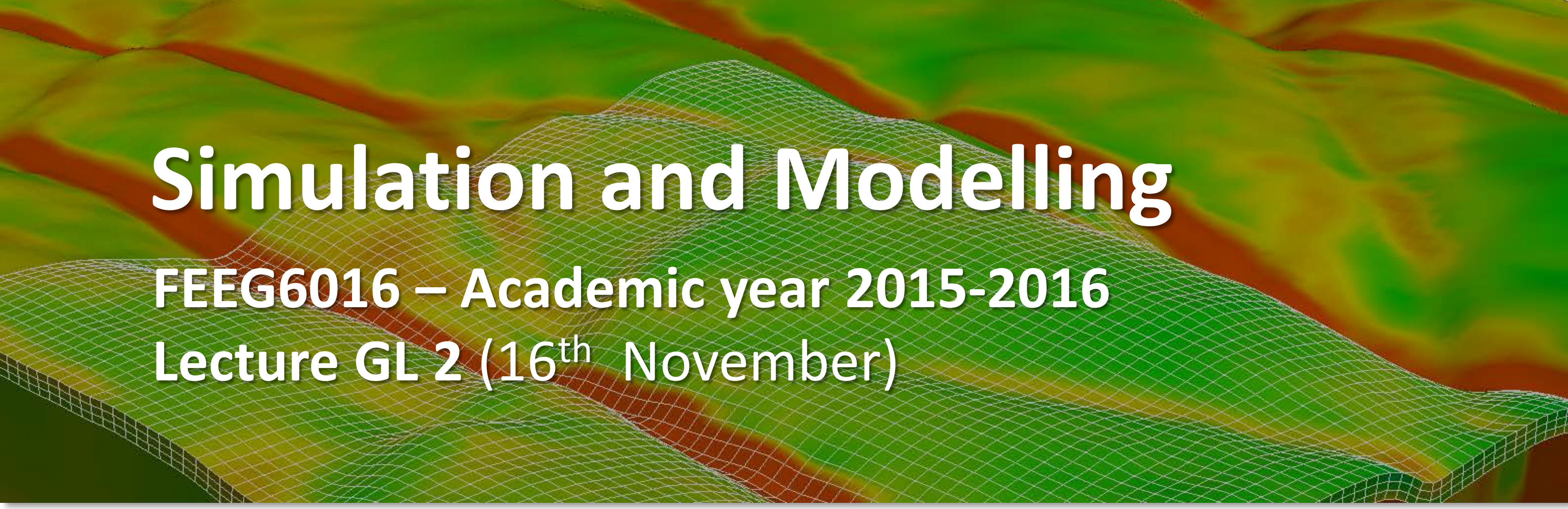


Simulation and Modelling

FEEG6016 – Academic year 2015-2016

Lecture GL 2 (16th November)

Georges Limbert [1, 2, 3]

g.limbert@soton.ac.uk | B7/4075

UNIVERSITY OF Southampton

[1] **national Centre for Advanced Tribology at Southampton**

www.southampton.ac.uk/ncats | www.biotribology.org

[2] **Bioengineering Science Research Group**

www.soton.ac.uk/ses/research/groups/bioeng.page?

Engineering Sciences, Faculty of Engineering and the Environment, University of Southampton, UK

[3] **Biomedical Engineering Association, IMechE**

Institution of Mechanical Engineers (IMechE), London, UK

Lecture 2

Derivation of a finite element formulation

Aims

- Present the general procedure to convert a boundary-value problem into its discretised version

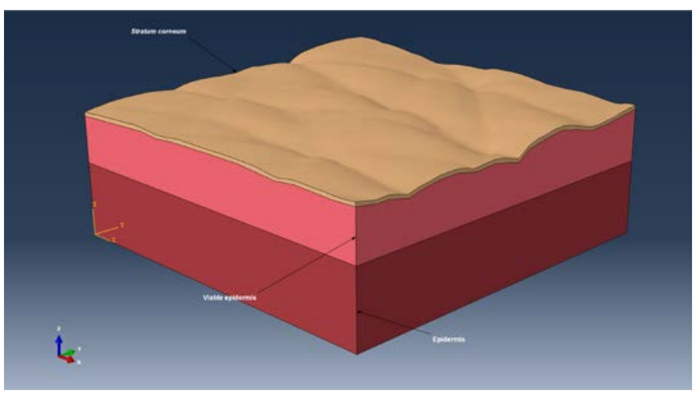
Outcomes

- Ability to derive the variational form of a boundary-value problem
- Ability to discretise the associated field variable using shape functions and derive the discrete version of the boundary-value problem
- Ability to write a procedure to solve a simple 2D elasticity problem

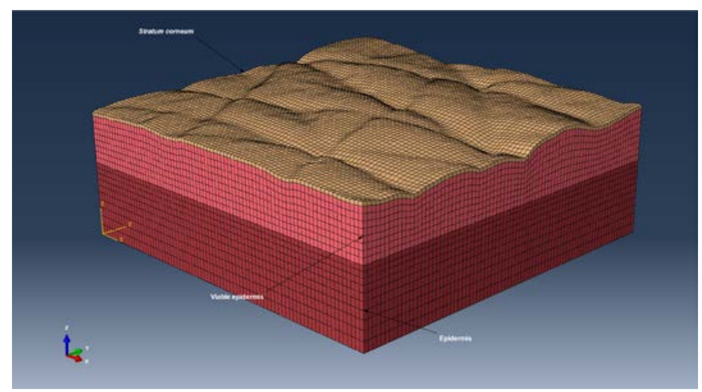
Reminder

The Finite Element Method in a nutshell

- ① Transform a PDE or a system of PDEs associated with time and boundary conditions into a variational problem
- ② Introduce a piece-wise approximation to the field variables (e.g. displacement, temperature, electric charge) in the governing equations
- ③ Discretise the physical domain into elements and write approximate equations for each element (**meshing** process). The local equations in each element can be expressed as matricial equations
- ④ Assemble the local equations of each element into a global matrix
- ⑤ Solve for the global response

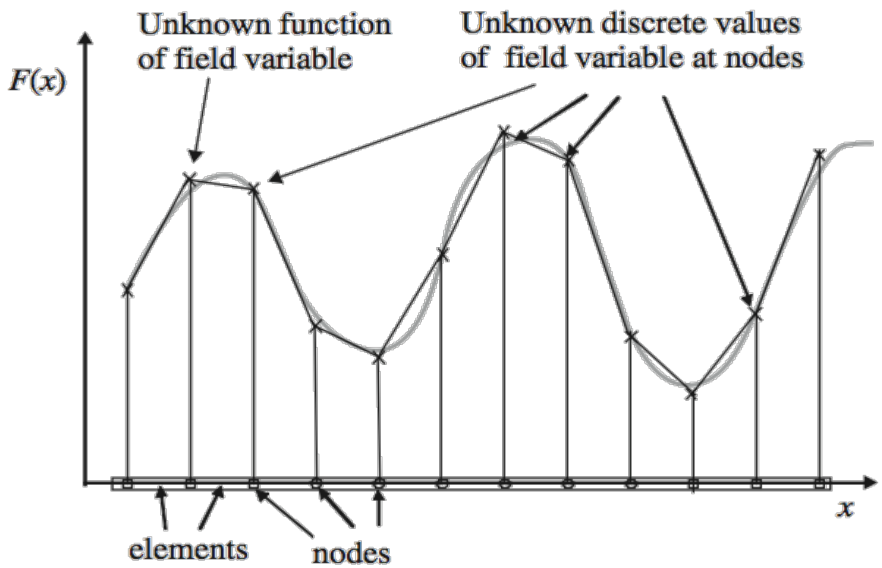


meshing



FEM overview: step ②

- In each element, one need to choose **interpolation functions** (or **shape functions**) to approximate the unknown fields within each element in terms of **nodal values (degrees of freedom)**
- In each element, the value of each field variable anywhere in the element is a linear combination of the shape functions and the nodal values
- Interpolation functions are usually polynomials (Lagrange, Hermite, B-Spline, NURBS, T-Spline)

