

Introduction to FEM

24

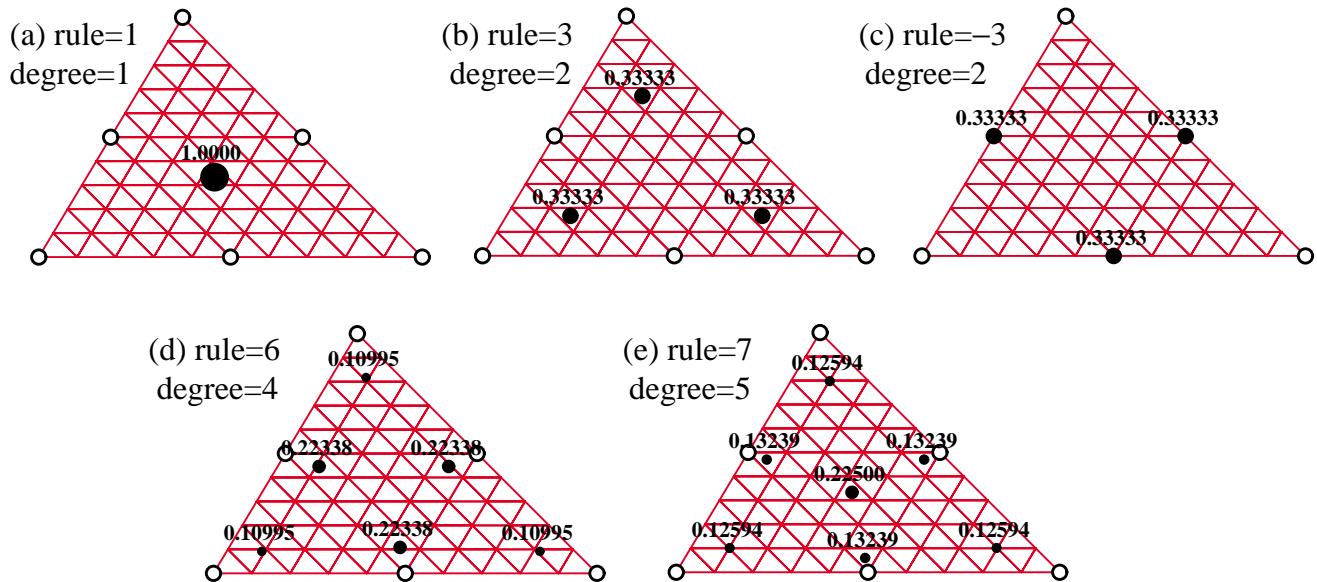
Implementation of Iso-P Triangular Elements

Peculiarities of Implementation of Iso-P Triangular Elements

- Special Gauss quadrature rules
- Computation of Jacobian and shape function x-y derivatives

