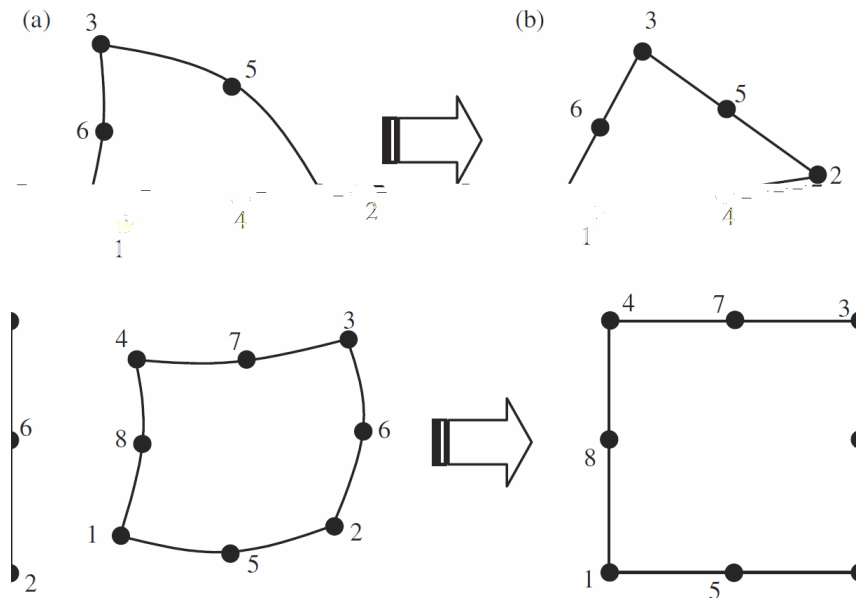
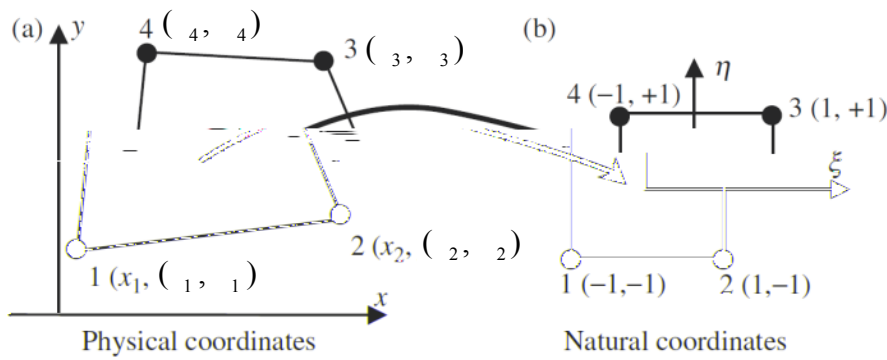


Chapter 4: Isoparametric Formulation

4.1 Introduction

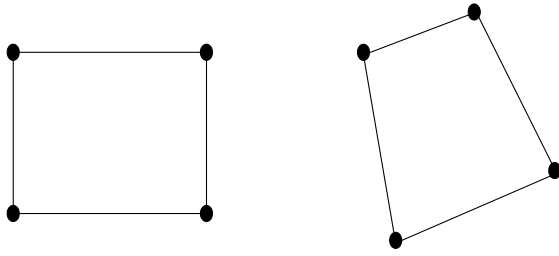
The isoparametric concept allows one to write a computer program systematically and enables one to construct elements with curved sides, which are very powerful in modeling many complex engineering structures



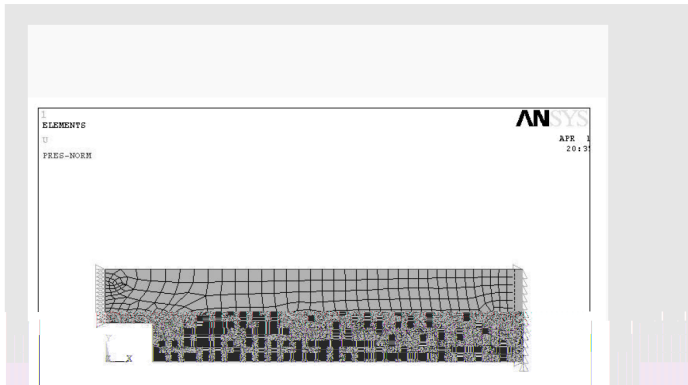
4.1 Four-Node Quadrilateral Element (Q4)