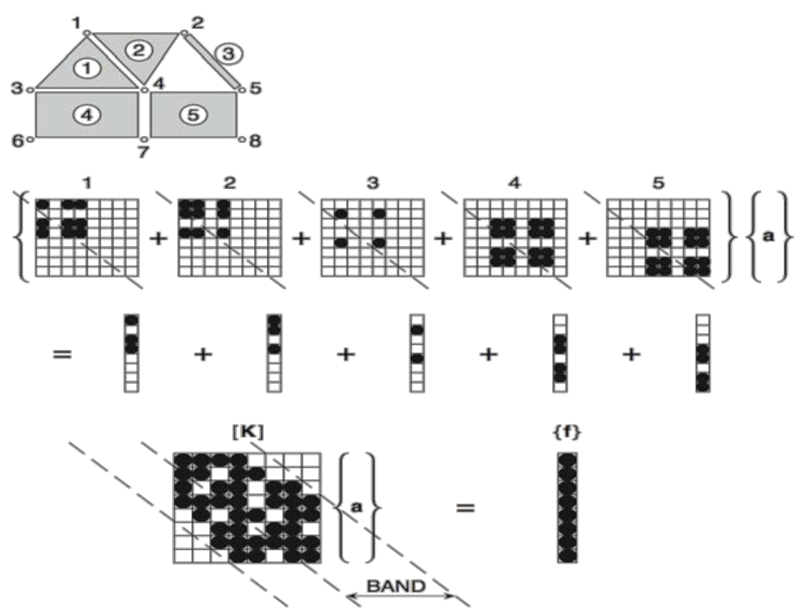


Step ③

- The field approximations are injected into the variational form introduced in step ①. This is defined at the element level.
- After algebraic manipulations one obtains a general matrix equation valid for each element of the mesh

Step ④

Assembly of local element-level equations into a global matrix system



Weak form

General principles
The principle of virtual work

Casting an IBVP into its weak form

Let's consider an arbitrary vector-valued function $\boldsymbol{\eta} = \boldsymbol{\eta}(\mathbf{x}) = \boldsymbol{\eta}(\chi(\mathbf{X}, t))$

This is the **test** of **weighting** function

- Time is assumed to be fixed
- η vanishes on the boundary of where displacements $\mathbf{u}$ are prescribed: $\partial\Omega_u$

Let's write a functional obtained by multiplying the strong form by the test function and integrating over the domain:

$$f(\mathbf{u}, \boldsymbol{\eta}) = \int_{\Omega_t} \left(\text{-div}\boldsymbol{\sigma} - \mathbf{b} + \rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta}\, dv = 0$$

Because the test function is arbitrary the weak and strong form are equivalent:

$$f(\mathbf{u}, \boldsymbol{\eta}) = \int_{\Omega_t} \left(\text{-div}\boldsymbol{\sigma} - \mathbf{b} + \rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta}\, dv = 0 \quad \Leftrightarrow \quad \text{div}\boldsymbol{\sigma} + \mathbf{b} = \rho\dot{\mathbf{v}} = \rho\ddot{\mathbf{u}}$$

Initial boundary-value problem

Strong form

$$\mathrm{div}\boldsymbol{\sigma} + \mathbf{b} = \rho\dot{\mathbf{v}} = \rho\ddot{\mathbf{u}}$$

$$\mathbf{u} = \overline{\mathbf{u}} \qquad on \quad \partial\Omega_u$$

$$\mathbf{t} = \overline{\mathbf{t}} \qquad on \quad \partial\Omega_\sigma$$

$$\mathbf{u}(\mathbf{x},t)\Big|_{t=0} = \mathbf{u}_0(\mathbf{X})$$

$$\dot{\mathbf{u}}(\mathbf{x},t)\Big|_{t=0} = \dot{\mathbf{u}}_0(\mathbf{X})$$

$$\begin{cases} \partial\Omega = \partial\Omega_u \cup \partial\Omega_\sigma \\ \partial\Omega_u \cap \partial\Omega_\sigma = \varnothing \end{cases}$$

Weak form

$$\int_{\Omega_t} \rho\ddot{\mathbf{u}}.\delta\mathbf{u}\, dv = \delta W_{ext}(\mathbf{u},\delta\mathbf{u}) - \delta W_{int}(\mathbf{u},\delta\mathbf{u})$$

$$\delta W_{\mathrm{ext}}(\mathbf{u},\delta\mathbf{u}) = \int_{\Omega_t} \mathbf{b}.\delta\mathbf{u}\, dv + \int_{\partial\Omega_\sigma} \overline{\mathbf{t}}.\delta\mathbf{u}\, ds$$

$$\delta W_{\mathrm{int}}(\mathbf{u},\delta\mathbf{u}) = \int_{\Omega_t} \boldsymbol{\sigma} : \mathrm{grad}\delta\mathbf{u}\, dv$$

$$\int_{\Omega_t} \mathbf{u}(\mathbf{x},t)\Big|_{t=0}.\delta\mathbf{u}\, dv = \int_{\Omega_t} \mathbf{u}_0(\mathbf{X}).\delta\mathbf{u}\, dv$$

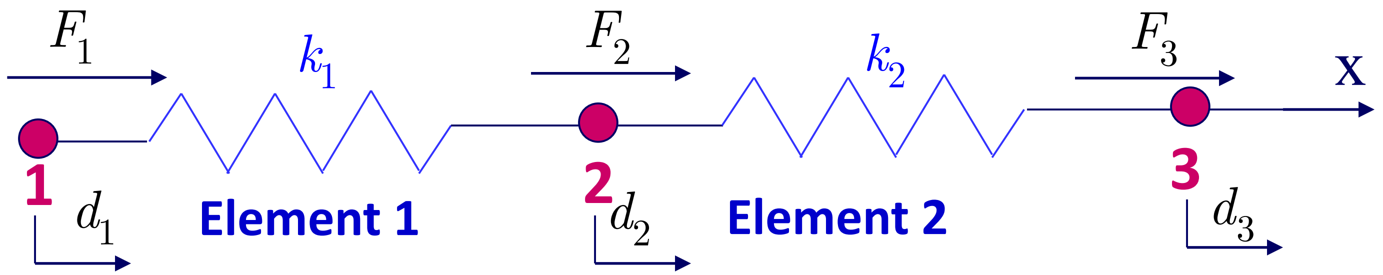
$$\int_{\Omega_t} \dot{\mathbf{u}}(\mathbf{x},t)\Big|_{t=0}.\delta\mathbf{u}\, dv = \int_{\Omega_t} \dot{\mathbf{u}}_0(\mathbf{X}).\delta\mathbf{u}\, dv$$

Field interpolation

Trial function

Previously…

In the **direct stiffness approach**, we derived the stiffness matrix of each element using elementary mechanics of solids (Hooke's law)



$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}$$

Now, the general FEM approach is as follows

- **Step 1:** approximate the displacement within each element

  Discretisation | shape functions

- **Step 2:** approximate the strain and stress within each element from the discretised displacement

  Constitutive relation

- **Step 3:** derive the stiffness matrix within each element using the weak form

  Weak form | Gauss integration

- **Step 4:** derive the global stiffness matrix and nodal force vector

  Assembly procedure

- **Step 5:** solve the system of algebraic equations

Trial solution

Guess a trial form for the solution (finite dimensional approximation):

$$u(x) = a_0\varphi_o(x) + a_1\varphi_1(x) + a_2\varphi_2(x) + \ldots = \sum_{i=0}^{m} a_i\varphi_i(x)$$

where $\varphi_o(x), \varphi_1(x), \varphi_2(x), \ldots, \varphi_m(x)$ are **known basis functions** and

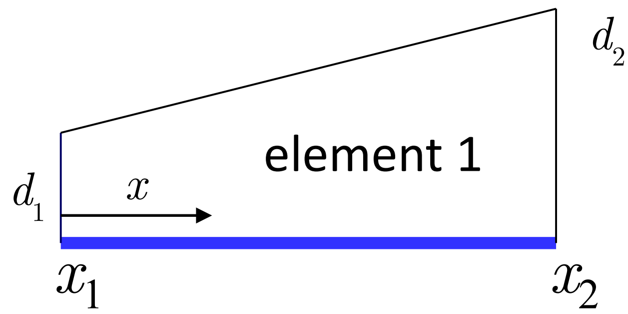
a_i are **undetermined coefficients** chosen such that the approximate solution:

1. Satisfies the boundary conditions (zero boundary residual)
2. Satisfies the governing PDE (zero domain residual)

Step 1: approximation of a field variable (displacement)

Simplest assumption: displacement is varying linearly within the element (1D)

$$\boxed{\mathrm{w}(x) = a_0 + a_1 x}$$



How to obtain a_0 and a_1 in terms of nodal displacements?

$$\begin{aligned} \mathrm{w}(x_1) &= a_0 + a_1 x_1 = d_1 \\ \mathrm{w}(x_2) &= a_0 + a_1 x_2 = d_2 \end{aligned} \Rightarrow \begin{aligned} a_1 &= -\frac{1}{x_2 - x_1} d_1 + \frac{1}{x_2 - x_1} d_2 \\ a_0 &= \frac{x_2}{x_2 - x_1} d_1 - \frac{x_1}{x_2 - x_1} d_2 \end{aligned}$$

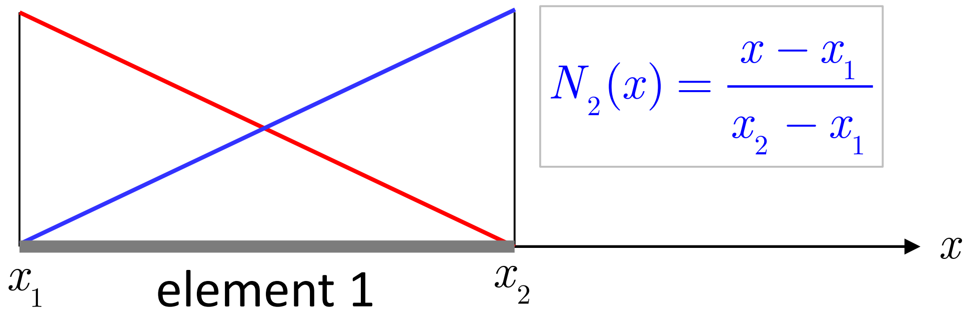
Shape functions

$$\mathrm{w}(x) = \underbrace{\frac{x_2 - x}{x_2 - x_1}}_{N_1(x)} d_1 + \underbrace{\frac{x - x_1}{x_2 - x_1}}_{N_2(x)} d_2 \Rightarrow \mathrm{w}(x) = \mathbf{N}^T \mathbf{d} = \begin{bmatrix} N_1(x) & N_2(x) \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}$$

Properties of shape functions

1/ Kronecker delta property: the shape function at any node has a value of 1 at that node and a value of zero at ALL other nodes

$$N_1(x) = \frac{x_2 - x}{x_2 - x_1}$$



$$N_2(x) = \frac{x - x_1}{x_2 - x_1}$$

$$N_1(x = x_1) = 1; \quad N_1(x = x_2) = 0$$
$$N_2(x = x_1) = 0; \quad N_2(x = x_2) = 1$$

2/ **Compatibility:** the displacement approximation is continuous across element boundaries

3/ **Completeness:**

$$\sum_i N_i(x) = N_1(x) + N_2(x) = 1$$
$$N_1(x)x_1 + N_2(x)x_2 = x \quad \text{for all } x$$

Partition of unity property

Step 2: Derive strain and stress within an element

Continuous strain:

$$\varepsilon(x) = \frac{d\mathrm{w}(x)}{dx}$$

Discretised strain:

$$\varepsilon^e(x) = \frac{d[\mathbf{N}^T(x)\mathbf{d}]}{dx} = \begin{bmatrix} \frac{dN_1(x)}{dx} & \frac{dN_2(x)}{dx} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} = \mathbf{Bd}$$

The strain-displacement matrix

$$\mathbf{B} = \frac{d\mathbf{N}^T}{dx}$$

$$\boxed{\varepsilon^e(x) = \mathbf{Bd}}$$

Step 2: Derive strain and stress within an element

Shape functions for a linear element

$$N_1(x) = \frac{x_2 - x}{x_2 - x_1}$$

$$N_2(x) = \frac{x - x_1}{x_2 - x_1}$$

The strain-displacement matrix for a linear element

$$\mathbf{B} = \frac{d\mathbf{N}^T}{dx} = \left[\frac{d}{dx}\left(\frac{x_2 - x}{x_2 - x_1}\right) \quad \frac{d}{dx}\left(\frac{x - x_1}{x_2 - x_1}\right) \right] = \left[\frac{-1}{x_2 - x_1} \quad \frac{1}{x_2 - x_1} \right]$$

Strain

$$\varepsilon^e = \left[\frac{-1}{x_2 - x_1} \quad \frac{1}{x_2 - x_1} \right] \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}$$

Strain is constant within a linearly interpolated displacement-based element

Step 2: Derive strain and stress within an element

1D Hooke's law (E is the Young's modulus of the material)

$$\sigma = E\varepsilon = E\frac{d\mathrm{w}}{dx}$$

Stress

$$\sigma^{e} = E\,\mathbf{B}\mathbf{d}$$

Stress is constant within a linearly interpolated displacement-based element

Generally (e.g. for Lagrange polynomials), stress and strain are **discontinuous across elements**

Summary

Displacement approximation

$$\mathrm{w}(x) = \mathbf{N}^T \mathbf{d}$$

Linear interpolation

$$\mathrm{w}(x) = \begin{bmatrix} \dfrac{x_2 - x}{x_2 - x_1} & \dfrac{x - x_1}{x_2 - x_1} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}$$

Strain approximation

$$\varepsilon(x) = \mathbf{Bd}$$

$$\varepsilon = \begin{bmatrix} \dfrac{-1}{x_2 - x_1} & \dfrac{1}{x_2 - x_1} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}$$

Stress approximation

$$\sigma = E\,\mathbf{Bd}$$

$$\sigma = \frac{E}{x_2 - x_1} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}$$

A small detour…

3D weak form of static elastic equilibrium

Weak form (static equilibrium equation)

$$\underbrace{\int_{\Omega_t} \boldsymbol{\sigma} : \mathrm{grad}\delta\mathbf{u}\, dv}_{\delta W_{\mathrm{int}}(\mathbf{u},\delta\mathbf{u})} = \underbrace{\int_{\Omega_t} \mathbf{b}\delta\mathbf{u}\, dv + \int_{\boxed{\partial\Omega_\sigma}} \overline{\mathbf{t}}.\delta\mathbf{u}\, \boxed{ds}}_{\delta W_{\mathrm{ext}}(\mathbf{u},\delta\mathbf{u})}$$

Constitutive equations (linear elasticity): to link strain to stress

$$\boxed{\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}}$$

$$(\boldsymbol{\sigma})_{ij} = \sigma_{ij} = (\mathbb{C} : \boldsymbol{\varepsilon})_{ij} = \mathbb{C}_{ijkl}\varepsilon_{kl} \qquad \{2\mathrm{D} : i,j,k,l : 1..2; \quad 3\mathrm{D} : i,j,k,l : 1..3\}$$

2D case (tensor to matrix equation)

$$\{\boldsymbol{\sigma}\} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \underbrace{\frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix}}_{[\mathbb{C}]} \underbrace{\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{Bmatrix}}_{\{\boldsymbol{\varepsilon}\}}$$

Task 3: stiffness matrix (3D)

Weak form (static equilibrium equation)

$$\int_{\Omega_t} \boldsymbol{\sigma} : \mathrm{grad}\delta\mathbf{w}\, dv = \int_{\Omega_t} \mathbf{b}\delta\mathbf{w}\, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{\mathbf{t}}.\delta\mathbf{w}\, \boxed{ds} \Rightarrow \int_{\Omega_t} \sigma_{ij}\delta\varepsilon_{ij}\, dv = \int_{\Omega_t} b_i\delta w_i\, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i\delta w_i\, \boxed{ds}$$

