UNIVERSIDAD POLITECNICA DE VALENCIA

DEPARTAMENTO DE INGENIERIA MECANICA Y DE MATERIALES

ELEMENTOS FINITOS

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

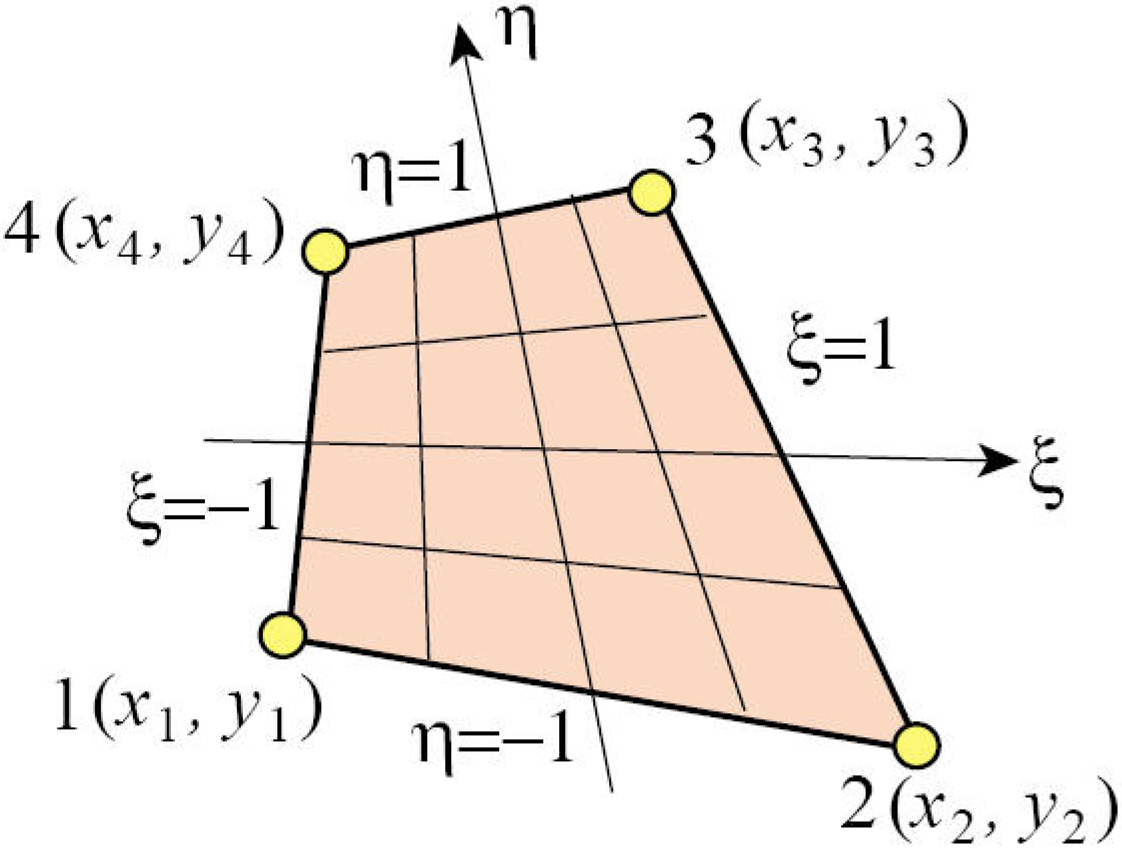
IMPLEMENTACION COMPUTACIONAL DEL METODO EF

LECCION 10.- IMPLEMENTACION DEL CUADRILATERO ISO P

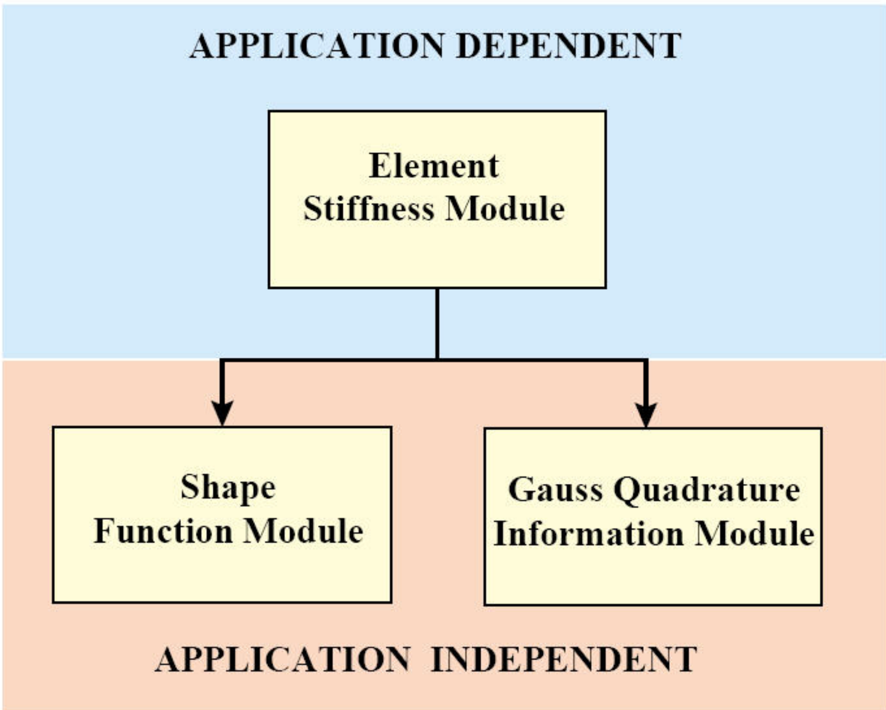
J. L. OLIVER
Dr. Ingeniero Industrial

Valencia, 2005

001 ELEMENTO CUADRILATERO BILINEAL D 4 NODOS ISOPARAMETRICO CURSO 2005-06



001 ORGANIZACIÓN DEL CÁLCULO DE LA MATRIZ DE RIGIDEZ CURSO 2005-06



Recall from §17.3 that Gauss quadrature rules for isoparametric quadrilateral elements have the canonical form

$$\int_{-1}^{1}\int_{-1}^{1} \mathbf{F}(\xi,\eta)\,d\xi\,d\eta = \int_{-1}^{1} d\eta \int_{-1}^{1} \mathbf{F}(\xi,\eta)\,d\xi \doteq \sum_{i=1}^{p_1}\sum_{j=1}^{p_2} w_i w_j \mathbf{F}(\xi_i,\eta_j). \tag{23.1}$$

Here $\mathbf{F} = h\mathbf{B}^T\mathbf{E}\mathbf{B}\,J$ is the matrix to be integrated, and p_1 and p_2 are the number of Gauss points in the ξ and η directions, respectively. Often, but not always, the same number $p = p_1 = p_2$ is chosen in both directions. A formula with $p_1 = p_2$ is called an *isotropic integration rule* because directions ξ and η are treated alike.

