UNIVERSIDAD POLITECNICA DE VALENCIA

DEPARTAMENTO DE INGENIERIA MECANICA Y DE MATERIALES

ELEMENTOS FINITOS

(E.T.S.I.I.V)

FORMULACION DE ELEMENTOS FINITOS

LECCION 6.- CUADRILATEROS ISOPARAMETRICOS

J. L. OLIVER
Dr. Ingeniero Industrial

Valencia, 2005

001 PASOS A SEGUIR PARA PODER FORMULAR UN ELEMENTOS CUADRILATERO CURSO 2004-5

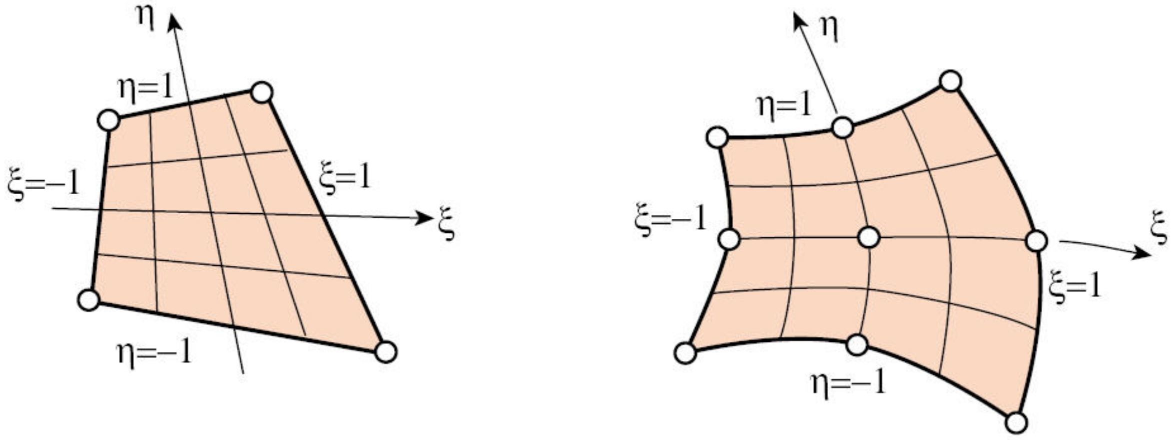
Isoparametric Quadrilaterals

Implementation Steps for Element Stiffness Matrix:

1. *Construct Shape Functions in Quad Coordinates (Chapter 18 is devoted to this topic)*
2. *Compute x-y Derivatives of Shape Functions and Build Strain-Displacement Matrix* **B**
3. *Integrate h* $\mathbf{B}^T\mathbf{E}\,\mathbf{B}$ *over element*

001 CALCULO DE LAS DERIVADAS PARCIALES DE LAS FUNCIONES DE FORMA CURSO 2004-5

Partial Derivative Computation



Shape functions are written in terms of ξ **and** η
But Cartesian partials (with respect to x, y**) are required to get strains & stresses**

001 NECESITAMOS CALCULAR LAS MATRICES JACOBIANA Y SU INVERSA CURSO 2004-5

The Jacobian and Inverse Jacobian

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}$$

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial x} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = \mathbf{J}^{-T} \begin{bmatrix} dx \\ dy \end{bmatrix}$$

in which

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

$$J = |\mathbf{J}| = \det \mathbf{J}$$

001 Y LAS DERIVADAS PARCIALES DE LAS FUNCIONES DE FORMA CURSO 2004-5

Shape Function Partial Derivatives

Using chain rule

$$\frac{\partial N_i^{(e)}}{\partial x} = \frac{\partial N_i^{(e)}}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial N_i^{(e)}}{\partial \eta}\frac{\partial \eta}{\partial x}$$

$$\frac{\partial N_i^{(e)}}{\partial y} = \frac{\partial N_i^{(e)}}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial N_i^{(e)}}{\partial \eta}\frac{\partial \eta}{\partial y}$$

Main problem is to get $\frac{\partial \xi}{\partial x} \quad \frac{\partial \eta}{\partial x} \quad \frac{\partial \xi}{\partial y} \quad \frac{\partial \eta}{\partial y}$

