Method

$p_t n_y$

$p_v = p_n n_y$

2-3. , p , .

F , (3.68) :

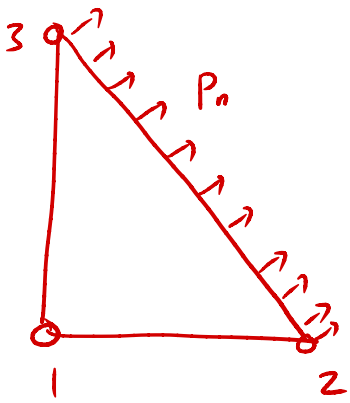
$$(3.69)\quad \begin{matrix} (f^e)_{x1} \\ (f^e)_{y1} \\ (f^e)_{x2} \\ (f^e)_{y2} \\ (f^e)_{x3} \\ (f^e)_{y3} \end{matrix} = t \int_{e} [N^e]^T \begin{matrix} p_x \\ p_y \end{matrix} d$$

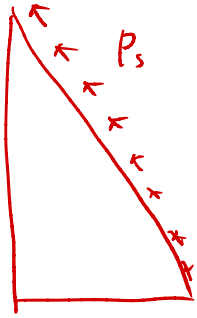
$$[N^e]^T = \begin{bmatrix} N_1^e & 0 \\ 0 & N_1^e \\ N_2^e & 0 \\ 0 & N_2^e \\ N_3^e & 0 \\ 0 & N_3^e \end{bmatrix}$$ 6x2

N_1^e 2-3,

F ( ) :

$$\begin{matrix} (f_!^e)_{x1} \\ (f_!^e)_{y1} \\ (f_!^e)_{x2} \\ (f_!^e)_{y2} \\ (f_!^e)_{x3} \\ (f_!^e)_{y3} \end{matrix} = \begin{matrix} 0 \\ 0 \\ t\int_2^3 N_2^e p_x \, d! \\ t\int_2^3 N_2^e p_y \, d! \\ t\int_2^3 N_3^e p_x \, d! \\ t\int_2^3 N_3^e p_y \, d! \end{matrix}$$





Example: G F ,

T3 Example

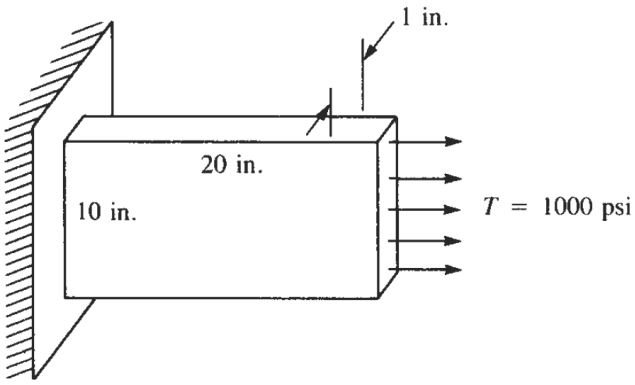
3

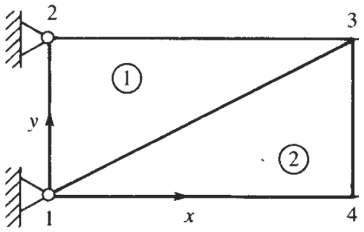
.

.

3 .

$t = 1$ in., $E =$ × , $\nu = 0.3$.





F .

,

.

$$\mathbf{d} = \begin{bmatrix} \bar{\mathbf{d}}_E \\ \mathbf{d}_F \end{bmatrix} = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \\ \bar{d}_3 \\ \bar{d}_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix}$$

$\mathbf{d}_F$.

Remark: A AB

.

```
          displacements=        (GDof,prescribedDof,stiffness,
force)
% function to find solution in terms of global displacements
      D  =       ( 1:GD    ,           D   );
 =         (      D  ,      D  )      (      D  );
displacements=zeros(GDof,1);
displacements(activeDof)=U;
```

A AB, `activeDof` (**d**),

(**d**)

($\bar{\mathbf{d}}_E$). **The MATLAB function allows this operation.**

$\mathbf{B}^e$ - ((3.61))

$$\mathbf{B}^e = \frac{1}{2A^e}\begin{bmatrix} y_2^e - y_3^e & 0 & y_3^e - y_1^e & 0 & y_1^e - y_2^e & 0 \\ 0 & x_3^e - x_2^e & 0 & x_1^e - x_3^e & 0 & x_2^e - x_1^e \\ x_3^e - x_2^e & y_2^e - y_3^e & x_1^e - x_3^e & y_3^e - y_1^e & x_2^e - x_1^e & y_1^e - y_2^e \end{bmatrix}$$

$$= \frac{1}{2}\begin{vmatrix} 1 & {}_1 & {}_1 \\ 1 & {}_2 & {}_2 \\ 1 & {}_3 & {}_3 \end{vmatrix} = \frac{1}{2}\det(\mathbf{M}) = ({}_2{}_3 - {}_3{}_2) + ({}_3{}_1 - {}_1{}_3) + ({}_1{}_2 - {}_2{}_1)$$

A :

$\mathbf{K}^e = tA^e\mathbf{B}^{eT}\mathbf{D}^e\mathbf{B}^e$.

$$\mathbf{D}^e = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

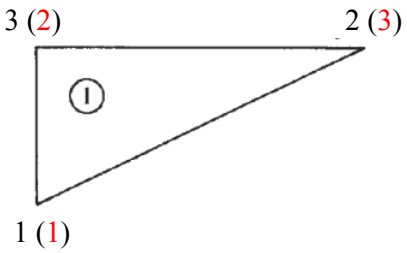
.

