Computational Structural Mechanics and Dynamics, Assignment 5

Jose Raul Bravo Martinez, MSc Computational Mechanics

March 12, 2019

Assignment 5.1
On "Convergence Requirements":

The isoparametric definitions of the straight-node bar element in its local system $\bar{x}$ is,

$$\begin{bmatrix} 1 \\ \bar{x} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} N_1^e(\xi) \\ N_2^e(\xi) \\ N_3^e(\xi) \end{bmatrix}$$

Here ξ is the isoparametric coordinate that takes the values -1, 1, 0 and 0 at nodes 1, 2 and 3 respectively, while N_1^e, N_2^e and N_3^e are the shape functions for a bar element.

For simplicity, take $x_1 = 0$, $x_2 = L$, $x_3 = \frac{1}{2}l + \alpha l$ Here l is the bar length and α a parameter that characterizes how far away is node 3 from the midpoint location $x = \frac{1}{2}l$.

Show that the minimum α (minimal in absolute value sense) for which $J = dx/d\xi$ at a point in the element are $\pm 1/4$ (the quarter points). Interpret this result as a singularity by showing that the axial strain becomes infinite at an end point.

Approximating the x coordinates as:

$$x(\xi) = x_1 N_1(\xi) + x_2 N_2(\xi) + x_3 N_3(\xi)$$

After substituting the expressions for the shape functions:

$$x(\xi) = \frac{1}{2}(x_1 - 2x_3 + x_2)\xi^2 + \frac{1}{2}(x_2 - x_1)\xi + x_3$$

Therefore, the expression of the Jacobian is:

$$J = \frac{dx}{d\xi} = (x_1 - 2x_3 + x_2)\xi + \frac{1}{2}(x_2 - x_1)$$

After substituting that $x_1 = 0$ and $x_2 = L$:

$$J = (-2\alpha l)\xi + \frac{L}{2}$$

Substituting that the Jacobian is zero, and the maximum value of $\xi = \pm 1$, one gets:

$$J = 0 \qquad \rightarrow \qquad \alpha = \pm\frac{1}{4}$$

The strain is calculated as follows:

$$\epsilon = \frac{du}{dx} = \frac{dN_1}{dx}u_1 + \frac{dN_2}{dx}u_2 + \frac{dN_3}{dx}u_3 = \frac{d\xi}{dx}\left(\frac{dN_1}{d\xi}u_1 + \frac{dN_2}{d\xi}u_2 + \frac{dN_3}{d\xi}u_3\right)$$

$$= \frac{1}{J}\left(\frac{dN_1}{d\xi}u_1 + \frac{dN_2}{d\xi}u_2 + \frac{dN_3}{d\xi}u_3\right)$$

Therefore, when $J = 0$ at the extreme of the bar elements, the strains there will be infinity.

Assignment 5.2

Extend the results obtained from the previous Exercise for a 9-node plane stress element. The element is initially a perfect square, nodes 5,6,7,8 are at the midpoint of the sides 1–2, 2–3, 3–4 and 4–1, respectively, and 9 at the center of the square.

Move node 5 tangentially towards 2 until the Jacobian determinant at 2 vanishes. This result is important in the construction of "singular elements" for fracture mechanics.

In order to attack this point, a MATLAB script was created (Included at the end of this report). What this script does is the following:

- The expressions of the Shape functions are readily entered (Symbolically):
  $N_1 = (1/4) * (1 - \xi) * (1 - \eta) * \xi * \eta;$,
  $N_2 = -(1/4) * (1 + \xi) * (1 - \eta) * \xi * \eta;$, ...
- The approximations of the coordinates are loaded(Symbolically):
  $x(\xi, \eta) = x_1 N_1(\xi, \eta) + x_2 N_2(\xi, \eta) + ... + x_9 N_9(\xi, \eta)$
  $y(\xi, \eta) = y_1 N_1(\xi, \eta) + y_2 N_2(\xi, \eta) + ... + y_9 N_9(\xi, \eta)$
- The derivatives are calculated by MATLAB, and they are saved in the Jacobian.
- The values of the coordinates of the nodes of the reference element are entered $(x_1, x_2, ...x_9, y_1, y_2, ..., y_9)$, and the values of the ξ and η coordinates are substituted for node 2. (1,-1)
- The script returns the expression of the Jacobian wrt x_5.
  $J = 0 = (1 - 2x_5)$