001 CON LO QUE LOS ELEMENTOS DE LA INVERSA DE LA MATRIZ JACOBIANA SERAN LOS QUE NECESITAMOS CURSO 2004-5

The Jacobian and Inverse-Jacobian Matrices

Compute the 2 x 2 Jacobian matrix

$$\mathbf{J} = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

Then invert to get

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

These are the quantities we need for the S.F. partials

001 PARA CALCULAR LOS ELEMENTOS DE LA MATRIZ JACOBIANA TENDREMOS QUE HACER LO SIGUIENTE CURSO 2004-5

Use the element geometry definition

$$x = \sum_{i=1}^{n} x_i N_i^{(e)} \qquad y = \sum_{i=1}^{n} y_i N_i^{(e)}$$

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

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

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

$$\mathbf{J} = \mathbf{PX} = \begin{bmatrix} \frac{\partial N_1^{(e)}}{\partial \xi} & \frac{\partial N_2^{(e)}}{\partial \xi} & \dots & \frac{\partial N_n^{(e)}}{\partial \xi} \\ \frac{\partial N_1^{(e)}}{\partial \eta} & \frac{\partial N_2^{(e)}}{\partial \eta} & \dots & \frac{\partial N_n^{(e)}}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}. \qquad (17.8)$$

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

The symbolic inversion of **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 factors that motivated the use of Gaussian numerical quadrature, as discussed below.

001 AQUÍ ESTA LA SECUENCIA DE PASOS A SEGUIR PARA OBTENER LAS DERIVADAS PARCIALES DE LAS FUNCIONES DE FORMA CURSO 2004-5

Partial Derivative Computation Sequence Summary

At a specific point of quad coordinates ξ and η:

Compute $\frac{\partial x}{\partial \xi} \quad \frac{\partial y}{\partial \xi} \quad \frac{\partial x}{\partial \eta} \quad \frac{\partial y}{\partial \eta}$ **from node coordinates and S.F.s**

Form J and invert to get $\mathbf{J}^{-1}$ and det J

Apply the chain rule to get the x , y **partials of the S.F.s**

001 CALCULO DE LA MATRIZ DEFORMACIONES DESPLAZAMIENTOS CURSO 2004-5

The Strain Displacement Matrix B

Use those S.F.s partials to build the strain-displacement matrix B:

$$\mathbf{e} = \begin{bmatrix} \frac{\partial N_1^{(e)}}{\partial x} & 0 & \frac{\partial N_2^{(e)}}{\partial x} & 0 & \dots & \frac{\partial N_n^{(e)}}{\partial x} & 0 \\ 0 & \frac{\partial N_1^{(e)}}{\partial y} & 0 & \frac{\partial N_2^{(e)}}{\partial y} & \dots & 0 & \frac{\partial N_n^{(e)}}{\partial y} \\ \frac{\partial N_1^{(e)}}{\partial y} & \frac{\partial N_1^{(e)}}{\partial x} & \frac{\partial N_2^{(e)}}{\partial y} & \frac{\partial N_2^{(e)}}{\partial x} & \dots & \frac{\partial N_n^{(e)}}{\partial y} & \frac{\partial N_n^{(e)}}{\partial x} \end{bmatrix} \mathbf{u}^{(e)} = \mathbf{B}\mathbf{u}^{(e)}$$

Unlike the 3-node triangle, here B = B(ξ,η) varies over quad

One Dimensional Gauss Integration Rules

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

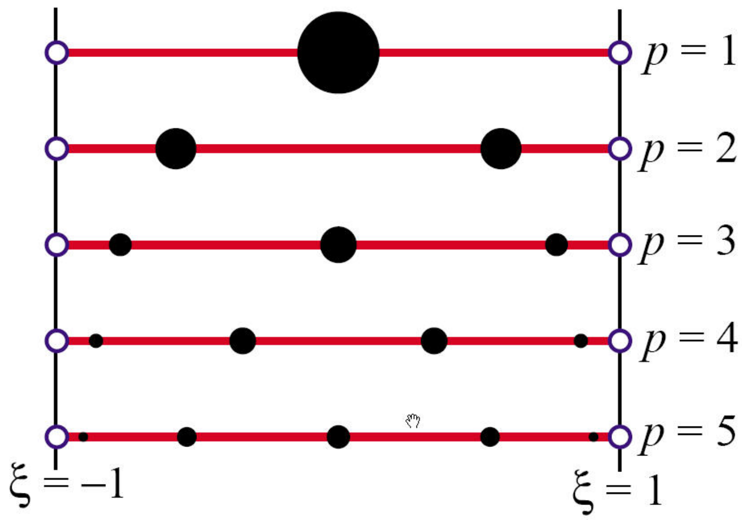
One point: $$\int_{-1}^{1} F(\xi)\, d\xi \doteq 2F(0),$$