$E =$ × $\nu = 0.3$ $\mathbf{D}^e$

, :

$$\mathbf{D}^e \quad \frac{30(10^6)}{0.91}\begin{bmatrix} 1 & 0.3 & 0 \\ 0.3 & 1 & 0 \\ 0 & 0 & 0.35 \end{bmatrix} (\text{psi})$$

E 1



local **(always** **)**

global .

$$A^{(1)} = 100$$

$$\mathbf{B}^{(1)} = \frac{1}{200}\begin{bmatrix} 0 & 0 & 10 & 0 & -10 & 0 \\ 0 & -20 & 0 & 0 & 0 & 20 \\ -20 & 0 & 0 & 10 & 20 & -10 \end{bmatrix} (\text{in.}^{-1})$$

$$\mathbf{K}^{(1)} = \frac{75000}{0.91}\begin{bmatrix} 140 & 0 & 0 & -70 & -140 & 70 \\ 0 & 400 & -60 & 0 & 60 & -400 \\ 0 & -60 & 100 & 0 & -100 & 60 \\ -70 & 0 & 0 & 35 & 70 & -35 \\ -140 & 60 & -100 & 70 & 240 & -130 \\ 70 & -400 & 60 & -35 & -130 & 435 \end{bmatrix} (\text{lb/in.})$$

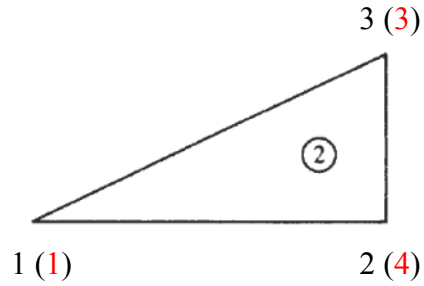
Remark: $\mathbf{K}^{(1)}$

. A AB, $\mathbf{K}^{(1)}$ **singular matrix** .

```
>> K = [140 0 0 -70 -140 70;
0 400 -60 0 60 -400;
0 -60 100 0 -100 60;
-70 0 0 35 70 -35;
-140 60 -100 70 240 -130;
70 -400 60 -35 -130 435];
```

```
>>      ( )

ans =

     0
```

E 2



A ,

.

$$A^{(2)} = 100$$

$$\mathbf{B}^{(2)} = \frac{1}{200}\begin{bmatrix} -10 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & -20 & 0 & 20 \\ 0 & -10 & -20 & 10 & 20 & 0 \end{bmatrix} (\text{in.}^{-1})$$

$$\mathbf{K}^{(2)} = \frac{75000}{0.91}\begin{bmatrix} 100 & 0 & -100 & 60 & 0 & -60 \\ 0 & 35 & 70 & -35 & -70 & 0 \\ -100 & 70 & 240 & -130 & -140 & 60 \\ 60 & -35 & -130 & 435 & 70 & -400 \\ 0 & -70 & -140 & 70 & 140 & 0 \\ -60 & 0 & 60 & 400 & 0 & 400 \end{bmatrix} (\text{lb/in.})$$

:

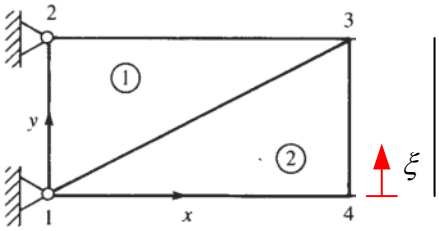
$$\mathbf{K} = \frac{375000}{0.91}\begin{bmatrix} 48 & 0 & -28 & 14 & 0 & -26 & -20 & 12 \\ 0 & 87 & 12 & -80 & -26 & 0 & 14 & -7 \\ -28 & 12 & 48 & -26 & -20 & 14 & 0 & 0 \\ 14 & -80 & -26 & 87 & 12 & -7 & 0 & 0 \\ 0 & -26 & -20 & 12 & 48 & 0 & -28 & 14 \\ -26 & 0 & 14 & -7 & 0 & 87 & 12 & -80 \\ -20 & 14 & 0 & 0 & -28 & 12 & 48 & -26 \\ 12 & -7 & 0 & 0 & 14 & -80 & -26 & 87 \end{bmatrix} \text{(lb/in.)}$$

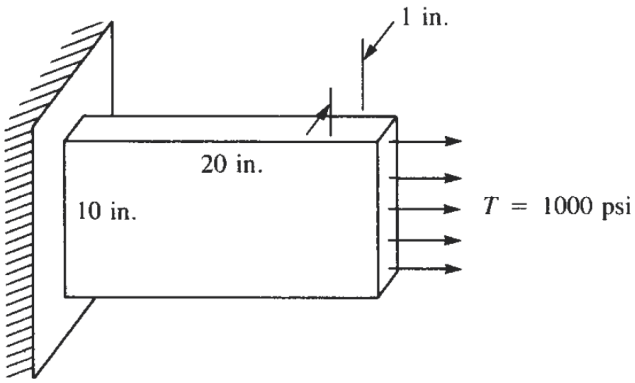
$\mathbf{f}^e$. , $\mathbf{f}^e$

$(\mathbf{f}_\Gamma^e)$ 3-4 1D

ξ:

$N_1^{edge43} = N_1 = 1 - \xi \qquad N_2 = \xi$





$$\mathbf{f}_\Gamma = \int_\Gamma \mathbf{N}^T \bar{\mathbf{t}}\, d\Gamma = \int_0^1 \begin{bmatrix} \bar{N}_1 \\ \bar{N}_1 \\ \bar{N}_2 \\ \bar{N}_2 \end{bmatrix} J d\xi = \int_0^1 \begin{bmatrix} (1-\xi)\times 1000 \\ 0 \\ \xi \times 1000 \\ 0 \end{bmatrix} J d\xi = \begin{bmatrix} \frac{1}{2} \\ 0 \\ \frac{1}{2} \\ 0 \end{bmatrix} 1000 \times 10 = \begin{bmatrix} 5000 \\ 0 \\ 5000 \\ 0 \end{bmatrix}$$

:

$$\frac{375000}{0.91}\begin{bmatrix} 48 & 0 & -28 & 14 & 0 & -26 & -20 & 12 \\ 0 & 87 & 12 & -80 & -26 & 0 & 14 & -7 \\ -28 & 12 & 48 & -26 & -20 & 14 & 0 & 0 \\ 14 & -80 & -26 & 87 & 12 & -7 & 0 & 0 \\ 0 & -26 & -20 & 12 & 48 & 0 & -28 & 14 \\ -26 & 0 & 14 & -7 & 0 & 87 & 12 & -80 \\ -20 & 14 & 0 & 0 & -28 & 12 & 48 & -26 \\ 12 & -7 & 0 & 0 & 14 & -80 & -26 & 87 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} = \begin{bmatrix} 0+r_1 \\ 0+r_2 \\ 0+r_3 \\ 0+r_4 \\ 5000 \\ 0 \\ 5000 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} = \frac{375000}{0.91}\left(\begin{bmatrix} 48 & 0 & -28 & 14 \\ 0 & 87 & 12 & -80 \\ -28 & 12 & 48 & -26 \\ 14 & -80 & -26 & 87 \end{bmatrix}\right)^{-1}\begin{bmatrix} 5000 \\ 0 \\ 5000 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} = \begin{bmatrix} 609.6 \\ 4.2 \\ 663.7 \\ 104.1 \end{bmatrix} \times 10^{-6}(\text{in.})$$

C

(3.20) (3.43):

(3.20) $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}$

(3.43) $\boldsymbol{\varepsilon}^e = \nabla_S \mathbf{u}^e = \nabla_S \mathbf{N}^e \mathbf{d}^e = \mathbf{B}^e \mathbf{d}^e$

$\Rightarrow \boldsymbol{\sigma} = \mathbf{D}\mathbf{B}^e \mathbf{d}^e$

E 1

$$\begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy}\end{bmatrix}=\frac{30(10^6)}{0.91}\begin{bmatrix}1 & 0.3 & 0\\ 0.3 & 1 & 0\\ 0 & 0 & 0.35\end{bmatrix}\times\frac{1}{200}\begin{bmatrix}0 & 0 & 10 & 0 & -10 & 0\\ 0 & -20 & 0 & 0 & 0 & 20\\ -20 & 0 & 0 & 10 & 20 & -10\end{bmatrix}\begin{bmatrix}0\\ 0\\ 609.6\\ 4.2\\ 0\\ 0\end{bmatrix}\times 10^{-6}$$

$$=\begin{bmatrix}1005\\ 301\\ 2.4\end{bmatrix}(\text{psi})$$

E 2

$$\begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy}\end{bmatrix}=\frac{30(10^6)}{0.91}\begin{bmatrix}1 & 0.3 & 0\\ 0.3 & 1 & 0\\ 0 & 0 & 0.35\end{bmatrix}\times\frac{1}{200}\begin{bmatrix}-10 & 0 & 10 & 0 & 0 & 0\\ 0 & 0 & 0 & -20 & 0 & 20\\ 0 & -10 & -20 & 10 & 20 & 0\end{bmatrix}\begin{bmatrix}0\\ 0\\ 663.7\\ 104.1\\ 609.6\\ 4.2\end{bmatrix}\times 10^{-6}$$

$$=\begin{bmatrix}995\\ -1.2\\ -2.4\end{bmatrix}(\text{psi})$$

Q: D σ , $!_{yy}$, σ_{xy} ?

A:

3.8 Programming Finite Element in 2D: A First Look

3 2D (-
).

remain the same:

Step1: D .

.

Step 2: D . F ,

$\mathbf{K}^e$, $\mathbf{f}^e$.

Step 3: A $\mathbf{Kd} = \mathbf{f}$.

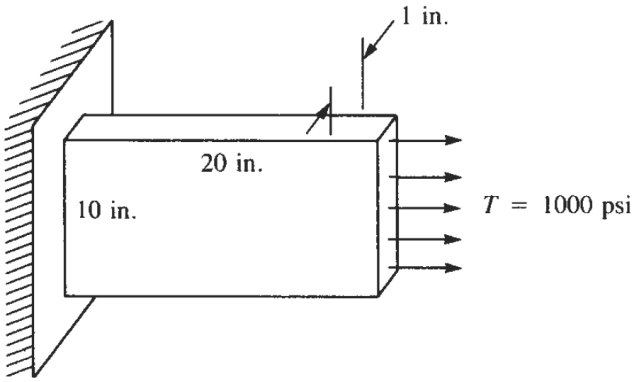
Step 4: .

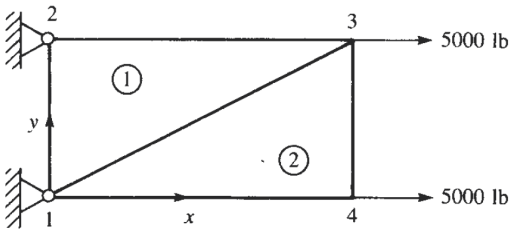
Step 5: $\mathbf{d}_F$.

