

The finite element method

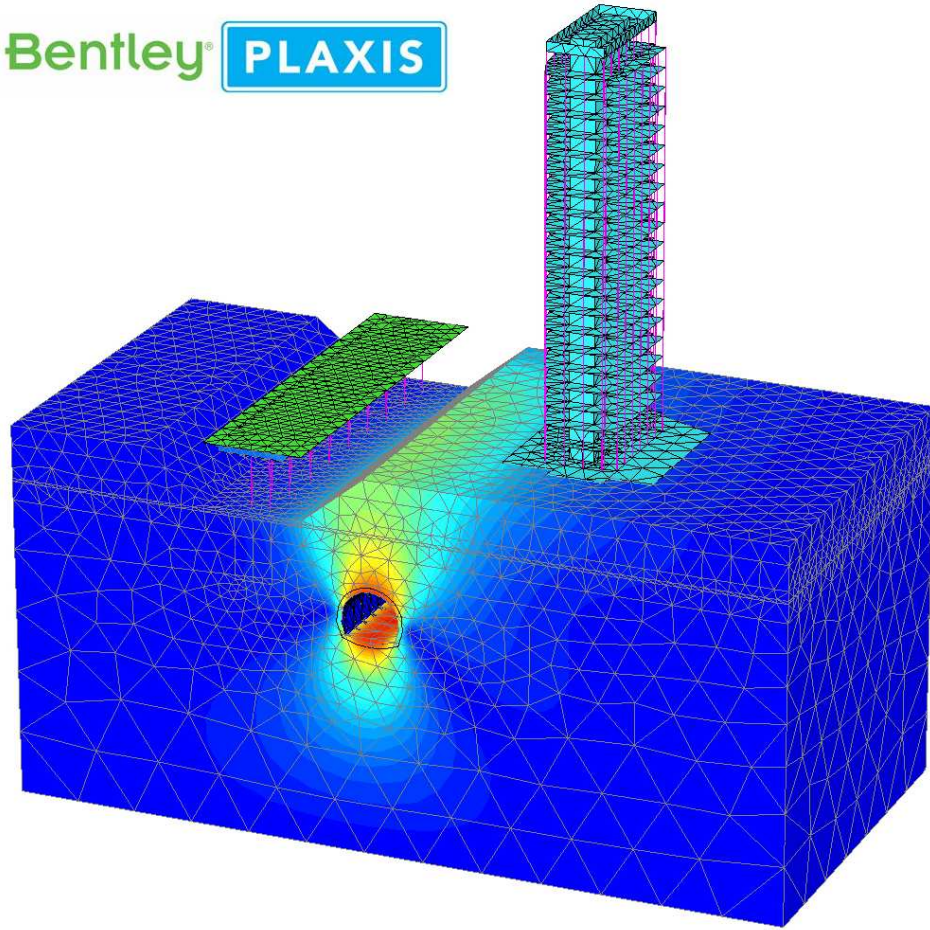
MUDE week 2.2

Frans van der Meer

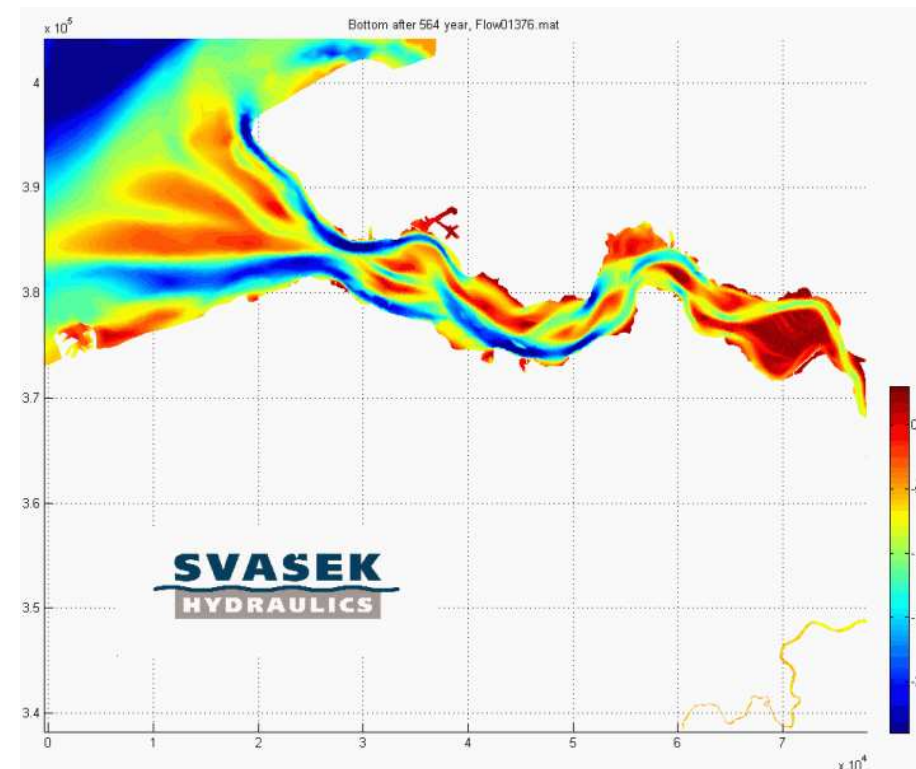
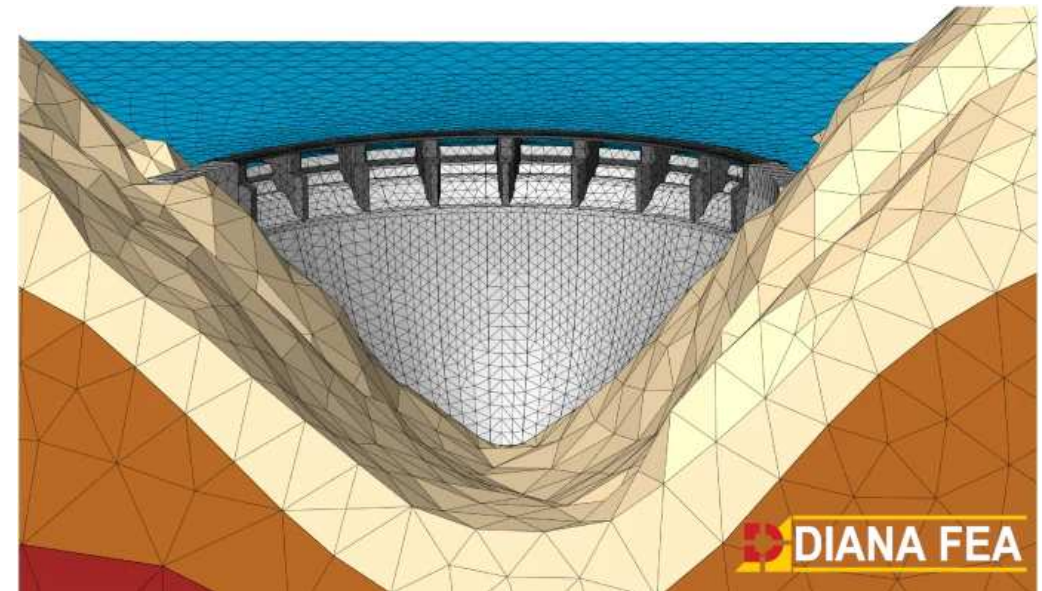
Finite elements and CEG

Three commercial codes with strong ties to this faculty

Bentley® **PLAXIS**



And several research codes



The Finite _____ Methods

Finite **difference** method: discretize the derivatives

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2}$$

Finite **volume** method: discretize the conservation

$$\frac{\partial u}{\partial t} = \nu \nabla^2 u \quad \rightsquigarrow \quad \frac{\partial}{\partial t} \int_{\Omega} u \, d\Omega = \nu \int_{\Gamma} \nabla u \cdot \mathbf{n} \, d\Gamma$$

