計算固体力学入門 (8)

Introduction to Computational Solid Mechanics (8)

Dept. of Sys. Desg. Eng.

Hiroshi Isakari

Today's topic

The contents of this course will be as follows:

1. FEM for a one-dimensional problem.

2. Mathematical preliminaries
   - Gauss-Legendre quadrature
   - Einstein's summation convention

3. Continuum mechanics
   - Deformation of continuum
   - Balance of continuum
   - Basic equations

4. Weak form

5. **Discretization**

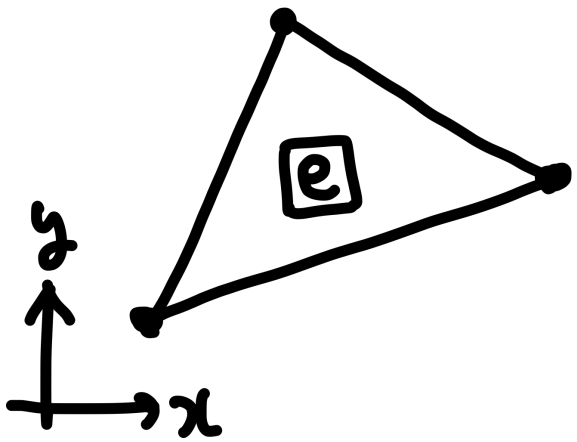
6. **FEM implementations**

Introduction

CST element

The displacement $\boldsymbol{u}(\boldsymbol{x}) = (u(\boldsymbol{x}), v(\boldsymbol{x}))^t$ is assumed to be linear in an element as

$$u(\boldsymbol{x}) \simeq a_1 + a_2 x + a_3 y \,.$$



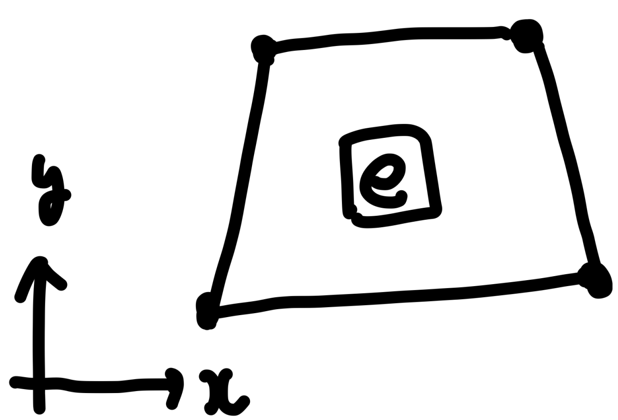
The strain ε_{ij} is constant in the element, e.g, :

$$\varepsilon_{11}(\boldsymbol{x}) = \frac{\partial u(\boldsymbol{x})}{\partial x} \simeq a_2$$

and so is the stress. The analysis with CST element is thus not so accurate.

How can we improve the accuracy?

➡ One possible way is to use **quadrangle elements**.



With such an element, we can approximate the displacement $\boldsymbol{u}(\boldsymbol{x}) = (u(\boldsymbol{x}), v(\boldsymbol{x}))^t$ as

$$u(\boldsymbol{x}) \simeq b_1 + b_2 x + b_3 y + b_4 xy \,.$$

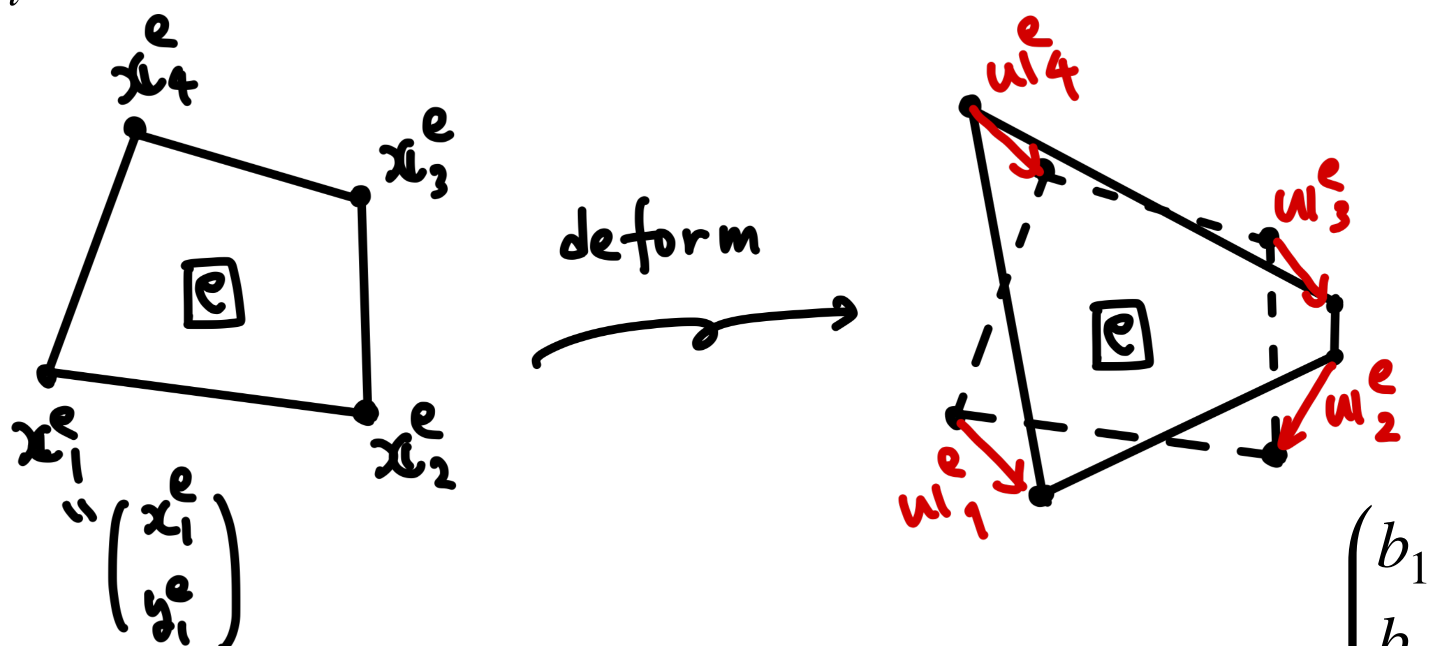
Thanks to the bilinear term, we can expect higher accuracy than CST element.

