ASEN 6367 Advanced FEM Spring '09 Quiz I Solutions

QUESTION 1.

(a) dx is the infinitesimal change in the independent variable x, whereas $dy = y'dx$ is the corresponding change in the dependent variable (assuming y' exists). The variation δy, which is not necessarily an infinitesimal, is a change in y for a fixed x when $y(x)$ is varied to $y(x) + \delta y(x)$. When performing the variation x is kept fixed. $\delta\Pi$ is the change in $\Pi[y]$ when y is varied by δy: $\delta\Pi = \Pi[y(x) + \delta y(x)] - \Pi[y(x)]$.

(b) By definition of variation, the independent variable x is frozen when $y(x)$ is varied. Hence $\delta x = 0$.

(c) AFEM Notes, §1.6.

(d) Seven: TPE, TCPE, HR, VHW, and three unnamed ones with masters: strains, stresses-strains and displacements-strains. The most useful ones in FEM are TPE and HR; VHW, TCPE and displacements-strains have also been used but less.

(e) Principles with multiple interior masters, such as HR, are used in FEM to try to alleviate the deficiencies of TPE. These include: (1) higher accuracy for displacements than stresses (which follow by differentiation) and (2) taking care of incompressibility. Use of hybrid principles with assumed interior stresses and boundary displacements follow similar motivations.

(f) Problem (I): Some terms on the stiffness formulation contain $1/r$ induced by the hoop strain u_r/r, so evaluations at $r = 0$ (as may occur over an element side or corner) should be avoided. To preclude this, Gauss integration rules over elements should have only interior points. Problem (II): exact- or near-incompressibility causes the elasticity matrix **E** to "blow up". (For an isotropic material that happens if Poisson's ratio $\nu \to \frac{1}{2}$.) If this matrix appears in the stiffness formulation, as happens in TPE-based elements, stiffness computations containing **E** are affected.

QUESTION 2.

(a) Start from the stress assumption

$$\begin{bmatrix} \sigma_{rr} \\ \sigma_{zz} \\ \sigma_{\theta\theta} \\ \sigma_{rz} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \bar{r} & \bar{z} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \bar{r} & 0 & 0 & 0 & \bar{z} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \bar{z} & \bar{r} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \bar{r} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{11} \end{bmatrix}. \tag{Q1.10}$$

Here α_1 through α_{11} are 11 stress parameters, while $\bar{r} = r - r_0$, and $\bar{z} = z - z_0$, in which $r_0 = (r_1 + r_2 + r_3 + r_4)/4$ and $z_0 = (z_1 + z_2 + z_3 + z_4)/4$ are the r and z coordinates of the element center, respectively. Inserting (Q1.10) into the equilibrium equations of §9.3.3 with zero body forces, show that if one imposes that internal equilibrium is to be satisfied strongly, (Q1.10) reduces to

$$\begin{bmatrix} \sigma_{rr} \\ \sigma_{zz} \\ \sigma_{\theta\theta} \\ \sigma_{rz} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \bar{r} & \bar{z} & 0 & 0 \\ 0 & 1 & -\bar{z}/r & 0 & 0 & \bar{r} & -(r+\bar{r})\bar{z}/r \\ 1 & 0 & 0 & r+\bar{r} & \bar{z} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \bar{r} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_7 \end{bmatrix} \quad \text{or} \quad \sigma = \mathbf{Sa}. \tag{Q1.11}$$

```
ClearAll[α1,α2,α3,α4,α5,α6,α7,α8,α9,α10,α11,r,z,z0,r0];
rb=r-r0; zb=z-z0;
S10={{1,0,0,0, rb,zb,0, 0, 0, 0, 0},
     {0,1,0,0, 0, 0,rb, 0, 0, 0,zb},
     {0,0,1,0, 0, 0, 0, 0,zb,rb, 0},
     {0,0,0,1, 0, 0, 0,rb, 0, 0, 0}};
Print["S10=",S10//MatrixForm];
αvec10={α1,α2,α3,α4,α5,α6,α7,α8,α9,α10,α11};
{sigrr,sigzz,sigtt,sigrz}=Simplify[S10.αvec10];
eq1=(1/r)*D[r*sigrr,r]+D[sigrz,z]-sigtt/r;
eq2=(1/r)*D[r*sigrz,r]+D[sigzz,z];
eq1=Simplify[eq1]; eq2=Simplify[eq2];
Print["eq1=",eq1]; Print["eq2=",eq2];
sol=Simplify[Solve[{eq1==0,eq2==0},{α10,α11}]];
Print["solution: ",sol];
```

