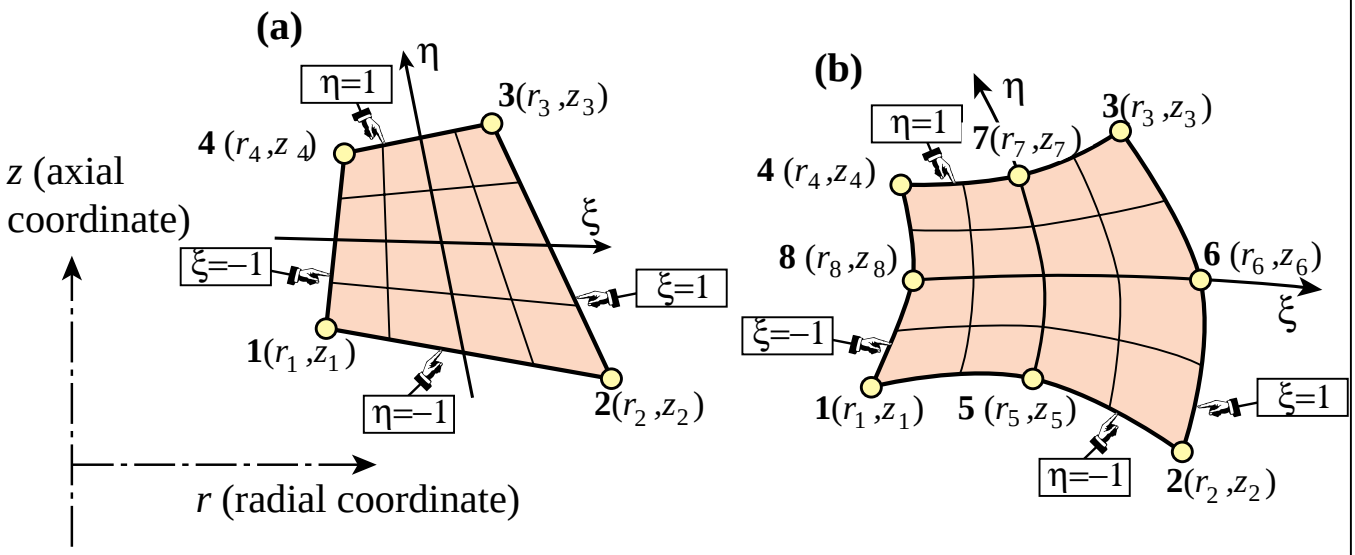
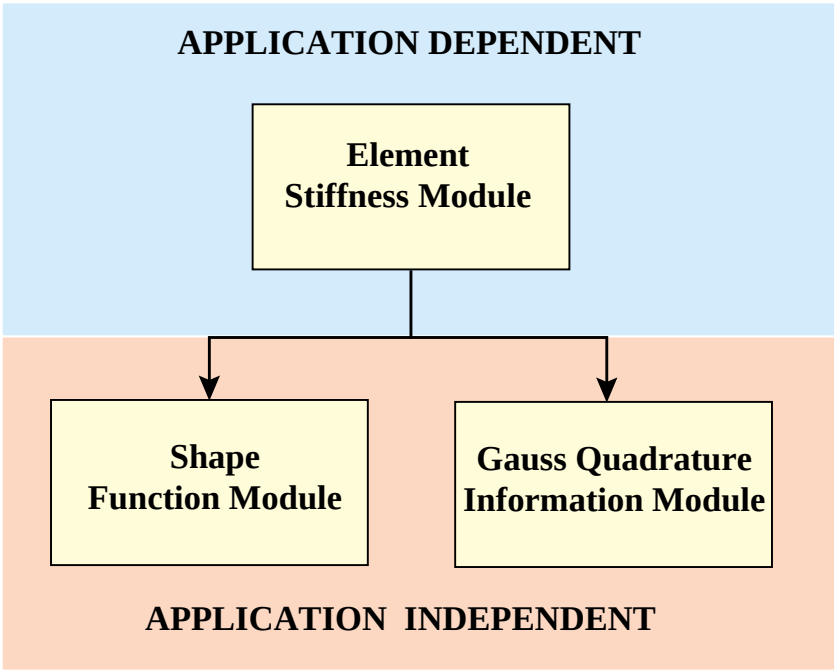
12