Finite **element** method: discretize the solution

$$u(x) \approx \sum_i N_i(x) u_i \quad \rightsquigarrow \quad ?$$

Discretizing the solution

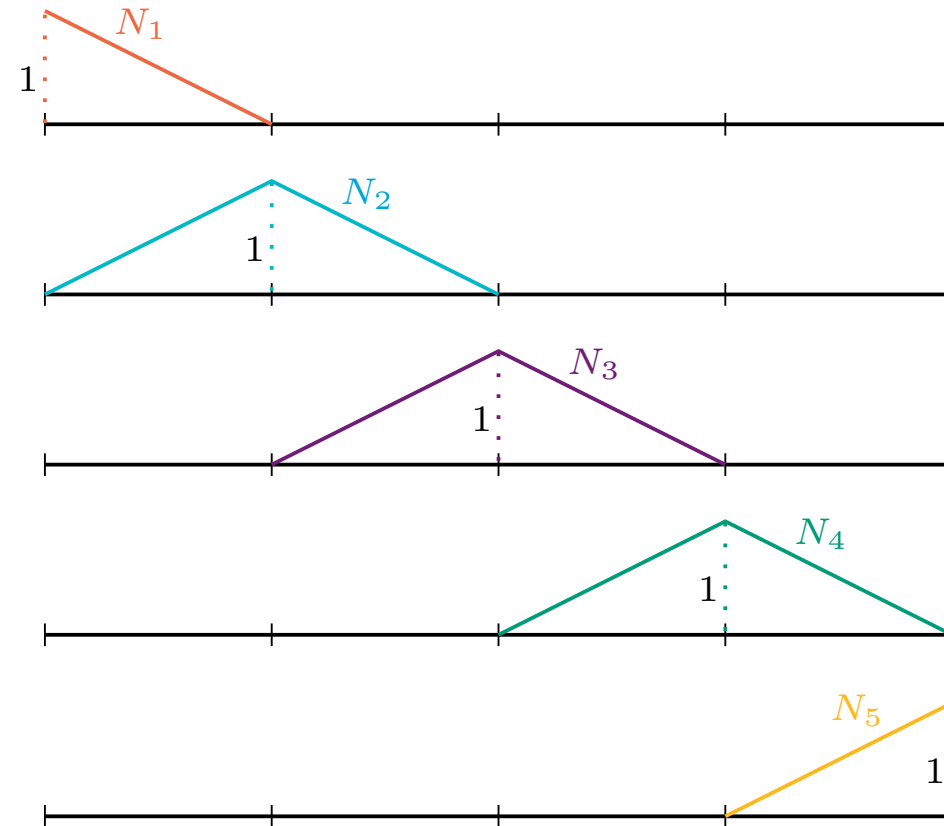
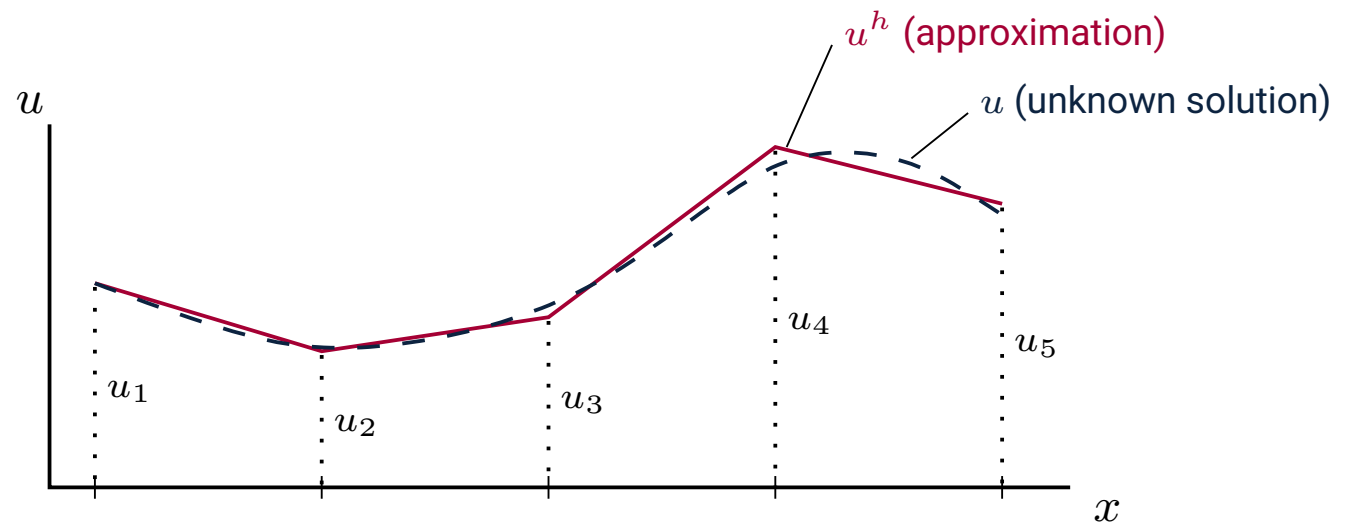
The Poisson equation in 1D

$$-\nu \frac{\partial^2 u}{\partial x^2} = f$$

Approximate u as u^h , with shape functions $N_i(x)$:

$$u^h(x) = \sum_i N_i(x) u_i = \mathbf{N} \mathbf{u}$$

How to find the *best* values u_i ?