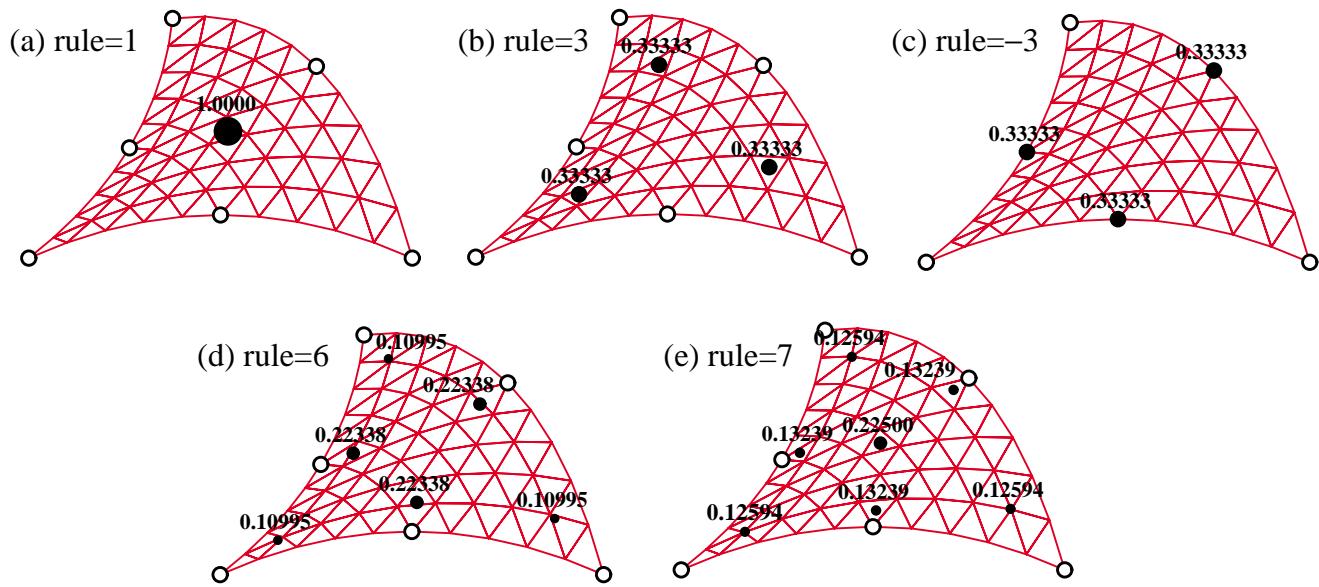
The 5 Simplest Gauss Rules Drawn over Straight Sided Triangles

(number annotated near sample point is the weight)



The 5 Simplest Gauss Rules Drawn over Arbitrary (Curved Side) Triangles

(number annotated near sample point is the weight)



Gauss Rules for Straight Sided Triangles

1-point (centroid) rule

$$\frac{1}{A} \int_{\Omega^{(e)}} F(\xi_1, \xi_2, \xi_3) d\Omega^{(e)} \approx F\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right),$$

3-internal-point rule

$$\frac{1}{A} \int_{\Omega^{(e)}} F(\xi_1, \xi_2, \xi_3) d\Omega^{(e)} \approx \frac{1}{3}F\left(\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\right) + \frac{1}{3}F\left(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right) + \frac{1}{3}F\left(\frac{1}{6}, \frac{1}{6}, \frac{2}{3}\right).$$

3 midpoint rule

$$\frac{1}{A} \int_{\Omega^{(e)}} F(\xi_1, \xi_2, \xi_3) d\Omega^{(e)} \approx \frac{1}{3}F\left(\frac{1}{2}, \frac{1}{2}, 0\right) + \frac{1}{3}F\left(0, \frac{1}{2}, \frac{1}{2}\right) + \frac{1}{3}F\left(\frac{1}{2}, 0, \frac{1}{2}\right).$$

In the above A is the triangle area

$$A = \int_{\Omega^{(e)}} d\Omega^{(e)} = \frac{1}{2} \det \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix}$$

For 6 and 7 point rules see notes

Introduction to FEM

Triangles of Arbitrary Geometry (e.g. Curved Sides)

Introduce the Jacobian determinant

$$d\Omega^{(e)} = J d\zeta_1 d\zeta_2 d\zeta_3, \quad J = \frac{1}{2} \det \begin{bmatrix} 1 & 1 & 1 \\ \sum_{i=1}^n x_i \frac{\partial N_i}{\partial \zeta_1} & \sum_{i=1}^n x_i \frac{\partial N_i}{\partial \zeta_2} & \sum_{i=1}^n x_i \frac{\partial N_i}{\partial \zeta_3} \\ \sum_{i=1}^n y_i \frac{\partial N_i}{\partial \zeta_1} & \sum_{i=1}^n y_i \frac{\partial N_i}{\partial \zeta_2} & \sum_{i=1}^n y_i \frac{\partial N_i}{\partial \zeta_3} \end{bmatrix}$$

Centroid rule

$$\int_{\Omega^{(e)}} F(\zeta_1, \zeta_2, \zeta_3) d\Omega^{(e)} \approx J\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) F\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