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon} \Rightarrow \sigma_{ij} = \mathbb{C}_{ijkl}\varepsilon_{kl}$$

$$\delta\boldsymbol{\varepsilon} = \mathrm{grad}_s(\delta\mathbf{w}) = \frac{1}{2}(\mathrm{grad}(\delta\mathbf{w}) + \mathrm{grad}(\delta\mathbf{w})^T) \Rightarrow \delta\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial\delta w_i}{\partial x_j} + \frac{\partial\delta w_j}{\partial x_i}\right)$$

$$\sigma_{ij}\delta\varepsilon_{ij} = \frac{1}{2}\sigma_{ij}\left(\frac{\partial\delta w_i}{\partial x_j} + \frac{\partial\delta w_j}{\partial x_i}\right) = \sigma_{ij}\frac{\partial\delta w_i}{\partial x_j} = \mathbb{C}_{ijkl}\frac{\partial u_k}{\partial x_l}\frac{\partial\delta w_i}{\partial x_j} = \mathbb{C}_{ijkl}\frac{\partial u_i}{\partial x_j}\frac{\partial\delta w_i}{\partial x_j}$$

Displacement

$$\mathbf{u} = N^A\mathbf{u}^A \Rightarrow u_i = N^A(\mathbf{x})u_i^A \qquad u_i^B = N^A(\mathbf{x}^B)u_i^A$$

$$N^A(\mathbf{x}^B) = \begin{cases} 1 & A = B \\ 0 & A \neq B \end{cases}$$

Test function

$$\delta w_i = N^A(\mathbf{x})\delta w_i^A$$

Task 3: stiffness matrix (3D)

Weak form (static equilibrium equation)

$$\int_{\Omega_t} \sigma_{ij}\delta\varepsilon_{ij}\, dv = \int_{\Omega_t} b_i \delta w_i\, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i \delta w_i \boxed{ds}$$

$$\sigma_{ij}\delta\varepsilon_{ij} = \mathbb{C}_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial \delta w_i}{\partial x_j} = \mathbb{C}_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial \left(N^A(\mathbf{x}) \delta w_i^A \right)}{\partial x_j} = \mathbb{C}_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial N^A(\mathbf{x})}{\partial x_j} \delta w_i^A$$

$$\Rightarrow \int_{\Omega_t} \mathbb{C}_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial N^B(\mathbf{x})}{\partial x_j} \delta w_i^B\, dv = \int_{\Omega_t} b_i N^A(\mathbf{x}) \delta w_i^A\, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i N^A(\mathbf{x}) \delta w_i^A \boxed{ds}$$

$$\Rightarrow \boxed{\delta w_i^A} \left[\int_{\Omega_t} \mathbb{C}_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial N^A(\mathbf{x})}{\partial x_j} dv - \left(\int_{\Omega_t} b_i N^A(\mathbf{x})\, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i N^A(\mathbf{x}) \boxed{ds} \right) \right] = 0$$

$$\Rightarrow \int_{\Omega_t} \mathbb{C}_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial N^A(\mathbf{x})}{\partial x_j} dv - \left(\int_{\Omega_t} b_i N^A(\mathbf{x})\, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i N^A(\mathbf{x}) \boxed{ds} \right) = 0$$

Task 3: stiffness matrix (3D)

Matrix form of the weak form

$$\int_{\Omega_t} \mathbb{C}_{ijkl} \frac{\partial N^A(\mathbf{x})}{\partial x_j} \frac{\partial N^B(\mathbf{x})}{\partial x_j} u_i^A \, dv - \left(\int_{\Omega_t} b_i N^A(\mathbf{x}) \, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i N^A(\mathbf{x}) \boxed{ds} \right) = 0$$

$$K_{AiBk} = \int_{\Omega_t} \mathbb{C}_{ijkl} \frac{\partial N^A(\mathbf{x})}{\partial x_j} \frac{\partial N^B(\mathbf{x})}{\partial x_j} dv$$

$$F_i^A = -\left(\int_{\Omega_t} b_i N^A(\mathbf{x}) \, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i N^A(\mathbf{x}) \boxed{ds} \right)$$

$$\Rightarrow \quad \boxed{K_{AiBk} u_i^A - F_i^A = 0}$$

$\mathbf{K}$ is the stiffness matrix

$\mathbf{F}$ is the nodal force vector

Task 3: stiffness matrix (3D)

Matrix form of the weak form

$$\int_{\Omega_t} \mathbb{C}_{ijkl} \frac{\partial N^A(\mathbf{x})}{\partial x_j} \frac{\partial N^B(\mathbf{x})}{\partial x_j} u_i^A \, dv - \left(\int_{\Omega_t} b_i N^A(\mathbf{x}) \, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i N^A(\mathbf{x}) \boxed{ds} \right) = 0$$

$$K_{AiBk} = \int_{\Omega_t} \mathbb{C}_{ijkl} B_{jA} B_{jB} \, dv$$

$$F_i^A = -\left(\int_{\Omega_t} b_i N^A(\mathbf{x}) \, dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{t}_i N^A(\mathbf{x}) \boxed{ds} \right)$$

$$\Rightarrow$$

$$K_{AiBk} u_i^A - F_i^A = 0$$

$$\mathbf{K} = \int_{\Omega_t} \mathbf{B} \mathbb{C} \mathbf{B}^T \, dv$$

$$\mathbf{F}^A = -\left(\int_{\Omega_t} \mathbf{b} \mathbf{N}^A dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{\mathbf{t}} \mathbf{N}^A \boxed{ds} \right)$$

$\mathbf{K}$ is the stiffness matrix

$\mathbf{F}$ is the nodal force vector

Task 3: stiffness matrix (1D)

Strong form (1D) (static, no body forces)

$$\mathrm{div}(\sigma) = 0$$

Weak form (3D) (static, no body forces)

$$\mathbf{K}\mathbf{u}^{A} - \mathbf{F}^{A} = \mathbf{0}$$

$$\mathbf{K} = \int_{\Omega_t} \mathbf{B}\mathbb{C}\mathbf{B}^{T}\, dv; \qquad \mathbf{F}^{A} = -\left(\int_{\Omega_t} \mathbf{b}\mathbf{N}^{A} dv + \int_{\boxed{\partial\Omega_\sigma}} \bar{\mathbf{t}}\mathbf{N}^{A} \boxed{ds} \right)$$

1D

$$\mathbf{N}^{T} = \begin{bmatrix} N_1(x) & N_2(x) \end{bmatrix} = \begin{bmatrix} \dfrac{x_2 - x}{x_2 - x_1} & \dfrac{x - x_1}{x_2 - x_1} \end{bmatrix}; \qquad \mathbf{B} = \begin{bmatrix} \dfrac{-1}{x_2 - x_1} & \dfrac{1}{x_2 - x_1} \end{bmatrix}$$

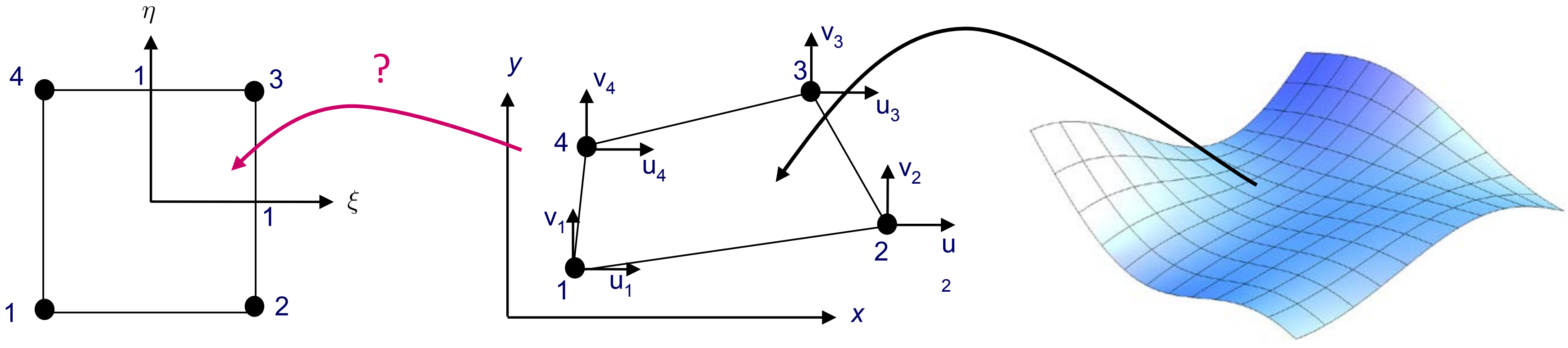
$$\frac{E}{x_2 - x_1}\begin{bmatrix} -1 & 1 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \frac{E}{x_2 - x_1}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

compare to Lecture 1

$$\begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} \end{Bmatrix} = \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix}\begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}$$