$$
S10=\begin{pmatrix} 1 & 0 & 0 & 0 & r-r0 & z-z0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & r-r0 & 0 & 0 & 0 & z-z0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & z-z0 & r-r0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & r-r0 & 0 & 0 & 0 \end{pmatrix}
$$

$$
eq1=\frac{\alpha1 - r\,\alpha10 + r0\,\alpha10 - \alpha3 + 2\,r\,\alpha5 - r0\,\alpha5 + z\,\alpha6 - z0\,\alpha6 - z\,\alpha9 + z0\,\alpha9}{r}
$$

$$
eq2=\frac{r\,\alpha11 + \alpha4 + 2\,r\,\alpha8 - r0\,\alpha8}{r}
$$

```
solution: {{α10 -> (α1 - α3 + 2*r*α5 - r0*α5 + z*α6 - z0*α6 - z*α9 + z0*α9)
 /(r - r0), α11 -> -((α4 + 2*r*α8 - r0*α8)/r)}}
```

Figure Q1.1. Script to do item (a) of Question 2.

The script shown in Figure Q1.1 does the item. It sets up the stress matrix (Q1.10) in S10, multiplies by the 11-vector of αs, gets stresses σ_{rr} through σ_{rz} and sets up the two equilibrium equations $(1/r)\partial(r\sigma_{rr})/\partial r + \partial\sigma_{rz}/\partial z - \sigma_{tt}/r = 0$ and $(1/r)\partial(r\sigma_{rz})/\partial r + \partial\sigma_{zz}/\partial z = 0$.

(to be completed)

(b) To form the flexibility matrix, the 4×10 **S** matrix is coded in *Mathematica* as shown in Figure Q1.2.

```
SMatrixQuad4StressHybrid[scoor_]:=Module[{r,z,r0,z0},
  {r,z,r0,z0}=scoor; rbar=r-r0; zbar=z-z0;
   S={{1,0,      0,  rbar,zbar,   0,               0},
      {0,1,-zbar/r,     0,   0,rbar,-(r+rbar)*zbar/r},
      {1,0,      0,r+rbar,zbar,   0,               0},
      {0,0,      1,     0,   0,   0,            rbar}};
   Return[S];
];
```

Figure Q1.2. Module that return stress matrix **S**

The module is referenced as

S=SMatrixQuad4StressHybrid[scoor];

where

scoor   List {r,z,r0,z0}, where r,z are the coordinates of the point at which **S** is to be evaluated, and r0,z0 are the coordinates of the center of the element.

S   List containing the 4×7 stress matrix.

The flexibility matrix **F** is formed by the module shown in Figure Q1.3.

```
FMatrixQuad4StressHybrid[ncoor_,Cmat_,opt_]:=
  Module[{i,k,l,p=2,num=False,h=1,qcoor,w,
    r1,r2,r3,r4,z1,z2,z3,z4,rc,zc,x,y,c,
    J11,J12,J21,J22,Jdet,Nf,Nξ,Nη,na,S,Fe},
  {{r1,z1},{r2,z2},{r3,z3},{r4,z4}}=ncoor;
  rc={r1,r2,r3,r4}; zc={z1,z2,z3,z4};
  r0=(r1+r2+r3+r4)/4; z0=(z1+z2+z3+z4)/4;
  {num,p,na}=opt;
  Fe=Table[0,{7},{7}];
  For [k=1, k<=p, k++,
     For [l=1, l<=p, l++,
        {{ξ,η},w}=QuadGaussRuleInfo[{p,num},{k,l}];
        Nf={(1-ξ)*(1-η),(1+ξ)*(1-η),(1+ξ)*(1+η),(1-ξ)*(1+η)}/4;
        Nξ= {-(1-η), (1-η),(1+η),-(1+η)}/4;
        Nη= {-(1-ξ),-(1+ξ),(1+ξ), (1-ξ)}/4;
        J11=Nξ.rc; J21=Nξ.zc; J12=Nη.rc; J22=Nη.zc;
        Jdet=Simplify[J11*J22-J12*J21];
        rg=Simplify[Nf.rc]; zg=Simplify[Nf.zc];
        S=SMatrixQuad4StressHybrid[{rg,zg,r0,z0}];
        Fe+=w*Jdet*rg*Transpose[S].(Cmat.S);
        ];
     ];  Return[Simplify[Fe]]
   ];
```