Basis function? 1/3

As before, let us find explicit representations for the basis functions.

Notations: $\boldsymbol{x}_i^e = (x_i^e, y_i^e)^t$: i^{th} nodal coordinate of element e.

$\boldsymbol{u}_i^e = (u_i^e, v_i^e)^t$: i^{th} nodal displacement of element e.



Since we have supposed $u(\boldsymbol{x}) = b_1 + b_2 x + b_3 y + b_4 xy = (1, x, y, xy)\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$, the (first component of) nodal displacements should satisfy the following:

$$\begin{pmatrix} 1 & x_1^e & y_1^e & x_1^e y_1^e \\ 1 & x_2^e & y_2^e & x_2^e y_2^e \\ 1 & x_3^e & y_3^e & x_3^e y_3^e \\ 1 & x_4^e & y_4^e & x_4^e y_4^e n \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} u_1^e \\ u_2^e \\ u_3^e \\ u_4^e \end{pmatrix} \Leftrightarrow \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} 1 & x_1^e & y_1^e & x_1^e y_1^e \\ 1 & x_2^e & y_2^e & x_2^e y_2^e \\ 1 & x_3^e & y_3^e & x_3^e y_3^e \\ 1 & x_4^e & y_4^e & x_4^e y_4^e n \end{pmatrix}^{-1} \begin{pmatrix} u_1^e \\ u_2^e \\ u_3^e \\ u_4^e \end{pmatrix}$$

Basis function? 2/3

with which we have:

$$u(\boldsymbol{x}) = \begin{pmatrix} 1 & x & y & xy \end{pmatrix} \begin{pmatrix} 1 & x_1^e & y_1^e & x_1^e y_1^e \\ 1 & x_2^e & y_2^e & x_2^e y_2^e \\ 1 & x_3^e & y_3^e & x_3^e y_3^e \\ 1 & x_4^e & y_4^e & x_4^e y_4^e \end{pmatrix}^{-1} \begin{pmatrix} u_1^e \\ u_2^e \\ u_3^e \\ u_4^e \end{pmatrix} = \begin{pmatrix} N_1^e(\boldsymbol{x}) & N_2^e(\boldsymbol{x}) & N_3^e(\boldsymbol{x}) & N_4^e(\boldsymbol{x}) \end{pmatrix} \begin{pmatrix} u_1^e \\ u_2^e \\ u_3^e \\ u_4^e \end{pmatrix},$$

where $N_i^e(\boldsymbol{x})$ is the basis function to be defined.

→ The **"only"** remaining task is to compute the vector-matrix product.

Basis function? 3/3

Here is the result: WolframAlpha 計算知能

{1, x, y, x*y}.{{1, 1, 1,1}, {x1, x2, x3, x4}, {y1,y2,y3,y4},{x1*y1,x2*y2,x3*y3,x4*y4}}^(-1)

拡張キーボード　アップロード　例を見る　ランダムな例を使う

入力:

$$\{1, x, y, x\,y\}.\begin{pmatrix} 1 & 1 & 1 & 1 \\ x1 & x2 & x3 & x4 \\ y1 & y2 & y3 & y4 \\ x1\,y1 & x2\,y2 & x3\,y3 & x4\,y4 \end{pmatrix}^{-1}$$

結果:

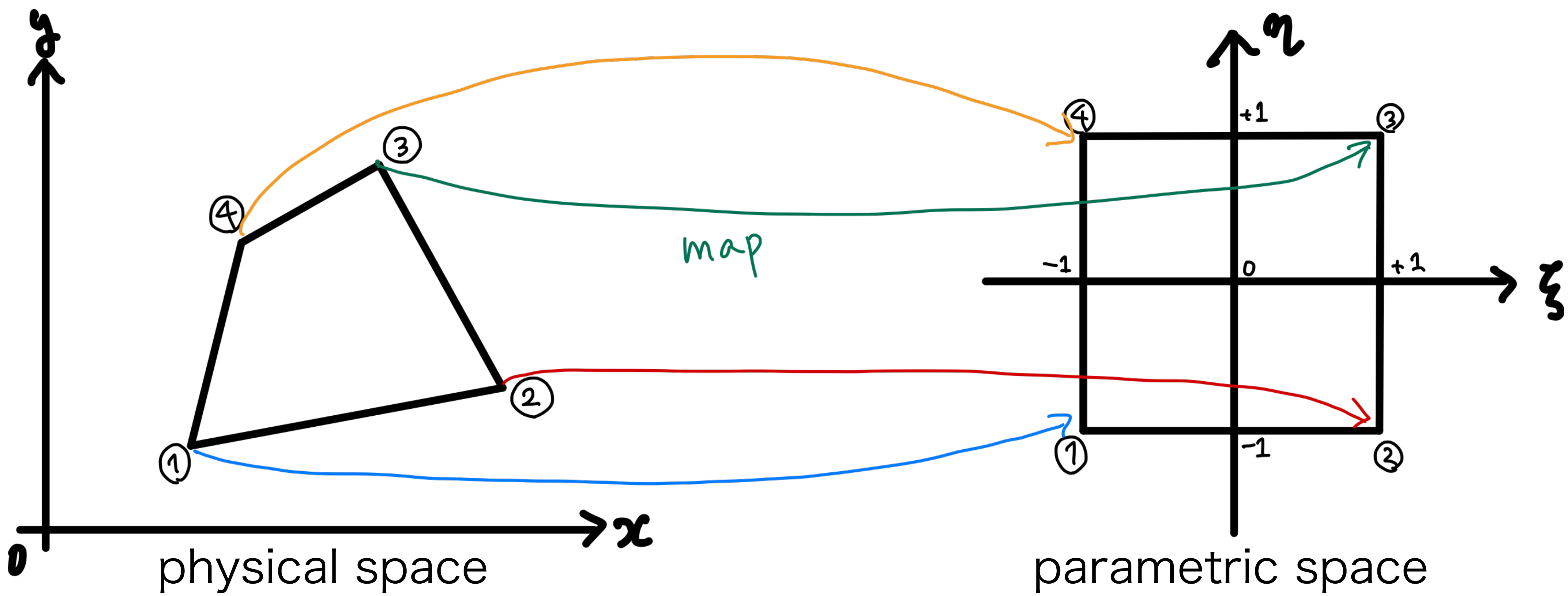
((x y (x1 x3 y1 y2 − x2 x3 y1 y2 −
x1 x2 y1 y3 + x2 x3 y1 y3 + x1 x2 y2 y3 − x1 x3 y2 y3)) /
(x1 x3 y1 y2 − x2 x3 y1 y2 − x1 x4 y1 y2 + x2 x4 y1 y2 − x1 x2 y1 y3 +
x2 x3 y1 y3 + x1 x4 y1 y3 − x3 x4 y1 y3 + x1 x2 y2 y3 − x1 x3 y2 y3 −
x2 x4 y2 y3 + x3 x4 y2 y3 + x1 x2 y1 y4 − x1 x3 y1 y4 − x2 x4 y1 y4 +
x3 x4 y1 y4 − x1 x2 y2 y4 + x2 x3 y2 y4 + x1 x4 y2 y4 − x3 x4 y2 y4 +
x1 x3 y3 y4 − x2 x3 y3 y4 − x1 x4 y3 y4 + x2 x4 y3 y4) +
(y (−x1 x4 y1 y2 + x2 x4 y1 y2 + x1 x2 y1 y4 − x2 x4 y1 y4 −
x1 x2 y2 y4 + x1 x4 y2 y4)) /
(x1 x3 y1 y2 − x2 x3 y1 y2 − x1 x4 y1 y2 + x2 x4 y1 y2 − x1 x2 y1 y3 +
x2 x3 y1 y3 + x1 x4 y1 y3 − x3 x4 y1 y3 + x1 x2 y2 y3 − x1 x3 y2 y3 −
x2 x4 y2 y3 + x3 x4 y2 y3 + x1 x2 y1 y4 − x1 x3 y1 y4 − x2 x4 y1 y4 +
x3 x4 y1 y4 − x1 x2 y2 y4 + x2 x3 y2 y4 + x1 x4 y2 y4 − x3 x4 y2 y4 +
x1 x3 y3 y4 − x2 x3 y3 y4 − x1 x4 y3 y4 + x2 x4 y3 y4) +
(x (x1 x4 y1 y3 − x3 x4 y1 y3 − x1 x3 y1 y4 + x3 x4 y1 y4 +
x1 x3 y3 y4 − x1 x4 y3 y4)) /
(x1 x3 y1 y2 − x2 x3 y1 y2 − x1 x4 y1 y2 + x2 x4 y1 y2 − x1 x2 y1 y3 +
x2 x3 y1 y3 + x1 x4 y1 y3 − x3 x4 y1 y3 + x1 x2 y2 y3 − x1 x3 y2 y3 −
x2 x4 y2 y3 + x3 x4 y2 y3 + x1 x2 y1 y4 − x1 x3 y1 y4 − x2 x4 y1 y4 +
x3 x4 y1 y4 − x1 x2 y2 y4 + x2 x3 y2 y4 + x1 x4 y2 y4 − x3 x4 y2 y4 +
x1 x3 y3 y4 − x2 x3 y3 y4 − x1 x4 y3 y4 + x2 x4 y3 y4) +
(−x2 x4 y2 y3 + x3 x4 y2 y3 + x2 x3 y2 y4 − x3 x4 y2 y4 −
x2 x3 y3 y4 + x2 x4 y3 y4) /
(x1 x3 y1 y2 − x2 x3 y1 y2 − x1 x4 y1 y2 + x2 x4 y1 y2 − x1 x2 y1 y3 +
x2 x3 y1 y3 + x1 x4 y1 y3 − x3 x4 y1 y3 + x1 x2 y2 y3 − x1 x3 y2 y3 −
x2 x4 y2 y3 + x3 x4 y2 y3 + x1 x2 y1 y4 − x1 x3 y1 y4 − x2 x4 y1 y4 +
x3 x4 y1 y4 − x1 x2 y2 y4 + x2 x3 y2 y4 + x1 x4 y2 y4 −
x3 x4 y2 y4 + x1 x3 y3 y4 − x2 x3 y3 y4 − x1 x4 y3 y4 + x2 x4 y3 y4),
(x y (−x1 y1 y2 + x2 y1 y2 + x1 y1 y3 − x3 y1 y3 − x2 y2 y3 + x3 y2 y3)) /
(x1 x3 y1 y2 − x2 x3 y1 y2 − x1 x4 y1 y2 + x2 x4 y1 y2 − x1 x2 y1 y3 +
x2 x3 y1 y3 + x1 x4 y1 y3 − x3 x4 y1 y3 + x1 x2 y2 y3 − x1 x3 y2 y3 −
x2 x4 y2 y3 + x3 x4 y2 y3 + x1 x2 y1 y4 − x1 x3 y1 y4 − x2 x4 y1 y4 +
x3 x4 y1 y4 − x1 x2 y2 y4 + x2 x3 y2 y4 + x1 x4 y2 y4 − x3 x4 y2 y4 +
x1 x3 y3 y4 − x2 x3 y3 y4 − x1 x4 y3 y4 + x2 x4 y3 y4) +

Who wants to do this?
This is too much complicated to be used in an actual computation!

→ Let us take another path to define the basis functions.

Parametric space

Let us map the element into a “parametric space”:



so that a general quadrangle becomes a square.

