Computer Implementation 7.2 *(Matlab)* *Notched beam using mapped quadrilateral elements (p. 516)*

In *Matlab*, the element equations for a quadrilateral element for a plane stress and plane strain problems can be generated in a manner similar to those presented for 2D BVP in Chapter 6. The following PlaneQuad4Element, PlaneQuad4LoadTerm and PlaneQuad4Results functions are developed for four node quadrilateral elements using 2×2 integration. Similar functions for 8 node quadrilateral element can easily be written.

MatlabFiles\Chap7\PlaneQuad4Element.m

```
function [k, r] = PlaneQuad4Element(type, e, nu, h, alpha, deltaT, bx, by, coord)
% [k, r] = PlaneQuad4Element(e, nu, h, alpha, deltaT, bx, by, coord)
% Generates for a triangular element for plane stress or plane strain problem
% e = Modulus of elasticity
% nu = Poisson's ratio
% h = Thickness
% alpha = coefficient of thermal expansion
% deltaT = temperature change
% bx, by = components of the body force
% coord = coordinates at the element ends

switch (type)
case 1
    e0 = alpha*deltaT*[1; 1; 0];
    c = e/(1 - nu^2)*[1, nu, 0; nu, 1, 0; 0, 0, (1 - nu)/2];
case 2
    e0 = (1 + nu)*alpha*deltaT*[1; 1; 0];
    c = e/((1 + nu)*(1 - 2*nu))*[1 - nu, nu, 0; nu, 1 - nu, 0;
        0, 0, (1 - 2*nu)/2];
end

% Use 2x2 integration. Gauss point locations and weights
pt=1/sqrt(3);
gpLocs = [-pt,-pt; -pt,pt; pt,-pt; pt,pt];
gpWts = [1,1,1,1];
k=zeros(8); r=zeros(8,1);
for i=1:length(gpWts)
    s = gpLocs(i, 1); t = gpLocs(i, 2); w = gpWts(i);
    n = [(1/4)*(1 - s)*(1 - t), (1/4)*(s + 1)*(1 -t), ...
            (1/4)*(s + 1)*(t + 1), (1/4)*(1 - s)*(t + 1)];
    dns=[(-1 + t)/4, (1 - t)/4, (1 + t)/4, (-1 - t)/4];
    dnt=[(-1 + s)/4, (-1 - s)/4, (1 + s)/4, (1 - s)/4];
    x = n*coord(:,1); y = n*coord(:,2);
    dxs = dns*coord(:,1); dxt = dnt*coord(:,1);
```

```
    dys = dns*coord(:,2); dyt = dnt*coord(:,2);
    J = [dxs, dxt; dys, dyt]; detJ = det(J);
    dnx = (J(2, 2)*dns - J(2, 1)*dnt)/detJ;
    dny = (-J(1, 2)*dns + J(1, 1)*dnt)/detJ;
    b = [dnx(1), 0, dnx(2), 0, dnx(3), 0, dnx(4), 0;
        0, dny(1), 0, dny(2), 0, dny(3), 0, dny(4);
        dny(1), dnx(1), dny(2), dnx(2), dny(3), dnx(3), dny(4), dnx(4)];
    n = [n(1),0,n(2),0,n(3),0,n(4),0;
        0,n(1),0,n(2),0,n(3),0,n(4)];
    k = k + h*detJ*w* b'*c*b;
    r = r + h*detJ*w*n'*[bx;by]+ h*detJ*w*b'*c*e0;
end
```

MatlabFiles\Chap7\PlaneQuad4Load.m

```
function rq = PlaneQuad4Load(side, qn, qt, h, coord)
% rq = PlaneQuad4Load(side, qn, qt, h, coord)
% Generates equivalent load vector for a triangular element
% side = side over which the load is specified
% qn, qt = load components in the normal and the tangential direction
% h = thickness
% coord = coordinates at the element ends

% Use 2 point integration. Gauss point locations and weights
pt=-1/sqrt(3);
gpLocs = [-pt, pt];
gpWts = [1,1];
rq=zeros(8,1);
for i=1:length(gpWts)
    a = gpLocs(i); w = gpWts(i);
    switch (side)
    case 1
        n = [(1 - a)/2, (1 + a)/2, 0, 0];
        dna = [-1/2, 1/2, 0, 0];
    case 2
        n = [0, (1 - a)/2, (1 + a)/2, 0];
        dna = [0, -1/2, 1/2, 0];
    case 3
        n = [0, 0, (1 - a)/2, (1 + a)/2];
        dna = [0, 0, -1/2, 1/2];
    case 4
        n = [(1 + a)/2, 0, 0, (1 - a)/2];
        dna = [1/2, 0, 0, -1/2];
    end
    dxa = dna*coord(:,1); dya = dna*coord(:,2);
    Jc=sqrt(dxa^2 + dya^2);
    nx = dya/Jc; ny = -dxa/Jc;
```

```
    qx = nx*qn - ny*qt;
    qy = ny*qn + nx*qt;
    n = [n(1),0,n(2),0,n(3),0,n(4),0;
        0,n(1),0,n(2),0,n(3),0,n(4)];
    rq = rq + h*Jc*w*n'*[qx; qy];
end
```