`QuadGaussRuleInfo` is an application independent module QuadGaussRuleInfo that implements the two-dimensional product Gauss rules with 1 through 4 points in each direction. The number of points in each direction may be the same or different. Use of this module was described in detail in §17.3.4. For the readers convenience it is listed, along with its subordinate module `LineGaussRuleInfo`, in Figure 23.3.

```
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]]];
];
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]},
   i,info=Null},  i=point;
  If [rule==1, info={0,2}];
  If [rule==2, info={g2[[i]],1}];
  If [rule==3, info={g3[[i]],w3[[i]]}];
  If [rule==4, info={g4[[i]],w4[[i]]}];
  If [numer, Return[N[info]], Return[Simplify[info]]];
];
```

Quad4IsoPShapeFunDer is an application independent module that computes the shape functions $N_i^{(e)}$, $i = 1, 2, 3, 4$ and its x-y partial derivatives at the sample integration points. The logic, listed in Figure 23.4, is straightforward and follows closely the description of Chapter 17.

The arguments of the module are the $\{x, y\}$ quadrilateral corner coordinates, which are passed in ncoor, and the two quadrilateral coordinates $\{\xi, \eta\}$, which are passed in qcoor. The former have the same configuration as described for the element stiffness module below.

The quadrilateral coordinates define the element location at which the shape functions and their derivatives are to be evaluated. For the stiffness formation these are Gauss points, but for strain and stress computations these may be other points, such as corner nodes.