[strategy to derive the basis functions $N_i^e(\boldsymbol{x})$]

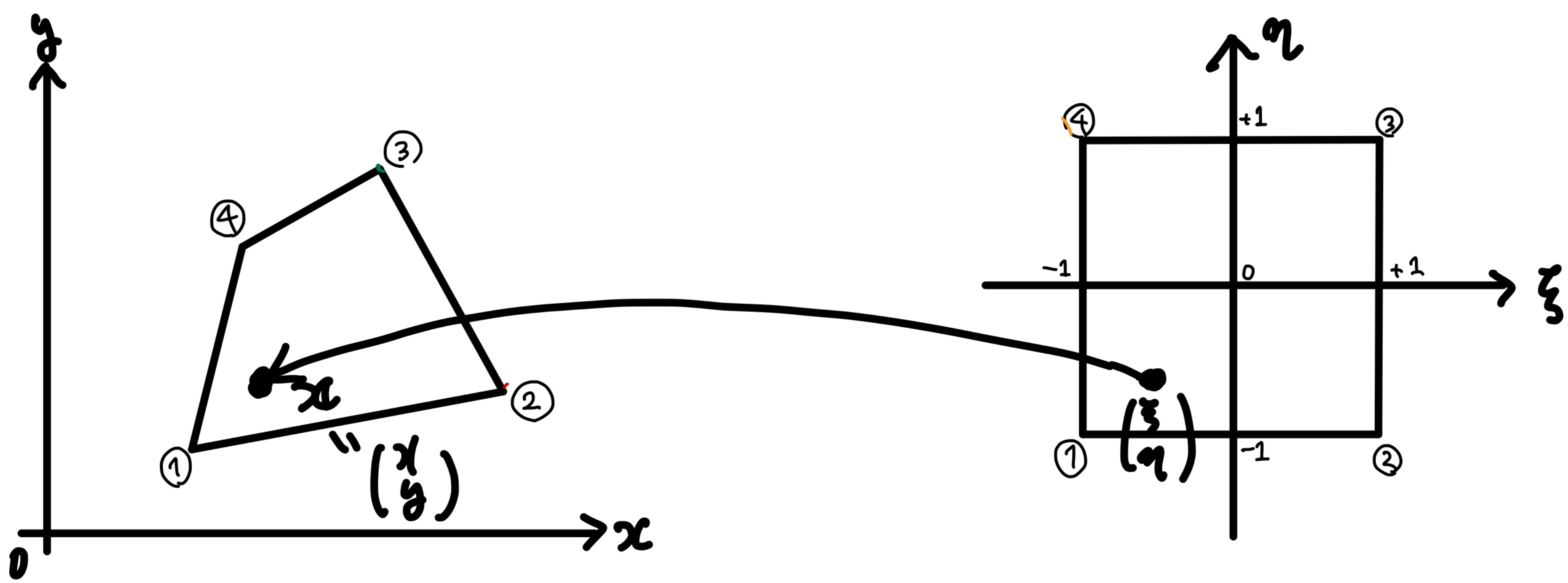
1. derive them in the parametric space ((ξ, η) space).
2. map them into the physical space ((x, y) space).

Shape function 1/2

Let us suppose that the map $(\xi, \eta) \to (x, y)$ is defined as

$$x = c_1 + c_2\xi + c_3\eta + c_4\xi\eta \text{ and } y = d_1 + d_2\xi + d_3\eta + d_4\xi\eta. \cdots \text{(A)}$$

(In other words, the coordinate of an arbitrary point (x, y) in the physical space is parametrized by the parameters ξ and η.)



The unknown coefficients in (A) are determined by

$$\begin{pmatrix} +1 & -1 & -1 & +1 \\ +1 & +1 & -1 & -1 \\ +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} x_1^e \\ x_2^e \\ x_3^e \\ x_4^e \end{pmatrix} \text{ and } \begin{pmatrix} +1 & -1 & -1 & +1 \\ +1 & +1 & -1 & -1 \\ +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} = \begin{pmatrix} y_1^e \\ y_2^e \\ y_3^e \\ y_4^e \end{pmatrix} \cdots \text{(B)}$$

Shape function 2/2

By substituting the sol. of (B) into (A), one has:

WolframAlpha 計算知能

```
{1,xi,eta,xi*eta}.{{1, -1,-1,1}, {1,1,-1,-1},{1,1,1,1},{1,-1,1,-1}}^(-1).{{x1},{x2},{x3},{x4}}
```

拡張キーボード　アップロード　例を見る　ランダムな例を使う

入力:

$$\{1, \xi, \eta, \xi\,\eta\}.\begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}^{-1}.\begin{pmatrix} \mathrm{x1} \\ \mathrm{x2} \\ \mathrm{x3} \\ \mathrm{x4} \end{pmatrix}$$

厳密な結果:

$$\left\{\mathrm{x1}\left(\frac{\eta\,\xi}{4} - \frac{\eta}{4} - \frac{\xi}{4} + \frac{1}{4}\right) + \mathrm{x2}\left(-\frac{\eta\,\xi}{4} - \frac{\eta}{4} + \frac{\xi}{4} + \frac{1}{4}\right) + \mathrm{x3}\left(\frac{\eta\,\xi}{4} + \frac{\eta}{4} + \frac{\xi}{4} + \frac{1}{4}\right) + \mathrm{x4}\left(-\frac{\eta\,\xi}{4} + \frac{\eta}{4} - \frac{\xi}{4} + \frac{1}{4}\right)\right\}$$

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

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

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

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

Thus, x-coordinate of a point in an element (in the physical space) is parametrized as

$$x = \sum_{i=1}^{4} N_i^e(\xi,\eta)x_i^e,$$

where $N_i^e(\xi,\eta)$ is called shape function. y-coordinate is similar.

Basis function

We then **recycle** the shape functions

$$
\begin{aligned}
N_1^e(\xi,\eta) &= \frac{1}{4}(\eta\xi - \eta - \xi + 1) = \frac{1}{4}(1-\eta)(1-\xi),\\
N_2^e(\xi,\eta) &= \frac{1}{4}(-\eta\xi - \eta + \xi + 1) = \frac{1}{4}(1-\eta)(1+\xi),\\
N_3^e(\xi,\eta) &= \frac{1}{4}(\eta\xi + \eta + \xi + 1) = \frac{1}{4}(1+\eta)(1+\xi),\\
N_4^e(\xi,\eta) &= \frac{1}{4}(-\eta\xi + \eta - \xi + 1) = \frac{1}{4}(1+\eta)(1-\xi),
\end{aligned}
$$

as the basis functions for the displacement:

$$
\begin{pmatrix} u(\boldsymbol{x}(\xi,\eta)) \\ v(\boldsymbol{x}(\xi,\eta)) \end{pmatrix} = \begin{pmatrix} N_1^e(\xi,\eta) & 0 & N_2^e(\xi,\eta) & 0 & N_3^e(\xi,\eta) & 0 & N_4^e(\xi,\eta) & 0 \\ 0 & N_1^e(\xi,\eta) & 0 & N_2^e(\xi,\eta) & 0 & N_3^e(\xi,\eta) & 0 & N_4^e(\xi,\eta) \end{pmatrix} \begin{pmatrix} u_1^e \\ v_1^e \\ u_2^e \\ v_2^e \\ u_3^e \\ v_3^e \\ u_4^e \\ v_4^e \end{pmatrix}.
$$

This is why such an element is called **iso**paramtric element.

