix \mathbf{E} is usually constant in applications, but we could in principle interpolate it as necessary should it vary over the element.

To integrate (17.19) numerically by a two-dimensional product Gauss rule, it must be first put in the canonical form (17.14), that is

$$\mathbf{K}^e = \int_{-1}^1 \int_{-1}^1 \mathbf{F}(\xi, \eta) d\xi d\eta. \quad (17.20)$$

With one exception, everything in (17.19) fits (17.20), since \mathbf{B} is a function of the natural coordinates while h and \mathbf{E} may be interpolated if necessary. The exception is the area differential $d\Omega^e$. To complete the reduction we need to express it in terms of the differentials $d\xi$ and $d\eta$. As shown in Remark 17.7 the desired relation is

$$d\Omega^e = dx dy = \det \mathbf{J} d\xi d\eta = J d\xi d\eta. \quad (17.21)$$

Accordingly, the canonicalized integrand is

$$\mathbf{F}(\xi, \eta) = h \mathbf{B}^T \mathbf{E} \mathbf{B} \det \mathbf{J}. \quad (17.22)$$

This matrix function is numerically integrated over the domain $-1 \leq \xi \leq +1$, $-1 \leq \eta \leq +1$ by a Gauss product rule. If the same number of Gauss points p are used in the ξ and η directions we have

$$\mathbf{K}^e = \sum_{i=1}^p \sum_{j=1}^p w_{ij} h(\xi_i, \eta_j) \mathbf{B}(\xi_i, \eta_j)^T \mathbf{E} \mathbf{B}(\xi_i, \eta_j) \det \mathbf{J}(\xi_i, \eta_j). \quad (17.23)$$

Here ξ_i and η_j denote sample point abscissas, w_{ij} the corresponding weight, and \mathbf{E} is assumed constant (if variable, it should be evaluated as $\mathbf{E}(\xi_i, \eta_j)$). The question of choosing p to achieve rank sufficiency is discussed at length in Chapter 19, but is investigated numerically in Exercise 17.1

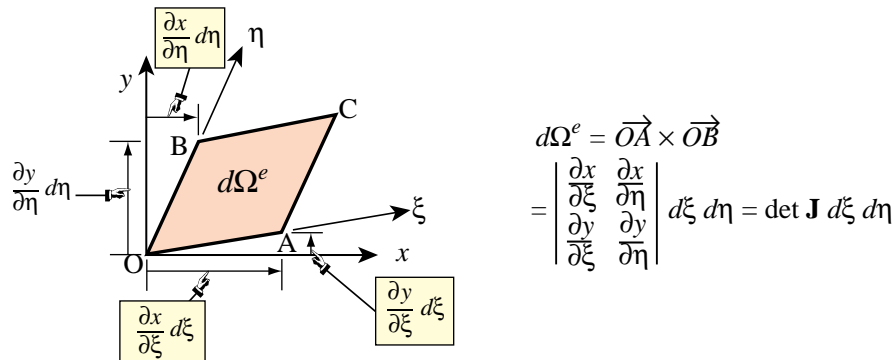


FIGURE 17.7. Geometric interpretation of the Jacobian determinant role in the formula (17.21).

```

Quad4IsoPMembStiffness[enccoor_,Emat_,th_,options_]:=
Module[{i,k,pG=2,numer=False,h=th,qcoor,c,w,Nf,
  dNx,dNy,Jdet,Be,Ke=Table[0,{8},{8}]},
If [Length[options]==2, {numer,pG}=options, {numer}=options];
If [pG<1||pG>4, Print["pG out of range"]; Return[Null]];
For [k=1, k<=pG*pG, k++,
  {qcoor,w}= QuadGaussRuleInfo[{pG,numer},k];
  {Nf,dNx,dNy,Jdet}=Quad4IsoPShapeFunDer[enccoor,qcoor];
  If [Length[th]==4, h=th.Nf]; c=w*Jdet*h;
  Be={Flatten[Table[{dNx[[i]], 0},{i,4}]],
    Flatten[Table[{0, dNy[[i]]},{i,4}]],
    Flatten[Table[{dNy[[i]],dNx[[i]]},{i,4}]]};
  Ke+=Simplify[c*Transpose[Be].(Emat.Be)];
]; Return[Simplify[Ke]]
1;

```

FIGURE 17.8. Module to compute the stiffness matrix of an isoparametric Quad4 element in plane stress.

Remark 17.7. To geometrically justify the transformation formula (17.21), consider the infinitesimal element of area OACB depicted in Figure 17.7. The area of the parallelogram can be computed using the outer product formula

$$d\Omega^e = \vec{OB} \times \vec{OA} = \frac{\partial x}{\partial \xi} d\xi \frac{\partial y}{\partial \eta} d\eta - \frac{\partial x}{\partial \eta} d\eta \frac{\partial y}{\partial \xi} d\xi = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} d\xi d\eta = \det \mathbf{J} d\xi d\eta. \quad (17.24)$$

This formula can be extended to any number of space dimensions, as shown in textbooks on differential geometry; for example [284,349,782].

