17

Isoparametric Quadrilaterals

TABLE OF CONTENTS

§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 elements. 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. The implementation of the algorithm in the form of element modules is partly explained in the Exercises of this Chapter, and covered more systematically 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:

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

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):

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

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:

$$\mathbf{J}^{-1}\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial\xi}\dfrac{\partial\xi}{\partial x} + \dfrac{\partial x}{\partial\eta}\dfrac{\partial\eta}{\partial x} & \dfrac{\partial y}{\partial\xi}\dfrac{\partial\xi}{\partial x} + \dfrac{\partial y}{\partial\eta}\dfrac{\partial\eta}{\partial x} \\ \dfrac{\partial x}{\partial\xi}\dfrac{\partial\xi}{\partial y} + \dfrac{\partial x}{\partial\eta}\dfrac{\partial\eta}{\partial y} & \dfrac{\partial y}{\partial\xi}\dfrac{\partial\xi}{\partial y} + \dfrac{\partial y}{\partial\eta}\dfrac{\partial\eta}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial x} & \dfrac{\partial y}{\partial x} \\ \dfrac{\partial x}{\partial y} & \dfrac{\partial y}{\partial y} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{17.3}$$

where 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 §16.5.1. The derivatives with respect to x and y are given by the chain rule:

$$\frac{\partial N_i^e}{\partial x} = \frac{\partial N_i^e}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial N_i^e}{\partial\eta}\frac{\partial\eta}{\partial x}, \qquad \frac{\partial N_i^e}{\partial y} = \frac{\partial N_i^e}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial N_i^e}{\partial\eta}\frac{\partial\eta}{\partial y}. \tag{17.4}$$

This can be put in matrix form as

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

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

§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:

$$x = \sum_{i=1}^{n} x_i N_i^e, \qquad y = \sum_{i=1}^{n} y_i N_i^e. \tag{17.6}$$

Differentiating with respect to the quadrilateral coordinates,

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

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

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

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.

Remark 17.3. The symbolic inversion of $\mathbf{J}$ for arbitrary ξ, η in general leads to extremely complicated expressions unless the element has a particularly simple geometry, (for example rectangles as in Exercises 17.1–17.3). This was one of the difficulties that motivated the use of Gaussian numerical quadrature, as discussed in §17.3 below.

§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:

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

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

```
Quad4IsoPShapeFunDer[ncoor_,qcoor_]:= Module[
  {Nf,dNx,dNy,dNξ,dNη,i,J11,J12,J21,J22,Jdet,ξ,η,x,y},
  {ξ,η}=qcoor;
   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=Table[ncoor[[i,1]],{i,4}]; y=Table[ncoor[[i,2]],{i,4}];
   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}]
];
```

FIGURE 17.1. A shape function module for the 4-node bilinear quadrilateral.

§17.2.5. *A Shape Function Implementation

To make the foregoing discussion more specific, Figure 17.1 shows the *shape function module* for the 4-node bilinear quadrilateral. This is a code fragment that returns the value of the shape functions and their $\{x, y\}$ derivatives at a given point of quadrilateral coordinates $\{\xi, \eta\}$. The module is invoked by saying

```
{Nf,Nfx,Nfy,Jdet}=Quad4IsoPShapeFunDer[ncoor,qcoor]
```
(17.10)

where the arguments are

`ncoor`	Quadrilateral node coordinates arranged in two-dimensional list form: `{{x1,y1},{x2,y2},{x3,y3},{x4,y4}}`.
`qcoor`	Quadrilateral coordinates $\{\xi,\ \eta\}$ of the point.

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.

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 `ncoor={{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 appropiate call is

```
{Nf,Nfx,Nfy,Jdet}=Quad4IsoPShapeFunDer[ncoor,{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`.

§17.3. Numerical Integration by Gauss Rules

Numerical integration is essential for practical evaluation of integrals over isoparametric element domains. 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 economy is important for efficient element calculations, since a *matrix product* is evaluated at each sample point. The fact that the location of the sample points in Gauss rules is usually given by non-rational numbers is of no concern in digital computation.

§17.3.1. One Dimensional Rules

The classical Gauss integration rules are defined by

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

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 abcissae in the interval $[-1,1]$. The use of the canonical interval $[-1,1]$ is no restriction, because an integral over another range, say from a to b, can be

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 $w_{35} = 512/900$.

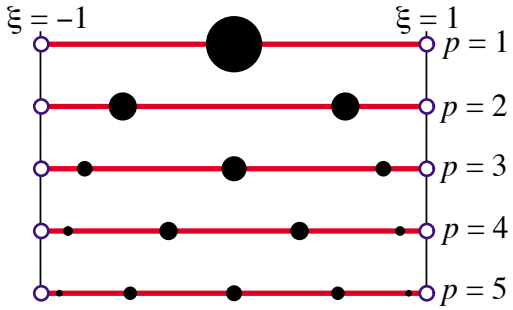

FIGURE 17.2. The first five one-dimensional Gauss rules $p = 1, 2, 3, 4, 5$ depicted over the line segment $\xi \in [-1, +1]$. Sample point locations are marked with black circles. The radii of those circles are proportional to the integration weights.

transformed to $[-1, +1]$ via a simple linear transformation of the independent variable, as shown in the Remark below.

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

Remark 17.4. A more general 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 $J = dx/d\xi = 1/2\ell$. Thus

$$\int_a^b F(x)\,dx = \int_{-1}^{1} F(\xi)\,J\,d\xi = \int_{-1}^{1} F(\xi)\,\tfrac{1}{2}\ell\,d\xi. \tag{17.12}$$

```
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,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 17.3. A *Mathematica* module that returns the first five one-dimensional Gauss rules.