Iso-P Quadrilateral Ring Elements

The 4-Node and 8-Node Iso-P Quadrilateral Ring Elements



Organization of Stiffness Computation



Gauss Quadrature Rule Information Modules

```
QuadGaussRuleInfo[{rule_,numer_},point_]:= Module[
 {ξ,η,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) ];
  {ξ,w1}=  LineGaussRuleInfo[{p1,numer},i1];
  {η,w2}=  LineGaussRuleInfo[{p2,numer},i2];
  info={{ξ,η},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=point,info=Null},
  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]]];
];
```

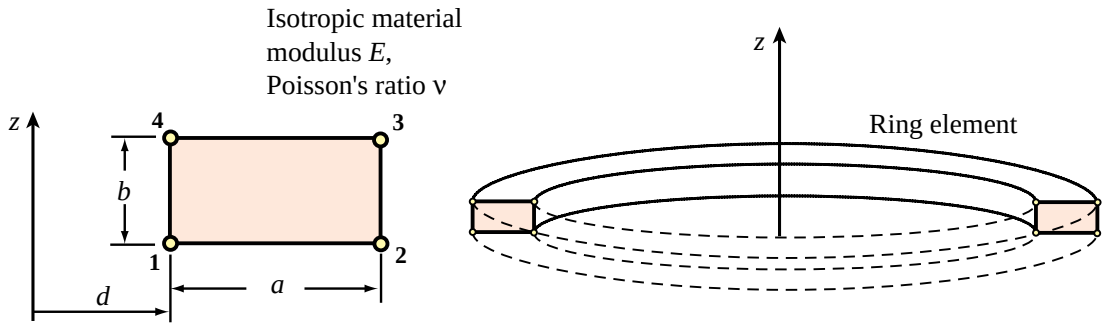
Shape Function Module

```
Quad4IsoPShapeFunDer[ncoor_,qcoor_]:= Module[
  {Nf,dNr,dNz,dNξ,dNη,i,J11,J12,J21,J22,Jdet,ξ,η,
   r1,r2,r3,r4,z1,z2,z3,z4,r,z},
  {ξ,η}=qcoor; {{r1,z1},{r2,z2},{r3,z3},{r4,z4}}=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;
   r={r1,r2,r3,r4}; z={z1,z2,z3,z4};
   J11=dNξ.r; J12=dNξ.z; J21=dNη.r; J22=dNη.z;
   Jdet=Simplify[J11*J22-J12*J21];
   dNr= ( J22*dNξ-J12*dNη)/Jdet;  dNr=Simplify[dNr];
   dNz= (-J21*dNξ+J11*dNη)/Jdet;  dNz=Simplify[dNz];
   Return[{Nf,dNr,dNz,Jdet}]
];
```

