ME 712 Sp '05 Final Solns

1) a) Governing equations (linear stress strain? Also DE approximation to physics)

b) Material properties

c) geometry errors (unknowns, and inability of elements to match)

d) FEM is approximate solution (limited shape functions)

e) Numerical integration is inexact

f) Boundary condition model

g) Applied load model

h) Numerical errors in linear algebra soln.

2) a) OK

b) Mass doesn't add to >0 (add 1st row)

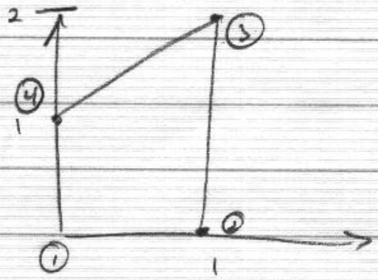
c) Negative eigenvalue (unstable) (negative stiffness)

d) Assymetric

3) $K = \frac{Et}{l}\begin{bmatrix} 5.5 & -6 & .5 \\ -6 & 8 & -2 \\ .5 & -2 & 1.5 \end{bmatrix}$

see next page

4)



From 6.2-11

$$\begin{bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_2}{\partial \xi} u_2 \\ \frac{\partial N_2}{\partial \eta} u_2 \\ \frac{\partial N_2}{\partial \xi} v_2 \\ \frac{\partial N_2}{\partial \eta} v_2 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} (1-\eta)u_2 \\ -(1+\xi)u_2 \\ (1-\eta)v_2 \\ -(1+\xi)v_2 \end{bmatrix}_{\xi=-1,\ \eta=-1} = \begin{bmatrix} \frac{1}{2} u_2 \\ 0\ u_2 \\ \frac{1}{2} v_2 \\ 0\ v_2 \end{bmatrix}_{\xi=-1,\ \eta=-1}$$

We need Jacobian @ $\xi=-1, \eta=-1$

$$J = \frac{1}{4}\begin{bmatrix} -2 & 2 & 0 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} .5 & 0 \\ 0 & .5 \end{bmatrix} \qquad \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} = J^{-1}\begin{bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{bmatrix} = \begin{bmatrix} u_2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{bmatrix} = J^{-1}\begin{bmatrix} \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{bmatrix} = \begin{bmatrix} v_2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} u_2 \\ 0 \\ v_2 \end{bmatrix}$$

```
In[1]:= J = {{1 / 2 (2 z - 1), -2 z, 1 / 2 (1 + 2 z)}}.{{0}, {L / 4}, {L}}
```

Out[1]= $\left\{\left\{-\frac{L\,z}{2} + \frac{1}{2} L\,(1 + 2\,z)\right\}\right\}$

```
In[2]:= J = Simplify[J[[1, 1]]]
```

Out[2]= $\frac{1}{2} L\,(1 + z)$

```
In[3]:= L = 1
```

Out[3]= 1

```
In[4]:= B = Simplify[(1 / J) * {{1 / 2 (2 z - 1), -2 z, 1 / 2 (1 + 2 z)}}]
```

Out[4]= $\left\{\left\{\frac{-1 + 2\,z}{1 + z},\ -\frac{4\,z}{1 + z},\ \frac{1 + 2\,z}{1 + z}\right\}\right\}$

```
In[5]:= K = MatrixForm[Simplify[(Transpose[B].B) * J /. z → 1 / Sqrt[3]] +
            Simplify[(Transpose[B].B) * J /. z → -1 / Sqrt[3]] // N]
```

Out[5]//MatrixForm=

$$\begin{pmatrix} 5.5 & -6. & 0.5 \\ -6. & 8. & -2. \\ 0.5 & -2. & 1.5 \end{pmatrix}$$

```
x1=0;x2=.25;x3=1;
x=[x1 x2 x3]';

K=zeros(3,3);
weights=[1 1];
gp=1/sqrt(3)*[-1 1];
np=9;[gp,weights]=gauss(np);
ngp=length(gp)

for i=1:ngp
  z=gp(i);w=weights(i);
  J=[1/2*(2*z-1) -2*z 1/2*(1+2*z)]*x;
  B=1/J*[1/2*(2*z-1) -2*z 1/2*(1+2*z)];
  B'*B
  B'*B*J
  K=K+w*B'*B*J;
end
K
```

5) a) 1) Guyan reduce Lagrange and constrained DOFs

2) Shift K

3) Subspace iteration

see code

```
% We don't have a lot of time to get fancy. Let's just use Guyan to
% remove DOFs 7-42, and the Lagrange 385-420 DOFs. Alternatively,
% we could try ditching the first 6 as well.
load  ~/websites/jslater/me712/Exams/mk.mat
Mmm=M([1:6,43:384],[1:6, 43:384]);
Kmm=K([1:6,43:384],[1:6, 43:384]);
Msm=M([7:42,385:420],[1:6,43:384]);
Ksm=K([7:42,385:420],[1:6,43:384]);
Mms=Msm';
Kms=Ksm';
Mss=M([7:42,385:420],[7:42,385:420]);
Kss=K([7:42,385:420],[7:42,385:420]);
T=[eye(size(Kmm)); -Kss\Kms'];

Mred=T'*[Mmm Mms;Msm Mss]*T;
Kred=T'*[Kmm Kms;Ksm Kss]*T;

X1=rand(size(Mred,1),20)-.5;
%Shift of 1.
shift=1;
Kredp=Kred+shift*Mred;
KinvM=real(Kredp\Mred);
%not required
%%%%%%%%%%%%%%
error=1;
i=0;
%%%%%%%%%%%%%%%%
% I did this with a for loop and "watched" during my "exam conditions"
while error>.00001
%%%%%%%%%%%%
  i=i+1;
  %%%%%%%%%%%%%%
  X2=real(KinvM*X1);
  Ksmall=X2'*Kredp*X2;
  Msmall=X2'*Mred*X2;
  disp('Msmall eig')
  min(eig(Msmall))
  [v,d]=eig(Msmall\Ksmall);
  X1=X2*v;
  freqs=sort(sqrt(real(diag(d))-shift)/2/pi)
  %not required
  if i>1
    i
      size(freqs)
  size(oldfreqs)

    error=abs((freqs(11)-oldfreqs(11))/freqs(11))
  end
  oldfreqs=freqs;
  size(freqs)
  size(oldfreqs)

  %%%%%%%%%%%%
  X1=X1/norm(X1);
```

```
    %If you don't normalize somehow, it breaks down due to
    %ill-conditioning pretty quickly (but not before you have two
    %places in the eigenvalues.
end
%X1'*Kred*X1
max(abs(X1'*Kred*X1-diag(X1'*Kred*X1)))
max(abs(X1'*Mred*X1-diag(X1'*Mred*X1)))
```