Finite Element Method

quadrilateral element



Typical FE discretization using quadrilateral elements



square element

coordinate mapping

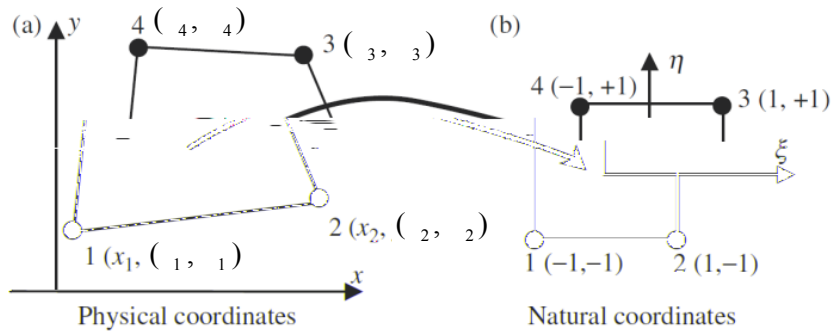
three-step procedure for developing element matrices

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e d\Omega \quad \mathbf{f} = \mathbf{f}_\Omega + \mathbf{f}_\Gamma = \int_{\Omega} \mathbf{N} \mathbf{b} \, \Omega + \int_{\Gamma} \mathbf{N} \bar{\mathbf{t}} \, \Gamma$$

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

Finite Element Method

4.1.1 Displacement Field Interpolation using N^{Q4}



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

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

$$N_1^{Q4}(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 - \eta)$$

$$N_2^{Q4}(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 - \eta)$$

$$N_3^{Q4}(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 + \eta)$$

$$N_4^{Q4}(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 + \eta)$$

Finite Element Method

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

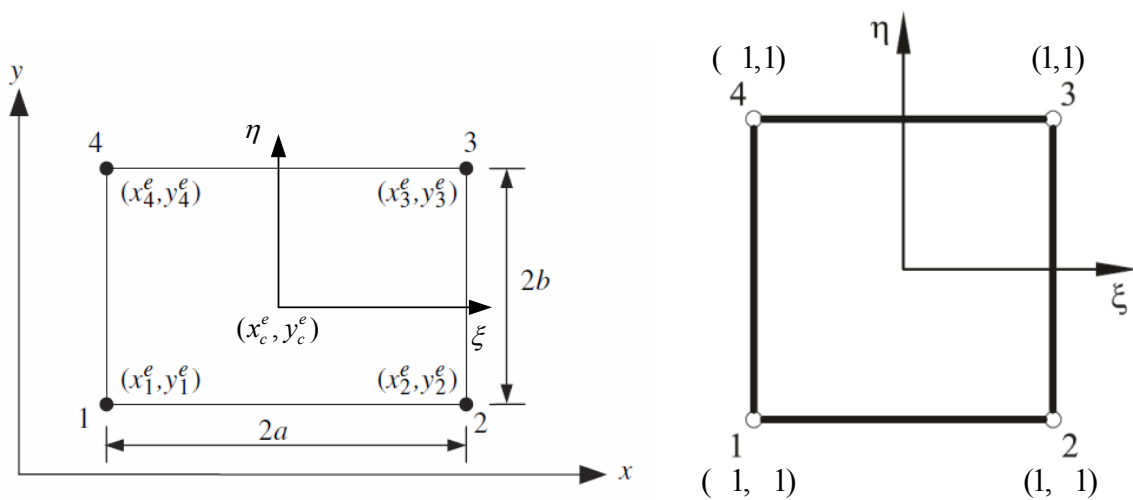
4.1.2 Strain-displacement matrix \mathbf{B}^e

\mathbf{B}^e

$x \quad y$

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

ξ, η



$$\xi = \frac{x - x_c^e}{a}$$

$$\eta = \frac{y - y_c^e}{b}$$

$$d\xi = \frac{dx}{a}$$

$$d\eta = \frac{dy}{b}$$

x

y

Finite Element Method

 ξ, η x, y

 f

$$\frac{\partial f}{\partial \xi} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \xi}$$

$$\frac{\partial f}{\partial \eta} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \eta}$$

$$\begin{aligned} \begin{bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \end{bmatrix} &= \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \\ &= \mathbf{J}^e \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \end{aligned}$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial}{\partial \xi} & \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} & \frac{\partial}{\partial \eta} \end{bmatrix} \quad \text{Jacobian}$$