Midpoint rule

$$\begin{aligned} \int_{\Omega^{(e)}} F(\zeta_1, \zeta_2, \zeta_3) d\Omega^{(e)} &\approx \frac{1}{3} J\left(\frac{1}{2}, \frac{1}{2}, 0\right) F\left(\frac{1}{2}, \frac{1}{2}, 0\right) + \frac{1}{3} J\left(0, \frac{1}{2}, \frac{1}{2}\right) F\left(0, \frac{1}{2}, \frac{1}{2}\right) \\ &\quad + \frac{1}{3} J\left(\frac{1}{2}, 0, \frac{1}{2}\right) F\left(\frac{1}{2}, 0, \frac{1}{2}\right). \end{aligned}$$

etc

Triangle Gauss Quadrature Module

```

TrigGaussRuleInfo[{rule_,numer_},point_]:= Module[
{zeta,p=rule,i=point,g1,g2,info=NULL},
  If [p== 1, info={{1/3,1/3,1/3},1}],
  If [p==-3, zeta={1/2,1/2,1/2}; zeta[[i]]=0; info={zeta,1/3}];
  If [p== 3, zeta={1/6,1/6,1/6}; zeta[[i]]=2/3; info={zeta,1/3}];
  If [p== 6,
    If [i<=3, g1=(8-Sqrt[10]+Sqrt[38-44*Sqrt[2/5]])/18;
      zeta={g1,g1,g1}; zeta[[i]]=1-2*g1;
      info={zeta,(620+Sqrt[213125-53320*Sqrt[10]])/3720}];
    If [i>3, g2=(8-Sqrt[10]-Sqrt[38-44*Sqrt[2/5]])/18;
      zeta={g2,g2,g2}; zeta[[i-3]]=1-2*g2;
      info={zeta,(620-Sqrt[213125-53320*Sqrt[10]])/3720}];
  If [p== 7,
    If [i==1,info={{1/3,1/3,1/3},9/40} ];
    If [i>1&&i<=4,zeta=Table[(6-Sqrt[15])/21,{3}];
      zeta[[i-1]]=(9+2*Sqrt[15])/21;
      info={zeta,(155-Sqrt[15])/1200}];
    If [i>4, zeta=Table[(6+Sqrt[15])/21,{3}];
      zeta[[i-4]]=(9-2*Sqrt[15])/21;
      info={zeta,(155+Sqrt[15])/1200}];
  If [numer, Return[N[info]], Return[Simplify[info]]];
];

```

rule = defines Gauss rule (1, 3, -3, 6 and 7 implemented)

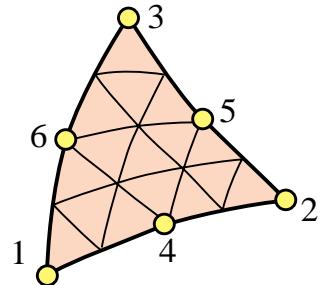
numer = True or False to get numeric or exact info

point = index of Gauss point, ranges from 1 through Abs[rule]

returns abscissa $\{\zeta_1, \zeta_2, \zeta_3\}$ and weight

Computation of Shape Function Derivatives

Illustrated for 6-node quadratic triangle:



Recall the iso-P element definition:

$$\begin{bmatrix} 1 \\ x \\ y \\ u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ u_{x1} & u_{x2} & u_{x3} & u_{x4} & u_{x5} & u_{x6} \\ u_{y1} & u_{y2} & u_{y3} & u_{y4} & u_{y5} & u_{y6} \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \end{bmatrix}$$

$$N_1 = \zeta_1(2\zeta_1 - 1), \quad N_4 = 4\zeta_1\zeta_2,$$

$$N_2 = \zeta_2(2\zeta_2 - 1), \quad N_5 = 4\zeta_2\zeta_3,$$

$$N_3 = \zeta_3(2\zeta_3 - 1), \quad N_6 = 4\zeta_3\zeta_1.$$

Partial Derivative Computation (Cont'd)