Two points: $$\int_{-1}^{1} F(\xi)\, d\xi \doteq F(-1/\sqrt{3}) + F(1/\sqrt{3}),$$

Three points: $$\int_{-1}^{1} F(\xi)\, d\xi \doteq \frac{5}{9}F(-\sqrt{3/5}) + \frac{8}{9}F(0) + \frac{5}{9}F(\sqrt{3/5})$$

For 4 and 5 points see Notes

Graphical Representation of the First Five One-Dimensional Gauss Integration Rules



$$\text{Four points:} \quad \int_{-1}^{1} F(\xi)\,d\xi \approx w_{14}F(\xi_{14}) + w_{24}F(\xi_{24}) + w_{34}F(\xi_{34}) + w_{44}F(\xi_{44}),$$

$$\text{Five points:} \quad \int_{-1}^{1} F(\xi)\,d\xi \approx w_{15}F(\xi_{15}) + w_{25}F(\xi_{25}) + w_{35}F(\xi_{35}) + w_{45}F(\xi_{45}) + w_{55}F(\xi_{55}). \tag{17.11}$$

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 five 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 $w_{55} = 512/900$.

The rules (17.11) 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.

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 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

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

Higher order Gauss rules are tabulated in standard manuals for numerical computation. For example, the widely used Handbook of Mathematical Functions [17.1] tabulates (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.

The following *Mathematica* module returns exact or floating-point information for the first five unidimensional Gauss rules:

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

To get information for the i^{th} point of the p^{th} rule, in which $1 \leq i \leq p$ and $p = 1, 2, 3, 4, 5$, call the module as `{ xi,wi }=LineGaussRuleInfo[{ p,numer },i]`. 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 `xi` and the weight w_i in `wi`. Example: `LineGaussRuleInfo[{ 3,False },2]` returns `{ 0,8/9 }`. If p is not in range 1 through 5, the module returns `{ Null,0 }`.

001 REGLAS DE GAUSS PARA PROBLEMAS BIDIMENSIONALES
REGLAS DEL PRODUCTO CURSO 2004-5

Two Dimensional Product Gauss Rules

Canonical form of integral:

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

Gauss integration rules with p_1 points in the ξ direction and p_2 points in the η direction:

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

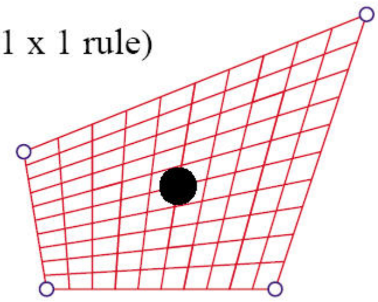
Usually $p_1 = p_2$

001 REPRESENTACION GRAFICAS REGLAS BIDIMENSIONALES CURSO 2004-5

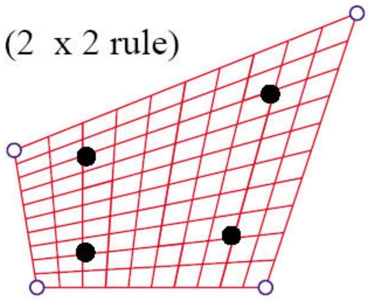
Graphical Representation of the First Four 2D Product-Type Gauss Integration Rules

with Equal # of Points p in Each Direction

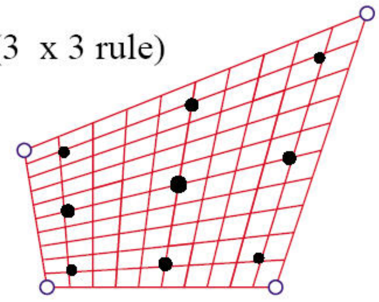
p = 1 (1 x 1 rule)



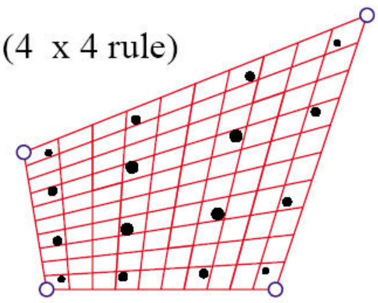
p = 2 (2 x 2 rule)



p = 3 (3 x 3 rule)



p = 4 (4 x 4 rule)