Stiffness Computation Module

```
Quad4IsoPRingStiffness[ncoor_,Emat_,Kfac_,options_]:=
  Module[{k,p=2,numer=False,qcoor,w,c,N1,N2,N3,N4,
    dNr1,dNr2,dNr3,dNr4,dNz1,dNz2,dNz3,dNz4,Jdet,
    r1,r2,r3,r4,z1,z2,z3,z4,rk,Be,Ke=Table[0,{8},{8}],
    modname="Quad4IsoPRingStiffness: "},
  If [Length[options]==2,{numer,p}=options,{numer}=options];
  If [p<1||p>4,Print[modname,"p out of range"];Return[Null]];
  {{r1,z1},{r2,z2},{r3,z3},{r4,z4}}=ncoor;
  For [k=1,k<=p*p,k++,
      {qcoor,w}= QuadGaussRuleInfo[{p,numer},k];
      {{N1,N2,N3,N4},{dNr1,dNr2,dNr3,dNr4},{dNz1,dNz2,
      dNz3,dNz4},Jdet}=Quad4IsoPShapeFunDer[ncoor,qcoor];
      rk=r1*N1+r2*N2+r3*N3+r4*N4;  c=Kfac*w*Jdet*rk;
      Be={{dNr1,   0,dNr2,   0,dNr3,   0,dNr4,   0},
          {   0,dNz1,   0,dNz2,   0,dNz3,   0,dNz4},
          {N1/rk,  0,N2/rk,  0,N3/rk,  0,N4/rk,  0},
          {dNz1,dNr1,dNz2,dNr2,dNz3,dNr3,dNz4,dNr4}};
      If [numer,Be=N[Be]];
      Ke+=Simplify[c*Transpose[Be].(Emat.Be)];
      ]; Return[Ke]
   ];
```

Test Element for Stiffness Computation



Test Element for Stiffness Computation

$$\mathbf{K}^{e}_{1\times 1} = \begin{bmatrix} 72 & 18 & 36 & -18 & -36 & -18 & 0 & 18 \\ 18 & 153 & -54 & 135 & -90 & -153 & -18 & -135 \\ 36 & -54 & 144 & -90 & 72 & 54 & -36 & 90 \\ -18 & 135 & -90 & 153 & -54 & -135 & 18 & -153 \\ -36 & -90 & 72 & -54 & 144 & 90 & 36 & 54 \\ -18 & -153 & 54 & -135 & 90 & 153 & 18 & 135 \\ 0 & -18 & -36 & 18 & 36 & 18 & 72 & -18 \\ 18 & -135 & 90 & -153 & 54 & 135 & -18 & 153 \end{bmatrix}$$

$$\mathbf{K}^{e}_{2\times 2} = \begin{bmatrix} 168 & -12 & 24 & 12 & -24 & -36 & 48 & 36 \\ -12 & 108 & -24 & 84 & -72 & -102 & -36 & -90 \\ 24 & -24 & 216 & -120 & 0 & 72 & -24 & 72 \\ 12 & 84 & -120 & 300 & -72 & -282 & 36 & -102 \\ -24 & -72 & 0 & -72 & 216 & 120 & 24 & 24 \\ -36 & -102 & 72 & -282 & 120 & 300 & -12 & 84 \\ 48 & -36 & -24 & 36 & 24 & -12 & 168 & 12 \\ 36 & -90 & 72 & -102 & 24 & 84 & 12 & 108 \end{bmatrix}$$

Test Element for Stiffness Computation

$$\mathbf{K}^{e\gamma}_{3\times 3} = \begin{bmatrix} 232 & -12 & 24 & 12 & -24 & -36 & 80 & 36 \\ -12 & 108 & -24 & 84 & -72 & -102 & -36 & -90 \\ 24 & -24 & 216 & -120 & 0 & 72 & -24 & 72 \\ 12 & 84 & -120 & 300 & -72 & -282 & 36 & -102 \\ -24 & -72 & 0 & -72 & 216 & 120 & 24 & 24 \\ -36 & -102 & 72 & -282 & 120 & 300 & -12 & 84 \\ 80 & -36 & -24 & 36 & 24 & -12 & 232 & 12 \\ 36 & -90 & 72 & -102 & 24 & 84 & 12 & 108 \end{bmatrix}$$

$$\mathbf{K}^{e}_{4\times 4} = \begin{bmatrix} 280 & -12 & 24 & 12 & -24 & -36 & 104 & 36 \\ -12 & 108 & -24 & 84 & -72 & -102 & -36 & -90 \\ 24 & -24 & 216 & -120 & 0 & 72 & -24 & 72 \\ 12 & 84 & -120 & 300 & -72 & -282 & 36 & -102 \\ -24 & -72 & 0 & -72 & 216 & 120 & 24 & 24 \\ -36 & -102 & 72 & -282 & 120 & 300 & -12 & 84 \\ 104 & -36 & -24 & 36 & 24 & -12 & 280 & 12 \\ 36 & -90 & 72 & -102 & 24 & 84 & 12 & 108 \end{bmatrix}$$