Example: a 4-noded quadrilateral element

Bilinear interpolation



Parent element (parametric domain) **Actual element** (physical domain) **Finite element mesh**

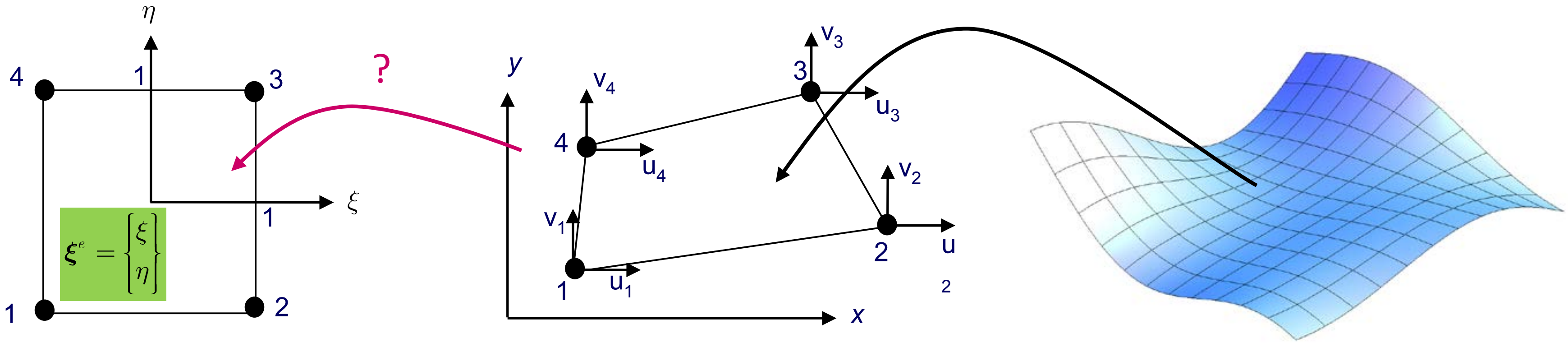
$$N_i = \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta) \qquad \xi_i, \eta_i = -1,1$$

Parametric coordinates $\boldsymbol{\xi}^e = \begin{Bmatrix} \xi \\ \eta \end{Bmatrix}$

$$\begin{Bmatrix} N_1 = \frac{1}{4}\left(1-\xi\right)\left(1-\eta\right) \\ N_2 = \frac{1}{4}\left(1+\xi\right)\left(1-\eta\right) \\ N_3 = \frac{1}{4}\left(1+\xi\right)\left(1+\eta\right) \\ N_4 = \frac{1}{4}\left(1-\xi\right)\left(1+\eta\right) \end{Bmatrix}$$

Example: a 4-noded quadrilateral element

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



Parent element (parametric domain) **Actual element** (physical domain) **Finite element mesh**

Interpolation of the position vector

$$\boxed{\mathbf{x}\left(\boldsymbol{\xi}^e, t\right) = \mathbf{x}_1(t)N_1\left(\boldsymbol{\xi}^e\right) + \mathbf{x}_2(t)N_2\left(\boldsymbol{\xi}^e\right) + \mathbf{x}_3(t)N_3\left(\boldsymbol{\xi}^e\right) + \mathbf{x}_4(t)N_4\left(\boldsymbol{\xi}^e\right)} \qquad \mathbf{x}_A(t) = \text{position}\,(node\ A) = \begin{Bmatrix} x_A \\ y_A \end{Bmatrix}$$

$$\mathbf{x}\left(\boldsymbol{\xi}^e, t\right) = \begin{Bmatrix} x(t) \\ y(t) \end{Bmatrix}_{node\,1} N_1\left(\boldsymbol{\xi}^e\right) + \begin{Bmatrix} x(t) \\ y(t) \end{Bmatrix}_{node\,2} N_2\left(\boldsymbol{\xi}^e\right) + \begin{Bmatrix} x(t) \\ y(t) \end{Bmatrix}_{node\,3} N_3\left(\boldsymbol{\xi}^e\right) + \begin{Bmatrix} x(t) \\ y(t) \end{Bmatrix}_{node\,4} N_4\left(\boldsymbol{\xi}^e\right)$$

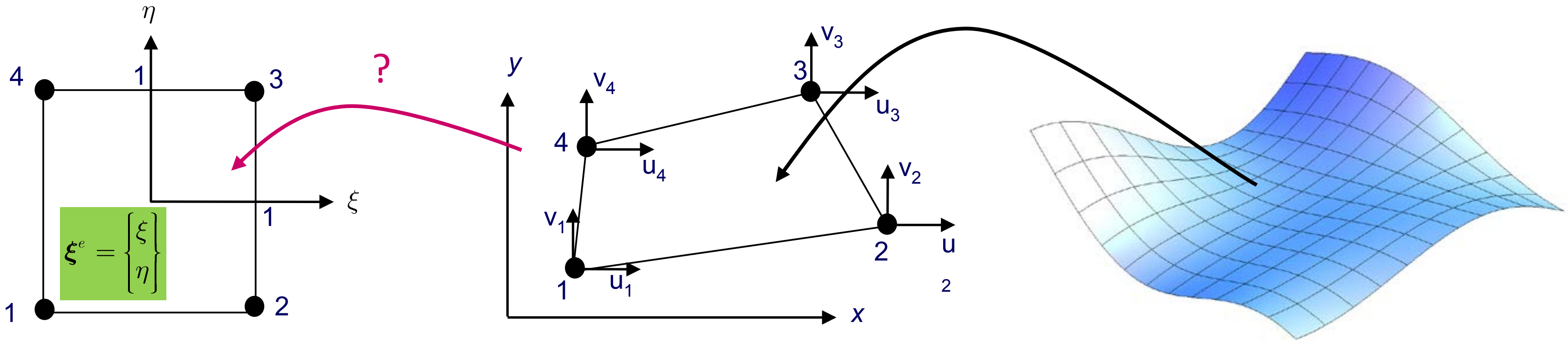
Interpolation of displacement vector

$$\boxed{\mathbf{d}\left(\boldsymbol{\xi}^e, t\right) = \mathbf{d}_1(t)N_1\left(\boldsymbol{\xi}^e\right) + \mathbf{d}_2(t)N_2\left(\boldsymbol{\xi}^e\right) + \mathbf{d}_3(t)N_3\left(\boldsymbol{\xi}^e\right) + \mathbf{d}_4(t)N_4\left(\boldsymbol{\xi}^e\right)} \qquad \mathbf{d}_A(t) = \text{displacement}\,(node\ A) = \begin{Bmatrix} u_A \\ v_A \end{Bmatrix}$$

$$\mathbf{d} = \{u_1, v_1, u_2, v_2, u_3, v_3, u_4, v_4\}^T$$

Example: a 4-noded quadrilateral element

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



Parent element (parametric domain) **Actual element** (physical domain) **Finite element mesh**

Interpolation of the position vector

$$\mathbf{x}\left(\boldsymbol{\xi}^e,t\right) = N_I\left(\boldsymbol{\xi}^e\right)\mathbf{x}_I(t) = \mathbf{N}^T\left(\boldsymbol{\xi}^e\right)\mathbf{x}_I(t)$$

Interpolation of displacement vector

$$\mathbf{d}\left(\boldsymbol{\xi}^e,t\right) = N_I\left(\boldsymbol{\xi}^e\right)\mathbf{d}_I(t) = \mathbf{N}^T\left(\boldsymbol{\xi}^e\right)\mathbf{d}_I(t)$$

