

17

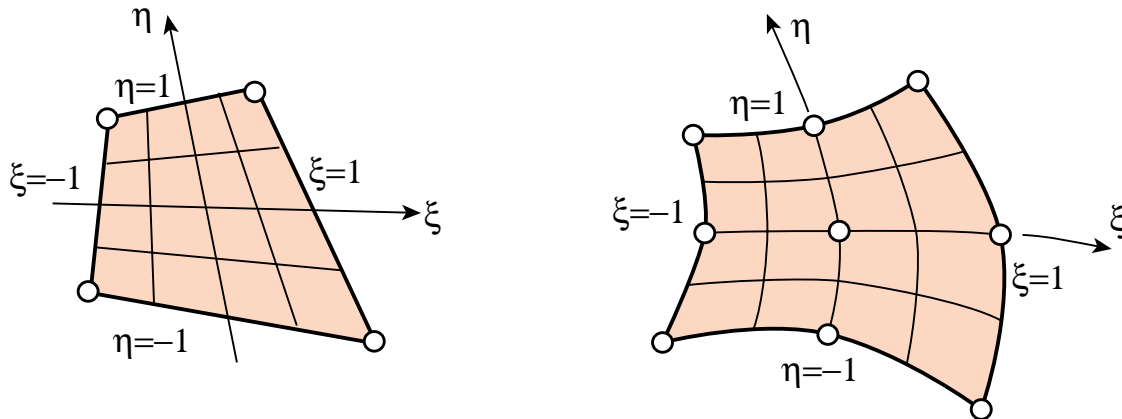
Isoparametric Quadrilaterals

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

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**

The Jacobian and Inverse Jacobian

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

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

in which

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

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 $\frac{\partial \xi}{\partial x}$ $\frac{\partial \eta}{\partial x}$ $\frac{\partial \xi}{\partial y}$ $\frac{\partial \eta}{\partial y}$

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

Computing the Jacobian Matrix

Use the element geometry definition (n : number of nodes)

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

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

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

Store as entries of \mathbf{J} and invert to get \mathbf{J}^{-1}

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

Partial Derivative Computation Sequence Summary

At a specific point of quad coordinates ξ and η :

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

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

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

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 $\mathbf{B} = \mathbf{B}(\xi, \eta)$ varies over quad

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} ]
];

```

One Dimensional Gauss Integration Rules

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

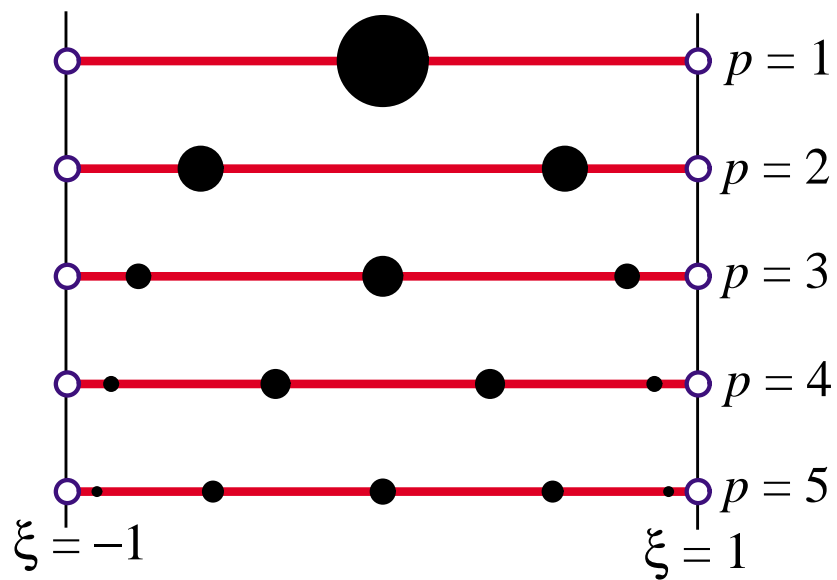
One point $\int_{-1}^1 F(\xi) d\xi \doteq 2F(0)$

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



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 $p = 1,2,3,4,5$

Two Dimensional Product Gauss Rules

Canonical form of integral:

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

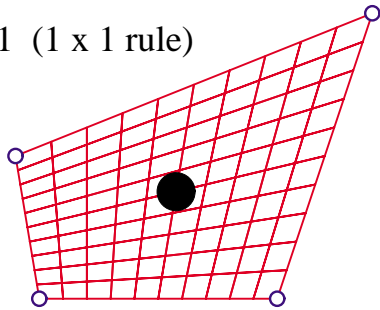
Gauss integration rules with p_1 points in the ξ direction and p_2 points in the η direction:

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

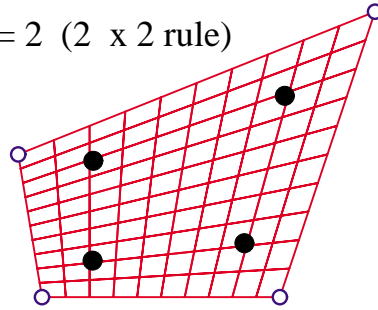
Usually $p_1 = p_2$

Graphical Representation of the First Four 2D Product-Type Gauss Integration Rules with Equal # of Points p in Each Direction

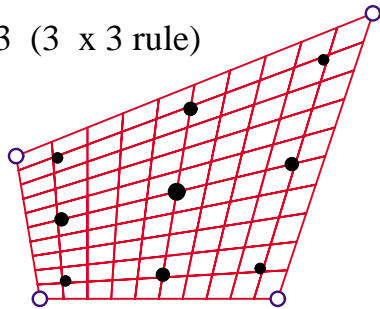
$p = 1$ (1 x 1 rule)



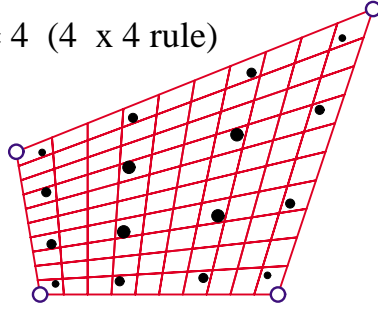
$p = 2$ (2 x 2 rule)



$p = 3$ (3 x 3 rule)



$p = 4$ (4 x 4 rule)



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