Quad4IsoPShapeFunDer returns the two-level list { Nf,Nx,Ny,Jdet }, in which the first three are 4-entry lists. List Nf collects the shape function values, Nx the shape function x-derivatives, Ny the shape function y-derivatives, and Jdet is the Jacobian determinant called J in Chapters 17.

```
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; J21=dNξ.y; J12=dNη.x; J22=dNη.y;
   Jdet=Simplify[J11*J22-J12*J21];
   dNx= ( J22*dNξ-J21*dNη)/Jdet;  dNx=Simplify[dNx];
   dNy= (-J12*dNξ+J11*dNη)/Jdet;  dNy=Simplify[dNy];
   Return[{Nf,dNx,dNy,Jdet}]
];
```

Module `Quad4IsoPMembraneStiffness` computes the stiffness matrix of a four-noded isoparametric quadrilateral element in plane stress. The module configuration is typical of isoparametric elements in any number of dimensions. It follows closely the procedure outlined in Chapter 17. The module logic is listed in Figure 23.5. The statements at the bottom of the module box (not shown in Figure 23.5) test it for a specific configuration.

The arguments of the module are:

`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:

$$\mathbf{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{12} & E_{22} & E_{23} \\ E_{13} & E_{23} & E_{33} \end{bmatrix} \tag{23.2}$$

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

$$\mathbf{E} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\nu) \end{bmatrix} \tag{23.3}$$

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`.

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. If omitted p = 2 is assumed.

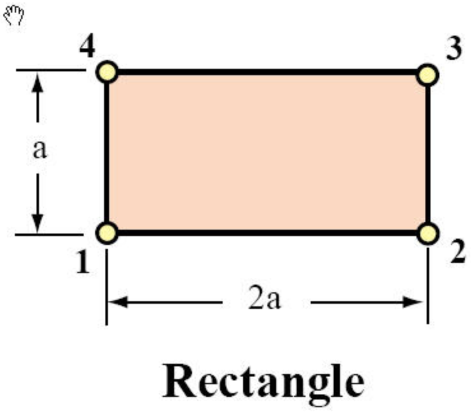
```
Quad4IsoPMembraneStiffness[ncoor_,mprop_,fprop_,options_]:=
  Module[{i,k,p=2,numer=False,Emat,th=1,h,qcoor,c,w,Nf,
   dNx,dNy,Jdet,B,Ke=Table[0,{8},{8}]},   Emat=mprop[[1]];
  If [Length[options]==2, {numer,p}=options, {numer}=options];
  If [Length[fprop]>0, th=fprop[[1]]];
  If [p<1||p>4, Print["p 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[Simplify[Ke]]
   ];
```

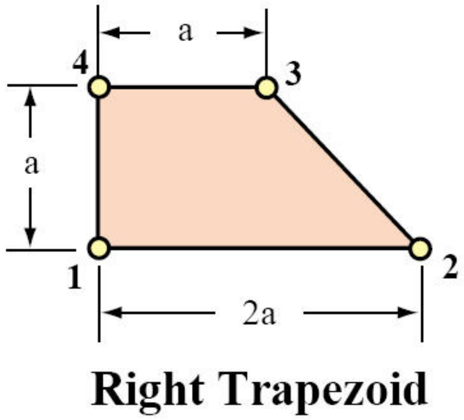
The module returns `Ke` as an 8 × 8 symmetric matrix pertaining to the following arrangement of nodal displacements:

$$\mathbf{u}^{(e)} = [\,u_{x1} \quad u_{y1} \quad u_{x2} \quad u_{y2} \quad u_{x3} \quad u_{y3} \quad u_{x4} \quad u_{y4}\,]^T. \tag{23.4}$$

001 MODELOS COMPROBACION MODULOS CURSO 2005-06

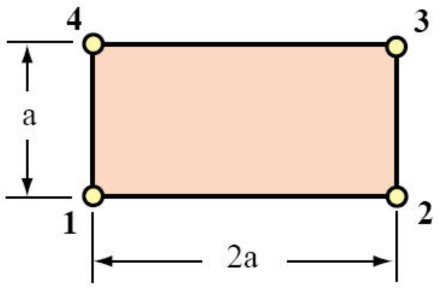
The stiffness module is tested on the two quadrilateral geometries shown in Figure 23.6. Both elements have unit thickness and isotropic material. The left one is a rectangle of base $2a$ and height a. The right one is a right trapezoid with base $2a$, top width a and height a. Both geometries will be used to illustrate the effect of the numerical integration rule.