Step 6: . F ,

(-).

3 A AB .





Step1: D .

.

:

( ).

```
% A Thin Plate Subjected to Uniform Traction
% T3 Implementation
% 2 elements
% clear memory and close all figures
clear all;
```

```
clc;
close all;

% materials
E  = 30e6;     poisson = 0.30;  thickness = 1;

% matrix D
D=E/(1-poisson^2)*[1 poisson 0;poisson 1 0;0 0 (1-poisson)/2];

% preprocessing
% numberElements: number of elements
numberElements = 2;
% numberNodes: number of nodes
numberNodes = 4;
% coordinates and connectivities
elementNodes=[1 3 2; 1 4 3];
nodeCoordinates=[0, 0; 0, 10; 20, 10; 20, 0];
drawingMesh(nodeCoordinates,elementNodes,'T3','k-o'); clear all;
```

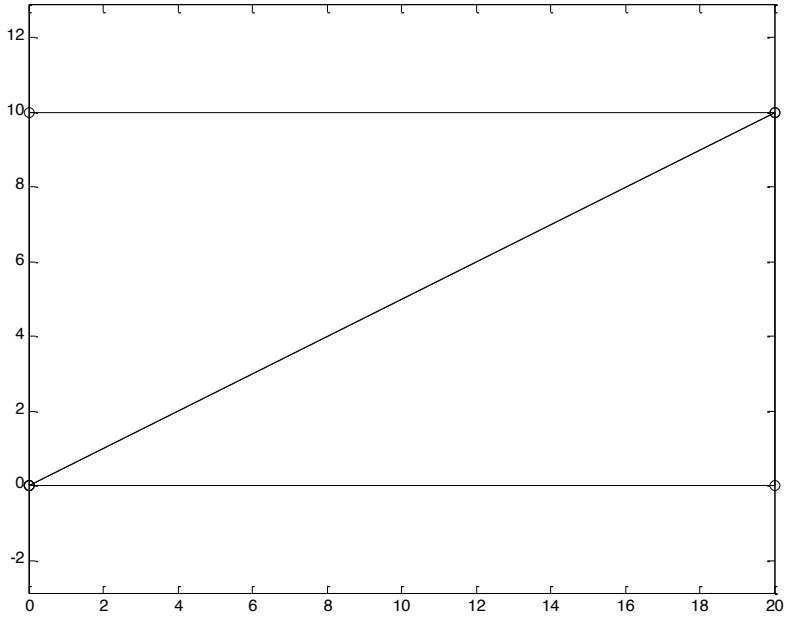
2D, .

drawingMesh.m

A AB. drawingMesh.m 3 4

.

```
function drawingMesh(nodeCoordinates, elementNodes, type, format)

    case 'T3'
       seg1 = [1,2,3,1];
    case 'Q4'
       seg1 = [1,2,3,4,1];
    otherwise
     disp('Type is not supported yet')


for e = 1:length(elementNodes(:,1))
        (    C          (            ( ,   1),1),
             C          (            ( ,   1),2),      )
    hold on
end

axis equal
hold off
```



A .

Step 2: D . F ,

$\mathbf{K}^e$, $\mathbf{f}^e$.

Step 3: A .

F 3 , $\mathbf{K}^e = tA^e\mathbf{B}^{eT}\mathbf{D}^e\mathbf{B}^e$. A AB,

formStiffness2D.m

```
function stiffness=              2D(GDof,numberElements,...
  elementNodes,numberNodes,nodeCoordinates,D,thickness)

% compute stiffness matrix for T3 elements

stiffness=zeros(GDof);

for e=1:numberElements
  numNodePerElement = length(elementNodes(e,:));
  numEDOF = 2*numNodePerElement;
  elementDof=zeros(1,numEDOF);
  for i = 1:numNodePerElement
    elementDof(2*i-1)=2*elementNodes(e,i)-1;
    elementDof(2*i)=2*elementNodes(e,i);
  end

  %  B matrix
  x1 = nodeCoordinates(elementNodes(e,1),1);
  y1 = nodeCoordinates(elementNodes(e,1),2);
```

```
  x2 = nodeCoordinates(elementNodes(e,2),1);
  y2 = nodeCoordinates(elementNodes(e,2),2);
  x3 = nodeCoordinates(elementNodes(e,3),1);
  y3 = nodeCoordinates(elementNodes(e,3),2);
  A = 1/2*det([1 x1 y1; 1 x2 y2; 1 x3 y3]);
  B = 1/(2*A).*[y2-y3 0 y3-y1 0 y1-y2 0;
                0 x3-x2 0 x1-x3 0 x2-x1;
                x3-x2 y2-y3 x1-x3 y3-y1 x2-x1 y1-y2]

% stiffness matrix
  stiffness(elementDof,elementDof)=...
    stiffness(elementDof,elementDof)+...
    A*          *B'*D*B;
end
```

G $\mathbf{f}^e$

. F :

```
% GDof: global number of degrees of freedom
GDof = 2*numberNodes;

% boundary conditions
prescribedDof = [1 2 3 4]';
% force vector
force = zeros(GDof,1);
     (5) = 5000;
     (7) = 5000;

% calculation of the system stiffness matrix
stiffness = formStiffness2D(GDof,numberElements,...
   elementNodes,numberNodes,nodeCoordinates,D,thickness);
```

Step 4: .