Figure Q1.3. Module that returns the flexibility matrix **F**

The module is referenced as

F=FMatrixQuad4StressHybrid[ncoor,Cmat,opt];

where

ncoor   Element node coordinates stored as {{r1,z1},{r2,z2},{r3,z3},{r4,z4}},

Cmat   Compliance matrix stored as {{C11,C12,C13,C14}, ... {C14,C24,C34,C44}}

opt   The list {num,p,na} of options. num is a logical flag: True to specify exact arithmetic, or False to specified floating-point airthmetic. p defines the Gauss integration rule: p=2, which specifies 2×2, is sufficient for this element. The last entry, na, is not used.

F   The 7×7 flexibility matrix.

The script to produce the flexibility matrix for the quiz element data and the output results are shown are shown in Figure Q1.4. All eigenvalues of **F** are nonzero and positive, which is a good check. The matrix has full rank.

(b) The computation of the connection matrix **G** is done with the help of the modules listed in Figures Q1.5 through Q1.8.

```
ClearAll[Em,ν,a,b,e,h,p,na]; na=7; p=2; Em=1000; ν=1/3;
a=4; b=10; h=2;
ncoor={{a,0},{b,0},{b,h},{a,h}};
Cmat={{1,-ν,-ν,0}, {-ν,1,-ν,0}, {-ν,-ν,1,0}, {0,0,0,2*(1+ν)}}/Em;
(*Print["Cmat=",Cmat//MatrixForm];*)
Fe=FMatrixQuad4StressHybrid[ncoor,Cmat,{False,p,na}];
Print["Fe=",Fe//MatrixForm];
Print["Fe=",Chop[N[Fe]]//MatrixForm];
Print["Eigs of Fe=",Simplify[Chop[Eigenvalues[N[Fe]]]]];
```

$$\text{Fe}=\begin{pmatrix} \frac{14}{125} & -\frac{7}{125} & 0 & \frac{58}{125} & 0 & -\frac{3}{125} & 0 \\ -\frac{7}{125} & \frac{21}{250} & 0 & -\frac{29}{125} & 0 & \frac{9}{250} & 0 \\ 0 & 0 & \frac{2583}{11500} & 0 & \frac{1}{375} & 0 & \frac{1147}{11500} \\ \frac{58}{125} & -\frac{29}{125} & 0 & \frac{147}{25} & 0 & -\frac{42}{125} & 0 \\ 0 & 0 & \frac{1}{375} & 0 & \frac{14}{375} & 0 & \frac{7}{375} \\ -\frac{3}{125} & \frac{9}{250} & 0 & -\frac{42}{125} & 0 & \frac{63}{250} & 0 \\ 0 & 0 & \frac{1147}{11500} & 0 & \frac{7}{375} & 0 & \frac{8071}{11500} \end{pmatrix}$$

$$\text{Fe}=\begin{pmatrix} 0.112 & -0.056 & 0 & 0.464 & 0 & -0.024 & 0 \\ -0.056 & 0.084 & 0 & -0.232 & 0 & 0.036 & 0 \\ 0 & 0 & 0.224609 & 0 & 0.00266667 & 0 & 0.0997391 \\ 0.464 & -0.232 & 0 & 5.88 & 0 & -0.336 & 0 \\ 0 & 0 & 0.00266667 & 0 & 0.0373333 & 0 & 0.0186667 \\ -0.024 & 0.036 & 0 & -0.336 & 0 & 0.252 & 0 \\ 0 & 0 & 0.0997391 & 0 & 0.0186667 & 0 & 0.701826 \end{pmatrix}$$

Eigs of Fe={5.94666, 0.72235, 0.235081, 0.204608, 0.11052, 0.0368093, 0.0357366}

Figure Q1.4. Computing **F** for the quiz element.

```
PMatrixQuad4StressHybrid[side_,sxi_]:=Module[
  {P=Table[0,{2},{8}],i,j},
  i={1,3,5,7}[[side]];j={3,5,7,1}[[side]];
  P[[1,i]]=P[[2,i+1]]=(1-sxi)/2;
  P[[1,j]]=P[[2,j+1]]=(1+sxi)/2;
  Return[P];
  ];
```

Figure Q1.5. Module to compute **P** for a quadrilateral element.