Main purpose is to illustrate effect of changing Gauss integration rule

001 MODELO RECTANGULO - DEFINICION CURSO 2005-06




```
ClearAll[Em,nu,a,b,e,h,p,num];  h=1;
Em=96; nu=1/3;    (* isotropic material *)
Emat=Em/(1-nu^2)*{{1,nu,0},{nu,1,0},{0,0,(1-nu)/2}};
Print["Emat=",Emat//MatrixForm];
ncoor={{0,0},{2*a,0},{2*a,a},{0,a}}; (* 2:1 rectangular geometry *)

p=2; (* 2 x 2 Gauss rule *) num=False; (* exact symbolic arithmetic *)
Ke=Quad4IsoPMembraneStiffness[ncoor,{Emat,0,0},{h},{num,p}];
Ke=Simplify[Chop[Ke]];  Print["Ke=",Ke//MatrixForm];
Print["Eigenvalues of Ke=",Chop[Eigenvalues[N[Ke]],.0000001]];
```

Uniform thickness $h = 1$, isotropic material with $E = 96$ and $\nu = 1/3$. Rectangle dimension a cancels out in forming stiffness.

$$\mathbf{E} = \begin{bmatrix} 108 & 36 & 0 \\ 36 & 108 & 0 \\ 0 & 0 & 36 \end{bmatrix}$$

001 MODELO RECTANGULO - RESULTADOS CURSO 2005-06

Note that the rectangle dimension a does not appear in (23.6). This is a general property: *the stiffness matrix of plane stress elements is independent of inplane dimension scalings*. This follows from the fact that entries of the strain-displacement matrix $\mathbf{B}$ have dimensions $1/L$, where L denotes a characteristic inplane length. Consequently entries of $\mathbf{B}^T\mathbf{B}$ have dimension $1/L^2$. Integration over the element area cancels out L^2.

Using a higher order Gauss integration rule, such as 3×3 and 4×4, reproduces exactly (23.6). This is a property characteristic of the rectangular geometry, since in that case the entries of $\mathbf{B}$ vary linearly in ξ and η, and J is constant. Therefore the integrand $h\,\mathbf{B}^T\,\mathbf{E}\,\mathbf{B}J$ is at most quadratic in ξ and η, and 2 Gauss points in each direction suffice to compute the integral exactly. Using a 1×1 rule yields a rank-deficiency matrix, a result illustrated in detail in §23.2.2.

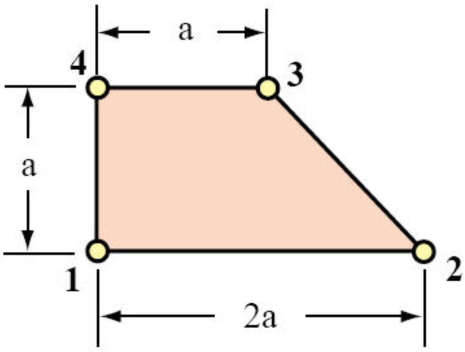
Stiffness matrix computed by *p x p* rule, *p* =2,3,4,...

$$\mathbf{K}^{(e)} = \begin{bmatrix} 42 & 18 & -6 & 0 & -21 & -18 & -15 & 0 \\ 18 & 78 & 0 & 30 & -18 & -39 & 0 & -69 \\ -6 & 0 & 42 & -18 & -15 & 0 & -21 & 18 \\ 0 & 30 & -18 & 78 & 0 & -69 & 18 & -39 \\ -21 & -18 & -15 & 0 & 42 & 18 & -6 & 0 \\ -18 & -39 & 0 & -69 & 18 & 78 & 0 & 30 \\ -15 & 0 & -21 & 18 & -6 & 0 & 42 & -18 \\ 0 & -69 & 18 & -39 & 0 & 30 & -18 & 78 \end{bmatrix}$$

