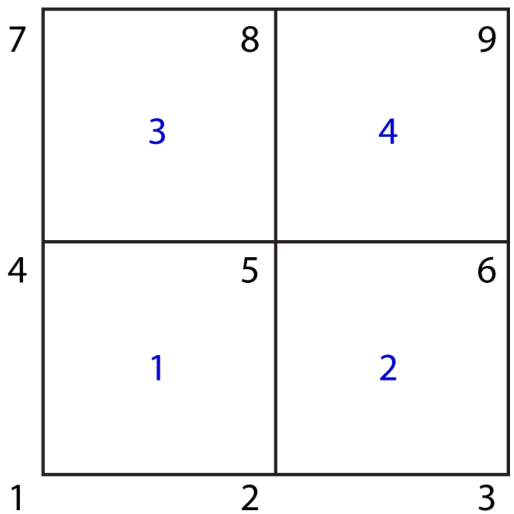
Multidimensional elastic solutions: FEM Implementation

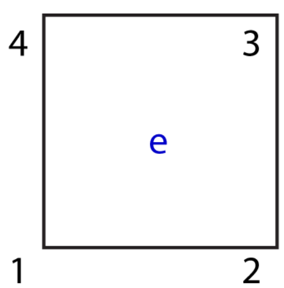
MAE 404/598 – Oswald

3/26/2015

Organization of nodal degrees of freedom



Global node numbering



Element node numbering

$$\mathbf{d} = \begin{bmatrix} u_{1x} & u_{1y} \\ u_{1y} & u_{2y} \\ u_{2x} & u_{3y} \\ \vdots & \vdots \\ u_{9x} & u_{9y} \end{bmatrix}$$

Nodal displacement matrix

$$\mathbf{d} = \begin{bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \\ \vdots \\ u_{9x} \\ u_{9y} \end{bmatrix} \xrightarrow{\text{Gather}} \mathbf{d}^e = \begin{bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \\ u_{4x} \\ u_{4y} \end{bmatrix}$$

Row major order

$$\mathbf{d} = \begin{bmatrix} u_{1x} \\ u_{2x} \\ u_{3x} \\ \vdots \\ u_{9x} \\ u_{1y} \\ u_{2y} \\ u_{3y} \\ \vdots \\ u_{9y} \end{bmatrix} \xrightarrow{\text{Gather}} \mathbf{d}^e = \begin{bmatrix} u_{1x} \\ u_{2x} \\ u_{3x} \\ u_{4x} \\ u_{1y} \\ u_{2y} \\ u_{3y} \\ u_{4y} \end{bmatrix}$$

Column major order

Weak form and discretization

$$\int_{\Omega} \nabla_S \mathbf{w} : \mathbf{C}^e : \nabla_S \mathbf{u} \, d\Omega = \int_{\Omega} \mathbf{w} \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_t} \mathbf{w} \cdot \bar{\mathbf{t}} \, d\Gamma$$

$$\nabla_S = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}$$

Symmetric gradient operator

$$\mathbf{w} = N_I \mathbf{w}_I \qquad \nabla_S \mathbf{w} = \mathbf{B}_I \mathbf{w}_I$$

$$\mathbf{u} = N_I \mathbf{u}_I \qquad \nabla_S \mathbf{u} = \mathbf{B}_I \mathbf{u}_I$$

$$\mathbf{B}_I = \begin{bmatrix} \frac{\partial N_I}{\partial x} & 0 \\ 0 & \frac{\partial N_I}{\partial y} \\ \frac{\partial N_I}{\partial y} & \frac{\partial N_I}{\partial x} \end{bmatrix}$$

General 2D B-matrix

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}$$

B-Matrix for a 4-node element.