Step 5: $\mathbf{d}_F$.

```
% solution
displacements=solution(GDof,prescribedDof,stiffness,force);

% output displacements
outputDisplacements(displacements, numberNodes, GDof);
```

`solution.m` 1D 2D !

(

(3.50)).

`outputDisplacements.m` .

```
function outputDisplacements...
    (displacements,numberNodes,GDof)

% output of displacements in tabular form

disp('Displacements')
jj=1:numberNodes; format
f=[jj; displacements(1:2:GDof)'; displacements(2:2:GDof)'];
fprintf('Node          UX          UY\n')
fprintf('%4d   %10.4e    %10.4e\n',f)
```

, :

```
Displacements
Node          UX          UY
   1   0.0000e+000    0.0000e+000
   2   0.0000e+000    0.0000e+000
   3   6.0958e-004    4.1633e-006
   4   6.6370e-004    1.0408e-004
```

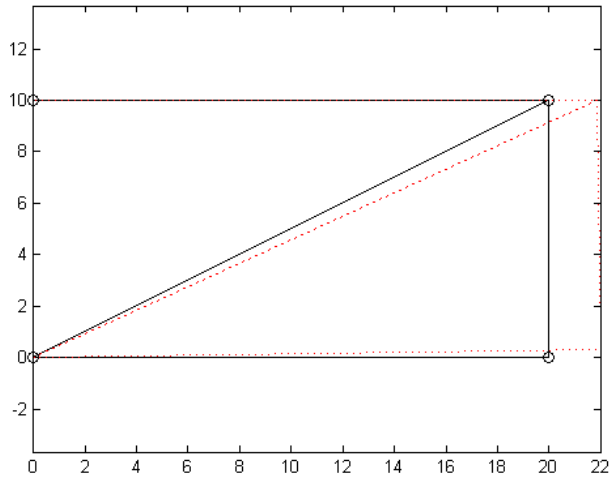
:

Bingo!

Step 6: . F ,

(-).

drawingMesh.m, :



3.9 Four-Node Rectangular Element (Bilinear Rectangle)

- . 4-
(4-) . A
3- .
.
.

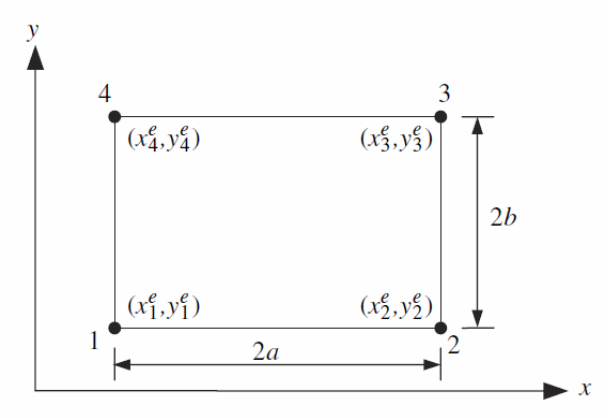
A , -
-
(3). , **three-step procedure for developing element matrices**:

1. C $\mathbf{N}^e$.
2. F - $\mathbf{B}^e$.
3. C $\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^{eT}\mathbf{D}^e\mathbf{B}^e d\Omega$ $\mathbf{f}^e = \mathbf{f}^e_\Omega + \mathbf{f}^e_\Gamma = \int_{\Omega^e} \mathbf{N}^{eT}\mathbf{b}\, d\Omega + \int_{\Gamma^e_t} \mathbf{N}^{eT}\bar{\mathbf{t}}\, d\Gamma$

$\mathbf{N}^e$ $\mathbf{B}^e$.

3.9.1 Construction of shape functions matrix $\mathbf{N}^e$

A $2a \times 2b$ (
90°) 1 4 ( ), .

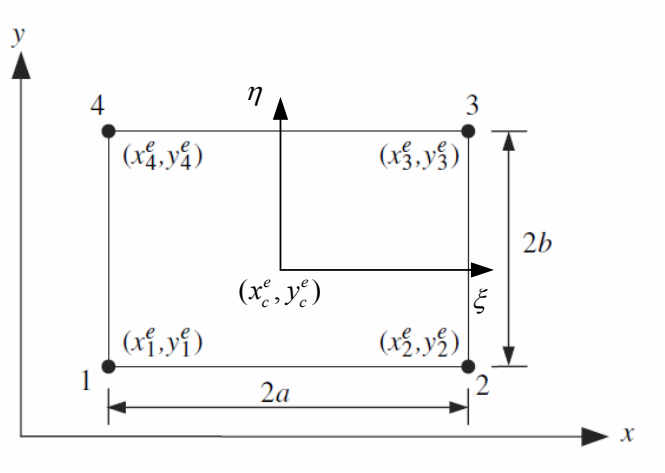


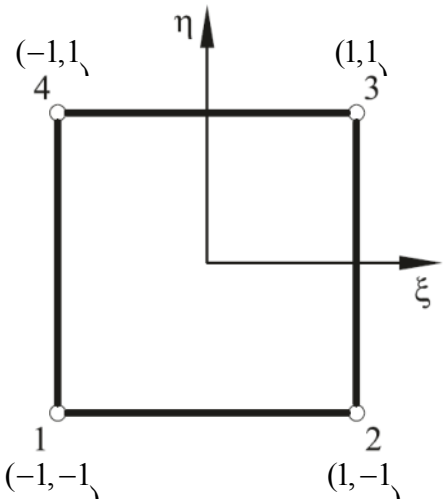
A , (ξ, η) . A

(ξ, η) . F

,

.