Remark:

 \mathbf{B}^e

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = (\mathbf{J}^e)^{-1} \begin{bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \end{bmatrix}$$

Finite Element Method

$$\mathbf{B}^e = \nabla_s \mathbf{N}^{Q4}$$

\mathbf{B}^e

$$\mathbf{B}^e = \nabla_s \mathbf{N}^{Q4}$$

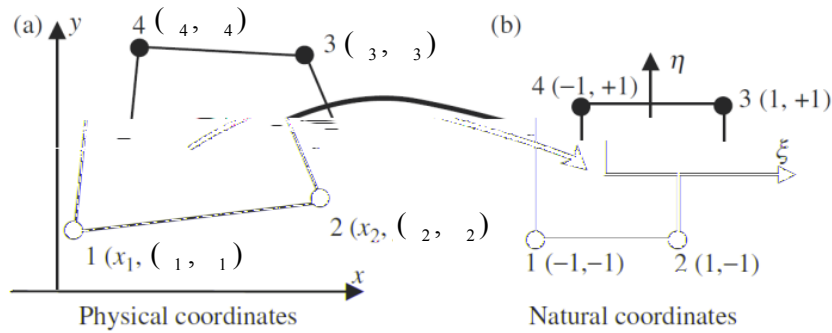
$$\begin{aligned}
 [\mathbf{B}^e]_{3 \times 8} &= \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1^{Q4} & 0 & N_2^{Q4} & 0 & N_3^{Q4} & 0 & N_4^{Q4} & 0 \\ 0 & N_1^{Q4} & 0 & N_2^{Q4} & 0 & N_3^{Q4} & 0 & N_4^{Q4} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{\partial N_1^{Q4}}{\partial x} & 0 & \frac{\partial N_2^{Q4}}{\partial x} & 0 & \frac{\partial N_3^{Q4}}{\partial x} & 0 & \frac{\partial N_4^{Q4}}{\partial x} & 0 \\ 0 & \frac{\partial N_1^{Q4}}{\partial y} & 0 & \frac{\partial N_2^{Q4}}{\partial y} & 0 & \frac{\partial N_3^{Q4}}{\partial y} & 0 & \frac{\partial N_4^{Q4}}{\partial y} \\ \frac{\partial N_1^{Q4}}{\partial y} & \frac{\partial N_1^{Q4}}{\partial x} & \frac{\partial N_2^{Q4}}{\partial y} & \frac{\partial N_2^{Q4}}{\partial x} & \frac{\partial N_3^{Q4}}{\partial y} & \frac{\partial N_3^{Q4}}{\partial x} & \frac{\partial N_4^{Q4}}{\partial y} & \frac{\partial N_4^{Q4}}{\partial x} \end{bmatrix}
 \end{aligned}$$

(Derivation Exercise) $(\mathbf{J}^e)^{-1} = \begin{bmatrix} J_{11}^{inv} & J_{12}^{inv} \\ J_{21}^{inv} & J_{22}^{inv} \end{bmatrix} \quad \mathbf{B}^e(1,1) \quad \mathbf{B}^e$

(Answer)

Finite Element Method

4.1.3 Jacobian matrix J^e



$$J = \begin{bmatrix} \frac{\partial}{\partial \xi} & \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} & \frac{\partial}{\partial \eta} \end{bmatrix}$$

the most important aspect
functions

use the same shape

element geometry

$x \quad y$

$$x(\xi, \eta) = [N_1^{Q4}(\xi, \eta) \quad N_2^{Q4}(\xi, \eta) \quad N_3^{Q4}(\xi, \eta) \quad N_4^{Q4}(\xi, \eta)] \begin{bmatrix} x_1^e \\ x_2^e \\ x_3^e \\ x_4^e \end{bmatrix}$$

$$= \sum_{i=1}^4 N_i^{Q4} x_i^e$$

$$y(\xi, \eta) = [N_1^{Q4}(\xi, \eta) \quad N_2^{Q4}(\xi, \eta) \quad N_3^{Q4}(\xi, \eta) \quad N_4^{Q4}(\xi, \eta)] \begin{bmatrix} y_1^e \\ y_2^e \\ y_3^e \\ y_4^e \end{bmatrix}$$

$$= \sum_{i=1}^4 N_i^{Q4} y_i^e$$

Q: $\xi = 1$

A:

Finite Element Method

Derivation Exercise

$$\xi = 1$$

Answer

Finite Element Method

Remarks:

Edge 2–3 in the physical coordinate system is mapped onto edge 2–3 in the natural coordinate system and vice versus.

$$\xi = \eta$$

ξ
a curved line

The shape functions used to interpolate the coordinates in Eqs. (4.7a) and (4.7b) are **the same as** those used for interpolation of the displacements (Eq. (4.1)).

isoparametric element

The isoparametric concept enables one to construct elements with curved sides, which are very powerful in modeling many complex engineering structures

boundaries

curved

$$\mathbf{J} = \begin{bmatrix} \frac{\partial}{\partial \xi} & \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} & \frac{\partial}{\partial \eta} \end{bmatrix}$$

Finite Element Method

$$\begin{aligned}
 \mathbf{J}^e &= \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^4 \frac{\partial N_i^{Q4}}{\partial \xi} x_i^e & \sum_{i=1}^4 \frac{\partial N_i^{Q4}}{\partial \xi} y_i^e \\ \sum_{i=1}^4 \frac{\partial N_i^{Q4}}{\partial \eta} x_i^e & \sum_{i=1}^4 \frac{\partial N_i^{Q4}}{\partial \eta} y_i^e \end{bmatrix} \\
 &= \begin{bmatrix} \frac{\partial N_1^{Q4}}{\partial \xi} & \frac{\partial N_2^{Q4}}{\partial \xi} & \frac{\partial N_3^{Q4}}{\partial \xi} & \frac{\partial N_4^{Q4}}{\partial \xi} \\ \frac{\partial N_1^{Q4}}{\partial \eta} & \frac{\partial N_2^{Q4}}{\partial \eta} & \frac{\partial N_3^{Q4}}{\partial \eta} & \frac{\partial N_4^{Q4}}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1^e & y_1^e \\ x_2^e & y_2^e \\ x_3^e & y_3^e \\ x_4^e & y_3^e \end{bmatrix} \\
 &= \begin{bmatrix} -\frac{1}{4}(1-\eta) & \frac{1}{4}(1-\eta) & \frac{1}{4}(1+\eta) & -\frac{1}{4}(1+\eta) \\ -\frac{1}{4}(1-\xi) & -\frac{1}{4}(1+\xi) & \frac{1}{4}(1+\xi) & \frac{1}{4}(1-\xi) \end{bmatrix} \begin{bmatrix} x_1^e & y_1^e \\ x_2^e & y_2^e \\ x_3^e & y_3^e \\ x_4^e & y_3^e \end{bmatrix}
 \end{aligned}$$

$$\left[B^e \right]_{3 \times 8} = \begin{bmatrix} \frac{\partial N_1^{Q4}}{\partial x} & 0 & \frac{\partial N_2^{Q4}}{\partial x} & 0 & \frac{\partial N_3^{Q4}}{\partial x} & 0 & \frac{\partial N_4^{Q4}}{\partial x} & 0 \\ 0 & \frac{\partial N_1^{Q4}}{\partial y} & 0 & \frac{\partial N_2^{Q4}}{\partial y} & 0 & \frac{\partial N_3^{Q4}}{\partial y} & 0 & \frac{\partial N_4^{Q4}}{\partial y} \\ \frac{\partial N_1^{Q4}}{\partial y} & \frac{\partial N_1^{Q4}}{\partial x} & \frac{\partial N_2^{Q4}}{\partial y} & \frac{\partial N_2^{Q4}}{\partial x} & \frac{\partial N_3^{Q4}}{\partial y} & \frac{\partial N_3^{Q4}}{\partial x} & \frac{\partial N_4^{Q4}}{\partial y} & \frac{\partial N_4^{Q4}}{\partial x} \end{bmatrix}$$

$$\mathbf{XYDerv}^e = (\mathbf{J}^e)^{-1} \nabla_{(\xi, \eta)} \mathbf{N}^{Q4}$$

$$\begin{bmatrix} \frac{\partial N_1^{Q4}}{\partial x} & \frac{\partial N_2^{Q4}}{\partial x} & \frac{\partial N_3^{Q4}}{\partial x} & \frac{\partial N_4^{Q4}}{\partial x} \\ \frac{\partial N_1^{Q4}}{\partial y} & \frac{\partial N_2^{Q4}}{\partial y} & \frac{\partial N_3^{Q4}}{\partial y} & \frac{\partial N_4^{Q4}}{\partial y} \end{bmatrix} = (\mathbf{J}^e)^{-1} \begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{bmatrix} [N_1^{Q4}(\xi, \eta) \quad N_2^{Q4}(\xi, \eta) \quad N_3^{Q4}(\xi, \eta) \quad N_4^{Q4}(\xi, \eta)]$$