Remark 17.5. Higher order Gauss rules are tabulated in standard manuals for numerical computation. For example, the widely used Handbook of Mathematical Functions [2] lists (in Table 25.4) rules with up to 96 points. 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. Implementation of 1D Rules

The *Mathematica* module shown 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 $1 \le i \le p$ and $p = 1, 2, 3, 4, 5$, call the module as

```
{xii,wi}=LineGaussRuleInfo[{p,numer},i]
```  (17.13)

Logical flag `numer` is `True` to get numerical (floating-point) information, or `False` to get exact information. The module returns the sample point abcissa ξ_i in `xii` and the weight w_i in `wi`. If `p` is not in the implemented range 1 through 5, the module returns `{Null,0}`.

Example 17.2. `{xi,w}=LineGaussRuleInfo[{3,False},2]` returns `xi=0` and `w=8/9`, whereas `{xi,w}=LineGaussRuleInfo[{3,True},2]` returns (to 16 places) `xi=0.` and `w=0.888888888888889`.

§17.3.3. Two Dimensional Rules

The simplest two-dimensional Gauss rules are called *product rules*. They are obtained by applying the one-dimensional rules to each independent variable in turn. To apply these rules we must first reduce the integrand to the canonical form:

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

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

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

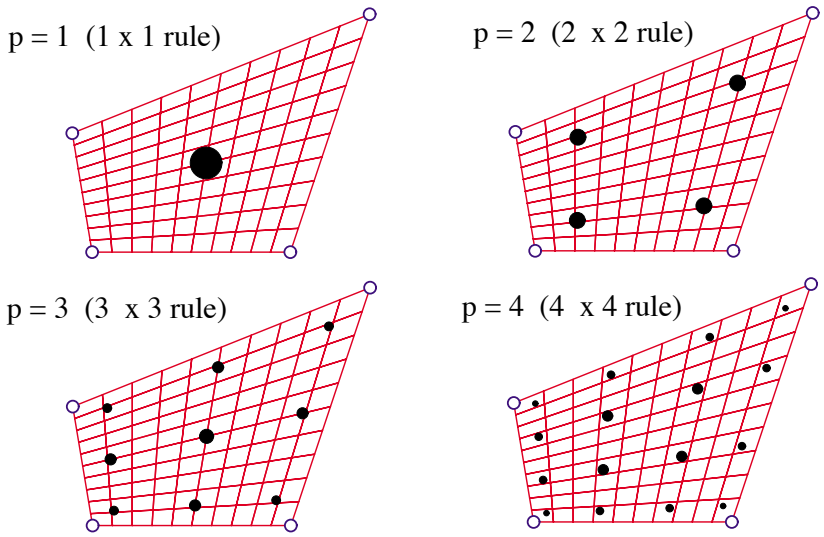

FIGURE 17.4. The first four two-dimensional Gauss product rules $p = 1, 2, 3, 4$ depicted over a straight-sided quadrilateral region. Sample points are marked with black circles. The areas of these circles are proportional to the integration weights.

where p_1 and p_2 are 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 two-dimensional Gauss product rules with $p = p_1 = p_2$ are illustrated in Figure 17.4.

§17.3.4. Implementation of 2D Gauss Rules

The *Mathematica* module listed in Figure 17.5 implements two-dimensional product Gauss rules having 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 of points p in both directions the module is called in either of two ways:

```
{{xii,etaj},wij}=QuadGaussRuleInfo[{p, numer}, {i,j}]
{{xii,etaj},wij}=QuadGaussRuleInfo[{p, numer},k ]
```
(17.16)

The first form is used to get information for point $\{i, j\}$ of the $p \times p$ rule, in which $1 \le i \le p$ and $1 \le j \le p$. 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[1] as `j=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:

```
{{xii,etaj},wij}=QuadGaussRuleInfo[{{p1,p2}, numer},{i,j}]
{{xii,etaj},wij}=QuadGaussRuleInfo[{{p1,p2}, numer},k ]
```
(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=Floor[(k-1)/p1]+1;`

[1] Indices i and j are denoted by `i1` and `i2`, respectively, inside the module.

```
QuadGaussRuleInfo[{rule_,numer_},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,numer},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,numer},i];
  {η,w2}=  LineGaussRuleInfo[{p2,numer},j];
  info={{ξ,η},w1*w2};
  If [numer, Return[N[info]], Return[Simplify[info]]];
];
```

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

`i=k-p1*(i-1)`. In all four forms, logical flag `numer` is set to `True` if numerical information is desired and to `False` if exact information is desired.