The following *Mathematica* module implements the two-dimensional product Gauss rules with 1 through 5 points in each direction. The number of points in each direction may be the same or different.

```
QuadGaussRuleInfo[{rule_,numer_},point_]:= Module[
 {xi,eta,p1,p2,i1,i2,w1,w2,k,info={{Null,Null},0},
  If [Length[rule] ==2, {p1,p2}=rule, p1=p2=rule];
  If [Length[point]==2, {i1,i2}=point,
      k=point; i2=Floor[(k-1)/p1]+1; i1=k-p1*(i2-1) ];
  {xi, w1}=  LineGaussRuleInfo[{p1,numer},i1];
  {eta,w2}=  LineGaussRuleInfo[{p2,numer},i2];
  info={{xi,eta},w1*w2};
  If [numer, Return[N[info]], Return[Simplify[info]]];
];
```

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 ]
```
(17.15)

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[2] 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 ]
```
(17.16)

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; 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}`.

Gauss Integration of Stiffness Matrix

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

Rewrite in canonical form:

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

where

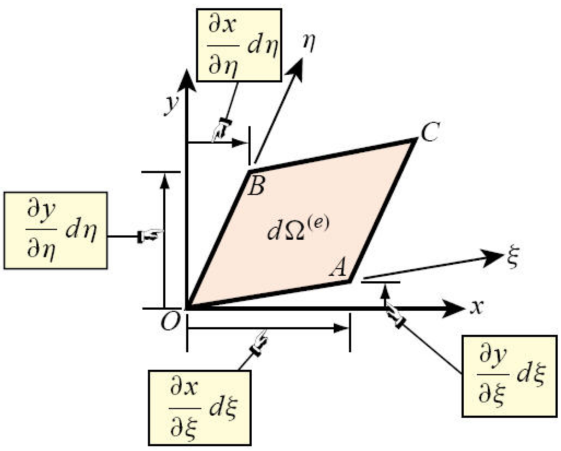
$$d\Omega^{(e)} = dx\,dy = \det\mathbf{J}\;d\xi\,d\eta.$$

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

Then apply the rule to **F** (a 2n x 2n matrix)

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

Geometric Interpretation of Jacobian Determinant det J = |J|



$$dA = \vec{O}B \times \vec{O}A = \frac{\partial x}{\partial \xi}d\xi\,\frac{\partial y}{\partial \eta}d\eta - \frac{\partial x}{\partial \eta}d\eta\,\frac{\partial y}{\partial \xi}d\xi = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix}\,d\xi\,d\eta = |\mathbf{J}|\,d\xi\,d\eta.$$

Quad 4 Element Formation in *Mathematica:* Stiffness Computation Module

```
Quad4IsoPMembraneStiffness[ncoor_,mprop_,fprop_,options_]:=
  Module[{i,k,p=2,numer=False,Emat,th=1,h,qcoor,c,w,Nf,
   dNx,dNy,Jdet,B,Ke=Table[0,{8},{8}]},  Emat=mprop[[1]];
  If [Length[options]==2, {numer,p}=options, {numer}=options];
  If [Length[fprop]>0, th=fprop[[1]]];
  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]==0, h=th, h=th.Nf]; c=w*Jdet*h;
        B={ 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[B].(Emat.B)];
      ]; Return[Ke]
   ];
```

Quad 4 Element Formation in *Mathematica:* Shape Functions and Their Derivatives

```
Quad4IsoPShapeFunDer[ncoor_,qcoor_]:= Module[
  {Nf,dNx,dNy,dNξ,dNη,i,J11,J12,J21,J22,Jdet,ξ,η,x1,x2,x3,x4,
   y1,y2,y3,y4,x,y},
  {ξ,η}=qcoor; {{x1,y1},{x2,y2},{x3,y3},{x4,y4}}=ncoor;
   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}]
];
```

Quad 4 Element Formation in *Mathematica:* 2D Gauss Quadrature Rule Information

```
QuadGaussRuleInfo[{rule_,numer_},point_]:= Module[
 {xi,eta,p1,p2,i1,i2,w1,w2,k,info=Null},
  If [Length[rule]==2,  {p1,p2}=rule, p1=p2=rule];
  If [Length[point]==2, {i1,i2}=point,
      k=point; i2=Floor[(k-1)/p1]+1; i1=k-p1*(i2-1) ];
  {xi, w1}=  LineGaussRuleInfo[{p1,numer},i1];
  {eta,w2}=  LineGaussRuleInfo[{p2,numer},i2];
  info={{xi,eta},w1*w2};
  If [numer, Return[N[info]], Return[Simplify[info]]];
];
```