Finite Element Method

$$[A][B][C] = ([A][B])[C] = [A]([B][C])$$

$$\begin{aligned} \begin{bmatrix} \frac{\partial N_1^{Q4}}{\partial x} & \frac{\partial N_2^{Q4}}{\partial x} & \frac{\partial N_3^{Q4}}{\partial x} & \frac{\partial N_4^{Q4}}{\partial x} \\ \frac{\partial N_1^{Q4}}{\partial y} & \frac{\partial N_2^{Q4}}{\partial y} & \frac{\partial N_3^{Q4}}{\partial y} & \frac{\partial N_4^{Q4}}{\partial y} \end{bmatrix} &= (\mathbf{J}_e)^{-1} \begin{bmatrix} \frac{\partial N_1^{Q4}}{\partial \xi} & \frac{\partial N_2^{Q4}}{\partial \xi} & \frac{\partial N_3^{Q4}}{\partial \xi} & \frac{\partial N_4^{Q4}}{\partial \xi} \\ \frac{\partial N_1^{Q4}}{\partial \eta} & \frac{\partial N_2^{Q4}}{\partial \eta} & \frac{\partial N_3^{Q4}}{\partial \eta} & \frac{\partial N_4^{Q4}}{\partial \eta} \end{bmatrix} \\ &= (\mathbf{J}_e)^{-1} \begin{bmatrix} -\frac{1}{4}(1-\eta) & \frac{1}{4}(1-\eta) & \frac{1}{4}(1+\eta) & -\frac{1}{4}(1+\eta) \\ -\frac{1}{4}(1-\xi) & -\frac{1}{4}(1+\xi) & \frac{1}{4}(1+\xi) & \frac{1}{4}(1-\xi) \end{bmatrix} \end{aligned}$$

4.1.4 Element Stiffness Matrix

$$\mathbf{B}^e$$

$$\mathbf{K}^e = t \int_{\Omega^e} \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e d\Omega = t \int_{-1}^{+1} \int_{-1}^{+1} \det(\mathbf{J}^e) \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e d\xi d\eta = t \sum_{i=1}^{n_{gp}} \sum_{j=1}^{n_{gp}} W_i W_j \det(\mathbf{J}^e) \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e$$

t

Remarks:

$$\begin{array}{ccc} & & \xi \quad \eta \\ & & \mathbf{B}^e \\ \xi \quad \eta & & (\mathbf{J}^e)^{-1} \\ & & (\mathbf{J}^e)^{-1} \end{array} \quad \text{dividing}$$

$$\det(\mathbf{J}^e) \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e$$

fraction functions

4.1.4 Element External Force Matrix

_____ t

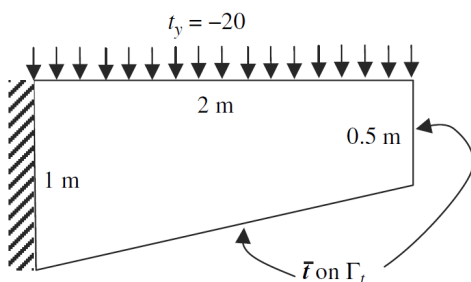
\mathbf{f}^e due to body forces (\mathbf{f}_Ω^e)

$$\mathbf{f}_\Omega^e = \int_{\Omega^e} \mathbf{N}^{eT} \mathbf{b} \, d\Omega = \int_{-1}^{+1} \int_{-1}^{+1} (\mathbf{N}^{Q4})^T \mathbf{b} \det(\mathbf{J}^e) \, d\xi d\eta = \sum_{l=1}^g \sum_{m=1}^g W_l W_m (\mathbf{N}^{Q4})^T \mathbf{b} \det(\mathbf{J}^e)$$

\mathbf{f}_Γ^e due to tractions (\mathbf{f}_Γ^e)