The module returns ξ_i and η_j in `xii` and `etaj`, respectively, and the weight product $w_i w_j$ in `wij`. This code is used in the Exercises at the end of the chapter. If the inputs are not in range, the module returns `{{Null,Null},0}`.

Example 17.3. `{{xi,eta},w}=QuadGaussRuleInfo[{3,False},{2,3}]` returns `xi=0`, `eta=Sqrt[3/5]` and `w=40/81`.

Example 17.4. `{{xi,eta},w}=QuadGaussRuleInfo[{3,True},{2,3}]` returns (to 16-place precision) `xi=0.`, `eta=0.7745966692414834` and `w=0.49382716049382713`.

§17.4. The Stiffness Matrix

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

$$\mathbf{K}^e = \int_{\Omega^e} h\, \mathbf{B}^T \mathbf{E} \mathbf{B}\, d\Omega^e \tag{17.18}$$

Of the terms that appear in (17.18) the strain-displacement matrix $\mathbf{B}$ has been discussed previously. The thickness h, if variable, may be interpolated via the shape functions. The stress-strain matrix $\mathbf{E}$ is usually constant in elastic problems, but we could in principle interpolate it as appropriate should it vary over the element. To integrate (17.18) numerically by a two-dimensional product Gauss rule, we have to reduce it to the canonical form (17.14), that is

$$\mathbf{K}^e = \int_{-1}^{1} \int_{-1}^{1} \mathbf{F}(\xi, \eta)\, d\xi\, d\eta. \tag{17.19}$$

If ξ and η are the quadrilateral coordinates, everything in (17.19) already fits this form, except the element of area $d\Omega^e$.

To complete the reduction we need to express $d\Omega^e$ in terms of the differentials $d\xi$ and $d\eta$. The desired relation is (see Remark below)

$$d\Omega^e = dx\, dy = \det \mathbf{J}\; d\xi\, d\eta = J\, d\xi\, d\eta. \tag{17.20}$$

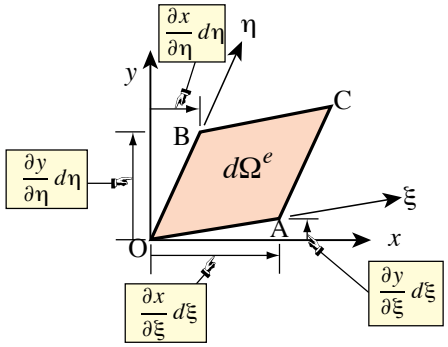

FIGURE 17.6. Geometric interpretation of the Jacobian-determinant formula.

We therefore have

$$\mathbf{F}(\xi, \eta) = h\, \mathbf{B}^T \mathbf{E} \mathbf{B}\, \det \mathbf{J}. \tag{17.21}$$

This matrix function can be numerically integrated over the domain $-1 \le \xi \le +1, -1 \le \eta \le +1$ by an appropriate Gauss product rule.

Remark 17.6. To geometrically justify the area transformation formula (17.20), consider the element of area OACB depicted in Figure 17.6. The area of this differential parallelogram can be computed as

$$\begin{aligned} dA = \vec{OB} \times \vec{OA} &= \frac{\partial x}{\partial \xi} d\xi \, \frac{\partial y}{\partial \eta} d\eta - \frac{\partial x}{\partial \eta} d\eta \, \frac{\partial y}{\partial \xi} d\xi \\ &= \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} d\xi \, d\eta = |\mathbf{J}|\, d\xi \, d\eta = \det \mathbf{J}\, d\xi \, d\eta. \end{aligned} \tag{17.22}$$

This formula can be extended to any number of dimensions, as shown in textbooks on differential geometry; for example [265,319,708].

§17.5. *Integration Variants

Several deviations from the standard integration schemes described in the foregoing sections are found in the FEM literature. Two variations are described below and supplemented with motivation Exercises.

§17.5.1. *Weighted Integration

It is sometimes useful to form the element stiffness as a linear combination of stiffnesses produced by two different integration rules Such schemes are known as *weighted integration* methods. They are distinguished from the selective-integration schemes described in the next subsection in that the constitutive properties are not modified.

For the 4-node bilinear element weighted integration is done by combining the stiffnesses $\mathbf{K}^e_{1\times1}$ and $\mathbf{K}^e_{2\times2}$ produced by 1×1 and 2×2 Gauss product rules, respectively:

$$\mathbf{K}^e_\beta = (1-\beta)\mathbf{K}^e_{1\times1} + \beta \mathbf{K}^e_{2\times2}. \tag{17.23}$$

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.[2]

[2] For programming the combination (17.23) may be regarded as a 5-point integration rule with weights $w_1 = 4(1-\beta)$ at the sample point at $\xi = \eta = 0$ and $w_i = \beta$ $(i = 2, 3, 4, 5)$ at the four sample points at $\xi = \pm 1/\sqrt{3}, \eta = \pm 1/\sqrt{3}$.