Figure Q1.5 list the module that returns matrix **P**. It is invoked as

`P=PMatrixQuad4StressHybrid[side,sxi];`

where

side Element side index: 1,2,3,4 for 1–2, 2–3, 3–4 and 4–1, respectively.

sxi Natural side coordinate.

P The 2×8 matrix that maps corner displacements to side displacements.

Figure Q1.6 list the module that returns matrix **T**. It is invoked as

`T=TMatrixQuad4StressHybrid[ncoor,ccoor,scoor];`

where

ncoor Element node coordinates stored as `{{r1,z1},{r2,z2},{r3,z3},{r4,z4}}`.

```
TMatrixQuad4StressHybrid[ncoor_,ccoor_,scoor_]:=Module[
  {r1,z1,r2,z2,r3,z3,r4,z4,rc,zc,r0,z0,r,z,
   rci,rcj,zci,zcj,Lij,S,T},
  {{r1,z1},{r2,z2},{r3,z3},{r4,z4}}=ncoor;
  rc={r1,r2,r3,r4}; zc={z1,z2,z3,z4};
  r0=(r1+r2+r3+r4)/4; z0=(z1+z2+z3+z4)/4;
  {{rci,zci},{rcj,zcj}}=ccoor;  {r,z}=scoor;
   Lij=PowerExpand[Sqrt[(rcj-rci)^2+(zcj-zci)^2]];
   cij=(zcj-zci)/Lij; sij=(rci-rcj)/Lij;
   S=SMatrixQuad4StressHybrid[{r,z,r0,z0}];
   T=Simplify[{{cij,0,0,sij},{0,sij,0,cij}}.S];
   Return[T];
];
```

Figure Q1.6. Module to compute **T** for a quadrilateral element.

ccoor   Coordinates of the corner nodes i, j of element side on which **T** will be evaluated: `{{rci,zci},{rc1,zcj}}`. This is used to compute side length and trig functions.

scoor   Coordinates `{r,z}` of the point i, j at which **T** will be evaluated.

T   The 2×7 matrix that maps tractions to stress parameters.

```
GMatrixQuad4StressHybrid[ncoor_,opt_]:=Module[
{rci,zci,rcj,zcj,L,scoor,num,p,na,sxi,w,rg,zg,side,k,T,G},
{num,p,na}=opt;   G=Table[0,{7},{8}];
 For [side=1,side<=4,side++,
   {rci,zci}=ncoor[[side]];
    If[side<4,{rcj,zcj}=ncoor[[side+1]],{rcj,zcj}=ncoor[[1]]];
    ccoor={{rci,zci},{rcj,zcj}};
    L=PowerExpand[Sqrt[(rcj-rci)^2+(zcj-zci)^2]];
    For [k=1,k<=p,k++,
       {sxi,w}=LineGaussRuleInfo[{p,num},k];
       If [num, {sxi,w}=N[{sxi,w}]];
       rg=Simplify[rci*(1-sxi)/2+rcj*(1+sxi)/2];
       zg=Simplify[zci*(1-sxi)/2+zcj*(1+sxi)/2];
       P=PMatrixQuad4StressHybrid[side,sxi];
       T=TMatrixQuad4StressHybrid[ncoor,ccoor,{rg,zg}];
       G+=w*rg*(L/2)*Simplify[Transpose[T].P];
       ];
    ]; Return[Simplify[G]];
  ];
```

Figure Q1.7. Module to compute **G** for a quadrilateral element.

Figure Q1.7 list the module that returns the connection matrix **G**. It is invoked as

`G=GMatrixQuad4StressHybrid[ncoor,opt];`

where

ncoor   Element node coordinates stored as `{{r1,z1},{r2,z2},{r3,z3},{r4,z4}}`.

opt   The list `{num,p,na}` of options. described above for `SMatrixQuad4StressHybrid`

G   The 7×8 connection matrix that links stress amplitudes to nodal displacements.