  From the expression returned by the script, one obtains that if the coordinate x of the node 5 of the quadrilateral is at 1/2, the Jacobian will be negative. This represents 1/4 of the side of the element, just like in the first point of this report for a bar element. Moreover, the factor $1/det(J)$ appears in the "B" Matrix, which will cause problems when $det(J) \to 0$.

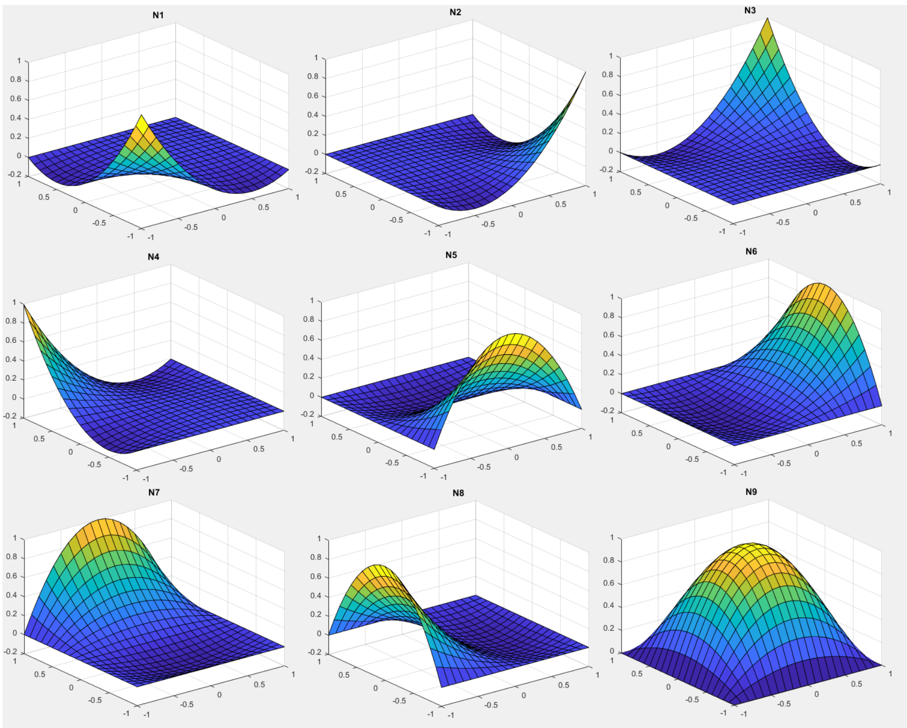

Figure 1: Shape Functions For the 9-Node Quad

Matlab Code

```
%%%% Script to calculate the determinant of the Jacobian of a 9-noded
%%%% Quadrilateral Element
%%%% Jose Raul Bravo Martinez
%%%% MSC Computational Mechanics
clear
clc
close all

%%%%%%%%%%%%%%%% Plotting the Shape Functions %%%%%%%%%%%%%%%%
x=[-1:.1:1];
y=[-1:.1:1];

[xx,yy]=meshgrid(x,y);
N1=xx;
for i=1:length(N1)
    for j=1:length(N1)
    N1(i,j)=(1/4)*(1 - xx(i,j))*(1- yy(i,j))*xx(i,j)*yy(i,j);
    end
end
figure(1)
surf(xx,yy,N1);
title 'N1'

N2=xx;
for i=1:length(N2)
    for j=1:length(N2)
    N2(i,j)=-(1/4)*(1 + xx(i,j))*(1- yy(i,j))*xx(i,j)*yy(i,j);
    end
end
figure(2)
surf(xx,yy,N2);
title 'N2'

N3=xx;
for i=1:length(N3)
    for j=1:length(N3)
    N3(i,j)=(1/4)*(1 + xx(i,j))*(1+ yy(i,j))*xx(i,j)*yy(i,j);
    end
end
figure(3)
surf(xx,yy,N3);
title 'N3'

N4=xx;
for i=1:length(N4)
    for j=1:length(N4)
    N4(i,j)=-(1/4)*(1 - xx(i,j))*(1+ yy(i,j))*xx(i,j)*yy(i,j);
    end
end
figure(4)
surf(xx,yy,N4);
title 'N4'
```

```
N5=xx;
for i=1:length(N5)
    for j=1:length(N5)
    N5(i,j)=-(1/2)*(1 - (xx(i,j)*xx(i,j)))*(1- yy(i,j))*yy(i,j);
    end
end
figure(5)
surf(xx,yy,N5);
title 'N5'

N6=xx;
for i=1:length(N6)
    for j=1:length(N6)
    N6(i,j)=(1/2)*(1 + xx(i,j))*(1- (yy(i,j)*yy(i,j)))*xx(i,j);