The idea behind (17.23) is that $\mathbf{K}^e_{1\times 1}$ is rank-deficient and too soft whereas $\mathbf{K}^e_{2\times 2}$ is rank-sufficient but too stiff. A combination of too-soft and too-stiff hopefully "balances" the stiffness. An application of this idea to the mitigation of *shear locking* for modeling in-plane bending is the subject of Exercise E17.4.

§17.5.2. *Selective Integration

In the FEM literature the term *selective integration* is used to described a scheme for forming $\mathbf{K}^e$ as the sum of two or more matrices computed with different integration rules *and* different constitutive properties.[3] We consider here the case of a two-way decomposition. Split the plane stress constitutive matrix $\mathbf{E}$ into two:

$$\mathbf{E} = \mathbf{E}_{\rm I} + \mathbf{E}_{\rm II} \tag{17.24}$$

This is called a *stress-strain splitting*. Inserting (17.24) into (17.13) the expression of the stiffness matrix becomes

$$\mathbf{K}^e = \int_{\Omega^e} h\,\mathbf{B}^T\mathbf{E}_{\rm I}\mathbf{B}\,d\Omega^e + \int_{\Omega^e} h\,\mathbf{B}^T\mathbf{E}_{\rm II}\mathbf{B}\,d\Omega^e = \mathbf{K}^e_{\rm I} + \mathbf{K}^e_{\rm II}. \tag{17.25}$$

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

In practice selective integration is mostly useful for the 4-node bilinear quadrilateral. For this element rules (I) and (II) are the 1×1 and 2×2 Gauss product rules, respectively. Exercises E17.5–7 investigate stress-strain splittings (17.24) that improve the in-plane bending performance of rectangular elements.

Notes and Bibliography

The 4-node quadrilateral has a checkered history. It was first derived as a rectangular panel with edge reinforcements (not included here) by Argyris in his 1954 *Aircraft Engineering* series [22, p. 49 in the Butterworths reprint]. Argyris used bilinear displacement interpolation in Cartesian coordinates.[4]

After much flailing, a conforming generalization to arbitrary geometry was published in 1964 by Taig and Kerr [719] using quadrilateral-fitted coordinates already denoted as $\{\xi, \eta\}$ but running from 0 to 1. (Reference [719] cites an 1961 English Electric Aircraft internal report as original source but [397, 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 [64,197,394,397].

Gauss integration is also called Gauss-Legendre quadrature. Gauss presented these rules, derived from first principles, in 1814; cf. Sec 4.11 of [310]. Legendre's name is often adjoined because the abcissas of the 1D sample points turned out to be the zeros of Legendre polynomials. A systematic description is given in [706]. For references in multidimensional numerical integration, see **Notes and Bibliography** in Chapter 24.

Selective and reduced integration in FEM developed in the early 1970s, and by now there is a huge literature. An excellent textbook source is [385].

References

Referenced items have been moved to Appendix R.

---

[3] This technique is also called "selective reduced integration" to reflect the fact that one of the rules (the "reduced rule") underintegrates the element.

[4] This work is probably the first derivation of a continuum-based finite element by assumed displacements. As noted in §1.7.1, Argyris was aware of the ongoing work in stiffness methods at Turner's group in Boeing, but the plane stress models presented in [758] 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

The *Mathematica* module `Quad4IsoPMembraneStiffness` listed in Figure E17.1 computes the element stiffness matrix of the 4-node bilinear quadrilateral. This module is useful as a tool for the Exercises that follow.

```
Quad4IsoPMembraneStiffness[ncoor_,Emat_,th_,options_]:=
  Module[{i,k,p=2,numer=False,h=th,qcoor,c,w,Nf,
    dNx,dNy,Jdet,Be,Ke=Table[0,{8},{8}]},
  If [Length[options]==2, {numer,p}=options,{numer}=options];
  If [p<1||p>4, Print["p out of range"]; Return[Null]];
  For [k=1, k<=p*p, k++,
       {qcoor,w}= QuadGaussRuleInfo[{p,numer},k];
       {Nf,dNx,dNy,Jdet}=Quad4IsoPShapeFunDer[ncoor,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]]
   ];
```

FIGURE E17.1. Mathematica module to compute the stiffness matrix of a 4-node bilinear quadrilateral in plane stress.

The module makes use of the shape function module `Quad4IsoPShapeFunDer` listed in Figure 17.1, and of the Gauss integration modules `QuadGaussRuleInfo` and (indirectly) `LineGaussRuleInfo`, listed in Figures 17.5 and are included in the web-posted Notebook `Quad4Stiffness.nb`.[5] The module is invoked as

```
Ke=Quad4IsoPMembraneStiffness[ncoor,Emat,thick,options]
```
(E17.1)

The arguments are:

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

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

$$\mathbf{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{12} & E_{22} & E_{23} \\ E_{13} & E_{23} & E_{33} \end{bmatrix} \tag{E17.2}$$

arranged as `{{E11,E12,E33},{E12,E22,E23},{E13,E23,E33}}`. Must be symmetric. If the material is isotropic with elastic modulus E and Poisson's ratio ν, this matrix becomes

$$\mathbf{E} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\nu) \end{bmatrix} \tag{E17.3}$$

`thick` The plate thickness specified either as a four-entry list: `{h1,h2,h3,h4}` or as a scalar: `h`.