From strong form to weak form equation

Weighted residual formulation:

$$-\nu \frac{\partial^2 u}{\partial x^2} = f \quad \Leftrightarrow \quad - \int_{\Omega} w \nu \frac{\partial^2 u}{\partial x^2} dx = \int_{\Omega} w f dx \quad \forall \quad w$$

Integration by parts:

$$\int_{\Omega} w \nu \frac{\partial^2 u}{\partial x^2} dx = - \int_{\Omega} \frac{\partial w}{\partial x} \nu \frac{\partial u}{\partial x} dx + \left[w \nu \frac{\partial u}{\partial x} \right]_0^L$$

Substitution of boundary conditions:

$$\int_{\Omega} \frac{\partial w}{\partial x} \nu \frac{\partial u}{\partial x} dx = \int_{\Omega} w f dx + w(L)h(L) + w(0)h(0) \quad \forall \quad w$$

From weak form to discretized form

Weak form equation

$$\int_{\Omega} \frac{\partial w}{\partial x} \nu \frac{\partial u}{\partial x} dx = \int_{\Omega} w f dx + \int_{\Gamma} w h d\Gamma \quad \forall w$$

Introduce discretization:

$$u \leftarrow u^h = \mathbf{N}\mathbf{u}, \quad w \leftarrow w^h = \mathbf{N}\mathbf{w} \quad (\text{Bubnov-Galerkin})$$

$$\frac{\partial u}{\partial x} \leftarrow \frac{\partial u^h}{\partial x} = \mathbf{B}\mathbf{u}, \quad \frac{\partial w}{\partial x} \leftarrow \frac{\partial w^h}{\partial x} = \mathbf{B}\mathbf{w}$$

Substitution gives:

$$\int_{\Omega} \mathbf{B}\mathbf{w} \nu \mathbf{B}\mathbf{u} dx = \int_{\Omega} \mathbf{N}\mathbf{w} f dx + \int_{\Gamma} \mathbf{N}\mathbf{w} h d\Gamma \quad \forall \mathbf{w} \quad \Rightarrow \quad \int_{\Omega} \mathbf{B}^T \nu \mathbf{B} dx \mathbf{u} = \int_{\Omega} \mathbf{N}^T f dx + \int_{\Gamma} \mathbf{N}^T h d\Gamma$$

The resulting system of equations

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad \text{with} \quad \mathbf{K} = \int_{\Omega} \mathbf{B}^T \nu \mathbf{B} \, dx \quad \text{and} \quad \mathbf{f} = \int_{\Omega} \mathbf{N}^T f \, dx + \int_{\Gamma} \mathbf{N}^T h \, d\Gamma$$

expanded as:
$$\mathbf{K} = \int_{\Omega} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} \nu \begin{bmatrix} B_1 & B_2 & \cdots & B_n \end{bmatrix} \, dx$$

with:
$$N_i = \begin{cases} 0, & x \leq x_{i-1} \\ \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \leq x < x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x_i \leq x < x_{i+1} \\ 0, & x > x_{i+1} \end{cases}$$

and:
$$B_i = \frac{\partial N_i}{\partial x} = \begin{cases} 0, & x \leq x_{i-1} \\ \frac{1}{x_i - x_{i-1}}, & x_{i-1} \leq x < x_i \\ \frac{-1}{x_{i+1} - x_i}, & x_i \leq x < x_{i+1} \\ 0, & x > x_{i+1} \end{cases}$$

After integration:

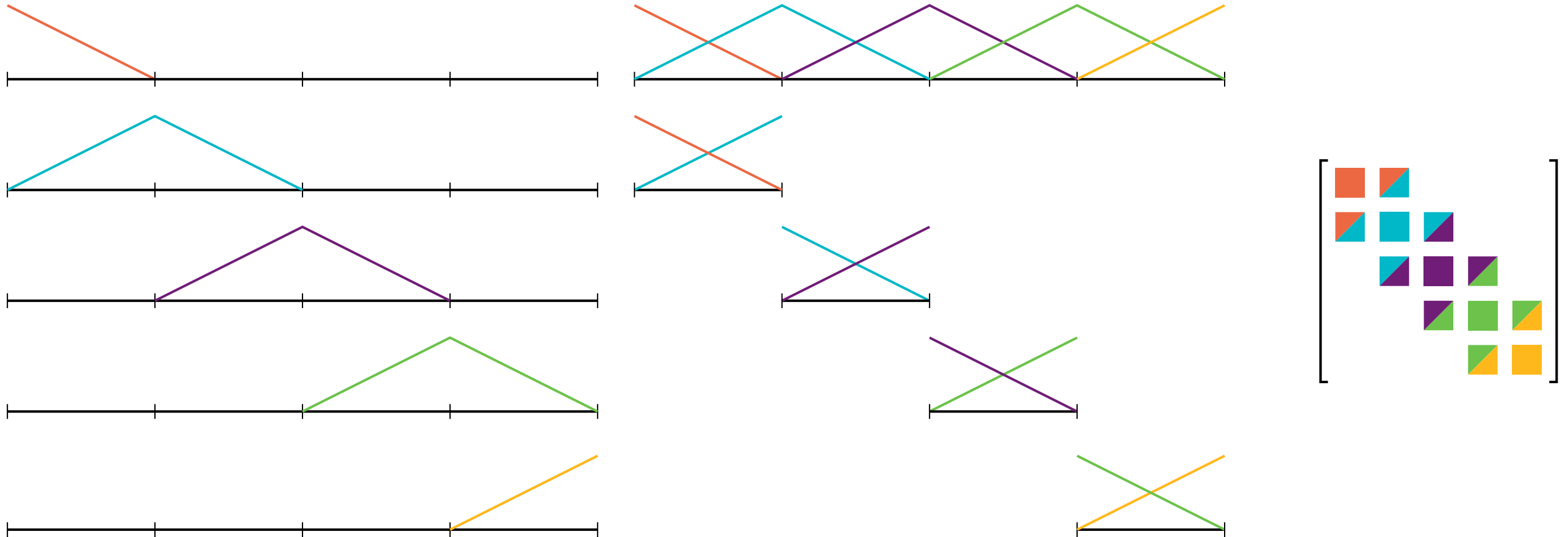
$$\frac{\nu}{\Delta x} \begin{bmatrix} 1 & -1 & 0 & & 0 & 0 & 0 \\ -1 & 2 & -1 & & 0 & 0 & 0 \\ 0 & -1 & 2 & \ddots & 0 & 0 & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & 0 & 0 & \ddots & 2 & -1 & 0 \\ 0 & 0 & 0 & & -1 & 2 & -1 \\ 0 & 0 & 0 & & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-2} \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} \frac{1}{2}q\Delta x + h(0) \\ q\Delta x \\ q\Delta x \\ \vdots \\ q\Delta x \\ q\Delta x \\ \frac{1}{2}q\Delta x + h(L) \end{bmatrix}$$