Weak form – stiffness matrix

$$\int_{\Omega} \nabla_S \mathbf{w} : \mathbf{C}^e : \nabla_S \mathbf{u} \, d\Omega = \int_{\Omega} \mathbf{w} \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_t} \mathbf{w} \cdot \bar{\mathbf{t}} \, d\Gamma$$

$$\int_{\Omega} \nabla_S \mathbf{w} : \mathbf{C}^e : \nabla_S \mathbf{u} \, d\Omega \Rightarrow \mathbf{w}^T \mathbf{K} \mathbf{d}$$

LHS of weak form

$$\mathbf{K} = \sum_e \mathbf{L}^{eT} \mathbf{K}^e \, \mathbf{L}^e$$

Stiffness matrix assembly

$$\mathbf{K}^e = \int_{\Omega} \mathbf{B}^{eT} \mathbf{C}^e \mathbf{B}^e \, d\Omega$$

Element stiffness matrix

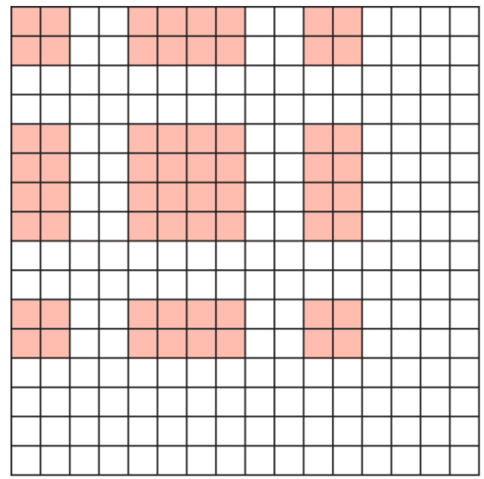
```
for c = mesh.conn
  xe = mesh.x(:,c);
  Ke = zeros(8);
  for q = quadrature_points   % [3xNQ] matrix, each column gives [ξ;η;wt]
    dNdp = grad_shape(q);      % compute gradient of shape functions
    J = xe * dNdp;             % J_ij = dx_i/dξ_j; J is not generally symmetric.
    dNdx = dNdp / J;
    B = zeros(3,8);
    B(1,1:2:end) = dNdx(:,1);   % assigns dNdx to odd  columns in first row
    B(2,2:2:end) = dNdx(:,2);   % assigns dNdx to even columns in second row
    B(3,1:1:end) = dNdx(:,2);   % assigns dNdy to odd  columns in third row
    B(3,2:2:end) = dNdx(:,1);   % assigns dNdx to even columns in third row
    Ke = Ke + B'*C*B * det(J)*q(3);
  end
  % scatter operation goes here…
end
```

Scattering of element stiffness matrix

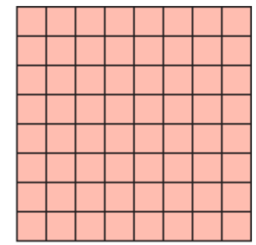
$K_{IJ} \Leftarrow K^e_{ij}$ What is the relationship between (i,j) and (I,J)?

$$\mathbf{d} = \begin{bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \\ \vdots \\ u_{9x} \\ u_{9y} \end{bmatrix} \xrightarrow{\text{Gather}} \mathbf{d}^e = \begin{bmatrix} u^e_{1x} \\ u^e_{1y} \\ u^e_{2x} \\ u^e_{2y} \\ u^e_{3x} \\ u^e_{3y} \\ u^e_{4x} \\ u^e_{4y} \end{bmatrix}$$

Row major order



K



Kᵉ

```
cc = [2*conn-1; 2*conn];
% e.g. c=[1;2;3;4]; becomes
% cc = [1;2;3;4;5;6;7;8];
K(cc,cc) = K(cc,cc) + Ke;
```

Procedure for row-major ordering (2D)

```
cc = [3*conn-2; 3*conn-1; e*conn];
K(cc,cc) = K(cc,cc) + Ke;
```

Procedure for row-major ordering (3D)

Vector operations vs for loops (scatter in 3D)

Where c is an array of the nodes within an element.

```
METHOD 1
cc = [3*c-2; 3*c-1; 3*c];
K(cc,cc) = K(cc,cc) + Ke;
```

```
METHOD 2
for i = 1:4
  for di = 1:3
     for j = 1:4
       for dj = 1:3
         I = 3*(c[i]-1) + di;
         J = 3*(c[j]-1) + dj;
         ii = 3*(i-1) + di;
         jj = 3*(j-1) + dj;
         K(I,J) = K(I,J) + Ke(ii,jj);
       end
    end
  end
end
```

Timing for 1000 elements
METHOD 1:   15 ms
METHOD 2: 850 ms
METHOD 2 is 60X slower!