[5] This Notebook does not include scripts for doing the Exercises below, although it has some text statements at the bottom of the cell. You will need to enter the Exercise scripts yourself.

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

`options` Processing options. This list may contain two items: {`numer,p`} or one: {`numer`}.

`numer` is a logical flag with value `True` or `False`. If `True`, the computations are done in floating point arithmetic. For symbolic or exact arithmetic work set `numer` to `False`.[6]

`p` specifies the Gauss product rule to have `p` points in each direction. `p` may be 1 through 4. For rank sufficiency, `p` must be 2 or higher. If `p` is 1 the element will be rank deficient by two.[7] If omitted `p` = 2 is assumed.

The module returns `Ke` as an 8×8 symmetric matrix pertaining to the following arrangement of nodal displacements:

$$\mathbf{u}^e = [\, u_{x1} \quad u_{y1} \quad u_{x2} \quad u_{y2} \quad u_{x3} \quad u_{y3} \quad u_{x4} \quad u_{y4} \,]^T . \tag{E17.4}$$

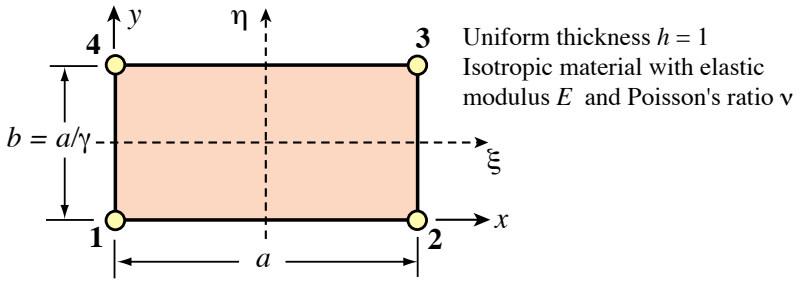

FIGURE E17.2. Element for Exercises 17.1 to 17.3.

For the following three exercises we 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 depicted in Figure E17.2. The element has uniform unit thickness h. The material is isotropic with elastic modulus E and Poisson's ratio ν and consequently $\mathbf{E}$ reduces to (E17.3). The stiffness matrix of this element can be expressed in closed form.[8] For convenience define $\gamma = a/b$ (rectangle aspect ratio), $\psi_1 = (1+\nu)\gamma$, $\psi_2 = (1-3\nu)\gamma$, $\psi_3 = 2 + (1-\nu)\gamma^2$, $\psi_4 = 2\gamma^2 + (1-\nu)$, $\psi_5 = (1-\nu)\gamma^2 - 4$, $\psi_6 = (1-\nu)\gamma^2 - 1$, $\psi_7 = 4\gamma^2 - (1-\nu)$ and $\psi_8 = \gamma^2 - (1-\nu)$. Then the stiffness matrix in closed form is

$$\mathbf{K}^e = \frac{Eh}{24\gamma(1-\nu^2)} \begin{bmatrix} 4\psi_3 & 3\psi_1 & 2\psi_5 & -3\psi_2 & -2\psi_3 & -3\psi_1 & -4\psi_6 & 3\psi_2 \\ & 4\psi_4 & 3\psi_2 & 4\psi_8 & -3\psi_1 & -2\psi_4 & -3\psi_2 & -2\psi_7 \\ & & 4\psi_3 & -3\psi_1 & -4\psi_6 & -3\psi_2 & -2\psi_3 & 3\psi_1 \\ & & & 4\psi_4 & 3\psi_2 & -2\psi_7 & 3\psi_1 & -2\psi_4 \\ & & & & 4\psi_3 & 3\psi_1 & 2\psi_5 & -3\psi_2 \\ & & & & & 4\psi_4 & 3\psi_2 & 4\psi_8 \\ & & & & & & 4\psi_3 & -3\psi_1 \\ \text{symm} & & & & & & & 4\psi_4 \end{bmatrix} . \tag{E17.5}$$

[6] 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.

[7] The rank of an element stiffness is discussed in Chapter 19.

[8] This closed form can be obtained by either exact integration, or numerical integration with a 2×2 or higher Gauss rule.

EXERCISE 17.1 [C:20] Exercise the *Mathematica* module of Figure E17.1 with the following script:

```
ClearAll[Em,nu,a,b,h];  Em=48; h=1; a=4; b=2; nu=0;
ncoor={{0,0},{a,0},{a,b},{0,b}};
Emat=Em/(1-nu^2)*{{1,nu,0},{nu,1,0},{0,0,(1-nu)/2}};
For [p=1, p<=4, p++,
     Ke= Quad4IsoPMembraneStiffness[ncoor,Emat,h,{True,p}];
     Print["Gauss integration rule: ",p," x ",p];
     Print["Ke=",Chop[Ke]//MatrixForm];
     Print["Eigenvalues of Ke=",Chop[Eigenvalues[N[Ke]]]]
];
```

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 stays 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 (E17.5). This may be done by hand (takes a while) or (quicker) running the script of Figure E17.3, which calls the *Mathematica* module of Figure E17.1.

```
ClearAll[Em,ν,a,b,h,γ];  b=a/γ;
ncoor={{0,0},{a,0},{a,b},{0,b}};
Emat=Em/(1-ν^2)*{{1,ν,0},{ν,1,0},{0,0,(1-ν)/2}};
Ke= Quad4IsoPMembraneStiffness[ncoor,Emat,h,{False,2}];
scaledKe=Simplify[Ke*(24*(1-ν^2)*γ/(Em*h))];
Print["Ke=",Em*h/(24*γ*(1-ν^2)),"*\n",scaledKe//MatrixForm];
```

FIGURE E17.3. Script suggested for Exercise E17.2.