(with uniform mesh $x_{i+1} - x_i = \Delta x$
and constant source $f(x) = q$)

Dirichlet BCs are applied before solving for $\mathbf{u}$

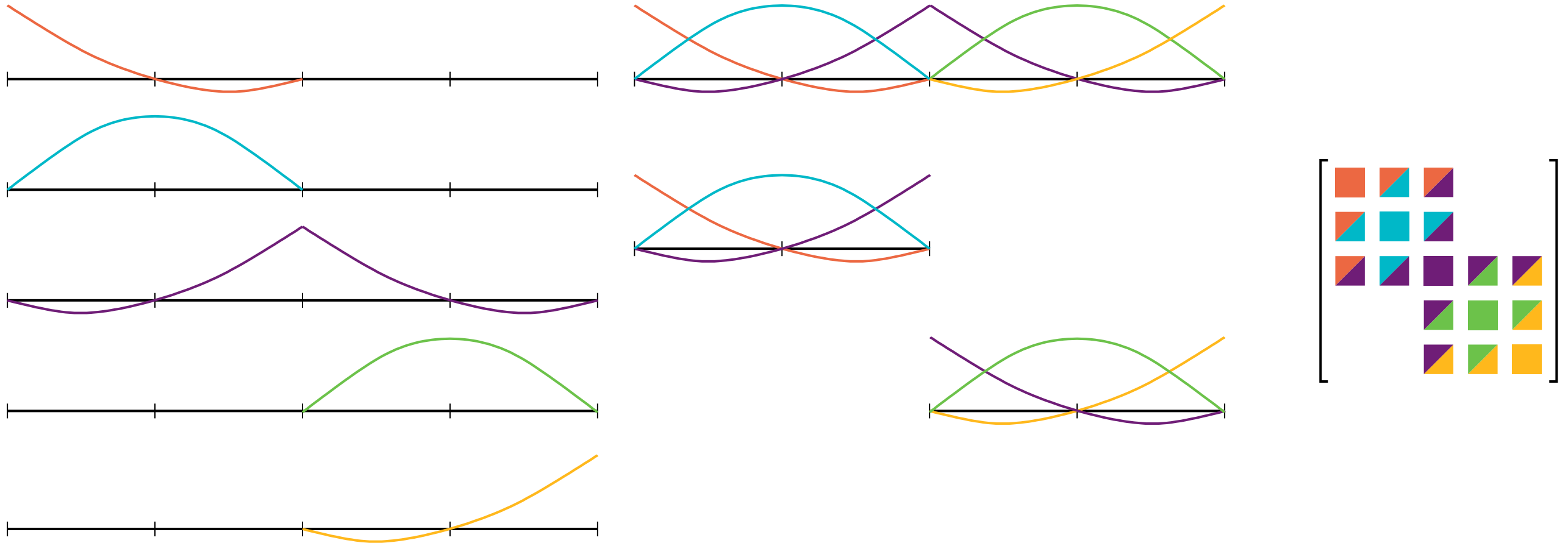
Now, what about these 'Elements'?