§17.5. Implementation of Quad4 Element

To illustrate a specific application of the tools described so far, the computer implementation of stiffness, body force and stress calculations are presented for the isoparametric Quad4 element. All three modules make use of the shape function module Quad4IsoPShapeFunDer listed in Figure 17.1. The stiffness and body force modules use the Gauss integration modules QuadGaussRuleInfo and (indirectly) LineGaussRuleInfo, which are listed in Figures 17.5 and 17.3, respectively. This material is also used for several end-of-Chapter Exercises.

§17.5.1. Quad4 Element Stiffness

The *Mathematica* module Quad4IsoPMembStiffness, listed in Figure 17.8, computes and returns the element stiffness matrix of the Quad4 element in plane stress. ("Memb" in its name is short for "Membrane", which is another term for plane stress.) The module is invoked as

$$\mathbf{Ke} = \text{Quad4IsoPMembStiffness}[\text{enccoor}, \text{Emat}, \text{th}, \text{options}] \quad (17.25)$$

The arguments are:

enccoor Element node coordinates arranged in two-dimensional list form:
 {{x1,y1},{x2,y2},{x3,y3},{x4,y4}}.

Emat A two-dimensional list that stores the 3×3 plane stress matrix of elastic moduli:

$$\mathbf{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{12} & E_{22} & E_{23} \\ E_{13} & E_{23} & E_{33} \end{bmatrix} \quad (17.26)$$

arranged as $\{ \{ E_{11}, E_{12}, E_{33} \}, \{ E_{12}, E_{22}, E_{23} \}, \{ E_{13}, E_{23}, E_{33} \} \}$. Must be symmetric. If the material is isotropic with elastic modulus E and Poisson's ratio ν , this matrix becomes

$$\mathbf{E} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1 - \nu) \end{bmatrix} \quad (17.27)$$

th The plate thickness specified as a 4-entry list: $\{ h_1, h_2, h_3, h_4 \}$ or as a scalar: h .

The first form is used to specify an element of variable thickness, in which case the entries are the four corner thicknesses and $h(\xi, \eta)$ is interpolated bilinearly. The second form specifies uniform thickness.

options Processing options. A list that may contain two items: $\{ \text{numer}, \text{pG} \}$ or one: $\{ \text{numer} \}$.

numer is a logical flag. If True, the computations are done in floating point arithmetic. For symbolic or exact arithmetic work, set **numer** to False.⁴

pG specifies the Gauss product rule to have **pG** points in each direction. **pG** may be 1 through 4. For rank sufficiency, **p** must be 2 or higher. If **pG** is 1 the element will be rank deficient by two.⁵ If omitted **pG** = 2 is assumed.⁶

The module returns \mathbf{K}_e as an 8×8 symmetric matrix pertaining to the following arrangement of nodal displacement freedoms:

$$\mathbf{u}^e = [u_{x1} \quad u_{y1} \quad u_{x2} \quad u_{y2} \quad u_{x3} \quad u_{y3} \quad u_{x4} \quad u_{y4}]^T. \quad (17.28)$$

Example 17.5. Consider the specialization of the general 4-node bilinear quadrilateral to a *rectangular* element dimensioned a and b in the x and y directions, respectively, as shown in Figure 17.9. The element has uniform thickness h . The material is isotropic and homogeneous with elastic modulus E and Poisson's ratio ν ; consequently \mathbf{E} reduces to (17.27). The stiffness matrix of this element can be expressed in closed form.⁷ For convenience define $\gamma = a/b$ (rectangle aspect ratio) and

$$\begin{aligned} \psi_1 &= (1 + \nu)\gamma, & \psi_2 &= (1 - 3\nu)\gamma, & \psi_3 &= 2 + (1 - \nu)\gamma^2, & \psi_4 &= 2\gamma^2 + (1 - \nu), \\ \psi_5 &= (1 - \nu)\gamma^2 - 4, & \psi_6 &= (1 - \nu)\gamma^2 - 1, & \psi_7 &= 4\gamma^2 - (1 - \nu), & \psi_8 &= \gamma^2 - (1 - \nu). \end{aligned} \quad (17.29)$$

Then the stiffness matrix in closed form reads

$$\mathbf{K}^e = \frac{Eh}{24\gamma(1 - \nu^2)} \begin{bmatrix} 4\psi_3 & 3\psi_1 & 2\psi_5 & -3\psi_2 & -2\psi_3 & -3\psi_1 & -4\psi_6 & 3\psi_2 \\ & 4\psi_4 & 3\psi_2 & 4\psi_8 & -3\psi_1 & -2\psi_4 & -3\psi_2 & -2\psi_7 \\ & & 4\psi_3 & -3\psi_1 & -4\psi_6 & -3\psi_2 & -2\psi_3 & 3\psi_1 \\ & & & 4\psi_4 & 3\psi_2 & -2\psi_7 & 3\psi_1 & -2\psi_4 \\ & & & & 4\psi_3 & 3\psi_1 & 2\psi_5 & -3\psi_2 \\ & & & & & 4\psi_4 & 3\psi_2 & 4\psi_8 \\ & & & & & & 4\psi_3 & -3\psi_1 \\ \text{symm} & & & & & & & 4\psi_4 \end{bmatrix}. \quad (17.30)$$

⁴ The reason for this option is speed. A symbolic or exact computation can take orders of magnitude more time than a floating-point evaluation. This becomes more pronounced as elements get more complicated.