The scaling introduced in the last two lines is for matrix visualization convenience. Verify (E17.5) by printout inspection and report any typos to instructor.

EXERCISE 17.3 [A/C:25=5+10+10] A Bernoulli-Euler plane beam of thin rectangular cross-section with span L, height b and thickness h (normal to the plane of the figure) is bent under end moments M as illustrated in Figure E17.4. The beam is fabricated of isotropic material with elastic modulus E and Poisson's ratio ν. The *exact* solution of the beam problem (from both the theory-of-elasticity and beam-theory standpoints) is a constant bending moment M along the span. Consequently the beam deforms with uniform curvature $\kappa = M/(EI_z)$, in which $I_z = \frac{1}{12}hb^3$ is the cross-section second moment of inertia about z.

The beam is modeled with *one layer* of identical 4-node iso-P bilinear quadrilaterals through its height. These are rectangles with horizontal dimension a; in the Figure $a = L/4$. The aspect ratio b/a 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 E17.4.

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 E17.4, are

$$u_x = -\kappa xy, \quad u_y = \tfrac{1}{2}\kappa(x^2 + \nu y^2), \tag{E17.6}$$

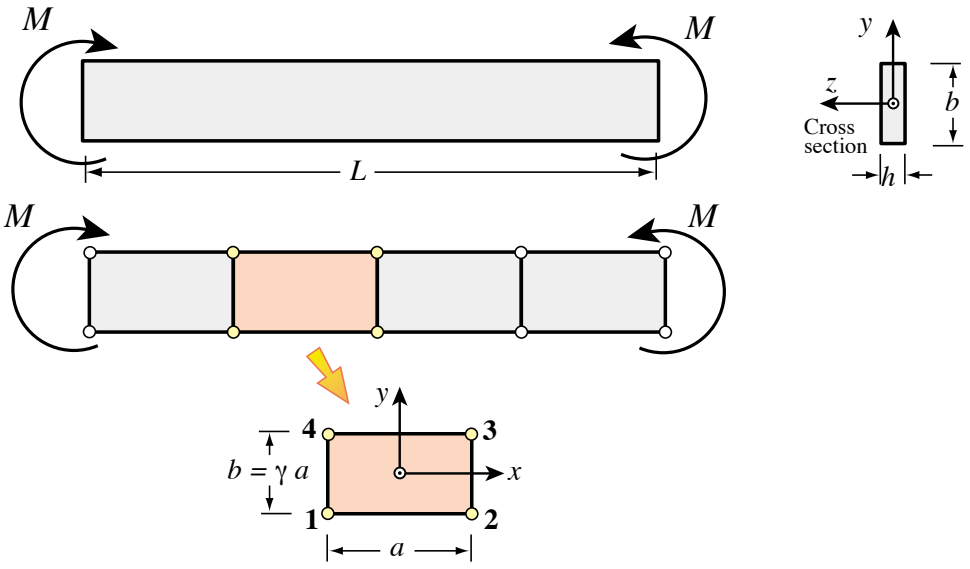

FIGURE E17.4. Pure bending of Bernoulli-Euler plane beam of thin rectangular cross section, for Exercises 17.3–7. The beam is modeled by one layer of 4-node iso-P bilinear quadrilaterals through its height.

where κ is the deformed beam curvature M/EI. The stiffness equations of the typical rectangular element are given by the close form expression (E17.5).

The purpose of this Exercise is to compare the in-plane bending response of the 4-node iso-P bilinear rectangle to that of a Bernoulli-Euler beam element (which would be exact for this configuration). The quadrilateral element will be called *x-bending exact* if it reproduces the beam solution for all $\{\gamma, \nu\}$. This comparison is distributed into three items.

(a) Check that (E17.6), as a plane stress 2D elasticity solution, is in full agreement with Bernoulli-Euler beam theory. This can be done by computing the strains $e_{xx} = \partial u_x/\partial x$, $e_{yy} = \partial u_y/\partial y$ and $2e_{xy} = \partial u_y/\partial x + \partial u_x/\partial y$. Then get the stresses σ_{xx}, σ_{yy} and σ_{xy} through the plane stress constitutive matrix (E17.3) of an isotropic material. Verify that both σ_{yy} and σ_{xy} vanish for any ν, and that $\sigma_{xx} = -E\,\kappa y = -My/I_z$, which agrees with equation (13.4) in Chapter 13.