D (,)

:

(3.70)

(3.71)
$$d\xi = \frac{dx}{a}$$
$$d\eta = \frac{dy}{b}$$

Remarks:

1. E (3.70) (3.71) **coordinate mapping**
.
.

2.
G . **This kind of coordinate mapping technique is one of the most often used techniques in the FEM.**
.

A D F , D F .
:

(3.72) $\mathbf{u}^e(\xi,\eta) = \mathbf{N}^{Q4}(\xi,\eta)\mathbf{d}^e$

(3.72)

$$\begin{bmatrix} u_x^e \\ u_y^e \end{bmatrix} = \begin{bmatrix} N_1^{Q4}(\xi,\eta) & 0 & N_2^{Q4}(\xi,\eta) & 0 & N_3^{Q4}(\xi,\eta) & 0 & N_4^{Q4}(\xi,\eta) & 0 \\ 0 & N_1^{Q4}(\xi,\eta) & 0 & N_2^{Q4}(\xi,\eta) & 0 & N_3^{Q4}(\xi,\eta) & 0 & N_4^{Q4}(\xi,\eta) \end{bmatrix} \begin{bmatrix} u_{x1}^e \\ u_{y1}^e \\ u_{x2}^e \\ u_{y2}^e \\ u_{x3}^e \\ u_{y3}^e \\ u_{x4}^e \\ u_{y4}^e \end{bmatrix}$$

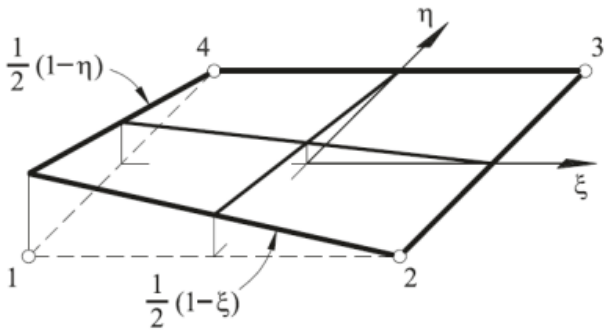
A ,
. , ,
.

1
ξ η
ξ^2 $\xi\eta$ η^2

Q: ?

A:

$N_i^{Q4}\ (i=1,\ 2,\ 3,\ 4)$ **tensor product method**.



F , $N_1^{Q4}(\xi,\eta)$ - $N_1(\xi)$

$N_1(\eta)$.

$$_1{}^4(\xi,\eta) = \ _1(\xi)\times \ _1(\eta) = \frac{1}{2}(1-\xi)\times\frac{1}{2}(1-\eta)$$

Q: D $_1{}^4(\xi,\eta)$ - ?

A:

Q: $^4(\xi,\eta)\ (\ =1,\ 2,\ 3,\ 4)$?

A:

A E (3.73) **bilinear** ,

ξ η .

ξ η . D i (ξ_i, η_i),

$N_i^{4Q}(\xi,\eta)\ (i=1,\ 2,\ 3,\ 4)$ - :

(3.74) $$N_i^{Q4}(\xi,\eta)=\frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)$$

3.9.2 Construction of strain-displacement matrix from shape functions matrix $\mathbf{N}^{Q4}$

3 , -

$\mathbf{B}^e$:

(3.75) $$\mathbf{B}^e = \nabla_S \mathbf{N}^{Q4}$$

Q: (3.75).

A:

(Derivation Exercise) D $\mathbf{B}^e(1,1)$ $\mathbf{B}^e$

(Answer)

$\mathbf{B}^e$:

$$(3.76)\quad \mathbf{B}^e = \frac{1}{4}\begin{bmatrix} -\frac{1-\eta}{a} & 0 & \frac{1-\eta}{a} & 0 & \frac{1+\eta}{a} & 0 & -\frac{1+\eta}{a} & 0 \\ 0 & -\frac{1-\xi}{b} & 0 & -\frac{1+\xi}{b} & 0 & \frac{1+\xi}{b} & 0 & \frac{1-\xi}{b} \\ -\frac{1-\xi}{b} & -\frac{1-\eta}{a} & -\frac{1+\xi}{b} & \frac{1-\eta}{a} & \frac{1+\xi}{b} & \frac{1+\eta}{a} & \frac{1-\xi}{b} & -\frac{1+\eta}{a} \end{bmatrix}$$

Remark:

-

. , ,

.

3.9.3 Construction of element matrices

(3.47):

$$(3.77)\qquad \mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^{eT}\mathbf{D}^e\mathbf{B}^e d\Omega = t\int_{-1}^{+1}\int_{-1}^{+1} ab\mathbf{B}^{eT}\mathbf{D}^e\mathbf{B}^e d\xi d\eta$$

F ,

. A **coordinate mapping** (

), - G

1D.

t

: :

- **$\mathbf{f}^e$ due to body forces ($\mathbf{f}^e_\Omega$)**

$$(3.78)\quad \mathbf{f}^e_\Omega = t\int_{\Omega^e} \mathbf{N}^{eT}\mathbf{b}\, d\Omega$$

. F

, . F ,

$\bar{\mathbf{b}} = \begin{bmatrix} \bar{b}_x \\ \bar{b}_y \end{bmatrix}$ $\bar{b}_x$ $\bar{b}_y$, :

$$\mathbf{f}_{\Omega}^{e} = t\left[\int_{\Omega^e} \mathbf{N}^e \, d\Omega\right]^T \bar{\mathbf{b}} = tab\left[\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N}^{Q4} d\xi d\eta\right]\bar{\mathbf{b}}$$

F 3 , $tab\begin{bmatrix}\bar{b}_x\\ \bar{b}_y\end{bmatrix}$

.