$$w = [w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6] \begin{bmatrix} \zeta_1(2\zeta_1 - 1) \\ \zeta_2(2\zeta_2 - 1) \\ \zeta_3(2\zeta_3 - 1) \\ 4\zeta_1\zeta_2 \\ 4\zeta_2\zeta_3 \\ 4\zeta_3\zeta_1 \end{bmatrix}$$

where w can be any interpolated quantity. Then

$$\frac{\partial w}{\partial x} = \sum w_i \frac{\partial N_i}{\partial x} = \sum w_i \left(\frac{\partial N_i}{\partial \zeta_1} \frac{\partial \zeta_1}{\partial x} + \frac{\partial N_i}{\partial \zeta_2} \frac{\partial \zeta_2}{\partial x} + \frac{\partial N_i}{\partial \zeta_3} \frac{\partial \zeta_3}{\partial x} \right)$$

$$\frac{\partial w}{\partial y} = \sum w_i \frac{\partial N_i}{\partial y} = \sum w_i \left(\frac{\partial N_i}{\partial \zeta_1} \frac{\partial \zeta_1}{\partial y} + \frac{\partial N_i}{\partial \zeta_2} \frac{\partial \zeta_2}{\partial y} + \frac{\partial N_i}{\partial \zeta_3} \frac{\partial \zeta_3}{\partial y} \right)$$

Partial Derivative Computation (Cont'd)

$$\left[\sum w_i \frac{\partial N_i}{\partial \zeta_1} \quad \sum w_i \frac{\partial N_i}{\partial \zeta_2} \quad \sum w_i \frac{\partial N_i}{\partial \zeta_3} \right] \begin{bmatrix} \frac{\partial \zeta_1}{\partial x} & \frac{\partial \zeta_1}{\partial y} \\ \frac{\partial \zeta_2}{\partial x} & \frac{\partial \zeta_2}{\partial y} \\ \frac{\partial \zeta_3}{\partial x} & \frac{\partial \zeta_3}{\partial y} \end{bmatrix} = \left[\frac{\partial w}{\partial x} \quad \frac{\partial w}{\partial y} \right]$$

Make $w \equiv 1, x, y$ and stack the results row-wise

$$\begin{bmatrix} \sum \frac{\partial N_i}{\partial \zeta_1} & \sum \frac{\partial N_i}{\partial \zeta_2} & \sum \frac{\partial N_i}{\partial \zeta_3} \\ \sum x_i \frac{\partial N_i}{\partial \zeta_1} & \sum x_i \frac{\partial N_i}{\partial \zeta_2} & \sum x_i \frac{\partial N_i}{\partial \zeta_3} \\ \sum y_i \frac{\partial N_i}{\partial \zeta_1} & \sum y_i \frac{\partial N_i}{\partial \zeta_2} & \sum y_i \frac{\partial N_i}{\partial \zeta_3} \end{bmatrix} \begin{bmatrix} \frac{\partial \zeta_1}{\partial x} & \frac{\partial \zeta_1}{\partial y} \\ \frac{\partial \zeta_2}{\partial x} & \frac{\partial \zeta_2}{\partial y} \\ \frac{\partial \zeta_3}{\partial x} & \frac{\partial \zeta_3}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial 1}{\partial x} & \frac{\partial 1}{\partial y} \\ \frac{\partial x}{\partial x} & \frac{\partial x}{\partial y} \\ \frac{\partial y}{\partial x} & \frac{\partial y}{\partial y} \end{bmatrix}$$

Partial Derivative Computation (Cont'd)

As shown in Notes, this system reduces to

$$\begin{bmatrix} 1 & 1 & 1 \\ \sum x_i \frac{\partial N_i}{\partial \xi_1} & \sum x_i \frac{\partial N_i}{\partial \xi_2} & \sum x_i \frac{\partial N_i}{\partial \xi_3} \\ \sum y_i \frac{\partial N_i}{\partial \xi_1} & \sum y_i \frac{\partial N_i}{\partial \xi_2} & \sum y_i \frac{\partial N_i}{\partial \xi_3} \end{bmatrix} \begin{bmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} \\ \frac{\partial \xi_3}{\partial x} & \frac{\partial \xi_3}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$


unknowns are grouped here

Partial Derivative Computation (Cont'd)