(b) Compute the strain energy $U_{\rm quad} = \frac{1}{2}(\mathbf{u}_{\rm beam})^T\mathbf{K}^e\mathbf{u}_{\rm beam}$ absorbed by the 4-node element under nodal displacements $\mathbf{u}_{\rm beam}$ constructed by evaluating (E17.6) at the nodes 1,2,3,4. To simplify this calculation, it is convenient to decompose that vector as follows:

$$\begin{aligned}\mathbf{u}_{\rm beam} = \mathbf{u}^x_{\rm beam} + \mathbf{u}^y_{\rm beam} &= \tfrac{1}{4}\kappa ab\,[-1\ \ 0\ \ 1\ \ 0\ \ -1\ \ 0\ \ 1\ \ 0\,]^T \\ &\quad + \tfrac{1}{8}\kappa(a^2+\nu b^2)\,[\,0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\,]^T\end{aligned} \tag{E17.7}$$

Explain why $\mathbf{K}^e\mathbf{u}^y_{\rm beam}$ must vanish and consequently

$$U_{\rm quad} = \tfrac{1}{2}(\mathbf{u}^x_{\rm beam})^T\mathbf{K}^e\mathbf{u}^x_{\rm beam}. \tag{E17.8}$$

This energy can be easily computed by *Mathematica* by using the first 4 lines of the script of the previous Exercise, except that here `ncoor={{-a,-b},{a,-b},{a,b},{-a,b}}/2`. If vector $\mathbf{u}^x_{\rm beam}$ is formed in `u` as a one-dimensional list, `Uquad=Simplify[u.Ke.u/2]`. This should come out as a function of M, E, ν, h, a and γ because $\kappa = M/(EI_z) = 12M/(Eha^3\gamma^3)$.

(c) From Mechanics of Materials, or equation (13.7) of Chapter 13, the strain energy absorbed by the beam segment of length a under a constant bending moment M is $U_{beam} = \frac{1}{2}M\kappa a = M^2a/(2EI_z) =$

$6M^2/(Eha^2\gamma^3)$. Form the *energy ratio* $r = U_{quad}/U_{beam}$ and show that it is a function of the rectangle aspect ratio $\gamma = b/a$ and of Poisson's ratio ν only:

$$r = r(\gamma, \nu) = \frac{1 + 2/\gamma^2 - \nu}{(2/\gamma^2)(1 - \nu^2)}. \tag{E17.9}$$

This happens to be the ratio of the 2D model solution to the exact (beam) solution. Hence $r = 1$ means that we get the exact answer, that is the 2D model is x-bending exact. If $r > 1$ the 2D model is overstiff, and if $r < 1$ the 2D model is overflexible. Evidently $r > 1$ for all γ if $0 \le \nu \le \frac{1}{2}$. Moreover if $b << a, r >> 1$; for example if $a = 10b$ and $\nu = 0, r \approx 50$ and the 2D model gives only about 2% of the correct solution. This phenomenon is referred to in the FEM literature as *shear locking*, because overstiffness is due to the bending motion triggering spurious shear energy in the element. Remedies to shear locking at the element level are studied in advanced FEM courses. Draw conclusions as to the adequacy or inadequacy of the 2D model to capture inplane bending effects, and comment on how you might improve results by modifying the discretization of Figure E17.4.[9]

EXERCISE 17.4 [A+C:20] A naive remedy to shear locking can be attempted with the weighted integration methodology outlined in §17.6.1. Let $\mathbf{K}^e_{1\times1}$ and $\mathbf{K}^e_{2\times2}$ denote the element stiffnesses produced by 1×1 and 2×2 Gauss product rules, respectively. Take

$$\mathbf{K}^e_\beta = (1 - \beta)\mathbf{K}^e_{1\times1} + \beta\mathbf{K}^e_{2\times2} \tag{E17.10}$$

where β is adjusted so that shear locking is reduced or eliminated. It is not difficult to find β if the element is rectangular and isotropic. For the definition of *x-bending exact* please read the previous Exercise. Inserting $\mathbf{K}^e_\beta$ into the test introduced there verify that

$$r = \frac{\beta(1 + 2\gamma^2 - \nu)}{(2/\gamma^2)(1 - \nu^2)}. \tag{E17.11}$$

Whence show that if

$$\beta = \frac{2/\gamma^2(1 - \nu^2)}{1 + 2/\gamma^2 - \nu}, \tag{E17.12}$$

then $r \equiv 1$ for all $\{\gamma, \nu\}$ and the element is x-bending exact. A problem with this idea is that it does not make it y-bending exact because $r(\gamma) \ne r(1/\gamma)$ if $\gamma \ne 1$. Moreover the device is not easily extended to non-rectangular geometries or non-isotropic material.

EXERCISE 17.5 [A+C:35] (Advanced) To understand this Exercise please begin by reading Exercise 17.3, and the concept of shear locking. The material is again assumed isotropic with elastic modules E and Poisson's ratio ν. The 4-node rectangular element will be said to be *bending exact* if $r = 1$ for any $\{\gamma, \nu\}$ if the bending test described in Exercise 17.3 is done in both x and y directions. A bending-exact element is completely shear-lock-free.