- **$\mathbf{f}^e$ due to tractions ($\mathbf{f}_{\Gamma}^{e}$)**

(3.79) $$\mathbf{f}_{\Gamma}^{e} = t\int_{\Gamma_t^e} \mathbf{N}^{eT}\bar{\mathbf{t}} \, d\Gamma$$

A 3 .

3.9.4 Gauss Quadrature in Two Dimensions

G 1D

$$\hat{I} = \int_{-1}^{1} f(\xi)d\xi = W_1 f(\xi_1) + W_2 f(\xi_2) + \ldots = \sum_{i=1}^{n_{gp}} W_i f(\xi_i)$$

W_i ξ_i (**Gauss points**). n_{gp}

G . F - , G

, :

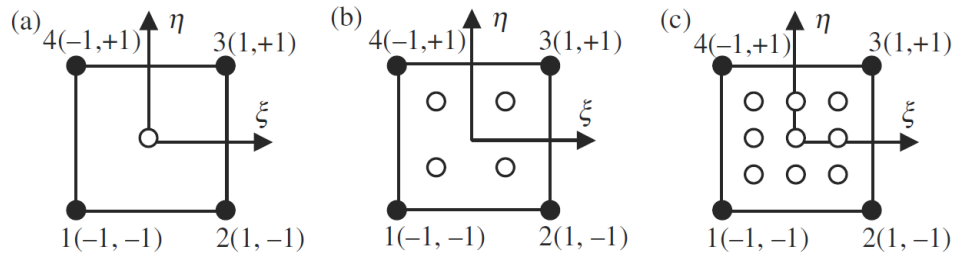
(3.80)
$$I = \int_{-1}^{+1}\int_{-1}^{+1} f(\xi,\eta)d\xi d\eta = \int_{-1}^{+1}\left(\sum_{i=1}^{n_{gp}} W_i f(\xi_i,\eta)\right)d\eta$$
$$= \sum_{i=1}^{n_{gp}}\sum_{j=1}^{n_{gp}} W_i W_j f(\xi_i,\eta_j)$$

, ,

-

.

.



Q: E (3.77) G .

A:

(3.81)

Q: ,

(3.75)?

A:

3.9.5 Stress Calculations

(3.20) (3.43):

(3.20) $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}$

(3.43) $\boldsymbol{\varepsilon} = \nabla \mathbf{u} = \nabla \mathbf{N}\, {}^{4}\mathbf{d} = \mathbf{B}\, \mathbf{d}$

$\Rightarrow \boldsymbol{\sigma} = \mathbf{D}\mathbf{B}^{e}\mathbf{d}^{e}$

G

.

3.10 Programming Finite Element in 2D: Four-Node Rectangular Element

2D (

-).

remain the same:

Step1: D .

.

Step 2: D . F ,

$\mathbf{K}^e$, $\mathbf{f}^e$.

Step 3: A $\mathbf{Kd} = \mathbf{f}$.

Step 4: .

Step 5: $\mathbf{d}_F$.

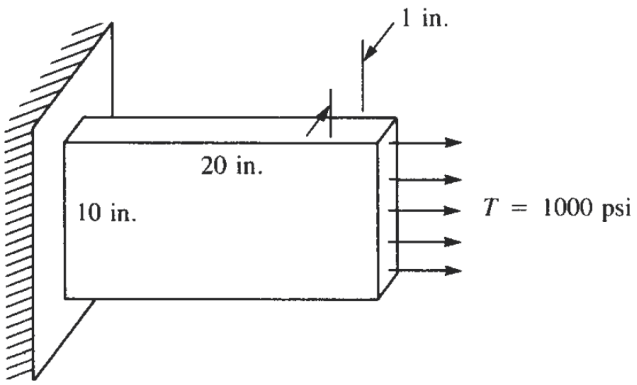
Step 6: . F ,

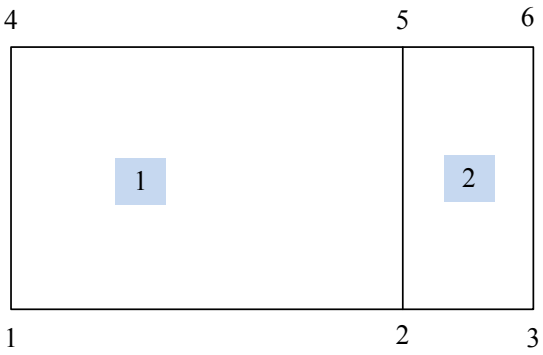
(-).

3. 4

3 . $t = 1$ in., $E =$ × ,

$\nu = 0.3$.





F .