Weak form

We then substitute the expanded displacement into the following element-wise weak form:

$$\int_e (\tilde{\varepsilon}_{11},\ \tilde{\varepsilon}_{22},\ \tilde{\gamma}_{12}) \begin{pmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{pmatrix} \mathrm{d}V = \sum_{i=1}^{4} \int_{\partial e_i} \tilde{\boldsymbol{u}}^t \boldsymbol{t}_i \mathrm{d}S$$

$$\Leftrightarrow \tilde{\boldsymbol{d}}^{e^t} \left(\int_e \mathsf{B}^{e^t} \mathsf{D}^e \mathsf{B}^e \mathrm{d}V \right) \boldsymbol{d}^e = \tilde{\boldsymbol{d}}^{e^t} \sum_{i=1}^{4} \int_{\partial e_i} \mathsf{N}^{e^t} \boldsymbol{t}_i \mathrm{d}S$$

To this end, we first need to compute the corresponding strain.

$$\begin{pmatrix} \varepsilon_{11}(\boldsymbol{x}) \\ \varepsilon_{22}(\boldsymbol{x}) \\ \gamma_{12}(\boldsymbol{x}) \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix} \begin{pmatrix} N_1^e(\xi,\eta) & 0 & \cdots & N_4^e(\xi,\eta) & 0 \\ 0 & N_1^e(\xi,\eta) & \cdots & 0 & N_4^e(\xi,\eta) \end{pmatrix}}_{= \mathsf{B}^e \text{ (B-matrix)}} \begin{pmatrix} u_1^e \\ v_1^e \\ u_2^e \\ v_2^e \\ u_3^e \\ v_3^e \\ u_4^e \\ v_4^e \end{pmatrix}$$

$$= \mathsf{B}^e \boldsymbol{d}^e$$

How to compute the B-matrix?

Derivative of shape functions

Naturally, we can use the chain rule to compute, for example, $\partial N_1(\xi,\eta)/\partial x$ as

$$\frac{\partial N_1(\xi,\eta)}{\partial x} = \frac{\partial N_1(\xi,\eta)}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial N_1(\xi,\eta)}{\partial \eta}\frac{\partial \eta}{\partial x}.$$

It is, however, difficult to compute $\partial\xi/\partial x$ and $\partial\eta/\partial x$ since we only have the following relation:

$$x = \sum_{i=1}^{4} N_i^e(\xi,\eta)x_i^e.$$

Is there any more simple way to compute $\partial N_1(\xi,\eta)/\partial x$? Let us instead compute as

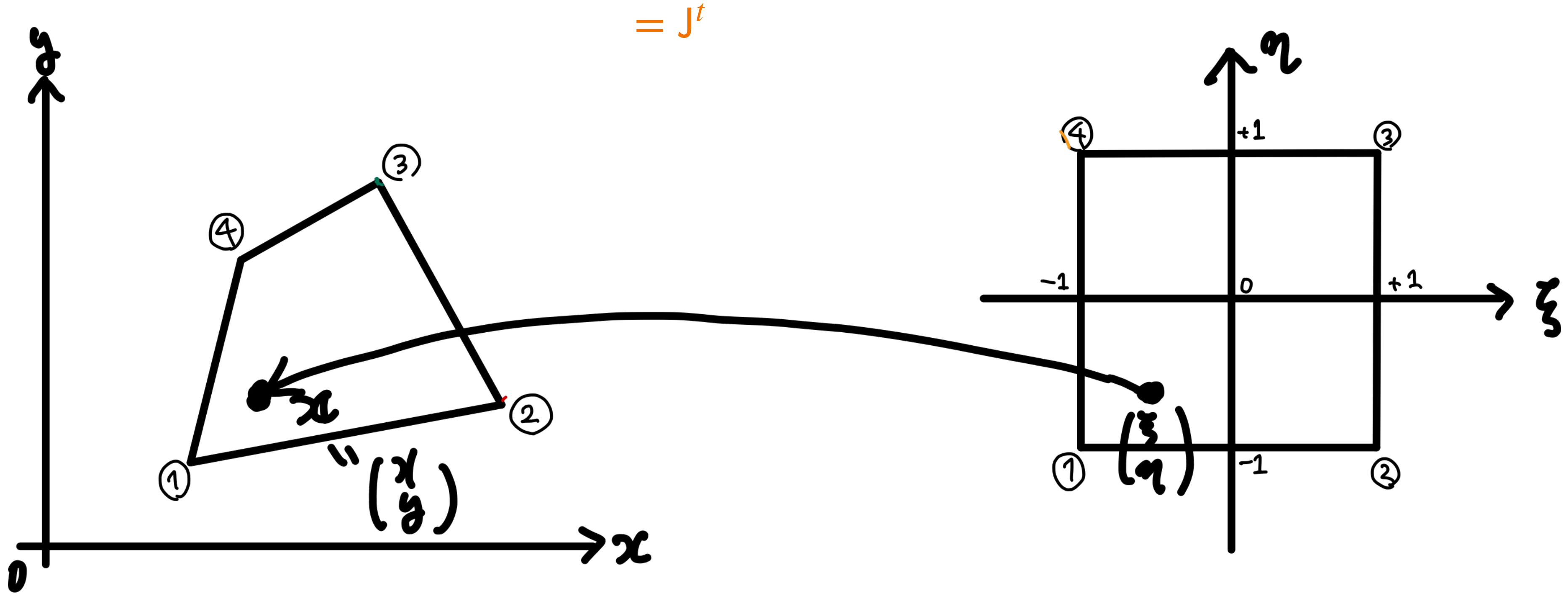
$$\begin{pmatrix} \frac{\partial N_1}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{\partial N_1}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_1}{\partial y}\frac{\partial y}{\partial \xi} \\ \frac{\partial N_1}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial N_1}{\partial y}\frac{\partial y}{\partial \eta} \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{pmatrix}}_{=\mathrm{J}} \begin{pmatrix} \frac{\partial N_1}{\partial x} \\ \frac{\partial N_1}{\partial y} \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} \frac{\partial N_1}{\partial x} \\ \frac{\partial N_1}{\partial y} \end{pmatrix} = \mathrm{J}^{-1} \begin{pmatrix} \frac{\partial N_1}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} \end{pmatrix} \qquad \mathrm{J}^{-1} = \frac{1}{\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}} \begin{pmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{pmatrix}$$