Rule	Eigenvalues of $\mathbf{K}^e$ for varying integration rule							
1×1	667.794	180.000	124.206	72.000	0	0	0	0
2×2	745.201	261.336	248.750	129.451	100.389	88.598	10.275	0
3×3	745.446	330.628	266.646	133.236	126.343	98.690	11.011	0
4×4	745.716	397.372	272.092	144.542	135.004	101.908	11.365	0

Consistent Body Force Module

```
Quad4IsoPRingBodyForces[ncoor_,Emat_,Kfac_,options_,bfor_]:=
  Module[{k,m,p=2,numer=False,qcoor,w,c,N1,N2,N3,N4,dNr,dNz,
    Jdet,r1,r2,r3,r4,z1,z2,z3,z4,rk,br,bz,br1,bz1,br2,bz2,
    br3,bz3,br4,bz4,brc,bzc,bk,fe=Table[0,{8}],
    modname="Quad4IsoPRingBodyForces: "}, m=Length[bfor];
  If [Length[options]==2,{numer,p}=options,{numer}=options];
  If [p<1||p>4,Print[modname,"p out of range"];Return[Null]];
  If [m==2,{br,bz}=bfor;br1=br2=br3=br4=br;bz1=bz2=bz3=bz4=bz];
  If [m==4,{{br1,bz1},{br2,bz2},{br3,bz3},{br4,bz4}}=bfor];
  {{r1,z1},{r2,z2},{r3,z3},{r4,z4}}=ncoor;
  For [k=1,k<=p*p,k++,
      {qcoor,w}= QuadGaussRuleInfo[{p,numer},k];
      {{N1,N2,N3,N4},dNr,dNz,Jdet}=Quad4IsoPShapeFunDer[ncoor,qcoor];
      rk=r1*N1+r2*N2+r3*N3+r4*N4; c=Kfac*w*Jdet*rk;
      brk=br1*N1+br2*N2+br3*N3+br4*N4;
      bzk=bz1*N1+bz2*N2+bz3*N3+bz4*N4;
      bk={N1*brk,N1*bzk,N2*brk,N2*bzk,N3*brk,N3*bzk,N4*brk,N4*bzk};
      If [numer,bk=N[bk]]; fe+=c*bk;
      ]; Return[Simplify[fe]]
];
```

Consistent Body Force Module Testing

```
ClearAll[Em,ν,a,b,d,h,p,num];
a=4; b=2; d=0; Kfac=2*Pi; Kfac=1; p=2;
ncoor={{d,0},{a+d,0},{a+d,b},{d,b}}; num=False;
fe=Quad4IsoPRingBodyForces[ncoor,Emat,Kfac,{num,p},9*{br,bz}];
Print["fe =",fe];
fe=Quad4IsoPRingBodyForces[ncoor,Emat,Kfac,{num,p},
  9*{{a*br,bz},{(a+d)*br,bz},{(a+d)*br,bz},{d*br,bz}}];
Print["fe =",fe];
```

$$\mathbf{f}^e = [\,24b_r \quad 24b_z \quad 48b_r \quad 48b_z \quad 48b_r \quad 48b_z \quad 24b_r \quad 24b_z\,]^T$$

$$\mathbf{f}^e = [\,80b_r \quad 24b_z \quad 176b_r \quad 48b_z \quad 160b_r \quad 48b_z \quad 64b_r \quad 24b_z\,]^T$$