The discretized Poisson equation: $\mathbf{K}\mathbf{a} = \mathbf{f}$ with $\mathbf{K} = \int_{\Omega} \mathbf{B}^T \nu \mathbf{B} dx$ and $\mathbf{f} = \int_{\Omega} \mathbf{N}^T f dx + \int_{\Gamma} \mathbf{N}^T h d\Gamma$



Higher order elements can also be formulated, 3-nodes per element for quadratic

The discretized Poisson equation: $\mathbf{K}\mathbf{a} = \mathbf{f}$ with $\mathbf{K} = \int_{\Omega} \mathbf{B}^T \nu \mathbf{B} dx$ and $\mathbf{f} = \int_{\Omega} \mathbf{N}^T f dx + \int_{\Gamma} \mathbf{N}^T h d\Gamma$



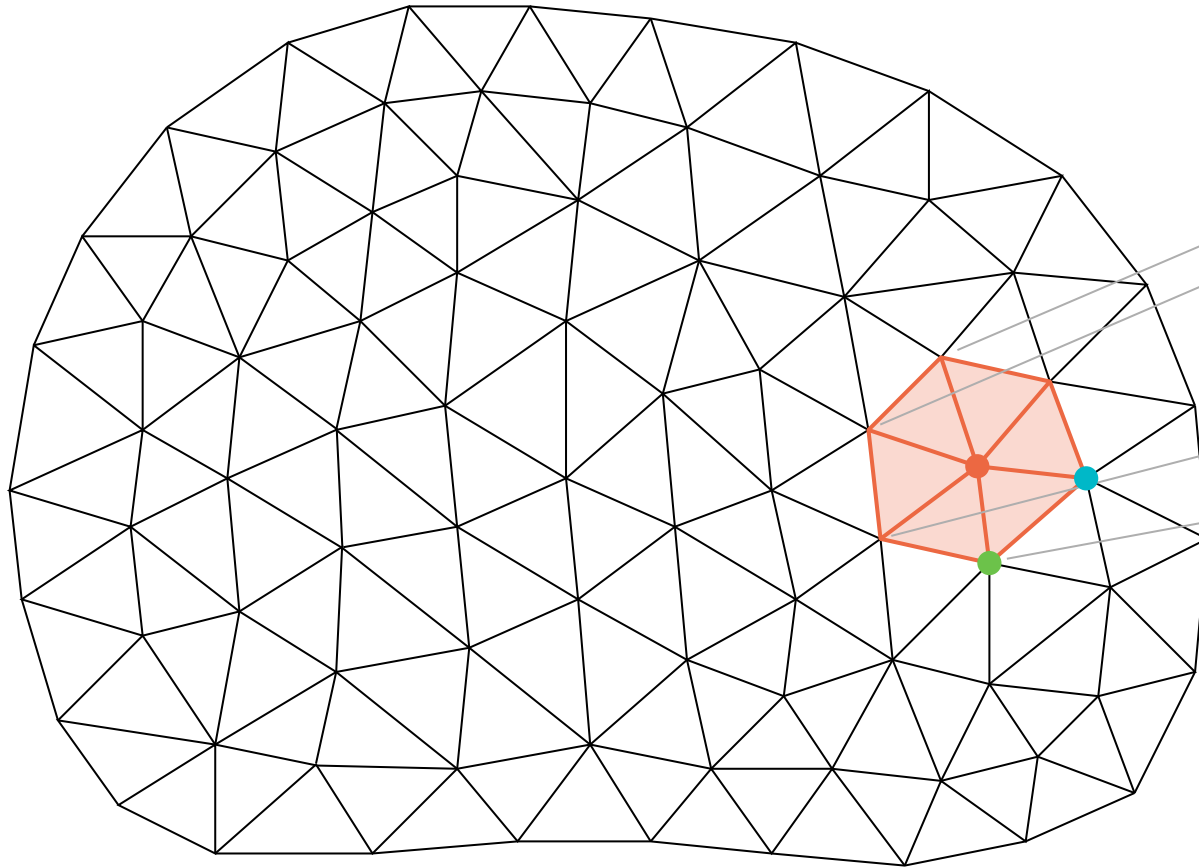
Shape function properties

Shape functions requirements for good performance:

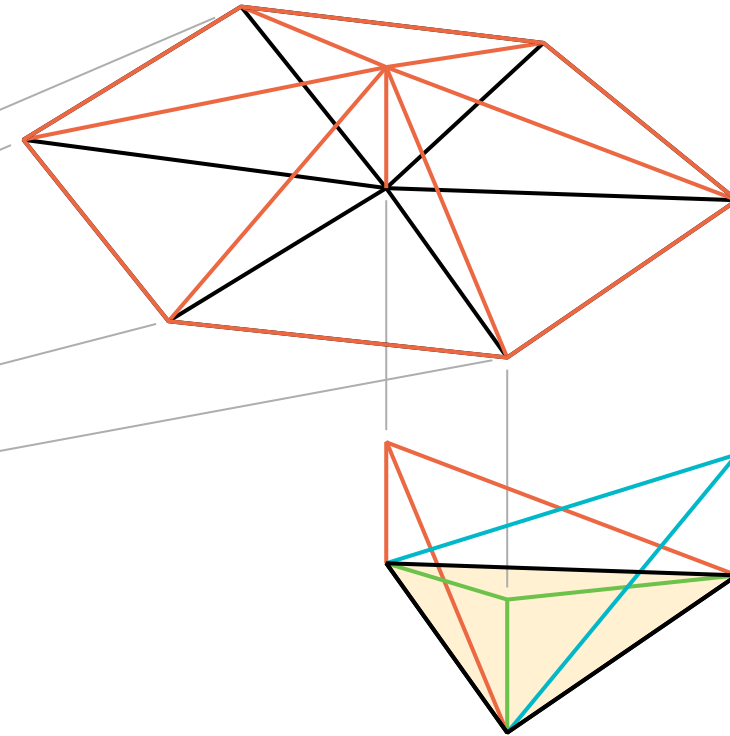
- Partition of unity: $\sum_i N_i(x) = 1$
 $\Rightarrow$ represent constant solutions exactly
- Kronecker delta property: $N_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$
 $\Rightarrow$ interpret degrees of freedom as nodal values
 $\Rightarrow$ apply boundary conditions directly

Discretizing a 2D solution with triangulation of the domain

$$u(x, y) \approx \sum_i N_i(x, y) u_i$$



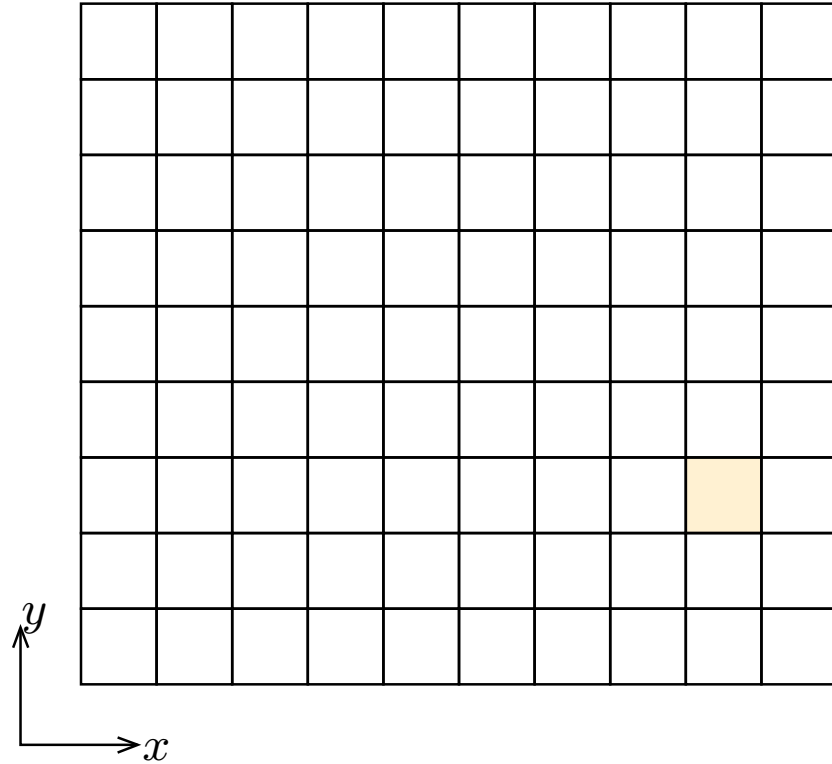
The nodal shape function spans multiple elements