    end
end
figure(6)
surf(xx,yy,N6);
title 'N6'

N7=xx;
for i=1:length(N7)
    for j=1:length(N7)
    N7(i,j)=(1/2)*(1 - (xx(i,j)*xx(i,j)))*(1+ yy(i,j))*yy(i,j);
    end
end
figure(7)
surf(xx,yy,N7);
title 'N7'


N8=xx;
for i=1:length(N8)
    for j=1:length(N8)
    N8(i,j)=-(1/2)*(1 - xx(i,j))*(1- (yy(i,j)*yy(i,j)))*xx(i,j);
    end
end
figure(8)
surf(xx,yy,N8);
title 'N8'


N9=xx;
for i=1:length(N9)
    for j=1:length(N9)
    N9(i,j)=(1-xx(i,j)*xx(i,j))*(1-yy(i,j)*yy(i,j));
    end
end
figure(9)
surf(xx,yy,N9);
title 'N9'



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%% Calculating the Determinant
```

```
    %%%%%%%%%%%%%%%%%%%%



syms xi eta X1 X2 X3 X4 X5 X6 X7 X8 X9 Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9
N1=(1/4)*(1-xi)*(1-eta)*xi*eta;
N2=-(1/4)*(1+xi)*(1-eta)*xi*eta;
N3=(1/4)*(1+xi)*(1+eta)*xi*eta;
N4=-(1/4)*(1-xi)*(1+eta)*xi*eta;
N5=-(1/2)*(1-xi^2)*(1-eta)*eta;
N6=(1/2)*(1+xi)*(1-eta^2)*xi;
N7=(1/2)*(1-xi^2)*(1+eta)*eta;
N8=-(1/2)*(1-xi)*(1-eta^2)*xi;
N9=(1-xi^2)*(1-eta^2);

Vect_X=N1*X1+N2*X2+N3*X3+N4*X4+N5*X5+N6*X6+N7*X7+N8*X8+N9*X9;
Vect_Y=N1*Y1+N2*Y2+N3*Y3+N4*Y4+N5*Y5+N6*Y6+N7*Y7+N8*Y8+N9*Y9;


J_11= diff(Vect_X,xi);
J_12= diff(Vect_Y,xi);
J_21= diff(Vect_X,eta);
J_22= diff(Vect_Y,eta);

Det_J=J_11*J_22-J_21*J_12

%%%% Substitute the Value Where you want to calculate the
    determinant [-1,1]^2 %%%%
%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                              xi=1;                              eta=-1;

%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


subs(Det_J)
X1=-1;X2=1;X3=1;X4=-1;X6=1;X7=0;X8=-1;X9=0;Y1=-1;Y2=-1;Y3=1;Y4=1;
    Y5=-1;Y6=0;Y7=1;Y8=0;Y9=0;
subs(Det_J)
```