```

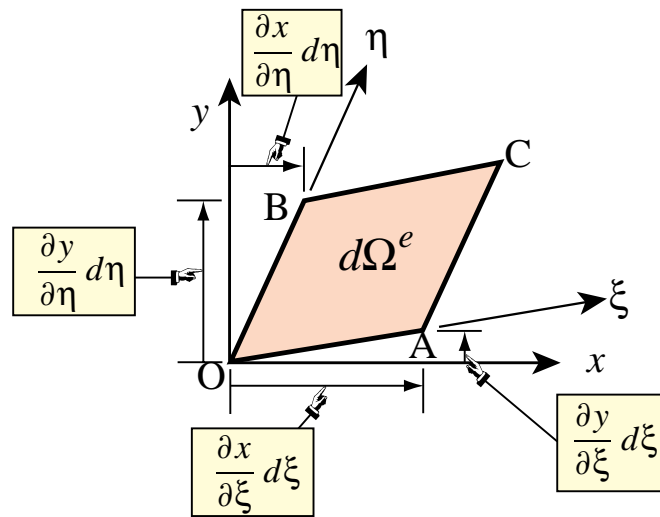
QuadGaussRuleInfo[{rule_,numer_},point_]:= Module[
  {xi,eta,p1,p2,i1,i2,w1,w2,k,info=NULL},
  If [Length[rule]==2, {p1,p2}=rule, p1=p2=rule];
  If [Length[point]==2, {i1,i2}=point,
    k=point; i2=Floor[(k-1)/p1]+1; i1=k-p1*(i2-1) ];
  {xi, w1}= LineGaussRuleInfo[{p1,numer},i1];
  {eta,w2}= LineGaussRuleInfo[{p2,numer},i2];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

Geometric Interpretation of Jacobian Determinant $J = \det \mathbf{J} = |\mathbf{J}|$



$$\begin{aligned}
 d\Omega^e &= \vec{OB} \times \vec{OA} = \frac{\partial x}{\partial \xi} d\xi \frac{\partial y}{\partial \eta} d\eta - \frac{\partial x}{\partial \eta} d\eta \frac{\partial y}{\partial \xi} d\xi = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} d\xi d\eta \\
 &= |\mathbf{J}| d\xi d\eta = J d\xi d\eta.
 \end{aligned}$$

Gauss Integration of Stiffness Matrix

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

Rewrite in canonical form:

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

where

$$d\Omega = dx dy = \det \mathbf{J} d\xi d\eta$$

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

Then apply the rule to \mathbf{F} (a $2n \times 2n$ matrix)

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

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

```

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]]
];

```

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