MatlabFiles\Chap7\PlaneQuad4Results.m

```
function se = PlaneQuad4Results(type, e, nu, alpha, deltaT, coord, dn)
% se = PlaneQuad4Results(type, e, nu, alpha, deltaT, coord, dn)
% Computes element solution for a plane stress/strain quad element
% e = modulus of elasticity
% nu = Poisson's ratio
% alpha = coefficient of thermal expansion
% deltaT = temperature change
% coord = nodal coordinates
% dn = nodal displacements
% Following are the output variables are at element center
% {strains, stresses, principal stresses, effective stress}
switch (type)
case 1
    e0 = alpha*deltaT*[1; 1; 0];
    c = e/(1 - nu^2)*[1, nu, 0; nu, 1, 0; 0, 0, (1 - nu)/2];
case 2
    e0 = (1 + nu)*alpha*deltaT*[1; 1; 0];
    c = e/((1 + nu)*(1 - 2*nu))*[1 - nu, nu, 0; nu, 1 - nu, 0;
        0, 0, (1 - 2*nu)/2];
end
s = 0; t = 0;
n = [(1/4)*(1 - s)*(1 - t), (1/4)*(s + 1)*(1 -t), ...
        (1/4)*(s + 1)*(t + 1), (1/4)*(1 - s)*(t + 1)];
dns=[(-1 + t)/4, (1 - t)/4, (1 + t)/4, (-1 - t)/4];
dnt=[(-1 + s)/4, (-1 - s)/4, (1 + s)/4, (1 - s)/4];
x = n*coord(:,1); y = n*coord(:,2);
dxs = dns*coord(:,1); dxt = dnt*coord(:,1);
dys = dns*coord(:,2); dyt = dnt*coord(:,2);
J = [dxs, dxt; dys, dyt]; detJ = det(J);
dnx = (J(2, 2)*dns - J(2, 1)*dnt)/detJ;
dny = (-J(1, 2)*dns + J(1, 1)*dnt)/detJ;
b = [dnx(1), 0, dnx(2), 0, dnx(3), 0, dnx(4), 0;
    0, dny(1), 0, dny(2), 0, dny(3), 0, dny(4);
    dny(1), dnx(1), dny(2), dnx(2), dny(3), dnx(3), dny(4), dnx(4)];
eps = b*dn;
sig = c*(eps-e0)
sx = sig(1); sy= sig(2); sxy=sig(3);
```

```
PrincipalStresses = eig([sx,sxy; sxy,sy])
se = sqrt((sx - sy)^2 + sy^2 + sx^2 + 6*sxy^2)/sqrt(2);
```

Using these functions finite element equations for any four node quadrilateral element for a plane stress or plane strain problem can easily be written. As an example we use these functions to solve the notched beam problem with three elements.

MatlabFiles\Chap7\PlaneQuad4Results.m

```
% Plane stress analysis of a notched beam
e = 3000*10^3; nu = 0.2; h = 4; q = 50;
nodes = [0, 5; 0, 12; 6, 0; 6, 5; 20, 0; 20, 12; 54, 0; 54, 12];
conn = [1, 4, 6, 2; 3, 5, 6, 4; 5, 7, 8, 6];
bx=0; by=0; alpha=0; deltaT = 0;
nel=size(conn,1); dof=2*size(nodes,1);
lmm=[];
for i=1:nel
   lm=[];
   for j=1:4
      lm=[lm, [2*conn(i,j)-1,2*conn(i,j)]];
   end
   lmm=[lmm; lm];
end
K=zeros(dof); R = zeros(dof,1);
% Generate equations for each element and assemble them.
for i=1:3
   con = conn(i,:);
   lm = lmm(i,:);
   [k, r] = PlaneQuad4Element(1, e, nu, h, alpha, deltaT, bx, by, nodes(con,:));
   K(lm, lm) = K(lm, lm) + k;
   R(lm) = R(lm) + r;
end
% Add the distributed load contributions
for i=1:2:3
   con = conn(i,:);
   lm = lmm(i,:);
   r = PlaneQuad4Load(3, -q, 0, h, nodes(con,:));
   R(lm) = R(lm) + r;
end
% Nodal solution and reactions
debc = [1,3,13,14,15,16]; ebcVals=zeros(length(debc),1);
[d, reactions] = NodalSoln(K, R, debc, ebcVals)
for i=1:3
   fprintf(1,'Results for element %3.0g \n',i)
   EffectiveStress=PlaneQuad4Results(1, e, nu, alpha, deltaT, ...
```

```
        nodes(conn(i,:),:), d(lmm(i,:)))
end

>> NotchedBeamEx

d =

            0
    -0.018316
            0
     -0.01832
    0.0027592
    -0.016649
    0.0011455
    -0.016463
      0.00305
    -0.011357
   -0.0021013
    -0.011625
            0
            0
            0
            0


reactions =

        -7932
        10361
       -19673
         5840
        17244
         4960

Results for element   1

sig =

      -104.57
       28.544
       166.98


PrincipalStresses =

      -217.77
       141.74
```

```
EffectiveStress =

   313.66

Results for element   2

sig =

  -22.085
  -19.167
  -43.656


PrincipalStresses =

  -64.306
   23.054


EffectiveStress =

   78.417

Results for element   3

sig =

    -50.6
  -43.717
   154.17


PrincipalStresses =

  -201.36
   107.05


EffectiveStress =

   271.22
```