Works for any combination of
p_1 = 1,2,3,4,5 and p_2 = 1,2,3,4,5

Calls 1D Gauss rule module of next slide twice

Quad 4 Element Formation in *Mathematica:* 1D Gauss Quadrature Rule Information

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

Works for p = 1,2,3,4,5

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

`mprop` Material properties supplied as the list `{ Emat,rho,alpha }`. `Emat` is a two-dimensional list storing the 3×3 plane stress matrix of elastic moduli:

$$\mathbf{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{12} & E_{22} & E_{23} \\ E_{13} & E_{23} & E_{33} \end{bmatrix} \tag{E17.1}$$

If the material is isotropic with elastic modulus E and Poisson's ratio ν, this matrix becomes

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

The other two items in `mprop` are not used in this module so zeros may be inserted as placeholders.

`fprop` Fabrication properties. The plate thickness specified as a four-entry list: `{ h1,h2,h3,h4 }`, a one-entry list: `{ h }`, or an empty list: `{ }`.

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 `h`. If an empty list appears the module assumes a uniform unit 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 forced to proceed in floating point arithmetic. For symbolic or exact arithmetic work set `numer` to `False`.[6]

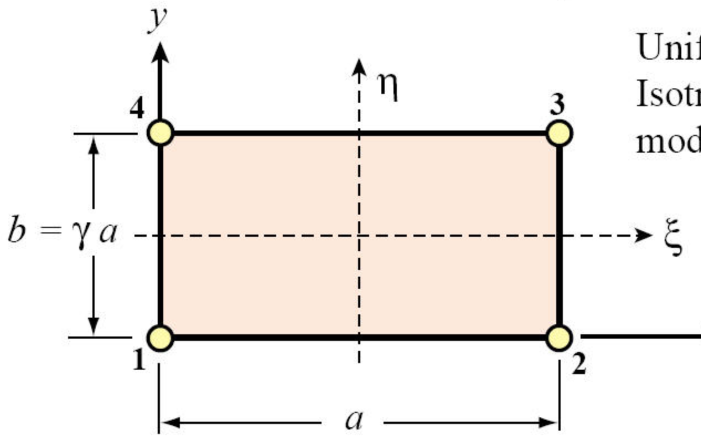
`p` specifies the Gauss product rule to have `p` points in each direction. `p` may be 1 through 4. For rank sufficiency, `p` must be 2 or higher. If `p` is 1 the element will be rank deficient by two.[7] If omitted `p` = 2 is assumed.

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

$$\mathbf{u}^{(e)} = [\, u_{x1} \quad u_{y1} \quad u_{x2} \quad u_{y2} \quad u_{x3} \quad u_{y3} \quad u_{x4} \quad u_{y4} \,]^T . \tag{E17.3}$$

Quad4 Element Formation in *Mathematica*

HW Exercises 17.1 through 17.3



Uniform thickness $h = 1$
Isotropic material with elastic modulus E and Poisson's ratio ν

$$\gamma = b/a,\ \psi_1 = (1+\nu)\gamma,\ \psi_2 = (1-3\nu)\gamma,$$

$\psi_3 = 1 - \nu + 2\gamma^2$, $\psi_4 = 2 + (1-\nu)\gamma^2$, $\psi_5 = 1 - \nu - 4\gamma^2$, $\psi_6 = 1 - \nu - \gamma^2$, $\psi_7 = 4 - (1-\nu)\gamma^2$ and $\psi_8 = 1 - (1-\nu)\gamma^2$. Then

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

001 **EJERCICIO 1** **CURSO 2004-5**

[C:15] 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,0,0},{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 discussed in detail 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?

001 **EJERCICIO 2** **CURSO 2004-5**

[C:20] Check the rectangular element stiffness closed form given in (E17.4). This may be done by hand (takes a few days) or running the following script that calls the *Mathematica* module of Figure E17.1:

```
ClearAll[Em,ν,a,b,h];  b=γ*a;
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,0,0},{h},{False,2}];
scaledKe=Simplify[Ke*(24*γ*(1-ν^2)/(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 (E17.4) by printout inspection and report any typos to instructor.