ME 712 Sp '11 Final Solns

1) 2x Shear + 1 tension, each face (18 checks)
2x bending each face (6 checks)
Pos-semidefinite 6 zero eigenvalues
1 torsion, each face (will not match C-T soln) (3 checks)
Mass is PD
Mass in each direction is same
Cube element has axis independence (redundant if matched closed-form)
Moment of inertia appropriate (3 checks + 6 products of inertia)
Patch test

2) Node 1 fixed but for rotation about z

3) 1st is zero, rigid body rotation about z
2nd is 0.011 Hz, bending ⊥ to z axis
3rd is 0.011 Hz, bending ⊥ to 2nd and z axis
See script. See next page

4) Mass moment of inertia about z axis divided by 1st element of mode shape squared.
1.46×10^5 kg m^2

5) See next page

Power Method - pre-shift (p3 extra)

$$(K - M\lambda)\,\underline{u} = \underline{0}$$

$$\left((K+\nu M) - M(\lambda+\nu)\right)\underline{u} = \underline{0}$$

$$\left[(\lambda+\nu)^{-1} I - \underbrace{(K+\nu M)^{-1} M}\right]\underline{u} = 0$$

$$A = (K+\nu M) \backslash M$$

$$\underline{x}_{i+1} = A\,\underline{x}_i$$

Iterate, with normalization.

Then shift A to shift $(\lambda+\nu)^{-1}$

$$A_{new} = A - \underline{x}_{i+1}\,\underline{x}_{i+1}^{T}\,(\lambda+\nu)_I$$

↑ 1st eig of A

```
disp('Problem 3')
det(Kr)
disp('That commonly fails')
disp('It is a good idea to assume rigid body modes, so always shift')
mu=1;
K=Kr+mu*Mr;%Shift eigenvalues by 1
A=K\Mr;
x=diag(Mr);
x=x/norm(x)
for i=1:10000

xn=A*x;
invlamplusmu=norm(xn)
x=xn/norm(xn);
end
u1=x;
lam1=1/invlamplusmu-mu
w1=sqrt(lam1)
A1=A-x*x'*invlamplusmu;
for i=1:10000

xn=A1*x;
invlamplusmu=norm(xn)
x=xn/norm(xn);
end
u2=x;
lam2=1/invlamplusmu-mu
w2=sqrt(lam2)
disp('These match matlab eigenanalysis.')


disp('Problem 4')
u1=u1/u1(1,1);
u1'*Kr*u1

u1'*Mr*u1
```

```
%Find shape functions
%ax^3+bx^2+cx+d
b=[-1 1 -1 1;%u(-1)
   -1/27 1/9 -1/3 1;%u(-1/3)
   1/27 1/9 1/3 1;%u(1/3)
   1 1 1 1]%u(1)
neval=eye(4)
N=b\neval
N1=N(:,1)',polyder(N1)
N2=N(:,2)',polyder(N2)
N3=N(:,3)',polyder(N3)
N4=N(:,4)',polyder(N4)

K=zeros(4,4)
w=[5 8 5]/9
x=[-sqrt(.6) 0 sqrt(.6)]
for i=1:3
   B=[polyval(polyder(N1),x(i)),polyval(polyder(N2),x(i)),polyval(polyder(N3),x(i)),polyval↙
(polyder(N4),x(i))]
   K=B'*B*w(i)+K
end
disp('Multiply by J=2/l and EA gives answer')
```