Replacing the shape functions of the 6-node triangle,

$$\begin{bmatrix} 1 & 1 \\ x_1(4\zeta_1 - 1) + 4x_4\zeta_2 + 4x_6\zeta_3 & x_2(4\zeta_2 - 1) + 4x_5\zeta_3 + 4x_4\zeta_1 \\ y_1(4\zeta_1 - 1) + 4y_4\zeta_2 + 4y_6\zeta_3 & y_2(4\zeta_2 - 1) + 4y_5\zeta_3 + 4y_4\zeta_1 \end{bmatrix} \begin{bmatrix} \frac{\partial \zeta_1}{\partial x} & \frac{\partial \zeta_1}{\partial y} \\ \frac{\partial \zeta_2}{\partial x} & \frac{\partial \zeta_2}{\partial y} \\ \frac{\partial \zeta_3}{\partial x} & \frac{\partial \zeta_3}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

or

$$\mathbf{JP} = \mathbf{R}$$

where \mathbf{J} and \mathbf{R} are known.

Partial Derivative Computation (Cont'd)

Solve $\mathbf{J} \mathbf{P} = \mathbf{R}$ for the partials

$$\mathbf{P} = \begin{bmatrix} \frac{\partial \zeta_1}{\partial x} & \frac{\partial \zeta_1}{\partial y} \\ \frac{\partial \zeta_2}{\partial x} & \frac{\partial \zeta_2}{\partial y} \\ \frac{\partial \zeta_3}{\partial x} & \frac{\partial \zeta_3}{\partial y} \end{bmatrix}$$

Plug these in the chain rule to get the x - y partial derivatives of the shape functions, and use these to form the strain-displacement matrix \mathbf{B}

Shape Function Module for 6-Node Triangle

```

Trig6IsoPShapeFunDer[ncoor_,tcoor_]:= Module[
{ξ1,ξ2,ξ3,x1,x2,x3,x4,x5,x6,y1,y2,y3,y4,y5,y6,
dx4,dx5,dx6,dy4,dy5,dy6,Jx21,Jx32,Jx13,Jy12,Jy23,Jy31,
Nf,dNx,dNy,Jdet}, {ξ1,ξ2,ξ3}=tcoor;
{{x1,y1},{x2,y2},{x3,y3},{x4,y4},{x5,y5},{x6,y6}}=ncoor;
dx4=x4-(x1+x2)/2; dx5=x5-(x2+x3)/2; dx6=x6-(x3+x1)/2;
dy4=y4-(y1+y2)/2; dy5=y5-(y2+y3)/2; dy6=y6-(y3+y1)/2;
Nf={ξ1*(2*ξ1-1),ξ2*(2*ξ2-1),ξ3*(2*ξ3-1),4*ξ1*ξ2,4*ξ2*ξ3,4*ξ3*ξ1};
Jx21= x2-x1+4*(dx4*(ξ1-ξ2)+(dx5-dx6)*ξ3);
Jx32= x3-x2+4*(dx5*(ξ2-ξ3)+(dx6-dx4)*ξ1);
Jx13= x1-x3+4*(dx6*(ξ3-ξ1)+(dx4-dx5)*ξ2);
Jy12= y1-y2+4*(dy4*(ξ2-ξ1)+(dy6-dy5)*ξ3);
Jy23= y2-y3+4*(dy5*(ξ3-ξ2)+(dy4-dy6)*ξ1);
Jy31= y3-y1+4*(dy6*(ξ1-ξ3)+(dy5-dy4)*ξ2);
Jdet = Jx21*Jy31-Jy12*Jx13;
dNx= {(4*ξ1-1)*Jy23,(4*ξ2-1)*Jy31,(4*ξ3-1)*Jy12,4*(ξ2*Jy23+ξ1*Jy31),
4*(ξ3*Jy31+ξ2*Jy12),4*(ξ1*Jy12+ξ3*Jy23)}/Jdet;
dNy= {(4*ξ1-1)*Jx32,(4*ξ2-1)*Jx13,(4*ξ3-1)*Jx21,4*(ξ2*Jx32+ξ1*Jx13),
4*(ξ3*Jx13+ξ2*Jx21),4*(ξ1*Jx21+ξ3*Jx32)}/Jdet;
Return[Simplify[{Nf,dNx,dNy,Jdet}]];
];