Eigenvalues of stiffness matrix:

$$[\,223.64 \quad 90 \quad 78 \quad 46.3603 \quad 42 \quad 0 \quad 0 \quad 0\,]$$

which verifies that $\mathbf{K}^{(e)}$ has the correct rank of five (8 freedoms minus 3 rigid body modes).




```
ClearAll[Em,nu,h,a,p];  h=1;
Em=48*63*13*107; nu=1/3;
Emat=Em/(1-nu^2)*{{1,nu,0},{nu,1,0},{0,0,(1-nu)/2}};
ncoor={{0,0},{2*a,0},{a,a},{0,a}};
For [p=1,p<=4,p++,
    Ke=Quad4IsoPMembraneStiffness[ncoor,{Emat,0,0},{h},{True,p}];
    Ke=Rationalize[Ke,0.0000001]; Print["Ke=",Ke//MatrixForm];
    Print["Eigenvalues of Ke=",Chop[Eigenvalues[N[Ke]],.0000001]]
    ];
```

Strange value of $E = 48 \text{ x } 63 \text{ x } 13 \text{ x } 107 = 4206384$ is to get exact entries in the stiffness matrix computed by Gauss rules $p = 1,2,3,4$.

The trapezoidal element geometry of Figure 23.6(b) is used to illustrate the effect of changing the $p \times p$ Gauss integration rule. Unlike the rectangular case, the element stiffness keeps changing as p is varied from 1 to 4. The element is rank sufficient, however, for $p \geq 2$ in agreement with the analysis of Chapter 19.

The computations are driven with the script shown in Figure 23.8. The value of p is changed in a loop. The flag `numer` is set to `True` to use floating-point computation for speed (see Remark 23.1). The computed entries of $\mathbf{K}^{(e)}$ are transformed to the nearest rational number (exact integers in this case) using the built-in function `Rationalize`. The strange value of $E = 48 \times 63 \times 13 \times 107 = 4206384$, in conjunction with $\nu = 1/3$, makes all entries of $\mathbf{K}^{(e)}$ exact integers when computed with the first 4 Gauss rules. This device facilitates visual comparison between the computed stiffness matrices:

1 x 1 rule

$$\mathbf{K}^{(e)}_{1\times1} = \begin{bmatrix} 1840293 & 1051596 & -262899 & -262899 & -1840293 & -1051596 & 262899 & 262899 \\ 1051596 & 3417687 & -262899 & 1314495 & -1051596 & -3417687 & 262899 & -1314495 \\ -262899 & -262899 & 1051596 & -525798 & 262899 & 262899 & -1051596 & 525798 \\ -262899 & 1314495 & -525798 & 1051596 & 262899 & -1314495 & 525798 & -1051596 \\ -1840293 & -1051596 & 262899 & 262899 & 1840293 & 1051596 & -262899 & -262899 \\ -1051596 & -3417687 & 262899 & -1314495 & 1051596 & 3417687 & -262899 & 1314495 \\ 262899 & 262899 & -1051596 & 525798 & -262899 & -262899 & 1051596 & -525798 \\ 262899 & -1314495 & 525798 & -1051596 & -262899 & 1314495 & -525798 & 1051596 \end{bmatrix}$$

2 x 2 rule

$$\mathbf{K}^{(e)}_{2\times2} = \begin{bmatrix} 2062746 & 1092042 & -485352 & -303345 & -1395387 & -970704 & -182007 & 182007 \\ 1092042 & 3761478 & -303345 & 970704 & -970704 & -2730105 & 182007 & -2002077 \\ -485352 & -303345 & 1274049 & -485352 & -182007 & 182007 & -606690 & 606690 \\ -303345 & 970704 & -485352 & 1395387 & 182007 & -2002077 & 606690 & -364014 \\ -1395387 & -970704 & -182007 & 182007 & 2730105 & 1213380 & -1152711 & -424683 \\ -970704 & -2730105 & 182007 & -2002077 & 1213380 & 4792851 & -424683 & -60669 \\ -182007 & 182007 & -606690 & 606690 & -1152711 & -424683 & 1941408 & -364014 \\ 182007 & -2002077 & 606690 & -364014 & -424683 & -60669 & -364014 & 2426760 \end{bmatrix}$$