The selective integration scheme outlined in §17.6.2 is more effective than weighted integration (covered in the previous exercise) to fully eliminate shear locking. Let the integration rules (I) and (II) be the 1×1 and 2×2 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] Consider the stress-strain splitting

$$\mathbf{E} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} = \frac{E}{1-\nu^2}\begin{bmatrix} \alpha & \beta & 0 \\ \beta & \alpha & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} + \frac{E}{1-\nu^2}\begin{bmatrix} 1-\alpha & \nu-\beta & 0 \\ \nu-\beta & 1-\alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{E}_{\rm I} + \mathbf{E}_{\rm II}, \tag{E17.13}$$

[9] Note that even if we make $a \to 0$ and $\gamma = b/a \to \infty$ by taking an infinite number of rectangular elements along x, the energy ratio r remains greater than one if $\nu > 0$ since $r \to 1/(1 - \nu^2)$. Thus the 2D model would not generally converge to the correct solution if we keep one layer through the height.

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

where α and β are scalars. Show that if

$$\chi = \sqrt{\frac{1-\nu^2}{3(1-\alpha)}} \tag{E17.14}$$

the resulting element stiffness $\mathbf{K}^e_{\rm I} + \mathbf{K}^e_{\rm II}$ is bending exact for any $\{\alpha, \beta\}$. As a corollary show that that if $\alpha = \nu^2$, which corresponds to the splitting

$$\mathbf{E} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} = \frac{E}{1-\nu^2}\begin{bmatrix} \nu^2 & \beta & 0 \\ \beta & \nu^2 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} + \frac{E}{1-\nu^2}\begin{bmatrix} 1-\nu^2 & \nu-\beta & 0 \\ \nu-\beta & 1-\nu^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{E}_{\rm I} + \mathbf{E}_{\rm II}, \tag{E17.15}$$

then $\chi = 1/\sqrt{3}$ and rule (II) becomes the standard 2×2 Gauss product rule. What are two computationally convenient settings for β?

EXERCISE 17.6 [A+C:35] (Advanced) A variation on the previous exercise on selective integration to make the isotropic rectangular 4-node element bending exact. Integration rule (I) is not changed. However rule (II) has four sample points located at $\{0, -\chi\}$, $\{\chi, 0\}$, $\{0, \chi\}$ and $\{-\chi, 0\}$ each with weight 1.[11] Show that if one selects the stress-strain splitting (E17.13) and

$$\chi = \sqrt{\frac{2(1-\nu^2)}{3(1-\alpha)}} \tag{E17.16}$$

the resulting element stiffness $\mathbf{K}^e_{\rm I} + \mathbf{K}^e_{\rm II}$ is bending exact for any $\{\alpha, \beta\}$. Discuss which choices of α reduce χ to $1/\sqrt{3}$ and $\sqrt{2/3}$, respectively.

EXERCISE 17.7 [A+C:40] (Advanced, research paper level, requires a CAS to be tractable) Extend Exercise 17.5 to consider the case of general anisotropic material:

$$\mathbf{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{12} & E_{22} & E_{23} \\ E_{13} & E_{23} & E_{33} \end{bmatrix} \tag{E17.17}$$

The rules for the selective integration scheme are as described in Exercise 17.5. The appropriate stress-strain splitting is

$$\mathbf{E} = \mathbf{E}_{\rm I} + \mathbf{E}_{\rm II} = \begin{bmatrix} E_{11}\,\alpha_1 & E_{12}\beta & E_{13} \\ E_{12}\,\beta & E_{22}\,\alpha_2 & E_{23} \\ E_{13} & E_{23} & E_{33} \end{bmatrix} + \begin{bmatrix} E_{11}(1-\alpha_1) & E_{12}(1-\beta) & 0 \\ E_{12}(1-\beta) & E_{22}(1-\alpha_2) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{E17.18}$$

in which β is arbitrary and

$$\begin{aligned} 1-\alpha_1 &= \frac{|\mathbf{E}|}{3\chi^2 E_{11}(E_{22}E_{33}-E_{23}^2)} = \frac{1}{3\chi^2 C_{11}}, \quad 1-\alpha_2 = \frac{|\mathbf{E}|}{3\chi^2 E_{22}(E_{11}E_{33}-E_{13}^2)} = \frac{1}{3\chi^2 C_{22}}, \\ |\mathbf{E}| &= \det(\mathbf{E}) = E_{11}E_{22}E_{33} + 2E_{12}E_{13}E_{23} - E_{11}E_{23}^2 - E_{22}E_{13}^2 - E_{33}E_{12}^2, \\ C_{11} &= E_{11}(E_{22}E_{33}-E_{13}^2)/|\mathbf{E}|, \quad C_{22} = E_{22}(E_{11}E_{33}-E_{13}^2)/|\mathbf{E}|. \end{aligned} \tag{E17.19}$$

Show that the resulting rectangular element is bending exact for any $\mathbf{E}$ and $\chi \neq 0$. (In practice one would select $\chi = 1/\sqrt{3}$.)

[11] This is called a 4-point median rule, since the four points are located on the quadrilateral medians.