Stress Recovery Module

```
Quad4IsoPRingStresses[ncoor_,Emat_,Kfac_,options_,ue_]:=
  Module[{k,m,numer=False,qcoor,w,c,N1,N2,N3,N4,
    dNr1,dNr2,dNr3,dNr4,dNz1,dNz2,dNz3,dNz4,Jdet,
    r1,r2,r3,r4,z1,z2,z3,z4,rk,Be,recovery="D",
    sigp,qctab={{-1,-1},{1,-1},{1,1},{-1,1}},
    TG={{1+Sqrt[3]/2,-1/2, 1-Sqrt[3]/2,-1/2},
        {-1/2, 1+Sqrt[3]/2,-1/2,1-Sqrt[3]/2},
        { 1-Sqrt[3]/2,-1/2,1+Sqrt[3]/2,-1/2},
        {-1/2, 1-Sqrt[3]/2,-1/2,1+Sqrt[3]/2}},
    sige=Table[0,{4},{3}]},
  m=Length[options]; If [m>0,numer=options[[1]]];
  If [m>1,recovery= options[[2]]];
  If [recovery=="G", qctab=qctab/Sqrt[3]];
  If [numer,qctab=N[qctab];TG=N[TG]];
  {{r1,z1},{r2,z2},{r3,z3},{r4,z4}}=ncoor;
  For [k=1,k<=Length[sige],k++, qcoor=qctab[[k]];
      If [numer, qcoor=N[qcoor]];
      {{N1,N2,N3,N4},{dNr1,dNr2,dNr3,dNr4},{dNz1,dNz2,
      dNz3,dNz4},Jdet}=Quad4IsoPShapeFunDer[ncoor,qcoor];
      rk=r1*N1+r2*N2+r3*N3+r4*N4;
      Be={{dNr1,   0,dNr2,   0,dNr3,   0,dNr4,   0},
          {   0,dNz1,   0,dNz2,   0,dNz3,   0,dNz4},
          {N1/rk, 0,N2/rk, 0,N3/rk, 0,N4/rk, 0},
          {dNz1,dNr1,dNz2,dNr2,dNz3,dNr3,dNz4,dNr4}};
      If [numer,Be=N[Be]];
      sige[[k]]=Simplify[Emat.(Be.ue)] ];
  If [recovery=="G", sigp=sige; sige=TG.sigp];
  Return[Simplify[sige]];
  ];
```

Stress Recovery Testing

```
ClearAll[Em,ν,a,b,d,p,num,err,ezz,grz,ur,uz];
Em=100; ν=0; a=5; b=2;  err=1/10; ezz=-5/100; grz=4/100;
ur[r_,z_]:=err*r; uz[r_,z_]:=ezz*z+grz*r;
ncoor={{d,0},{a+d,0},{a+d,b},{d,b}}; num=False;
Emat=Em/((1+ν)*(1-2*ν))*{{1+ν,ν,ν,0},{ν,1+ν,ν,0},
        {ν,ν,1+ν,0},{0,0,0,1/2-ν}};
        Print["Emat=",Emat//MatrixForm];
ue={}; For [n=1,n<=4,n++,{rn,zn}=ncoor[[n]]; urn=ur[rn,zn];
      uzn=uz[rn,zn];AppendTo[ue,urn]; AppendTo[ue,uzn] ];
Print["ue=",ue];
sige=Quad4IsoPRingStresses[ncoor,Emat,Kfac,{num,"D"},ue];
Print["corner stresses=",sige//MatrixForm];
```

$$\text{Emat}=\begin{pmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 50 \end{pmatrix}$$

$$\text{ue}=\left\{\frac{d}{10}, \frac{d}{25}, \frac{5+d}{10}, \frac{5+d}{25}, \frac{5+d}{10}, -\frac{1}{10}+\frac{5+d}{25}, \frac{d}{10}, -\frac{1}{10}+\frac{d}{25}\right\}$$

$$\text{corner stresses}=\begin{pmatrix} 10 & -5 & 10 & 2 \\ 10 & -5 & 10 & 2 \\ 10 & -5 & 10 & 2 \\ 10 & -5 & 10 & 2 \end{pmatrix}$$