The script to produce the connection matrix for the quiz element data and the output results are shown are shown in Figure Q1.8. The matrix should have rank 7. The *mathematica* function `NullSpace[G]` is used to verify that requirement.

```
ClearAll[Em,nu,a,b,e,h,p,na]; p=2; a=4; b=10; h=2;
ncoor={{a,0},{b,0},{b,h},{a,h}};
G=GMatrixQuad4StressHybrid[ncoor,{False,p,na}];
Print["G=",G//MatrixForm];
Print["chk RBM=",Simplify[G.Transpose[{
        {0,1,0,1,0,1,0,1}}]]];
Print["null space of G=",NullSpace[G]//MatrixForm];
```

$$
G=\begin{pmatrix}
-4 & 0 & 10 & 0 & 10 & 0 & -4 & 0 \\
0 & -18 & 0 & -24 & 0 & 24 & 0 & 18 \\
-18 & -7 & -24 & 7 & 24 & 7 & 18 & -7 \\
12 & 0 & 30 & 0 & 30 & 0 & 12 & 0 \\
\frac{4}{3} & 0 & -\frac{10}{3} & 0 & \frac{10}{3} & 0 & -\frac{4}{3} & 0 \\
0 & 12 & 0 & -30 & 0 & 30 & 0 & -12 \\
12 & -3 & -30 & 3 & 30 & 3 & -12 & -3
\end{pmatrix}
$$

```
chk RBM={{0}, {0}, {0}, {0}, {0}, {0}, {0}}
null space of G=( 0  1  0  1  0  1  0  1 )
```

Figure Q1.8. Computing **G** for the quiz element.

(d) The computation of the element stiffness matrix **K** is done with the script of the script listed in Figure Q1.9. The result is shown in the bottom os the figure. The eigenvalue computation verifies that $\mathbf{K}^e$ has rank 7, with a sigle rigid body mode.

```
ClearAll[Em,ν,a,b,e,h,p,na]; na=7; p=2; Em=1000; ν=1/3;
a=4; b=10; h=2;
ncoor={{a,0},{b,0},{b,h},{a,h}};
Cmat={{1,-ν,-ν,0}, {-ν,1,-ν,0}, {-ν,-ν,1,0}, {0,0,0,2*(1+ν)}}/Em;
opt={False,p,na};
F=FMatrixQuad4StressHybrid[ncoor,Cmat,opt];
G=GMatrixQuad4StressHybrid[ncoor,opt];
Ke=Simplify[Transpose[G].Inverse[F].G];  kfac=(1-ν^2)*h*(b-a)*24;
Print["Ke=",N[Ke]//MatrixForm];
Print["eigs of Ke=",Chop[Eigenvalues[N[Ke]]]];
```

$$
Ke=\begin{pmatrix}
2512.07 & 1360.71 & 641.892 & 139.291 & -1888.82 & -1272.26 & -1644.75 & -227.743 \\
1360.71 & 7751.37 & -928.431 & 5748.63 & -2406.95 & -6185.33 & 227.743 & -7314.67 \\
641.892 & -928.431 & 4682.17 & -2821.57 & -1933.4 & 1343.05 & -1888.82 & 2406.95 \\
139.291 & 5748.63 & -2821.57 & 12251.4 & -1343.05 & -11814.7 & 1272.26 & -6185.33 \\
-1888.82 & -2406.95 & -1933.4 & -1343.05 & 4682.17 & 2821.57 & 641.892 & 928.431 \\
-1272.26 & -6185.33 & 1343.05 & -11814.7 & 2821.57 & 12251.4 & -139.291 & 5748.63 \\
-1644.75 & 227.743 & -1888.82 & 1272.26 & 641.892 & -139.291 & 2512.07 & -1360.71 \\
-227.743 & -7314.67 & 2406.95 & -6185.33 & 928.431 & 5748.63 & -1360.71 & 7751.37
\end{pmatrix}
$$

```
eigs of Ke={33404.9, 9042.57, 6890.45, 2593.67, 2229.9, 222.955, 9.5522, 0}
```

Figure Q1.9. Script computing the stiffness matrix $\mathbf{K}^e$ and results for the quiz element.

For completeness, Figure Q1.10 lists the Gauss integration rule modules used in the integration over the element domain and on its boundary.

```
QuadGaussRuleInfo[{rule_,numer_},point_]:= Module[
 {ξ,η,p1,p2,i,j,w1,w2,m,info={{Null,Null},0}},
  If [Length[rule]==2,  {p1,p2}=rule, p1=p2=rule];
  If [p1<0, Return[QuadNonProductGaussRuleInfo[
      {-p1,numer},point]]];
  If [Length[point]==2, {i,j}=point, m=point;
      j=Floor[(m-1)/p1]+1; i=m-p1*(j-1) ];
  {ξ,w1}=  LineGaussRuleInfo[{p1,numer},i];
  {η,w2}=  LineGaussRuleInfo[{p2,numer},j];
  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]},
   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,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]]];
];
```

Figure Q1.10.   Gauss integration rule module.