3 x 3 rule

$$\mathbf{K}^{(e)}_{3\times3} = \begin{bmatrix} 2067026 & 1093326 & -489632 & -304629 & -1386827 & -968136 & -190567 & 179439 \\ 1093326 & 3764046 & -304629 & 968136 & -968136 & -2724969 & 179439 & -2007213 \\ -489632 & -304629 & 1278329 & -484068 & -190567 & 179439 & -598130 & 609258 \\ -304629 & 968136 & -484068 & 1397955 & 179439 & -2007213 & 609258 & -358878 \\ -1386827 & -968136 & -190567 & 179439 & 2747225 & 1218516 & -1169831 & -429819 \\ -968136 & -2724969 & 179439 & -2007213 & 1218516 & 4803123 & -429819 & -70941 \\ -190567 & 179439 & -598130 & 609258 & -1169831 & -429819 & 1958528 & -358878 \\ 179439 & -2007213 & 609258 & -358878 & -429819 & -70941 & -358878 & 2437032 \end{bmatrix}$$

4 x 4 rule

$$\mathbf{K}^{(e)}_{4\times4} = \begin{bmatrix} 2067156 & 1093365 & -489762 & -304668 & -1386567 & -968058 & -190827 & 179361 \\ 1093365 & 3764124 & -304668 & 968058 & -968058 & -2724813 & 179361 & -2007369 \\ -489762 & -304668 & 1278459 & -484029 & -190827 & 179361 & -597870 & 609336 \\ -304668 & 968058 & -484029 & 1398033 & 179361 & -2007369 & 609336 & -358722 \\ -1386567 & -968058 & -190827 & 179361 & 2747745 & 1218672 & -1170351 & -429975 \\ -968058 & -2724813 & 179361 & -2007369 & 1218672 & 4803435 & -429975 & -71253 \\ -190827 & 179361 & -597870 & 609336 & -1170351 & -429975 & 1959048 & -358722 \\ 179361 & -2007369 & 609336 & -358722 & -429975 & -71253 & -358722 & 2437344 \end{bmatrix}$$

Eigenvalues of stiffness matrices obtained by different Gauss rules:

Rule	Eigenvalues (scaled by 10^{-6}) of $\mathbf{K}^{(e)}$							
1×1	8.77276	3.68059	2.26900	0	0	0	0	0
2×2	8.90944	4.09769	3.18565	2.64528	1.54678	0	0	0
3×3	8.91237	4.11571	3.19925	2.66438	1.56155	0	0	0
4×4	8.91246	4.11627	3.19966	2.66496	1.56199	0	0	0

Rank deficient
by two

The formation of the trapezoidal element stiffness using floating-point computation by setting `numer=True` took 0.017, 0.083, 0.15 and 0.25 seconds for $p = 1, 2, 3, 4$, respectively, on a Mac G4/867. Changing `numer=False` to do exact computation increases the formation time to 0.033, 1.7, 4.4 and 44.6 seconds, respectively. (The unusually high value for $p = 4$ is due to the time spent in the simplification of the highly complex exact expressions produced by the Gauss quadrature rule.) This underscores the speed advantage of using floating-point arithmetic when exact symbolic and algebraic calculations are not required.

001 MODULO VECTOR FUERZAS NODALES CONSISTENTES PARA CARGAS EN CUERPO — CURSO 2005-06

The module `Quad4IsoPMembraneBodyForces` listed in Figure 23.8 computes the consistent force associated with a body force field $\vec{\mathbf{b}} = \{b_x, b_y\}$ given over a four-node iso-P quadrilateral in plane stress. The field is specified per unit of volume in componentwise form. For example if the element is subjected to a gravity acceleration field (self-weight) in the $-y$ direction, $b_x = 0$ and $b_y = -\rho g$, where ρ is the mass density.

The arguments of the module are are exactly the same as for `Quad4IsoPMembraneStiffness` except for the following differences.

- `mprop` — Not used; retained as placeholder.
- `bfor` — Body forces per unit volume. Specified as a two-item one-dimensional list: `{ bx,by }`, or as a four-entry two-dimensional list: `{ bx1,by1 },{ bx2,by2 }, { bx3,by3 },{ bx4,by4 }`. In the first form the body force field is taken to be constant over the element. The second form assumes body forces to vary over the element and specified by values at the four corners, from which the field is interpolated bilinearly.