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

4.2 Q4 MATLAB Isoparametric Implementation



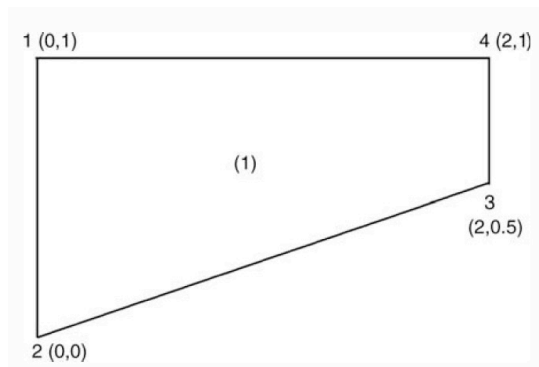
Finite Element Method

$$t_y = -20 \text{ Nm}^{-1}$$

$$E = 3 \times 10^7$$

$$\nu = 0.3$$

ONLY



(Rule of Thumb 經驗法則)

Finite Element Method

$$\begin{aligned}
 \mathbf{J}^e &= \begin{bmatrix} \frac{\partial N_1^{Q4}}{\partial \xi} & \frac{\partial N_2^{Q4}}{\partial \xi} & \frac{\partial N_3^{Q4}}{\partial \xi} & \frac{\partial N_4^{Q4}}{\partial \xi} \\ \frac{\partial N_1^{Q4}}{\partial \eta} & \frac{\partial N_2^{Q4}}{\partial \eta} & \frac{\partial N_3^{Q4}}{\partial \eta} & \frac{\partial N_4^{Q4}}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1^e & y_1^e \\ x_2^e & y_2^e \\ x_3^e & y_3^e \\ x_4^e & y_3^e \end{bmatrix} \\
 &= \begin{bmatrix} -\frac{1}{4}(1-\eta) & \frac{1}{4}(1-\eta) & \frac{1}{4}(1+\eta) & -\frac{1}{4}(1+\eta) \\ -\frac{1}{4}(1-\xi) & -\frac{1}{4}(1+\xi) & \frac{1}{4}(1+\xi) & \frac{1}{4}(1-\xi) \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 2 & 0.5 \\ 2 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0.125\eta - 0.375 \\ 1 & 0.125\xi + 0.125 \end{bmatrix}
 \end{aligned}$$

$$|\mathbf{J}^e| = -0.125\eta + 0.375,$$

$$(\mathbf{J}^e)^{-1} = \begin{bmatrix} \frac{1+\xi}{3-\eta} & 1 \\ \frac{8}{\eta-3} & 0 \end{bmatrix}.$$

$$\mathbf{B}^e = \begin{bmatrix} \frac{N_1^{Q4}}{x} & 0 & \frac{N_2^{Q4}}{x} & 0 & \frac{N_3^{Q4}}{x} & 0 & \frac{N_4^{Q4}}{x} & 0 \\ 0 & \frac{N_1^{Q4}}{y} & 0 & \frac{N_2^{Q4}}{y} & 0 & \frac{N_3^{Q4}}{y} & 0 & \frac{N_4^{Q4}}{y} \\ \frac{N_1^{Q4}}{y} & \frac{N_1^{Q4}}{x} & \frac{N_2^{Q4}}{y} & \frac{N_2^{Q4}}{x} & \frac{N_3^{Q4}}{y} & \frac{N_3^{Q4}}{x} & \frac{N_4^{Q4}}{y} & \frac{N_4^{Q4}}{x} \end{bmatrix}$$

2x2

$$\begin{aligned}
 \xi_1 &= -\frac{1}{\sqrt{3}}, & \xi_2 &= \frac{1}{\sqrt{3}}, \\
 \eta_1 &= -\frac{1}{\sqrt{3}}, & \eta_2 &= \frac{1}{\sqrt{3}}, & W_1 &= W_2 = 1.
 \end{aligned}$$

Finite Element Method

$$\mathbf{K}^e = t \int_{\Omega^e} \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e d\Omega = t \int_{-1}^{+1} \int_{-1}^{+1} \det(\mathbf{J}^e) \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e d\xi d\eta = t \sum_{i=1}^2 \sum_{j=1}^2 W_i W_j \det(\mathbf{J}^e) \mathbf{B}^{eT} \mathbf{D}^e \mathbf{B}^e$$