⁵ The rank of an element stiffness is discussed in Chapter 19.

⁶ The name **pG** for this variable is used instead of **p** to void confusion with pressure.

⁷ This result can be obtained either by exact integration, or numerical integration with a 2×2 or higher order Gauss rule.

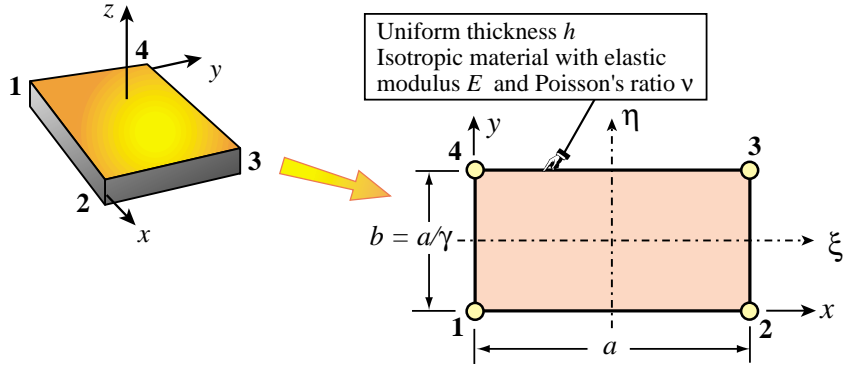


FIGURE 17.9. Rectangular specialization of Quad4 for Example 17.5.

The eigenvalues of this matrix, roughly ordered by ascending value, are

$$\lambda_{1,2,3} = 0, \quad \lambda_4 = S \frac{a^2 + b^2 - a^2\nu}{6ab}, \quad \lambda_5 = S \frac{a^2 + b^2 - b^2\nu}{6ab},$$

$$\lambda_6 = S \frac{a^2 + b^2 - (a^2 + b^2)\nu}{6ab}, \quad \lambda_{7,8} = S \frac{a^2 + b^2 \pm \sqrt{(a^2 - b^2)^2 + 4a^2b^2\nu^2}}{2ab},$$

in which $S = Eh/(1 - \nu^2)$. The three zero eigenvalues correspond to the 3 rigid body plane motions. The other five are guaranteed to be positive if E , a , b , and h are positive, and $0 \leq \nu \leq \frac{1}{2}$.

These results can be verified by the test script given in Exercise 17.2.

§17.5.2. Quad4 Element Body Force Vector

The *Mathematica* module `Quad4IsoPMembBodyForces`, listed in Figure 17.10, computes and returns the consistent node force vector associated with a body force field given for a Quad4 element in plane stress. The module is invoked as

$$\mathbf{Ke} = \text{Quad4IsoPMembBodyForces}[\text{enccor}, \mathbf{Emat}, \text{th}, \text{options}, \text{bfor}] \quad (17.31)$$

Arguments `enccor`, `Emat`, `th`, and `options` are exactly the same as for the stiffness module described in §17.5.1. 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: $\{\mathbf{bx}, \mathbf{by}\}$, where \mathbf{bx} and \mathbf{by} are the body force values (force per unit of volume) along the x and y directions, respectively.

Field varies over the element and is specified in terms of node values. Then `bfor` is a two-dimensional list: $\{\{\mathbf{bx}_1, \mathbf{by}_1\}, \{\mathbf{bx}_2, \mathbf{by}_2\}, \{\mathbf{bx}_3, \mathbf{by}_3\}, \{\mathbf{bx}_4, \mathbf{by}_4\}\}$, in which \mathbf{bx}_i and \mathbf{by}_i denote the body force values along the x and y directions, respectively, at the local node i . The field is interpolated to the Gauss points through the bilinear shape functions.

The module returns `fe` as an force 8-vector with components conjugate to the arrangement (17.28) of nodal displacements.


```

Quad4IsoPMembBodyForces[enccoor_,Emat_,th_,options_,bfor_]:=
Module[{i,k,pG=2,numer=False,h=th,
  bx,by,bx1,by1,bx2,by2,bx3,by3,bx4,by4,bxc,byc,qcoor,
  c,w,Nf,dNx,dNy,Jdet,fe=Table[0,{8}]},
If [Length[options]==2, {numer,pG}=options, {numer}=options];
If [Length[bfor]==2,{bx,by}=bfor;bx1=bx2=bx3=bx4=bx;by1=by2=by3=by4=by];
If [Length[bfor]==4,{bx1,by1},{bx2,by2},{bx3,by3},{bx4,by4}=bfor];
If [pG<1||pG>4, Print["pG out of range"]; Return[Null]];
bxc={bx1,bx2,bx3,bx4}; byc={by1,by2,by3,by4};
For [k=1, k<=pG*pG, k++,
  {qcoor,w}= QuadGaussRuleInfo[{pG,numer},k];
  {Nf,dNx,dNy,Jdet}=Quad4IsoPShapeFunDer[enccoor,qcoor];
  bx=Nf.bxc; by=Nf.byc; If [Length[th]==4, h=th.Nf];
  c=w*Jdet*h;
  bk=Flatten[Table[{Nf[[i]]*bx,Nf[[i]]*by},{i,4}]];
  fe+=c*bk;
]; Return[fe]
];