Jacobian

Note that the matrix J is nothing but the transposed Jacobi matrix of the map

$$\begin{pmatrix} \mathrm{d}x \\ \mathrm{d}y \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix}}_{=\mathrm{J}^t} \begin{pmatrix} \mathrm{d}\xi \\ \mathrm{d}\eta \end{pmatrix},$$



whose determinant is

$$|\mathrm{J}|^t = |\mathrm{J}| = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}.$$

B- and K-matrices

We now can compute the B-matrix:

$$\mathsf{B}^e = \begin{pmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix} \begin{pmatrix} N_1^e(\xi,\eta) & 0 & \cdots & N_4^e(\xi,\eta) & 0 \\ 0 & N_1^e(\xi,\eta) & \cdots & 0 & N_4^e(\xi,\eta) \end{pmatrix},$$

and the K-matrix:

$$\begin{aligned} \mathsf{K}^e &= \int_e \mathsf{B}^t\mathsf{D}\mathsf{B}\,\mathrm{d}x\mathrm{d}y \\ &= \int_{\xi=-1}^{1}\int_{\eta=-1}^{1} \mathsf{B}^t\mathsf{D}\mathsf{B}\,|\,\mathsf{J}\,|\,\mathrm{d}\xi\mathrm{d}\eta\,. \end{aligned}$$

Note that the component of B can be linear in either ξ or η, and the component of $\mathsf{B}^t\mathsf{D}\mathsf{B}$ can at most be quadratic. $\rightarrow$ We use the Gauss-Legendre quadrature of second degree to evaluate each integral. Recall that 2nd order GL quadrature is exact when the integrand is of order up to 3 (=2x2-1).

The remaining tasks

$$
\begin{aligned}
\mathsf{K}^e &= \int_e \mathsf{B}^t\mathsf{D}\mathsf{B}\,\mathrm{d}x\mathrm{d}y \\
&= \int_{\xi=-1}^{1}\int_{\eta=-1}^{1} \mathsf{B}^t\mathsf{D}\mathsf{B}\,|\,\mathsf{J}\,|\,\mathrm{d}\xi\mathrm{d}\eta\,. \\
&= \mathsf{B}^t\mathsf{D}\mathsf{B}\,|_{\xi=-1/\sqrt{3},\eta=-1/\sqrt{3}} \times 1 + \mathsf{B}^t\mathsf{D}\mathsf{B}\,|_{\xi=1/\sqrt{3},\eta=-1/\sqrt{3}} \times 1 + \mathsf{B}^t\mathsf{D}\mathsf{B}\,|_{\xi=-1/\sqrt{3},\eta=1/\sqrt{3}} \times 1 + \mathsf{B}^t\mathsf{D}\mathsf{B}\,|_{\xi=1/\sqrt{3},\eta=1/\sqrt{3}} \times 1
\end{aligned}
$$

We then combine all the element stiffness matrices to obtain a system of algebraic equations (whose coefficient matrix is the global stiffness matrix) as in the case of CST element, and solve the equations to obtain the displacement field.

After that, if you want, you can compute the stress by

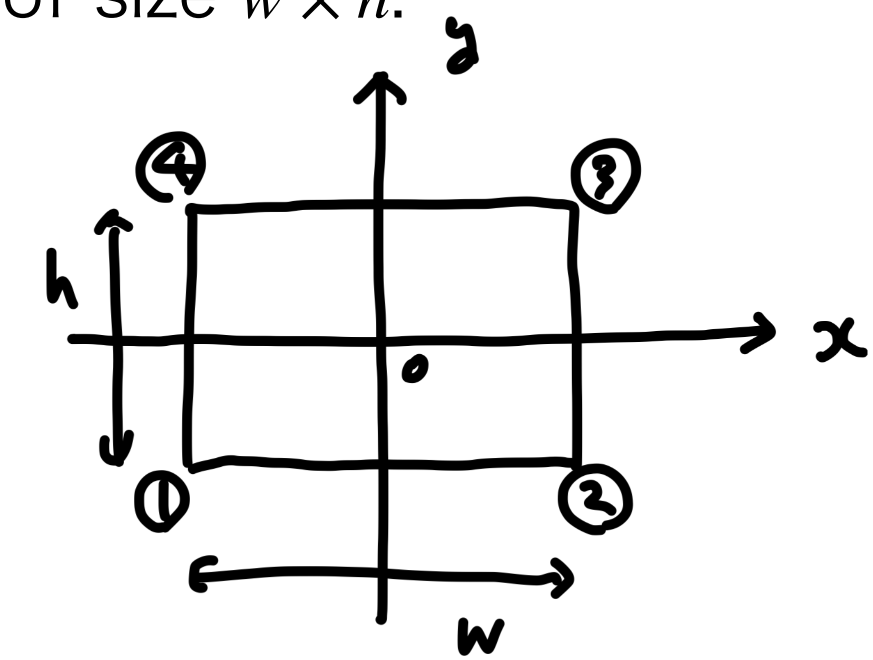
$$
\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{pmatrix} = \mathsf{D}\mathsf{B}\boldsymbol{d}^e.
$$

Usually, the stress is evaluated at the Gauss points because

- the accuracy is good at these points (super-convergence point),
- B-matrices at these points have already been computed.

Remarks on isopararametric elem. 1/4

Let us consider a single element of rectangular shape of size $w \times h$.