—

$$(\xi_1, \eta_1) = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) \quad \mathbf{B}^e(\xi_1, \eta_1)$$

$$\mathbf{K}^e(\xi_1, \eta_1)$$

Q: $\mathbf{B}^e(\xi_1, \eta_1)$

Answer:

Finite Element Method

$$(\xi_1, \eta_1) = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right)$$

$$t = 1$$

$$\mathbf{K}^e(\xi_1, \eta_1) = W_1 W_1 \mathbf{B}^{eT}(\xi_1, \eta_1) \mathbf{D}^e \mathbf{B}^e(\xi_1, \eta_1) |\mathbf{J}^e(\xi_1, \eta_1)|.$$

$$\mathbf{K}^e = \sum_{i=1}^2 \sum_{j=1}^2 \mathbf{K}^e(\xi_i, \eta_j)$$

$$= 10^7 \begin{bmatrix} 1.49 & -0.74 & -0.66 & 0.16 & -0.98 & 0.65 & 0.15 & -0.08 \\ & 2.75 & 0.24 & -2.46 & 0.66 & -1.68 & -0.16 & 1.39 \\ & & 1.08 & 0.33 & 0.15 & -0.16 & -0.56 & -0.41 \\ & & & 2.6 & -0.08 & 1.39 & -0.41 & -1.53 \\ & & & & 2 & -0.82 & -1.18 & 0.25 \\ & \text{SYM} & & & & 3.82 & 0.33 & -3.53 \\ & & & & & & 1.59 & 0.25 \\ & & & & & & & 3.67 \end{bmatrix}.$$

element force matrix

Q:

Answer:

Finite Element Method

$$\mathbf{f}_\Gamma^e + \mathbf{r}^e = \begin{bmatrix} r_{x1} \\ r_{y1} - 20 \\ r_{x2} \\ r_{y2} \\ 0 \\ 0 \\ 0 \\ -20 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ u_{x3} \\ u_{y3} \\ u_{x4} \\ u_{y4} \end{bmatrix}$$

$$10^7 \begin{bmatrix} 1.49 & -0.74 & -0.66 & 0.16 & -0.98 & 0.65 & 0.15 & -0.08 \\ & 2.75 & 0.24 & -2.46 & 0.66 & -1.68 & -0.16 & 1.39 \\ & & 1.08 & 0.33 & 0.15 & -0.16 & -0.56 & -0.41 \\ & & & 2.6 & -0.08 & 1.39 & -0.41 & -1.53 \\ & & & & 2 & -0.82 & -1.18 & 0.25 \\ & \text{SYM} & & & & 3.82 & 0.33 & -3.53 \\ & & & & & & 1.59 & 0.25 \\ & & & & & & & 3.67 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ u_{x3} \\ u_{y3} \\ u_{x4} \\ u_{y4} \end{bmatrix} = \begin{bmatrix} r_{x1} \\ r_{y1} - 20 \\ r_{x2} \\ r_{y2} \\ 0 \\ 0 \\ 0 \\ -20 \end{bmatrix}$$

$$10^7 \begin{bmatrix} 2 & -0.82 & -1.18 & 0.25 \\ & 3.82 & 0.33 & -3.53 \\ & & 1.59 & 0.25 \\ \text{SYM} & & & 3.67 \end{bmatrix} \begin{bmatrix} u_{x3} \\ u_{y3} \\ u_{x4} \\ u_{y4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -20 \end{bmatrix}$$

$$\begin{bmatrix} u_{x3} \\ u_{y3} \\ u_{x4} \\ u_{y4} \end{bmatrix} = 10^{-6} \begin{bmatrix} -1.17 \\ -9.67 \\ 2.67 \\ -9.94 \end{bmatrix} \quad \text{or} \quad \mathbf{d}^e = 10^{-6} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1.17 \\ -9.67 \\ 2.67 \\ -9.94 \end{bmatrix}$$

Finite Element Method

```
clear all;

% materials
E = 3e7;    poisson = 0.30;  thickness = 1;

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

% numberElements: number of elements
numberElements=1;
% numberNodes: number of nodes
numberNodes=4;
% coordinates and connectivities
elementNodes=[1 2 3 4];
nodeCoordinates=[0,1; 0, 0; 2, 0.5; 2, 1];

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

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

% boundary conditions
prescribedDof=[1 2 3 4]';

% force vector
force=[0 -20 0 0 0 0 0 -20]';

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

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