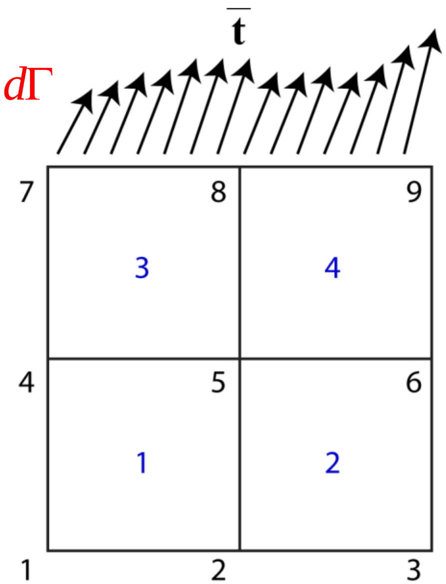
Integration of applied traction

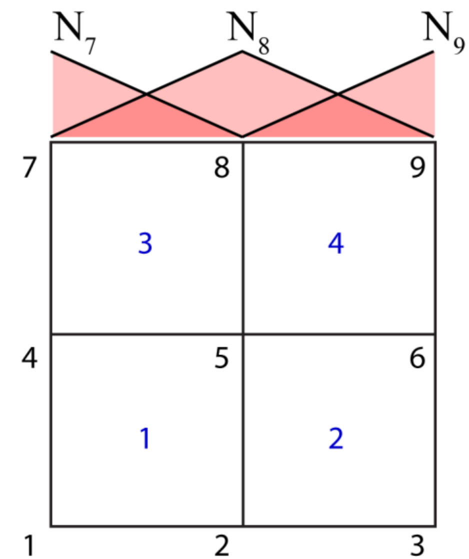
$$\int_{\Omega} \nabla_S \mathbf{w} : \mathbf{C}^e : \nabla_S \mathbf{u}\, d\Omega = \int_{\Omega} \mathbf{w} \cdot \mathbf{b}\, d\Omega + \int_{\Gamma_t} \mathbf{w} \cdot \bar{\mathbf{t}}\, d\Gamma$$

$$\int_{\Gamma_t} \mathbf{w} \cdot \bar{\mathbf{t}}\, d\Gamma = \mathbf{w}_I \sum_e \mathbf{L}^{eT} \int_{\Gamma_t^e} \mathbf{N}_I^e \cdot \bar{\mathbf{t}}\, d\Gamma = \mathbf{w}_I \mathbf{f}_I^t$$

$$\int_{\Gamma_t^e} \mathbf{N}_I^e \cdot \bar{\mathbf{t}}\, d\Gamma = \int_{-1}^{1} \mathbf{N}_I^e(\xi) \cdot \bar{\mathbf{t}}\left(\mathbf{x}(\xi)\right) \left\| \frac{\partial \mathbf{x}}{\partial \xi} \right\| d\xi$$

$$J^{edge} = \left\| \frac{\partial \mathbf{x}}{\partial \xi} \right\| = \frac{\text{length on element edge}}{\text{length on parent element edge}}$$





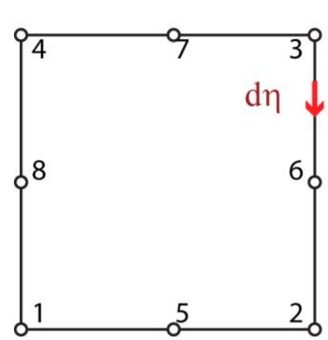
Simple case

When the element edge is straight (and nodes I, J are the corner nodes on the edge)

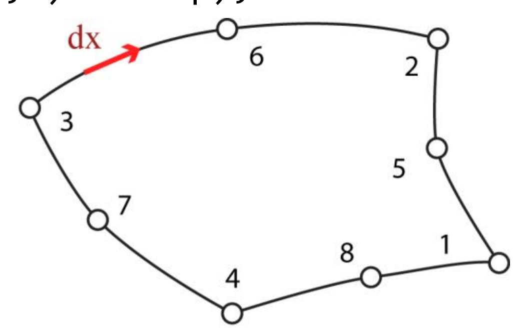
$$\left\| \frac{\partial \mathbf{x}}{\partial \xi} \right\| = \frac{\left\| \mathbf{x}_I^e - \mathbf{x}_J^e \right\|}{2}$$

General case *(where c are the nodes on the edge)*

```
dNdp = shape1D(ξ);  % 1D edge shape function
J_edge = norm(xe(:,c) * dNdp);
```



parent coordinates



global coordinates