```

ncoor = x - y node coordinates

tcoor = $\{\zeta_1, \zeta_2, \zeta_3\}$ of point at which S.F.s are to be evaluated

returns shape functions, x - y derivatives and Jacobian determinant

Element Stiffness Module

```

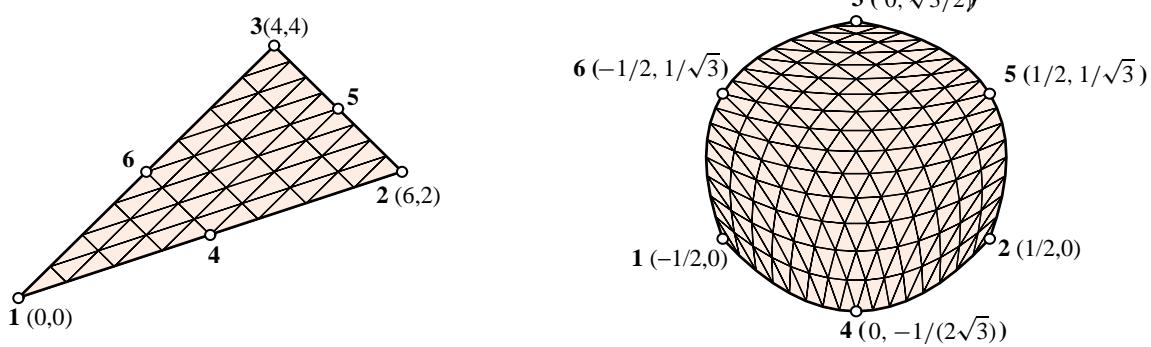
Trig6IsoPMembraneStiffness[ncoor_,mprop_,fprop_,opt_]:=Module[{i,k,l,p=3,num=False,Emat,th={fprop},h,tcoor,w,c,Nf,dNx,dNy,Jdet,Be,Ke=Table[0,{12},{12}]},Emat=mprop[[1]]; If [Length[fprop]>0, th=fprop[[1]]]; If [Length[opt]>0, num=opt[[1]]]; If [Length[opt]>1, p= opt[[2]]]; If [p!=-3&&p!=1&&p!=3&&p!=7, Print["Illegal p"];Return[Null]]; For [k=1, k<=Abs[p], k++, {tcoor,w}=TrigGaussRuleInfo[{p,num},k]; {Nf,dNx,dNy,Jdet}=Trig6IsoPShapeFunDer[ncoor,tcoor]; If [Length[th]==0, h=th, h=th.Nf]; c=w*Jdet*h/2; Be={Flatten[Table[{dNx[[i]], 0},{i,6}]], Flatten[Table[{0, dNy[[i]]},{i,6}]], Flatten[Table[{dNy[[i]],dNx[[i]]},{i,6}]]}; Ke+=c*Transpose[Be].(Emat.Be)]; Return[Ke]
];

```

For argument & function-return description see Notes

Test of 6-Node Triangle Stiffness Module

Two geometries tested



Mathematica test statements for left element (superparametric triangle) with $E = 288$, $\nu=1/3$ and $h = 1$

```
ClearAll[Em,nu,h]; h=1; Em=288; nu=1/3;
ncoor={{0,0},{6,2},{4,4},{3,1},{5,3},{2,2}};
Emat=Em/(1-nu^2)*{{1,nu,0},{nu,1,0},{0,0,(1-nu)/2}};
Ke=Trig6IsoPMembraneStiffness[ncoor,{Emat,0,0},{h},{False,-3}];
Ke=Simplify[Ke]; Print[Chop[Ke]//MatrixForm];
Print["eigs of Ke=",Chop[Eigenvalues[N[Ke]]]];
```