```

FIGURE 17.10. Module to compute node forces consistent with a given body force field for an isoparametric Quad4 element in plane stress.

```

Quad4IsoPMembStresses[enccoor_,Emat_,th_,options_,udis_]:=
Module[{i,k,numer=False,qcoor,Nf,
  dNx,dNy,Jdet,Be,qctab,ue=udis,sige=Table[0,{4},{3}]},
qctab={{-1,-1},{1,-1},{1,1},{-1,1}};
numer=options[[1]];
If [Length[udis]==4, ue=Flatten[udis]];
For [k=1, k<=Length[sige], k++,
  qcoor=qctab[[k]]; If [numer, qcoor=N[qcoor]];
  {Nf,dNx,dNy,Jdet}=Quad4IsoPShapeFunDer[enccoor,qcoor];
  Be={Flatten[Table[{dNx[[i]], 0},{i,4}]],
    Flatten[Table[{0, dNy[[i]]},{i,4}]],
    Flatten[Table[{dNy[[i]],dNx[[i]]},{i,4}]]};
  sige[[k]]=Emat.(Be.ue);
]; Return[sige]
];

```

FIGURE 17.11. Module to compute nodal stress values for an isoparametric Quad4 element in plane stress, given its node displacements.

§17.5.3. Quad4 Element Stresses

The *Mathematica* module Quad4IsoPMembStresses, listed in Figure 17.11, recovers the nodal stresses for a Quad4 element in plane stress, given its nodal displacements. Initial stresses are assumed to vanish. The module is invoked as

$$Ke = \text{Quad4IsoPMembStresses}[\text{enccoor}, \text{Emat}, \text{th}, \text{options}, \text{udis}] \quad (17.32)$$

Arguments enccoor, Emat, th, and options are exactly the same as for the stiffness module described in §17.5.1. The additional argument is

- udis Element node displacements. These may be provided in either of two formats:
- A flat 8-vector as one-dimensional list: {ux1, uy1, ux2, ... uy4}
 - A node-by-node two-dimensional list: {{ux1, uy1}, ... {ux4, uy4}}

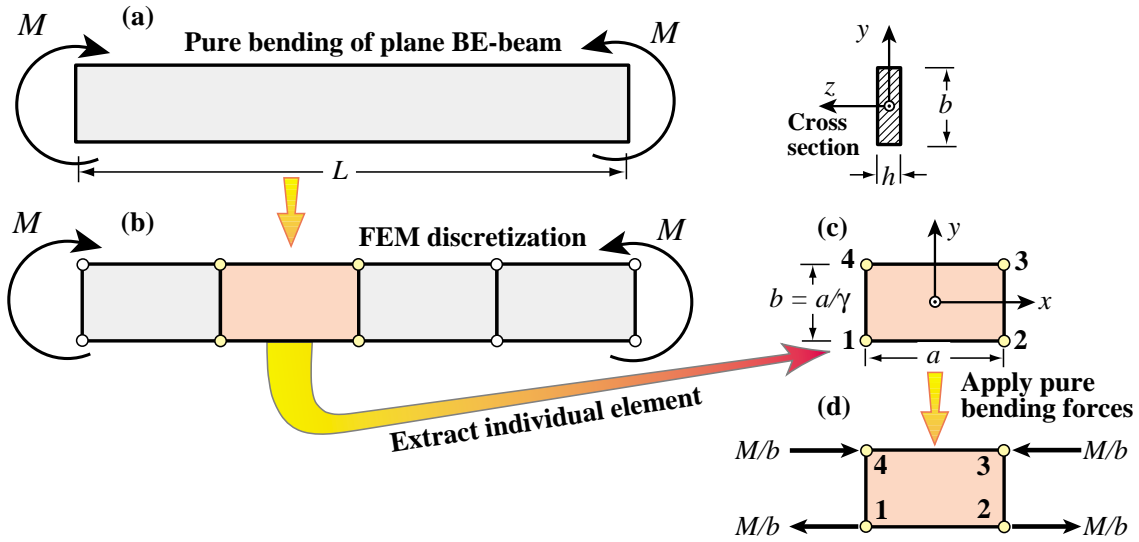


FIGURE 17.12. Pure bending of Bernoulli-Euler plane beam of thin rectangular cross section, to illustrate shear locking. The beam is modeled by one layer of Quad4 plane stress elements through its height.

The module returns `sig`, which is a two-dimensional list storing computed node stresses in the node-by-node arrangement $\{\{\text{sigxx1}, \text{sigyy1}, \text{sigxy1}\}, \dots, \{\text{sigxx4}, \text{sigyy4}, \text{sigxy4}\}\}$.

In the module of Figure 17.11, the nodal stresses are evaluated directly at the node locations. The extrapolation-from-Gauss-points procedure described in Chapter 30 is not implemented in this code.

§17.6. *Quad4 Shear Locking

An inherent deficiency of *low-order, displacement-assumed* plane stress elements was quickly noticed in the late 1960s when applied to problems in which *inplane bending* was dominant (or at least important). Those elements absorbed *parasitic shear energy*, resulting in overstiffness. The phenomenon was given the name *shear locking*. That effect is magnified when the element aspect ratio (the ratio of the element longest to smallest dimension) becomes larger. As a result, the computed deflections (and associated stresses) could be too small by orders of magnitude, leading to unsafe designs. The problem becomes especially acute for mesoscale composite beam analysis, in which elements often have large aspect ratios; e.g., 10 or more.