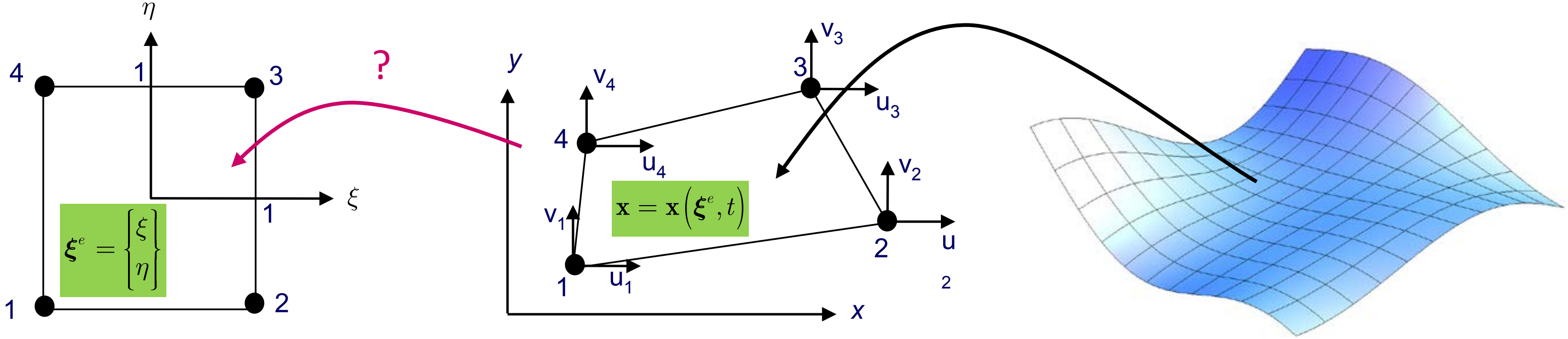
Interpolation of test function

$$\boldsymbol{w}\left(\boldsymbol{\xi}^e,t\right) = N_I\left(\boldsymbol{\xi}^e\right)w_I(t) = \mathbf{N}^T\left(\boldsymbol{\xi}^e\right)w_I(t)$$

Shape function vector

$$\mathbf{N}^T\left(\boldsymbol{\xi}^e\right) = \left\{N_1(\boldsymbol{\xi}^e), N_2(\boldsymbol{\xi}^e), N_3(\boldsymbol{\xi}^e), N_4(\boldsymbol{\xi}^e)\right\}$$

$$\boxed{\mathbf{N}^T\left(\boldsymbol{\xi}^e\right)=\left\{N_1(\boldsymbol{\xi}^e),N_2(\boldsymbol{\xi}^e),N_3(\boldsymbol{\xi}^e),N_4(\boldsymbol{\xi}^e)\right\}}\qquad N_i=\frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)\qquad \xi_i,\eta_i=-1,1$$



Parent element (parametric domain) **Actual element** (physical domain) **Finite element mesh**

There is a dependency between the parametric and actual coordinates that must be accounted for the calculus of derivatives of the shape functions but $\mathbf{x}=\mathbf{x}\left(\xi^e,t\right)$ **is generally not invertible**

$$\frac{\partial N_I\left(\boldsymbol{\xi}^e\right)}{\partial\mathbf{x}}=\frac{\partial N_I\left(\boldsymbol{\xi}^e\right)}{\partial\boldsymbol{\xi}^e}\frac{\partial\boldsymbol{\xi}^e}{\partial\mathbf{x}}=\frac{\partial N_I\left(\boldsymbol{\xi}^e\right)}{\partial\boldsymbol{\xi}^e}\underbrace{\left(\frac{\partial\mathbf{x}}{\partial\boldsymbol{\xi}^e}\right)^{-1}}_{\mathbf{J}^{-1}_{\xi^e}}$$

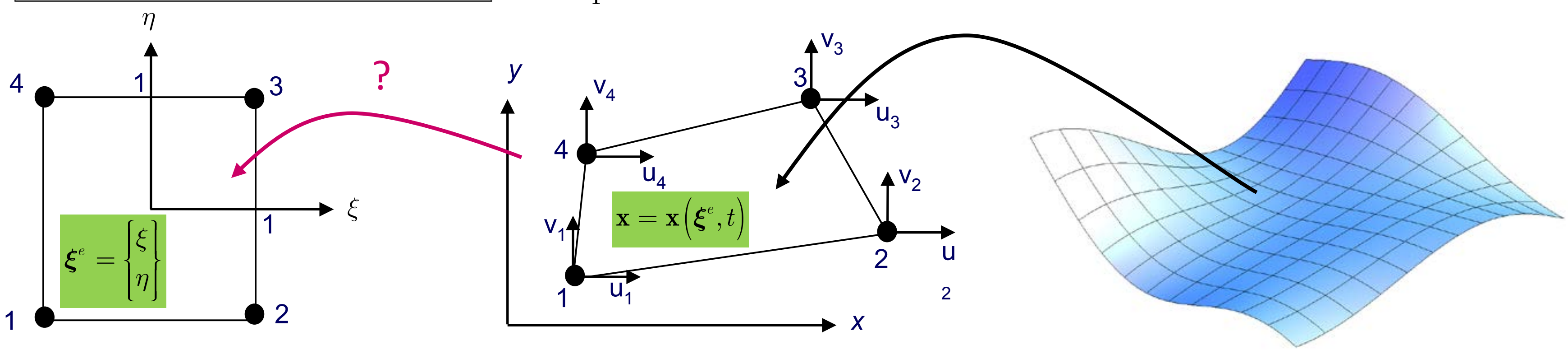
$$\frac{\partial\mathbf{x}}{\partial\boldsymbol{\xi}^e}=\mathbf{x}_I\otimes\frac{\partial N_I\left(\boldsymbol{\xi}^e\right)}{\partial\boldsymbol{\xi}^e}$$

$\mathbf{J}$ **is the Jacobian of the geometric transformation from the parent element to the actual element**

$$\mathbf{J}\;=\frac{\partial\mathbf{x}}{\partial\boldsymbol{\xi}^e}=\begin{bmatrix}\dfrac{\partial\left(N_Ix_I\right)}{\partial\xi} & \dfrac{\partial\left(N_Ix_I\right)}{\partial\eta}\\ \dfrac{\partial\left(N_Iy_I\right)}{\partial\xi} & \dfrac{\partial\left(N_Iy_I\right)}{\partial\eta}\end{bmatrix}=\mathbf{x}_I\otimes\frac{\partial N_I\left(\boldsymbol{\xi}^e\right)}{\partial\boldsymbol{\xi}^e}=;\;J\;=\det(\mathbf{J}\;)>0$$

Derivatives of the shape functions

$\mathbf{N}^T\left(\boldsymbol{\xi}^e\right)=\left\{N_1(\boldsymbol{\xi}^e),N_2(\boldsymbol{\xi}^e),N_3(\boldsymbol{\xi}^e),N_4(\boldsymbol{\xi}^e)\right\}$ $\quad N_i = \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta) \quad \xi_i,\eta_i = -1,1$