A point (x, y) in the element is parametrized as

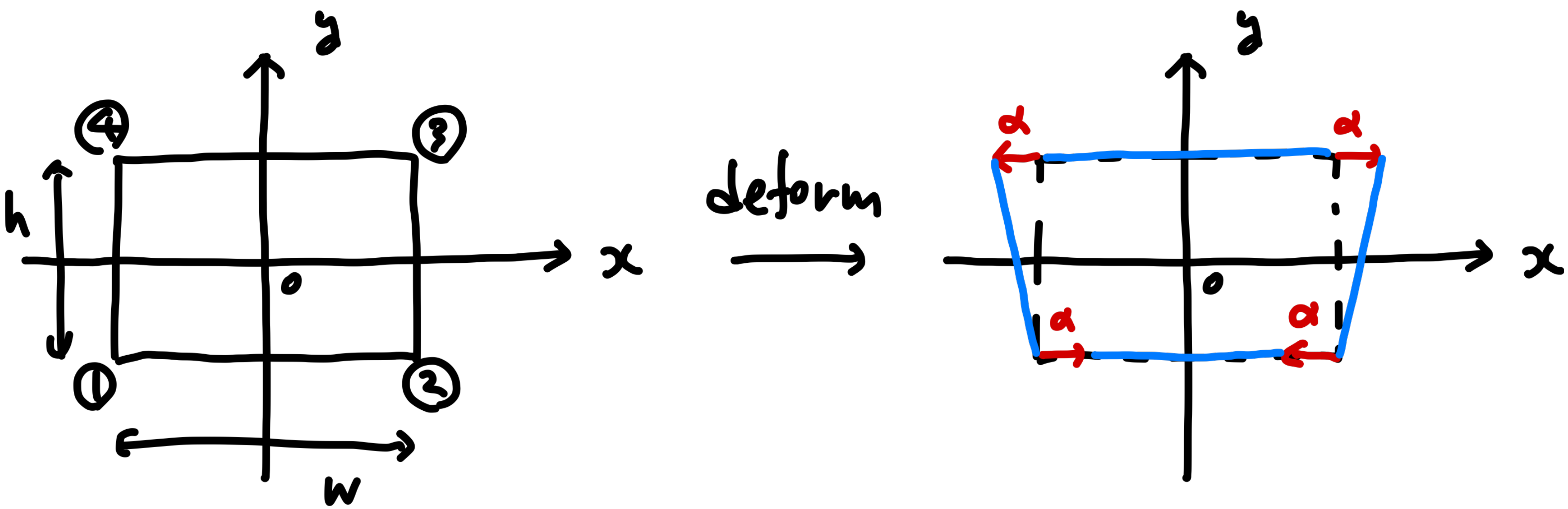
$$
\begin{aligned}
x &= \sum_{i=1}^{4} x_i N_i(\xi, \eta) \\
&= -\frac{w}{2}\frac{1}{4}(1-\xi)(1-\eta) + \frac{w}{2}\frac{1}{4}(1+\xi)(1-\eta) + \frac{w}{2}\frac{1}{4}(1+\xi)(1+\eta) - \frac{w}{2}\frac{1}{4}(1-\xi)(1+\eta) \\
&= \frac{w}{2}\xi
\end{aligned}
$$

$$
\begin{aligned}
y &= \sum_{i=1}^{4} y_i N_i(\xi, \eta) \\
&= -\frac{h}{2}\frac{1}{4}(1-\xi)(1-\eta) - \frac{h}{2}\frac{1}{4}(1+\xi)(1-\eta) + \frac{h}{2}\frac{1}{4}(1+\xi)(1+\eta) + \frac{h}{2}\frac{1}{4}(1-\xi)(1+\eta) \\
&= \frac{h}{2}\eta
\end{aligned}
$$

Remarks on isopararametric elem. 2/4

Let us now consider the following deformation (bending):

$$u_1 = +\alpha,\ u_2 = -\alpha,\ u_3 = +\alpha,\ u_4 = -\alpha$$
$$v_1 = v_2 = v_3 = v_4 = 0$$



The displacement in the element is evaluated as

$$u = \sum_{i=1}^{4} u_i N_i(\xi, \eta) = \alpha\xi\eta$$

$$v = \sum_{i=1}^{4} v_i N_i(\xi, \eta) = 0$$

Remarks on isopararametric elem. 3/4

Let us first compute the strain:

$$\begin{cases} x = \frac{w}{2}\xi \\ y = \frac{h}{2}\eta \end{cases} \rightarrow \begin{array}{ll} \frac{\partial x}{\partial \xi} = \frac{w}{2}, & \frac{\partial x}{\partial \eta} = 0 \\ \frac{\partial y}{\partial \xi} = 0, & \frac{\partial y}{\partial \eta} = \frac{h}{2} \end{array}$$

$$\begin{cases} u = \alpha \xi \eta \\ v = 0 \end{cases} \rightarrow \begin{array}{ll} \frac{\partial u}{\partial \xi} = \alpha \eta, & \frac{\partial u}{\partial \eta} = \alpha \xi \\ \frac{\partial v}{\partial \xi} = 0, & \frac{\partial v}{\partial \eta} = 0 \end{array}$$

On the other hand,

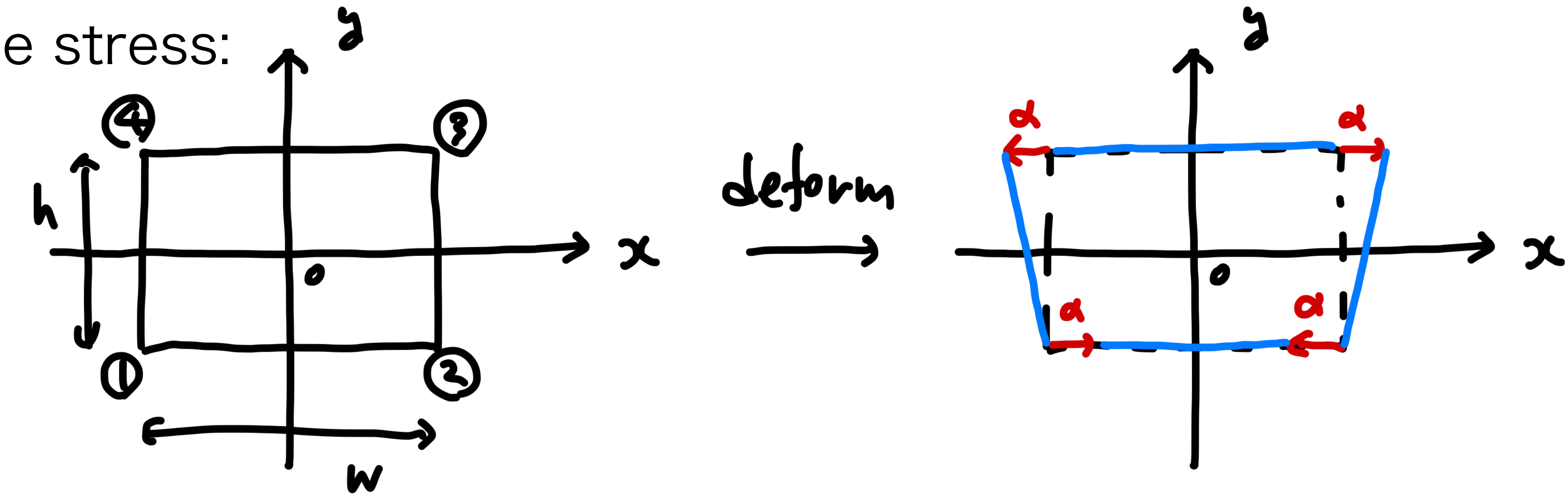
$$\frac{\partial u}{\partial \xi} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial u}{\partial y}\overbrace{\frac{\partial y}{\partial \xi}}^{=0} = \frac{\partial u}{\partial x}\frac{w}{2}$$