§17.6.1. *Plane Beam In Pure Bending

To illustrate shear locking and its dependence on element aspect ratio, consider a Bernoulli-Euler prismatic, homogeneous, plane beam of thin rectangular cross-section with span L , height b and thickness h as shown in Figure 17.12(a), and subjected to pure bending under end moments M as illustrated there. The beam is fabricated of isotropic material with elastic modulus E and Poisson's ratio ν . Body forces are assumed to vanish. The *exact* solution of the beam problem is a constant bending moment M along the span. Consequently the beam deforms with uniform curvature $\kappa = M/(EI_{zz})$, in which $I_{zz} = \frac{1}{12}hb^3$ is the cross-section second moment of inertia about z .

The beam is modeled with *one layer* of identical Quad4 plane stress elements through its height. These are rectangles with horizontal dimension a ; in the Figure $a = L/4$. The aspect ratio a/b

is denoted by γ . By analogy with the exact solution, all rectangles in the finite element model will undergo the same deformation. We can therefore isolate a typical element as illustrated in Figure 17.12(b,c).

The exact displacement field for the beam segment referred to the $\{x, y\}$ axes placed at the element center as shown in the bottom of Figure 17.12(c), are

$$u_x = -\kappa xy, \quad u_y = \frac{1}{2}\kappa(x^2 + y^2), \quad (17.33)$$

in which κ is the deformed beam curvature M/EI_{zz} , with $I_{zz} = bh^3/12$. The axial stress is $\sigma_{xx} = E e_{xx} = E (\partial u_x / \partial x) = -E \kappa y$, and is easily verified that all other stresses are zero. Integrating σ_{xx} over the cross reproduces $E I_{zz} \kappa = M$. Since the stresses satisfy all equilibrium equations for zero body force, (17.33) is the exact elasticity solution.

The quadrilateral element will be called *bending exact* in the x direction if it reproduces the solution (17.33) for all $\{\gamma, \nu\}$. This feature can be conveniently established by comparing exact and FEM bending energies, since those are scalar quantities.

The strain energy absorbed by the element if deformed in accordance to the elasticity solution, is⁸

$$U_{exact}^e = \frac{1}{2} \int_0^a M \kappa dx = \frac{M^2}{2E I_{zz}} \int_0^a dx = \frac{6 M^2 a}{b^3 h}. \quad (17.34)$$

If the energy actually absorbed by the element under study is U_{FEM}^e , the ratio $r = U_{FEM}^e / U_{exact}^e$ provides a simple measure of locking. This is the *bending energy ratio*, or BER. If $r < 1$ the element is overflexible whereas if $r > 1$ it is overstiff. If $r \gg 1$ the element is said to *lock*.

§17.6.2. *The Quad4 Bending Energy Ratio

The stiffness equations of the Quad4 rectangular element are given by the closed form expression (17.30) if the material is isotropic. The moment pair of Figure 17.12(a) is applied at the element level by the force system shown in Figure 17.12(d). The strain energy stored under nodal displacements \mathbf{u}_{beam} built by evaluating (17.33) at the nodes 1,2,3,4, is

$$U_{Quad4}^e = \frac{1}{2} (\mathbf{u}_{beam})^T \mathbf{K}^e \mathbf{u}_{beam}. \quad (17.35)$$

To simplify this calculation, it is convenient to decompose that vector as follows:

$$\begin{aligned} \mathbf{u}_{beam} = \mathbf{u}_{beam}^x + \mathbf{u}_{beam}^y &= \frac{1}{4} \kappa a b [-1 \ 0 \ 1 \ 0 \ -1 \ 0 \ 1 \ 0]^T \\ &\quad + \frac{1}{8} \kappa (a^2 + y b^2) [0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1]^T \end{aligned} \quad (17.36)$$

(It should be noted that \mathbf{u}_{beam}^y is a rigid body motion along y , which produces no energy because the element does such motions correctly; thus that term could be dropped.) The calculation carried out in Exercise 17.3 shows that

$$r = r(\gamma, \nu) = \frac{1 + 2/\gamma^2 - \nu}{(2/\gamma^2)(1 - \nu^2)} \quad (17.37)$$

⁸ A reader familiar with the variational principles of continuum mechanics would notice that (17.34) is actually the complementary strain energy, sometimes called “stress energy.”

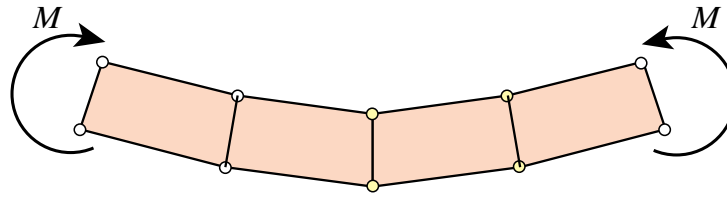


FIGURE 17.13. Deformation of the Quad4 mesh of Figure 17.12(b) under pure bending. Because element sides are forced to remain straight, a spurious shear strain develops inside each element. That shear strain is zero at the element center and maximum at the top and bottom sides.