```
Displacements
Node      UX              UY
1      0.0000e+000      0.0000e+000
2      0.0000e+000      0.0000e+000
3     -1.1778e-006     -9.6697e-006
4      2.6743e-006     -9.9353e-006
```

formStiffness2D

Finite Element Method

```
function stiffness=formStiffness2D(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness)

% compute stiffness matrix
% for plane stress Q4 elements

stiffness=zeros(GDof);

% 2 by 2 quadrature
[gaussWeights,gaussLocations]=gauss2d('2x2');

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

    % 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);

% Jacobian matrix, inverse of Jacobian,
% derivatives w.r.t. x,y
        [Jacob,invJacobian,XYderivatives]=...
Jacobian(nodeCoordinates(elementNodes(e,:),:),naturalDerivatives)
;

% 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)+...
        B'*D*thickness*B*gaussWeights(q)*det(Jacob);
    end
end
```

```
function [weights,locations]=gauss2d(option)
% Gauss quadrature in 2D
```

Finite Element Method

```
% option '2x2'
% option '1x1'
% locations: Gauss point locations
% weights: Gauss point weights

switch option
  case '2x2'

    locations=...
    [ -0.577350269189626 -0.577350269189626;
      0.577350269189626 -0.577350269189626;
      0.577350269189626 0.577350269189626;
      -0.577350269189626 0.577350269189626];
    weights=[ 1;1;1;1];

  case '1x1'

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

end % end function gauss2d
```

```
% .....
function [shape,naturalDerivatives]=shapeFunctionQ4(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];

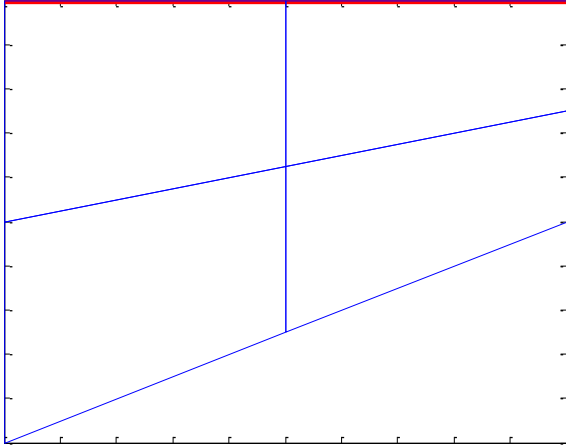
end % end function shapeFunctionQ4
```

```
function [JacobianMatrix,invJacobian,XYDerivatives]=...
Jacobian(nodeCoordinates,naturalDerivatives)

% JacobianMatrix : Jacobian matrix
% invJacobian : inverse of Jacobian Matrix
% XYDerivatives : derivatives w.r.t. x and y
% naturalDerivatives : derivatives w.r.t. xi and eta
% nodeCoordinates : nodal coordinates at element level
JacobianMatrix=naturalDerivatives*nodeCoordinates;
invJacobian=inv(JacobianMatrix);
XYDerivatives=invJacobian*naturalDerivatives;

end % end function Jacobian
```

Finite Element Method



```
% Clamped Taper Plate with Vertical Traction
% Q4 Isoparametric Formulation Implementation
% 4 elements
% clear memory
clear all;

% materials
E = 3e7;    poisson = 0.30;  thickness = 1;

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

% Brute force preprocessing
% numberElements: number of elements
numberElements=4;
% numberNodes: number of nodes
numberNodes=9;
% coordinates and connectivities
elementNodes=[1 2 5 4; 2 3 6 5; 4 5 8 7;5 6 9 8];
nodeCoordinates=[0,0; 1,0.25; 2,0.5; 0,0.5; 1,0.625; 2,0.75; 0,1;
1,1; 2,1];

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

% boundary conditions
prescribedDof=[1 2 7 8 13 14]';
% force vector
force=zeros(GDof,1);
force(14)=-10; force(16) = -20; force(18)=-10;
```

Finite Element Method

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

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

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

Displacements

Node	UX	UY
1	0.0000e+000	0.0000e+000
2	-2.7321e-006	-6.7654e-006
3	-1.2329e-006	-1.8611e-005
4	0.0000e+000	0.0000e+000
5	4.2819e-007	-6.6965e-006
6	1.9583e-006	-1.8676e-005
7	0.0000e+000	0.0000e+000
8	3.9956e-006	-7.1385e-006
9	5.0961e-006	-1.8767e-005

NOT