$$\rightarrow \varepsilon_{11} = \frac{\partial u}{\partial x} = \frac{2}{w}\alpha\eta \text{ . Similary, } \varepsilon_{22} = 0 \ \& \ \gamma_{12} = \frac{2}{h}\alpha\xi$$

Remarks on isopararametric elem. 4/4

Let us then compute the stress:

$$\sigma_{11} = E\varepsilon_{11} = E\frac{2}{w}\alpha\eta \sim \frac{1}{w}$$
$$\sigma_{22} = E\varepsilon_{22} = 0$$
$$\sigma_{12} = \mu\gamma_{12} = \mu\frac{2}{h}\alpha\xi \sim \frac{1}{h}$$



If we use a very “thin” element i.e. $h \ll w$, then we have $\sigma_{12} \gg \sigma_{11}$. Thus, the shear stress is much greater than the normal one, which is contradict to the beam theory by Bernoulli and Euler.

Thus, the FEM with the isoparametric element overestimate the shear stress, which is called the shear locking.

In order to avoid the shear locking, some improved elements have been proposed such as higher-order elements, (selective) reduced-integration elements, non-conforming elements, etc.

An FEM implementation

We have a nice free software called FreeFEM which is super easy to use.
If you are interested in the software, visit https://freefem.org

freefem.org

FREEFEM DOCUMENTATION COMMUNITY MODULES SOURCE CODE GALLERY EVENTS TRY IT ONLINE DONATE

Example Scripts

- Laplacian
- Stokes
- Elasticity
- Thermal conduction
- Acoustics

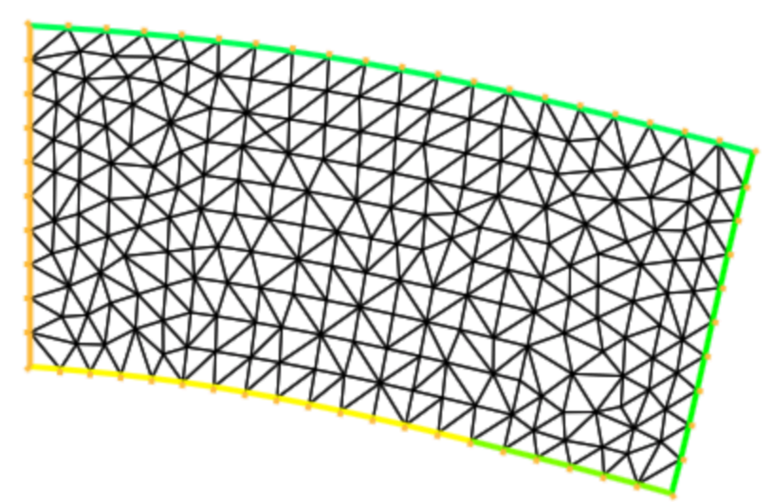
FreeFEM JS only works in 2D. Created by Antoine Le Hyaric.

FreeFEM on GitHub
356 stars - 109 forks

Code SAVE AS EDP

```
// 平面応力状態
lambda=2*lambda*mu/(lambda+2*mu);

/* 荷重を設定 */
func t1 = 0.0;
func t2 = -500.0*1000.0; //下向きに500kN


/* 領域形状を設定 */
border a1(t = 0.05, 0) { x=0; y=t; label=1; };
border a2(t = 0, 0.07) { x=t; y=0; label=2; };
border a3(t = 0.07, 0.10) { x=t; y=0; label=3; };
border a4(t = 0, 0.05) { x=0.1; y=t; label=4; };
border a5(t = 0.1, 0) { x=t; y=0.05; label=5; };

/* 有限要素分割, 有限要素空間, 変数の定義 */
int n=2;
mesh Sh;
Sh = buildmesh(a1(5*n) + a2(7*n) + a3(3*n) + a4(5*n) + a5(10*n));
//plot(Sh,wait=1);
fespace Vh1(Sh,[P1,P1]);
Vh1 [u1, u2]; // 変位
Vh1 [v1, v2]; // 変位（試験関数）

/* 弱形式を定義 */
problem elasticity([u1,u2],[v1,v2]) =
        int2d(Sh)(
        (lambda+2*mu)*dx(v1)*dx(u1)+lambda*dx(v1)*dy(u2)
        +lambda*dy(v2)*dx(u1)+(lambda+2*mu)*dy(v2)*dy(u2)
        +mu*(dy(v1)+dx(v2))*(dy(u1)+dx(u2)))
        -int1d(Sh,4)(v1*t1+v2*t2)
        +on(1,u1=0,u2=0);

/* 有限要素解析を実行*/
elasticity;

/* 変位を1000倍して表示*/
real scale=1000.0;
mesh Th;
Th = movemesh (Sh,[x+scale*u1,y+scale*u2]);
plot(Th,wait=1);
```

Result



Console

On the final report

Final report assignments will be posted on Canvas LMS. Submit your report via Canvas. The due date is 23:59 (JST) on Tuesday 20 July. If you have any question, you can email me: isakari at sd.keio.ac.jp

I hope that you have enjoyed this lecture. Thank you.