Parent element (parametric domain) **Actual element** (physical domain) **Finite element mesh**

$$\frac{\partial N_I\left(\boldsymbol{\xi}^e\right)}{\partial \mathbf{x}}=\frac{\partial N_I\left(\xi,\eta\right)}{\partial(x,y)}=\begin{bmatrix}\frac{\partial\xi}{\partial x} & \frac{\partial\xi}{\partial y}\\ \frac{\partial\eta}{\partial x} & \frac{\partial\eta}{\partial y}\end{bmatrix}\begin{Bmatrix}\frac{\partial N_I}{\partial\xi}\\ \frac{\partial N_I}{\partial\eta}\end{Bmatrix}=\mathbf{J}^{-1}\begin{Bmatrix}\frac{\partial N_I}{\partial\xi}\\ \frac{\partial N_I}{\partial\eta}\end{Bmatrix}$$

$$\frac{\partial x}{\partial\xi}=x_I\frac{\partial N_I}{\partial\xi}; \qquad \frac{\partial x}{\partial\eta}=x_I\frac{\partial N_I}{\partial\eta}$$
$$\frac{\partial y}{\partial\xi}=y_I\frac{\partial N_I}{\partial\xi}; \qquad \frac{\partial y}{\partial\eta}=y_I\frac{\partial N_I}{\partial\eta}$$

$$\begin{bmatrix}\frac{\partial x}{\partial\xi} & \frac{\partial x}{\partial\eta}\\ \frac{\partial y}{\partial\xi} & \frac{\partial y}{\partial\eta}\end{bmatrix}=\begin{bmatrix}\frac{\partial N_1}{\partial\xi} & \frac{\partial N_2}{\partial\xi} & \frac{\partial N_3}{\partial\xi} & \frac{\partial N_4}{\partial\xi}\\ \frac{\partial N_1}{\partial\eta} & \frac{\partial N_2}{\partial\eta} & \frac{\partial N_3}{\partial\eta} & \frac{\partial N_4}{\partial\eta}\end{bmatrix}\begin{bmatrix}x_1 & y_1\\ x_2 & y_2\\ x_3 & y_3\\ x_4 & y_4\end{bmatrix};$$

B matrix and 2D strain tensor

$$\begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial N_1}{\partial \xi} & \dfrac{\partial N_2}{\partial \xi} & \dfrac{\partial N_3}{\partial \xi} & \dfrac{\partial N_4}{\partial \xi} \\ \dfrac{\partial N_1}{\partial \eta} & \dfrac{\partial N_2}{\partial \eta} & \dfrac{\partial N_3}{\partial \eta} & \dfrac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix};$$

$$\boxed{\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{xx} \\ 2\varepsilon_{xy} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & 0 & \dfrac{\partial N_2}{\partial x} & 0 & \dfrac{\partial N_3}{\partial x} & 0 & \dfrac{\partial N_4}{\partial x} & 0 \\ 0 & \dfrac{\partial N_1}{\partial y} & 0 & \dfrac{\partial N_2}{\partial y} & 0 & \dfrac{\partial N_3}{\partial y} & 0 & \dfrac{\partial N_4}{\partial y} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial y} & \dfrac{\partial N_2}{\partial x} & \dfrac{\partial N_3}{\partial y} & \dfrac{\partial N_3}{\partial x} & \dfrac{\partial N_4}{\partial y} & \dfrac{\partial N_4}{\partial x} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} = \mathbf{Bd}}$$

$$\boxed{\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{xx} \\ 2\varepsilon_{xy} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial x} & \dfrac{\partial N_3}{\partial x} & \dfrac{\partial N_4}{\partial x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_2}{\partial y} & \dfrac{\partial N_3}{\partial y} & \dfrac{\partial N_4}{\partial y} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial y} & \dfrac{\partial N_2}{\partial x} & \dfrac{\partial N_3}{\partial y} & \dfrac{\partial N_3}{\partial x} & \dfrac{\partial N_4}{\partial y} & \dfrac{\partial N_4}{\partial x} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{Bmatrix} = \mathbf{B}^{*}\mathbf{d}}$$

How to numerically calculate integrals?

Classical Gauss integration rule (1D)

$$\boxed{\int_{-1}^{1} f(\xi)d\xi \approx \sum_{i=1}^{p} w_i f(\xi_i)}$$

w_i are called integration or Gauss **weights**

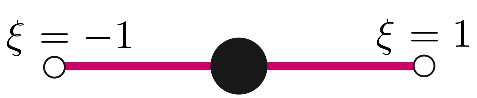
ξ_i are sample-point abscissae in the interval [-1,1] or Gauss/integration **points**

There are a large number of integration schemes (Gauss, Lobatto, etc) and associated formulas that provide weights and points (e.g. *Handbook of Mathematical Functions*)

Examples:

1 point rule

$$\boxed{\int_{-1}^{1} f(\xi)d\xi \approx \sum_{i=1}^{p=1} w_i f(0) = 2f(0)}$$



2 points rule

$$\boxed{\int_{-1}^{1} f(\xi)d\xi \approx \sum_{i=1}^{p=2} w_i f(\xi_i) = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)}$$

How to numerically calculate integrals?

Gauss integration over the parametric domain (2D)

$$\boxed{\int_{\square} f(\boldsymbol{\xi})\ d\square = \int_{-1}^{1}\int_{-1}^{1} f(\boldsymbol{\xi}) d\xi d\eta = w_i w_j f(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)}$$

$$\square = \{\{-1,1\},\{-1,1\}\}$$

w_i are called integration or Gauss **weights**

$\boldsymbol{\xi}_i$ are the integration of Gauss **points**

How to transform an integral over the parametric to the physical domain?

Change of variable!

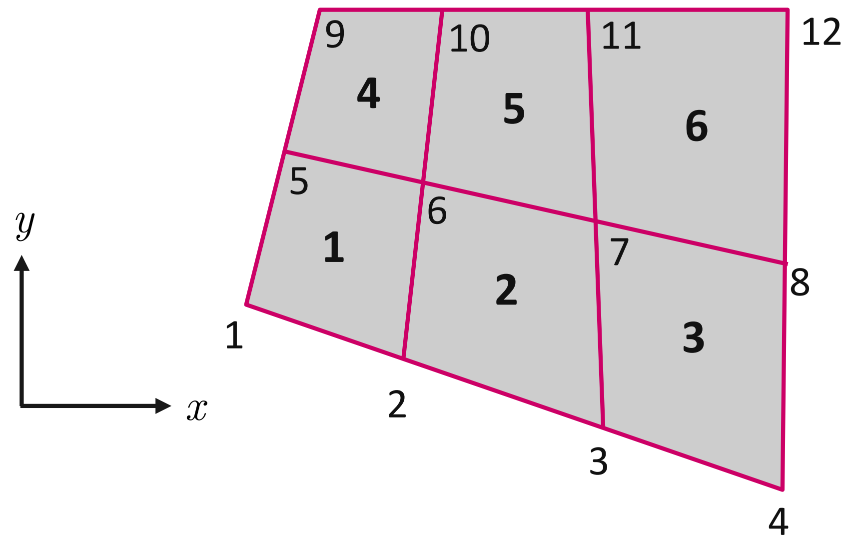
$$d\Omega_e = dxdy = \det(\mathbf{J}_{\boldsymbol{\xi}^e})d\xi d\eta$$

$$\boxed{\int_{\Omega^e} f(\mathbf{x})\ d\Omega = \int_{\square} f(\mathbf{x}(\boldsymbol{\xi}^e)) \underbrace{\det\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}^e}\right)}_{J_{\boldsymbol{\xi}^e} = \det(\mathbf{J}_{\boldsymbol{\xi}^e})} d\square = w_i w_j J_{\boldsymbol{\xi}^e}({\boldsymbol{\xi}^e}_i, {\boldsymbol{\xi}^e}_j) f({\boldsymbol{\xi}^e}_i, {\boldsymbol{\xi}^e}_j)}$$

Example

$$\mathbf{G}_1 = \left\{-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, w_1 = 1\right\}; \mathbf{G}_2 = \left\{\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, w_2 = 1\right\}; \mathbf{G}_3 = \left\{\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, w_3 = 1\right\}; \mathbf{G}_4 = \left\{-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, w_4 = 1\right\}$$

Assembly procedure: example



$$\mathbf{NodeCoordinates} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{12} & y_{12} \end{bmatrix}$$