The module returns `fe` as an 8×1 one dimensional array arranged `{ fx1,fy1,fx2,fy2,fx3,fy3, fx4,fy4 }` to represent the vector

$$\mathbf{f}^{(e)} = [\, f_{x1} \quad f_{y1} \quad f_{x2} \quad f_{y2} \quad f_{x3} \quad f_{y3} \quad f_{x4} \quad f_{y4} \,]^T . \tag{23.13}$$

```
Quad4IsoPMembraneBodyForces[ncoor_,mprop_,fprop_,options_,bfor_]:=
  Module[{i,k,p=2,numer=False,Emat,th=1,h,
    bx,by,bx1,by1,bx2,by2,bx3,by3,bx4,by4,bxc,byc,qcoor,
    c,w,Nf,dNx,dNy,Jdet,B,qctab,fe=Table[0,{8}]},
  If [Length[options]==2, {numer,p}=options, {numer}=options];
  If [Length[fprop]>0, th=fprop[[1]]];
  If [Length[bfor]==2,{bx,by}=bfor;bx1=bx2=bx3=bx4=bx;by1=by2=by3=by4=b
  If [Length[bfor]==4,{{bx1,by1},{bx2,by2},{bx3,by3},{bx4,by4}}=bfor];
  If [p<1||p>4, Print["p out of range"]; Return[Null]];
  bxc={bx1,bx2,bx3,bx4}; byc={by1,by2,by3,by4};
  For [k=1, k<=p*p, k++,
       {qcoor,w}= QuadGaussRuleInfo[{p,numer},k];
       {Nf,dNx,dNy,Jdet}=Quad4IsoPShapeFunDer[ncoor,qcoor];

       bk=Flatten[Table[{Nf[[i]]*bx,Nf[[i]]*by},{i,4}]];
       fe+=c*bk;
      ]; Return[fe]
   ];
```

001 MODULO TENSIONES EN NODOS ESQUINA CURSO 2005-06

Akthough the subject of stress recovery is treated in further detail in a later chapter, for completeness a stress computation module for the 4-node quad is shown in Figure 23.10.

The arguments of the module are are exactly the same as for `Quad4IsoPMembraneStiffness` except for the following differences.

`fprop` Not used; retained as placeholder.

`udis` The 8 corner displacements components. these may be specified as a 8-entry one-dimensional list form:
`{ux1,uy1, ux2,uy2, ux3,uy3, ux4,uy4}`,
or as a 4-entry two-dimensional list:
`{ux1,uy1},{ux2,uy2},{ux3,uy3},{ux4,uy4}`.

The module returns the corner stresses stored in a 4-entry, two-dimensional list:
`{{sigxx1,sigyy1,sigxy1},{sigxx2,sigyy2,sigxy2}, {sigxx3,sigyy3,sigxy3}, {sigxx4,sigyy4,sigxy4}}` to represent the stress array

$$\sigma^{(e)} = \begin{bmatrix} \sigma_{xx1} & \sigma_{xx2} & \sigma_{xx3} & \sigma_{xx4} \\ \sigma_{yy1} & \sigma_{yy2} & \sigma_{yy3} & \sigma_{yy4} \\ \sigma_{xy1} & \sigma_{xy2} & \sigma_{xy3} & \sigma_{xy4} \end{bmatrix} \tag{23.14}$$

The stresses are directly evaluated at the corner points without invoking any smoothing procedure. A more elaborated recovery scheme is presented in a later Chapter.

```
Quad4IsoPMembraneStresses[ncoor_,mprop_,fprop_,options_,udis_]:=
  Module[{i,k,numer=False,Emat,th=1,h,qcoor,Nf,
    dNx,dNy,Jdet,B,qctab,ue=udis,sige=Table[0,{4},{3}]},
  qctab={{-1,-1},{1,-1},{1,1},{-1,1}};
  Emat=mprop[[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[ncoor,qcoor];
        B={ 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.(B.ue);
      ]; Return[sige]
   ];
```