Evidently $r > 1$ for all γ if $0 \leq \nu \leq \frac{1}{2}$. Moreover if $b \ll a$ so $\gamma \gg 1$, $r \approx \gamma^2/(1 + \nu)$. For example, if $a = 10b$ so $\gamma = 10$, and $\nu = 0$, $r \approx 50$ and the Quad4 model gives only about 2% of the correct solution. This ratio easily overcomes typical safety factors.

In the FEM literature the phenomenon is called *shear locking*, because the over stiffness is due to the FEM mesh motion triggering spurious or parasitic shear energy. A picture is worth 10^3 words: since the Quad4 sides are forced to deform as straight segments, spurious shear strain develops in each element; see Figure 17.13.

Even if we make $a \rightarrow 0$ so $\gamma = b/a \rightarrow 0$ by taking an infinite number of rectangular elements along x , the energy ratio r remains greater than one if $\nu > 0$ since $r \rightarrow 1/(1 - \nu^2)$. Thus the Quad4 model *would not generally converge to the correct solution if we keep one layer through the height*.

§17.6.3. *Shear Unlocking Methods

Several approaches to alleviate shear locking were proposed and tried starting in the mid 1960s:

- Equilibrium stress hybrid models (1964)
- Mixed models based on the Hellinger-Reissner principle (1965)
- Modified Gauss integration rules (1971)
- Injection of incompatible displacement modes (1973)
- Assumed strain elements (1978)
- The Free Formulation (1984)
- Templates (1994)

As discussed in [261], all approaches lead to the same pure-bending-exact Quad4 element if the geometry is rectangular. Only the modified integration method will be covered below because it requires relatively minor changes to the conventional isoparametric formulation. Other methods require fancier mathematical tools, and are better left to more advanced courses.

§17.7. *Modified Gauss Integration

This section title spans a variety of techniques in which the Gauss product integration rules described in §17.3.2 for quadrilaterals are modified to improve element performance. The following terminology will be needed in the sequel.

Reduced integration (RI): a single Gauss rule is used, which is not sufficient to attain the full rank of the element stiffness matrix.⁹

⁹ The topic of element rank sufficiency is covered in detail in Chapter 19

Weighted integration (WI): the stiffness matrix is formed by a combination of two or more rules that use the *same constitutive properties*, but are given weights that depend on element properties.

Selective integration (SI): the stiffness matrix is split into two or more components associated with *different constitutive properties*, with each component integrated by a different rule.

Selective reduced integration (SRI): one of the SI rules is rank-deficient when applied by itself.

The specific application below will be mitigating or eliminating shear locking in Quad4 elements using WI and SRI. It should be noted, however, that the foregoing methods are also widely used to treat other defects, such as distortion sensitivity, as well as the so-called “pressure locking” that affects 3D elements with incompressible or near-incompressible material.

§17.7.1. *Shear Unlocking By WI

For the Quad4 element, weighted integration (WI) is done by combining the two stiffness matrices: $\mathbf{K}_{1 \times 1}^e$ and $\mathbf{K}_{2 \times 2}^e$ produced by the 1×1 and 2×2 Gauss product rules, respectively:

$$\mathbf{K}_\beta^e = (1 - \beta)\mathbf{K}_{1 \times 1}^e + \beta\mathbf{K}_{2 \times 2}^e. \quad (17.38)$$

Here β is a scalar in the range $[0, 1]$. If $\beta = 0$ or $\beta = 1$ one recovers the element integrated by the 1×1 or 2×2 rule, respectively.

The idea behind (§17.7.1) is that $\mathbf{K}_{1 \times 1}^e$ is rank-deficient and too soft whereas $\mathbf{K}_{2 \times 2}^e$ is rank-sufficient but too stiff. A Goldilocks compromise of too-soft and too-stiff hopefully “balances” the stiffness. The analysis of Exercise 17.4 for an isotropic rectangular Quad4 shows that the choice

$$\beta = \frac{2/\gamma^2(1 - \nu^2)}{1 + 2/\gamma^2 - \nu}, \quad (17.39)$$

produces a pure-bending-exact rectangle in the x -direction. This result, however, has limited practical use for three reasons: (1) it is not extendible to anisotropic materials, (2) exactness in the y -direction is not met unless $\gamma = 1$, and (3) extension to non-rectangular shapes is not obvious.

Remark 17.8. For programming, the combination (17.38) may be viewed as a 5-point integration rule with weights $w_1 = 4(1 - \beta)$ at the sample point with $\xi = \eta = 0$ and $w_i = \beta$ ($i = 2, 3, 4, 5$) at the four sample points with $\xi = \pm 1/\sqrt{3}$, $\eta = \pm 1/\sqrt{3}$. This differs from ordinary Gauss rules in that the weights are no longer problem independent.