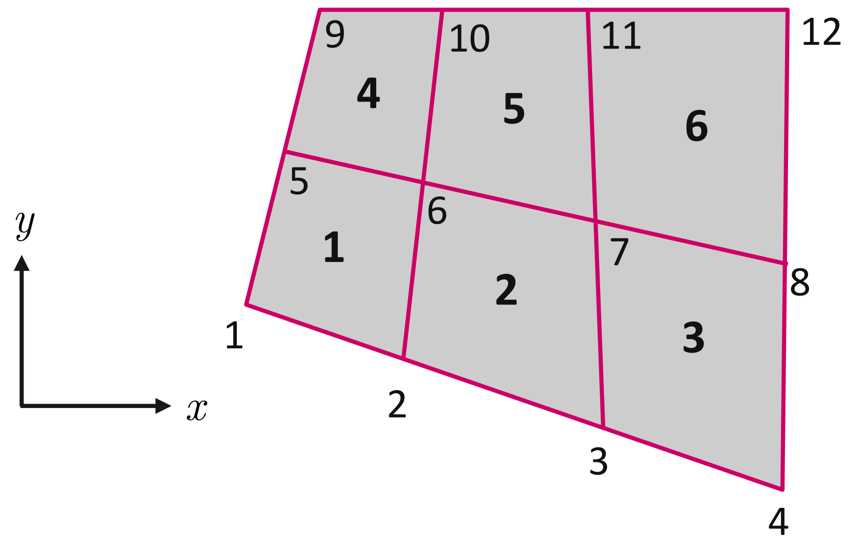
$$\mathbf{IEN} = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 2 & 3 & 6 & 7 \\ 3 & 4 & 7 & 8 \\ 5 & 6 & 9 & 10 \\ 6 & 7 & 10 & 11 \\ 7 & 8 & 11 & 12 \end{bmatrix}$$

- The geometry is defined by storing the physical coordinates of each node in a matrix where rows represent node number (**NodeCoordinates** array)
- The element topology is also known as the **IEN** ("element nodes") array and is a matrix connecting the nodes to the elements (connectivity matrix). Rows represent the elements and columns represent the nodes that support the element
- The physical node coordinates can be found by extracting by using the row number of the IEN array. **Example:** for element number 4

$$\mathbf{NodeCoordinates}(\mathrm{IEN\_4}) = \begin{bmatrix} x_5 & y_5 \\ x_6 & y_6 \\ x_9 & y_9 \\ x_{10} & y_{10} \end{bmatrix}$$

$$\mathrm{IEN\_4} = \begin{bmatrix} 5 & 6 & 9 & 10 \end{bmatrix}$$

Assembly procedure: boundary and loading conditions



$$\textbf{NodeCoordinates} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{12} & y_{12} \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} u_1 \\ \vdots \\ u_{12} \\ v_1 \\ \vdots \\ v_{12} \end{bmatrix} = \text{degrees of freedom}$$

- **Displacement** prescribed (**supressed**) on nodes 2, 1, 5 and 9 $\boxed{u = v = 0}$

$$\textbf{PrescribedDOF} = \begin{bmatrix} 2 & 1 & 5 & 9 & 2+12 & 1+12 & 5+12 & 9+12 \end{bmatrix}$$

- **Displacement** prescribed (**value**) on nodes 4, 8 and 12 $\boxed{v = 0.47}$

$$\mathbf{D}(16) = \mathbf{D}(20) = \mathbf{D}(24) = 0.47$$

Global system matrix

Initialise $\boxed{\mathbf{K} = \mathbf{0}}$

The dimension of $\mathbf{K}$ is n^2 $\boxed{n = \text{number of degrees of freedom}}$

$$\begin{bmatrix} \text{dof}_1 & \text{dof}_2 & \cdots & \text{dof}_n \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ K_{n1} & K_{n2} & \cdots & K_{nn} \end{bmatrix} \begin{bmatrix} \text{dof}_1 \\ \text{dof}_2 \\ \vdots \\ \text{dof}_n \end{bmatrix}$$

The **global** stiffness matrix is then formed by assembling the **individual element** stiffness matrices according to the element topology:

Place the stiffness contributions at the DOF numbers according to the IEN array

Global system matrix

After the global system is created, $\boxed{\mathbf{KD} = \mathbf{F}}$ is usually modified by separating the free (active) DOFs $\mathbf{D}_f$ and supressed (prescribed) DOFs $\mathbf{D}_s$

$$\boxed{\mathbf{K} = \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fs} \\ \mathbf{K}_{sf} & \mathbf{K}_{ss} \end{bmatrix} \begin{bmatrix} \mathbf{D}_f \\ \mathbf{D}_s \end{bmatrix} = \begin{bmatrix} \mathbf{F}_f \\ \mathbf{F}_s \end{bmatrix}}$$

$$\boxed{\begin{aligned} \mathbf{D}_f &= \mathbf{D}(\text{All dofs} - \mathbf{PrescribedDOF}) \\ \mathbf{D}_s &= \mathbf{D}(\mathbf{PrescribedDOF}) \end{aligned}}$$

The **global** stiffness matrix is then formed by assembling the **individual element** stiffness matrices according to the element topology:

Place the stiffness contributions at the DOF numbers according to the IEN array

It follows:

$$\boxed{\mathbf{K}_{ff}\mathbf{D}_f = \mathbf{F}_f - \mathbf{K}_{fs}\mathbf{D}_s = \hat{\mathbf{F}}_f}$$

$\mathbf{K}_{fs}$ is obtained by extracting the right contributions from $\mathbf{K}$

$$\boxed{\mathbf{K}_{fs} = \mathbf{K}(\text{All dofs}, \mathrm{Pr}\,\mathbf{escribedDOF}); \qquad \mathbf{F}_f = \mathbf{F}_f(\text{All dofs})}$$

Reaction forces may be calculated by evaluating $\mathbf{F} = \mathbf{KD}$ after the system is solved

Work flow of a typical structural mechanics FE code

READ INPUT (material properties, node coordinates, element topology, structure of global system, loads, boundary conditions) | **INITIALISATION** (**K** = **0**)

SOLVING

```
FOR  e = 1: number of elements
        set element stiffness matrix k = 0
        FOR g = 1:number of Gauss points
                call Gauss quadrature points
                call shape functions and their derivatives
                call Jacobian matrix and physical derivatives
                Form strain-displacement matrix B
                Form element stiffness matrix k
        END FOR

        Assemble k to global stiffness matrix K
END FOR
```

Modify **K** for boundary conditions

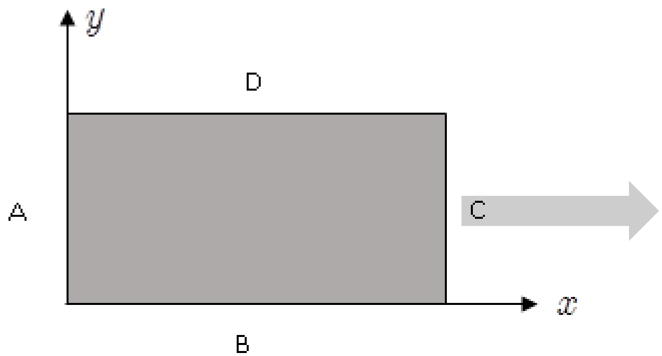
Solve global system **KD**=**R** with respect to **D**

POST-PROCESSING

Write output (e.g. displacements, reaction forces, strains, stresses and energies)

Practical example: 2D elasticity

2D rectangular solid domain $\{E = 210 \text{ GPa};\ \nu = 0.3\}$ $L_x = 1\,\text{m};$ $L_y = 1\,\text{m}$



$$\mathbf{F} = \begin{Bmatrix} F_x \\ 0 \end{Bmatrix}$$

Boundary conditions

$$u(\text{side A}) = v(\text{side B}) = 0$$

Loading conditions

$$\mathbf{F}(\text{side C})[\text{Newton}] = \begin{Bmatrix} F_x(\text{side C}) \\ F_y(\text{side C}) \end{Bmatrix} = \begin{Bmatrix} 10^6 \\ 0 \end{Bmatrix}$$

The boundary and loading conditions are homogeneous and therefore do not introduce shear coupling. The Cauchy stress tensor has only one non-null component, σ_{xx}.

Laboratory workshop 2

2D elasticity problem

- Derivation of the weak form
- Derivation of a 2D bi-linear 4-noded quadrilateral element

Development of a Python programme to:

- Compute the element stiffness matrix and nodal force vectors
- To apply boundary and loading conditions
- To mesh a 2D rectangular domain with 4-noded quadrilateral element
- To assemble the global stiffness matrix and load vector
- To solve a boundary-value problem