, 4 - . ,

:

$$\varepsilon_{xx} = \sigma_{xx} / E = 1000/(30\times10^6) = 3.3333\times10^{-5}$$
$$(u_x)_{\text{tip}} = L\varepsilon_{xx} = 6.6667\times10^{-6}\text{in.}$$

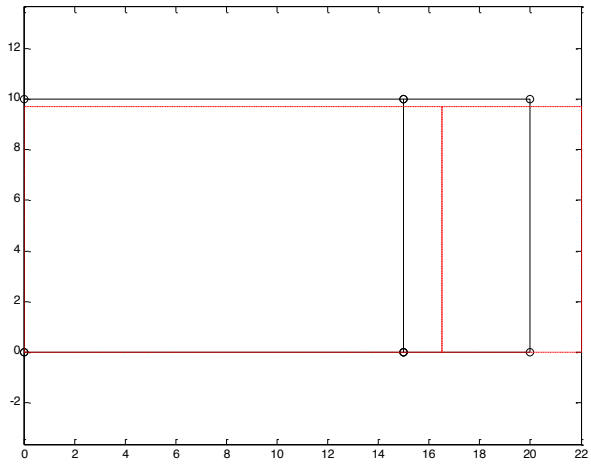
A AB

```
% A Thin Plate Subjected to Uniform Traction
% Rectangular Element Implementation
% 2 elements
% clear memory
clear all;
clc;
close all;

% materials
E  = 30e6;    poisson = 0.30;  thickness = 1;

% matrix D for plane stress
D=E/(1-poisson^2)*[1 poisson 0;poisson 1 0;0 0 (1-poisson)/2];

% preprocessing
% numberElements: number of elements
numberElements = 2;
% numberNodes: number of nodes
numberNodes = 6;
% coordinates and connectivities
elementNodes=[1 2 5 4; 2  3 6 5];
nodeCoordinates=[0, 0; 15, 0; 20 0; 0 10; 15 10; 20 10];
          (    C          ,            ,' 4',' - ');

% GDof: global number of degrees of freedom
GDof = 2*numberNodes;

% boundary conditions
         D   =  1 2 7 ';
% force vector
force = zeros(GDof,1);
     (5) = 5000;
     (11) = 5000;

% calculation of the system stiffness matrix
stiffness =                 (GDof,numberElements,...
   elementNodes,nodeCoordinates,D,thickness);

% solution
displacements = solution(GDof,prescribedDof,stiffness,force);

% output displacements
outputDisplacements(displacements, numberNodes, GDof);
```

```
  =    (   (    C          )-
   (    C          ))/   (   (             ))*0.1;
% reform displacement vector to fit nodeCoordinates
     =    2   (             ,2)* ;
hold on
           (    C          +    ,            ,' 4',' --');
```

(!):

```
Displacements
Node          UX          UY
   1   0.0000e+00   0.0000e+00
   2   5.0000e-04   -3.2903e-19
   3   6.6667 -04   -5.4760e-19
   4   0.0000e+00   -1.0000e-04
   5   5.0000e-04   -1.0000e-04
   6   6.6667 -04   -1.0000e-04
```



formStiffnessRec .

.

A AB.

```
function stiffness=                (GDof,numberElements,...
    elementNodes,nodeCoordinates,D,thickness)

% compute stiffness matrix
% for plane stress rectangular elements

stiffness=zeros(GDof);

% 2 by 2 quadrature
[gaussWeights,gaussLocations]=     2 ('2x2');

for e=1:numberElements
  numEDOF = 8;
```

```
  elementDof=zeros(1,numEDOF);
  for i = 1:4
     elementDof(2*i-1)=2*elementNodes(e,i)-1;
     elementDof(2*i)=2*elementNodes(e,i);
  end
  %
  % THIS IS A HACK: we assume node 1 and node 2 align
  % with x-axis and node 2 and node3 align with y-axis
  %
  a = 0.5*abs(nodeCoordinates(elementNodes(e,2),1) -
nodeCoordinates(elementNodes(e,1),1));
  b = 0.5*abs(nodeCoordinates(elementNodes(e,3),2) -
nodeCoordinates(elementNodes(e,2),2));
  % cycle for Gauss point
  for q=1:size(gaussWeights,1)
    GaussPoint=gaussLocations(q,:);
    xi=GaussPoint(1);
    eta=GaussPoint(2);

% shape functions and derivatives
    [shapeFunction,naturalDerivatives]=shapeFunctionQ4(xi,eta);
    XYderivatives(1,:) = 1/a * naturalDerivatives(1,:);
    XYderivatives(2,:) = 1/b * naturalDerivatives(2,:);

%  B matrix
    B=zeros(3,numEDOF);
    B(1,1:2:numEDOF)  = XYderivatives(1,:);
    B(2,2:2:numEDOF)  = XYderivatives(2,:);
    B(3,1:2:numEDOF)  = XYderivatives(2,:);
    B(3,2:2:numEDOF)  = XYderivatives(1,:);

% stiffness matrix
    stiffness(elementDof,elementDof)=...
       stiffness(elementDof,elementDof)+...
       thickness*a*b*B'*D*B*gaussWeights(q);
  end
```

```
        weights=[ 1;1;1;1];

        case '1x1'

        locations=[0 0];
        weights=[4];
    end

    end  % end function gauss2d
```

F 4.

```
% .....................................................................
    function [shape,naturalDerivatives]=     F        4(xi,eta)

    % shape function and derivatives for Q4 elements
    % shape : Shape functions
    % naturalDerivatives: derivatives w.r.t. xi and eta
    % xi, eta: natural coordinates (-1 ... +1)

    shape=1/4*[ (1-xi)*(1-eta), (1+xi)*(1-eta), ...
           (1+xi)*(1+eta), (1-xi)*(1+eta)];
    naturalDerivatives=...
          1/4*[-(1-eta),  1-eta, 1+eta, -(1+eta);
              -(1-xi), -(1+xi),  1+xi,  1-xi];

    end % end function shapeFunctionQ4
```

3.10 Concluding Remarks

2D FE .

, ,

.

3- (3) 4-

.

, **the use of an isoparametric formulation**

. .