§17.7.2. *Shear Unlocking By SRI: Isotropic Material

The SRI unlocking approach is more powerful than WI since it may cover anisotropic material and produces exactness in both x and y directions. Split the plane stress elasticity matrix \mathbf{E} into two:

$$\mathbf{E} = \mathbf{E}_I + \mathbf{E}_{II} \quad (17.40)$$

Inserting (17.40) into (17.29) the expression of the stiffness matrix becomes

$$\mathbf{K}^e = \int_{\Omega^e} h \mathbf{B}^T \mathbf{E}_I \mathbf{B} d\Omega^e + \int_{\Omega^e} h \mathbf{B}^T \mathbf{E}_{II} \mathbf{B} d\Omega^e = \mathbf{K}_I^e + \mathbf{K}_{II}^e. \quad (17.41)$$

If these were done through the same integration rule, the stiffness would be identical to that obtained by integrating $h \mathbf{B}^T \mathbf{E} \mathbf{B} d\Omega^e$ because $\mathbf{B}^T \mathbf{E}_I \mathbf{B} + \mathbf{B}^T \mathbf{E}_{II} \mathbf{B} = \mathbf{B}^T \mathbf{E} \mathbf{B}$. The trick is to use two different rules: rule (I) for the first integral and rule (II) for the second one.

For Quad4, rules (I) and (II) are the 1×1 and 2×2 Gauss product rules, respectively. However the latter is generalized so the sample points are located at $\{-\chi, \chi\}$, $\{\chi, -\chi\}$, $\{\chi, \chi\}$ and $\{-\chi, \chi\}$, with weight 1.¹⁰ The elasticity matrix splitting is

$$\mathbf{E} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} \alpha & \beta & 0 \\ \beta & \alpha & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} + \frac{E}{1-\nu^2} \begin{bmatrix} 1-\alpha & \nu-\beta & 0 \\ \nu-\beta & 1-\alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{E}_I + \mathbf{E}_{II}, \quad (17.42)$$

where α and β are scalars. Exercise 17.4 shows that if

$$\chi = \sqrt{\frac{1-\nu^2}{3(1-\alpha)}} \quad (17.43)$$

the resulting element stiffness $\mathbf{K}_I^e + \mathbf{K}_{II}^e$ is bending exact for any $\{\alpha, \beta\}$. In particular, if one takes $\alpha = \nu^2$, which corresponds to the splitting

$$\mathbf{E} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} \nu^2 & \beta & 0 \\ \beta & \nu^2 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} + \frac{E}{1-\nu^2} \begin{bmatrix} 1-\nu^2 & \nu-\beta & 0 \\ \nu-\beta & 1-\nu^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{E}_I + \mathbf{E}_{II}, \quad (17.44)$$

then $\chi = 1/\sqrt{3}$ and rule (II) becomes the standard 2×2 Gauss product rule. The work is minimized by taking $\beta = \nu$ since if so \mathbf{E}_{II} , which is used at 4 of 5 points, becomes diagonal.

§17.7.3. *Shear Unlocking By SRI: Anisotropic Material

The extension to anisotropic material requires a computer algebra system (CAS) to be tractable. It is worked out in Exercise 17.6. Assume that \mathbf{E} is given by (17.26). The appropriate splitting is

$$\mathbf{E} = \mathbf{E}_I + \mathbf{E}_{II} = \begin{bmatrix} E_{11}\alpha_1 & E_{12}\beta & E_{13} \\ E_{12}\beta & E_{22}\alpha_2 & E_{23} \\ E_{13} & E_{23} & E_{33} \end{bmatrix} + \begin{bmatrix} E_{11}(1-\alpha_1) & E_{12}(1-\beta) & 0 \\ E_{12}(1-\beta) & E_{22}(1-\alpha_2) & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (17.45)$$

in which β is arbitrary and

$$\begin{aligned} 1 - \alpha_1 &= \frac{|\mathbf{E}|}{3\chi^2 E_{11}(E_{22}E_{33} - E_{23}^2)} = \frac{1}{3\chi^2 C_{11}}, & 1 - \alpha_2 &= \frac{|\mathbf{E}|}{3\chi^2 E_{22}(E_{11}E_{33} - E_{13}^2)} = \frac{1}{3\chi^2 C_{22}}, \\ |\mathbf{E}| &= \det(\mathbf{E}) = E_{11}E_{22}E_{33} + 2E_{12}E_{13}E_{23} - E_{11}E_{23}^2 - E_{22}E_{13}^2 - E_{33}E_{12}^2, \\ C_{11} &= E_{11}(E_{22}E_{33} - E_{23}^2)/|\mathbf{E}|, & C_{22} &= E_{22}(E_{11}E_{33} - E_{13}^2)/|\mathbf{E}|. \end{aligned} \quad (17.46)$$

in which C_{ij} denote the entries of the compliance matrix $\mathbf{C} = \mathbf{E}^{-1}$. The resulting rectangular element is bending exact for any \mathbf{E} and $\chi \neq 0$. In practice one would select $\chi = 1/\sqrt{3}$.

¹⁰ For a rectangular geometry these sample points lie on the diagonals. In the case of the standard 2-point Gauss product rule $\chi = 1/\sqrt{3}$.