Test of 6-Node Superparametric Triangle

Computed stiffness matrix for Gauss integration rules $p = 3, -3, 6$ or 7
 (the $p = 1$ rule returns a rank deficient matrix):

$$\begin{bmatrix} 54 & 27 & 18 & 0 & 0 & 9 & -72 & 0 & 0 & 0 & 0 & -36 \\ 27 & 54 & 0 & -18 & 9 & 36 & 0 & 72 & 0 & 0 & -36 & -144 \\ 18 & 0 & 216 & -108 & 54 & -36 & -72 & 0 & -216 & 144 & 0 & 0 \\ 0 & -18 & -108 & 216 & -36 & 90 & 0 & 72 & 144 & -360 & 0 & 0 \\ 0 & 9 & 54 & -36 & 162 & -81 & 0 & 0 & -216 & 144 & 0 & -36 \\ 9 & 36 & -36 & 90 & -81 & 378 & 0 & 0 & 144 & -360 & -36 & -144 \\ -72 & 0 & -72 & 0 & 0 & 0 & 576 & -216 & 0 & -72 & -432 & 288 \\ 0 & 72 & 0 & 72 & 0 & 0 & -216 & 864 & -72 & -288 & 288 & -720 \\ 0 & 0 & -216 & 144 & -216 & 144 & 0 & -72 & 576 & -216 & -144 & 0 \\ 0 & 0 & 144 & -360 & 144 & -360 & -72 & -288 & -216 & 864 & 0 & 144 \\ 0 & -36 & 0 & 0 & 0 & -36 & -432 & 288 & -144 & 0 & 576 & -216 \\ -36 & -144 & 0 & 0 & -36 & -144 & 288 & -720 & 0 & 144 & -216 & 864 \end{bmatrix}$$

Eigenvalues of stiffness matrix:

[1971.66 1416.75 694.82 545.72 367.7 175.23 157.68 57.54 12.899 0 0 0]

which verifies that the computed stiffness has the correct rank.

Test of "Parabolic Circle" Triangle for 4 Rules

Mathematica test statements with $E = 504$, $v=0$ and $h = 1$ going over 4 rank-sufficient integration rules (3, -3, 6 and 7):

```
ClearAll[Em,nu,h]; Em=7*72; nu=0; h=1;
{x1,y1}={-1,0}/2; {x2,y2}={1,0}/2; {x3,y3}={0,Sqrt[3]}/2;
{x4,y4}={0,-1/Sqrt[3]}/2; {x5,y5}={1/2,1/Sqrt[3]}; {x6,y6}={-1/2,1/Sqrt[3]};
ncoor= {{x1,y1},{x2,y2},{x3,y3},{x4,y4},{x5,y5},{x6,y6}};
Emat=Em/(1-nu^2)*{{1,nu,0},{nu,1,0},{0,0,(1-nu)/2}};
For [i=2,i<=5,i++, p={1,-3,3,6,7}[[i]];
  Ke=Trig6IsoPMembraneStiffness[ncoor,{Emat,0,0},{h},{True,p}];
  Ke=Chop[Simplify[Ke]];
  Print["Ke=",SetPrecision[Ke,4]//MatrixForm];
  Print["Eigenvalues of Ke=",Chop[Eigenvalues[N[Ke]],.0000001]]
];

```

For stiffness matrices see Notes. The eigenvalues are

Rule	Eigenvalues of $\mathbf{K}^{(e)}$											
3	702.83	665.11	553.472	553.472	481.89	429.721	429.721	118.391	118.391	0	0	0
-3	1489.80	1489.80	702.833	665.108	523.866	523.866	481.890	196.429	196.429	0	0	0
6	1775.53	1775.53	896.833	768.948	533.970	533.970	495.570	321.181	321.181	0	0	0
7	1727.11	1727.11	880.958	760.719	532.750	532.750	494.987	312.123	312.123	0	0	0