EXERCISE 23.1

[C:15] Figures E23.1–2 show the *Mathematica* implementation of the stiffness modules for the 5-node, "bilinear+bubble" iso-P quadrilateral of Figure E18.3. Module `Quad5IsoPMembraneStiffness` returns the 10×10 stiffness matrix whereas module `Quad5IsoPShapeFunDer` returns shape function values and Cartesian derivatives. (The Gauss quadrature module is reused.) Both modules follow the style of the 4-node quadrilateral implementation listed in Figures 23.4–5. The only differences in argument lists is that `ncoor` has five node coordinates: `{{x1,y1},{x2,y2},{x3,y3},{x4,y4},{x5,y5}}`, and that a variable plate thickness in `fprop` (one of the 3 possible formats) is specified as `{h1,h2,h3,h4,h5}`.

```
Quad5IsoPMembraneStiffness[ncoor_,mprop_,fprop_,options_]:=
  Module[{i,j,k,p=2,numer=False,Emat,th=1,h,qcoor,c,w,Nf,
   dNx,dNy,Jdet,B,Ke=Table[0,{10},{10}]},
  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}=Quad5IsoPShapeFunDer[ncoor,qcoor];
        If [Length[th]==0, h=th, h=th.Nf]; c=w*Jdet*h;
        B={ Flatten[Table[{dNx[[i]],       0},{i,5}]],
            Flatten[Table[{0,       dNy[[i]]},{i,5}]],
            Flatten[Table[{dNy[[i]],dNx[[i]]},{i,5}]]};
        Ke+=Simplify[c*Transpose[B].(Emat.B)];
      ]; Return[Ke];
   ];
```

```
Quad5IsoPShapeFunDer[ncoor_,qcoor_]:= Module[
  {Nf,dNx,dNy,dNξ,dNη,Nb,dNbξ,dNbη,J11,J12,J21,J22,Jdet,ξ,η,x,y},
  {ξ,η}=qcoor; Nb=(1-ξ^2)*(1-η^2); (* Nb: node-5 "bubble" function *)
   dNbξ=2*ξ(η^2-1); dNbη=2*η*(ξ^2-1);
   Nf= { ((1-ξ)*(1-η)-Nb)/4,((1+ξ)*(1-η)-Nb)/4,
         ((1+ξ)*(1+η)-Nb)/4,((1-ξ)*(1+η)-Nb)/4, Nb};
   dNξ={-(1-η+dNbξ)/4, (1-η-dNbξ)/4,
         (1+η-dNbξ)/4,-(1+η+dNbξ)/4, dNbξ};
   dNη={-(1-ξ+dNbη)/4,-(1+ξ+dNbη)/4,
         (1+ξ-dNbη)/4, (1-ξ-dNbη)/4, dNbη};
   x=Table[ncoor[[i,1]],{i,5}]; y=Table[ncoor[[i,2]],{i,5}];
   J11=dNξ.x; J21=dNξ.y; J12=dNη.x; J22=dNη.y;
   Jdet=Simplify[J11*J22-J12*J21];
   dNx= ( J22*dNξ-J21*dNη)/Jdet;  dNx=Simplify[dNx];
   dNy= (-J12*dNξ+J11*dNη)/Jdet;  dNy=Simplify[dNy];
   Return[{Nf,dNx,dNy,Jdet}]
];
```

Test `Quad5IsoPMembraneStiffness` for the 2:1 rectangular element studied in §23.3.1, with node 5 placed at the element center. Use Gauss rules 1×1, 2×2 and 3×3. Take $E = 96 \times 30 = 2880$ in lieu of $E = 96$ to get exact integer entries in $\mathbf{K}^{(e)}$ for all Gauss rules while keeping $\nu = 1/3$ and $h = 1$. Report on which rules give rank sufficiency. Partial result: $K_{22} = 3380$ and 3588 for the 2×2 and 3×3 rules, respectively.

001 **EJERCICIO 2** **CURSO 2005-06**

EXERCISE 23.4

[C:25] Implement the 9-node biquadratic element for plane stress to get its 18×18 stiffness matrix. Follow the style of Figures 23.3–4 or E23.1–2. (The Gauss quadrature module may be reused without change.) Test it for the 2:1 rectangular element studied in §23.3.1, with nodes 5–8 placed at the side midpoints, and node 9 at the element center. For the elastic modulus take $E = 96\times39\times11\times55\times7 = 15855840$ instead of $E = 96$, along with $\nu = 1/3$ and $h = 1$, so as to get exact integer entries in $\mathbf{K}^{(e)}$. Use both 2×2 and 3×3 Gauss integration rules and show that the 2×2 rule produces a rank deficiency of 3 in the stiffness. (If the computation with `num=False` takes too long on a slow PC, set `num=True` and `Rationalize` entries as in Figure 23.8.) Partial result: $K_{11} = 5395390$ and 6474468 for the 2×2 and 3×3 rules, respectively.