Notes and Bibliography

The 4-node quadrilateral — which is the focus of this Chapter — has a checkered history (pun intended). It was first derived as a *rectangular* panel with edge reinforcements (not included here) by Argyris in his 1954 *Aircraft Engineering* series [26]; see p. 49 in the Butterworths reprint. Argyris used bilinear displacement interpolation in Cartesian coordinates.¹¹

After much flailing, a conforming generalization to arbitrary geometry was published in 1964 by Taig and Kerr [793] using quadrilateral-fitted coordinates already denoted as $\{\xi, \eta\}$ but running from 0 to 1. (Reference [793] cites an 1961 English Electric Aircraft internal report as original source but [437, p. 520] remarks that the work goes back to 1957.) Bruce Irons, who was aware of Taig’s work while at Rolls Royce, changed the $\{\xi, \eta\}$ range to $[-1, 1]$ to fit Gauss quadrature tables. He proceeded to create the seminal isoparametric family as a far-reaching extension upon moving to Swansea [69,211,432,433,434].

Gauss integration is also called Gauss-Legendre quadrature. Gauss presented these rules, derived from first principles, in 1814; cf. [334, §4.11]. Legendre’s name is often adjoined because the 1D sample point abscissas turned out to be zeros of Legendre polynomials. A systematic collection of all rules known until 1965 is given in [780]. Several of the non-product Gauss rules in Table 17.2 will be found there although without the extra flexibility of user-specified abscissas and weights for parameter studies. For additional references in multidimensional numerical integration, see **Notes and Bibliography** in Chapter 24.

The “unlocking” techniques enumerated in §17.6.3 appeared in the following chronological order. Equilibrium stress hybrid models in [639], mixed models in [389], (although the focus there was on incompressible materials), modified integration almost simultaneously in [634,905], incompatible displacement modes in [881], assumed strains in [503]; the Free Formulation in [92], and templates in [246]. An excellent textbook source for several of these methods is [424].

References

Referenced items have been moved to Appendix R.

¹¹ This work is probably the first derivation of a continuum-based finite element by assumed displacements. As noted in Appendix O, Argyris was aware of the ongoing work in stiffness methods at Turner’s group in Boeing, but the plane stress models presented in [834] were derived by interelement flux assumptions. Argyris used the unit displacement theorem, displacing each DOF in turn by one. The resulting displacement pattern is now called a shape function.

Homework Exercises for Chapter 17

Isoparametric Quadrilaterals

EXERCISE 17.1 [C:20] Exercise the *Mathematica* module of Figure 17.8 with the following script:

```
ClearAll[Em,v,a,b,h,γ]; b=a/γ;
ncoor={{0,0},{a,0},{a,b},{0,b}};
Emat=Em/(1-v^2)*{{1,v,0},{v,1,0},{0,0,(1-v)/2}};
Ke= Quad4IsoPMembStiffness[ncoor,Emat,h,{False,2}];
scaledKe=Simplify[Ke*(24*(1-v^2)*γ/(Em*h))];
Print["Ke=",Em*h/(24*γ*(1-v^2)), "\n", scaledKe//MatrixForm];
```

FIGURE E17.1. Script for Exercise 17.1.

Verify that for integration rules $p=2,3,4$ the stiffness matrix does not change and has three zero eigenvalues, which correspond to the three two-dimensional rigid body modes. On the other hand, for $p=1$ the stiffness matrix is different and displays five zero eigenvalues, which is physically incorrect. (This phenomenon is analyzed further in Chapter 19.) Question: why does the stiffness matrix stay exactly the same for $p \geq 2$? Hint: take a look at the entries of the integrand $h \mathbf{B}^T \mathbf{E} \mathbf{B} J$; for a *rectangular geometry* are those polynomials in ξ and η , or rational functions? If the former, of what polynomial order in ξ and η are the entries?

EXERCISE 17.2 [C:20] Check the rectangular element stiffness closed form given in Example 17.5. This may be done by hand (takes a while) or (quicker) running the following script, which calls the *Mathematica* module of Figure 17.8:

```
ClearAll[Em,v,a,b,h,γ]; b=a/γ;
ncoor={{0,0},{a,0},{a,b},{0,b}};
Emat=Em/(1-v^2)*{{1,v,0},{v,1,0},{0,0,(1-v)/2}};
Ke= Quad4IsoPMembStiffness[ncoor,Emat,h,{False,2}];
scaledKe=Simplify[Ke*(24*(1-v^2)*γ/(Em*h))];
Print["Ke=",Em*h/(24*γ*(1-v^2)), "\n", scaledKe//MatrixForm];
```

FIGURE E17.2. Script suggested for Exercise 17.2.

The scaling introduced in the last two lines is for matrix visualization convenience. Verify (17.30) by printout inspection and report any typos to instructor.

EXERCISE 17.3 [A/C:20] Verify the Quad4 bending energy ratio (17.37) obtained in the analysis of §17.6.1.

EXERCISE 17.4 [A+C:25] Verify the WI shear unlocking result (17.39) of §17.7.1 for a rectangular Quad4 element of isotropic material.

EXERCISE 17.5 [A+C:30] (Advanced) Verify the SRI shear unlocking results (17.42) in §17.7.2 for a rectangular Quad4 element of isotropic material.

EXERCISE 17.6 [A+C:40] (Advanced, research paper level, requires a CAS to be tractable) Verify the SRI shear unlocking results (17.42) in §17.7.3 for a rectangular Quad4 element of arbitrary anisotropic material.