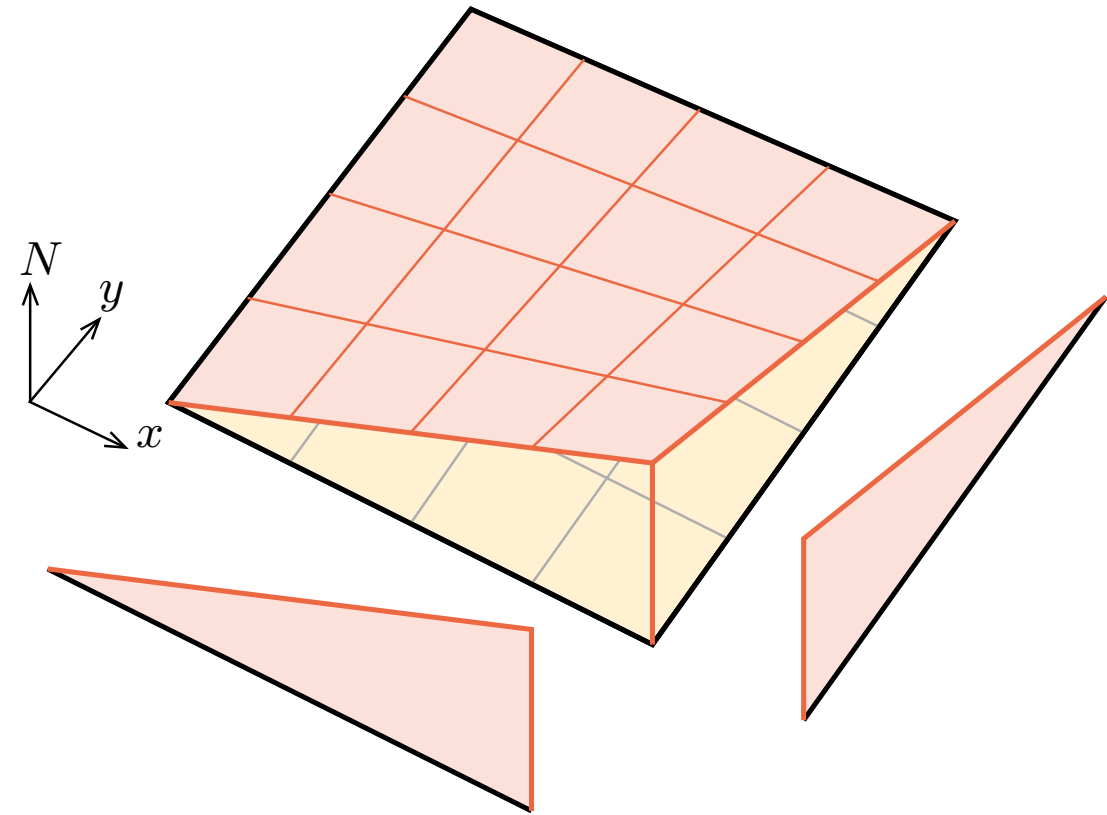
Every element has 3 shape functions:

$$N_i = a_i + b_i x + c_i y$$

2D shape functions on a quadrilateral elements

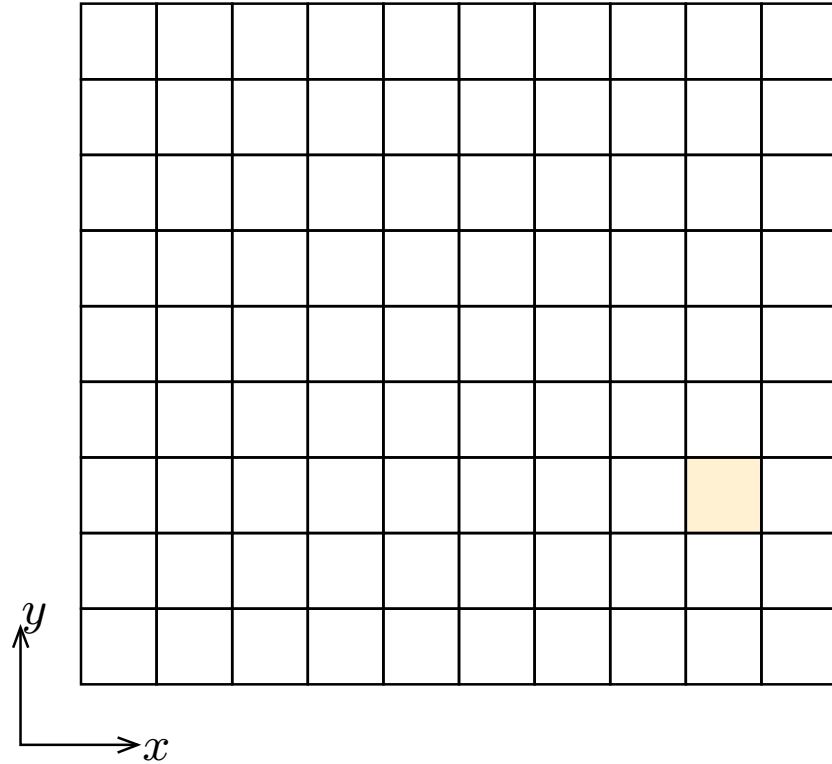


Quad-4 element

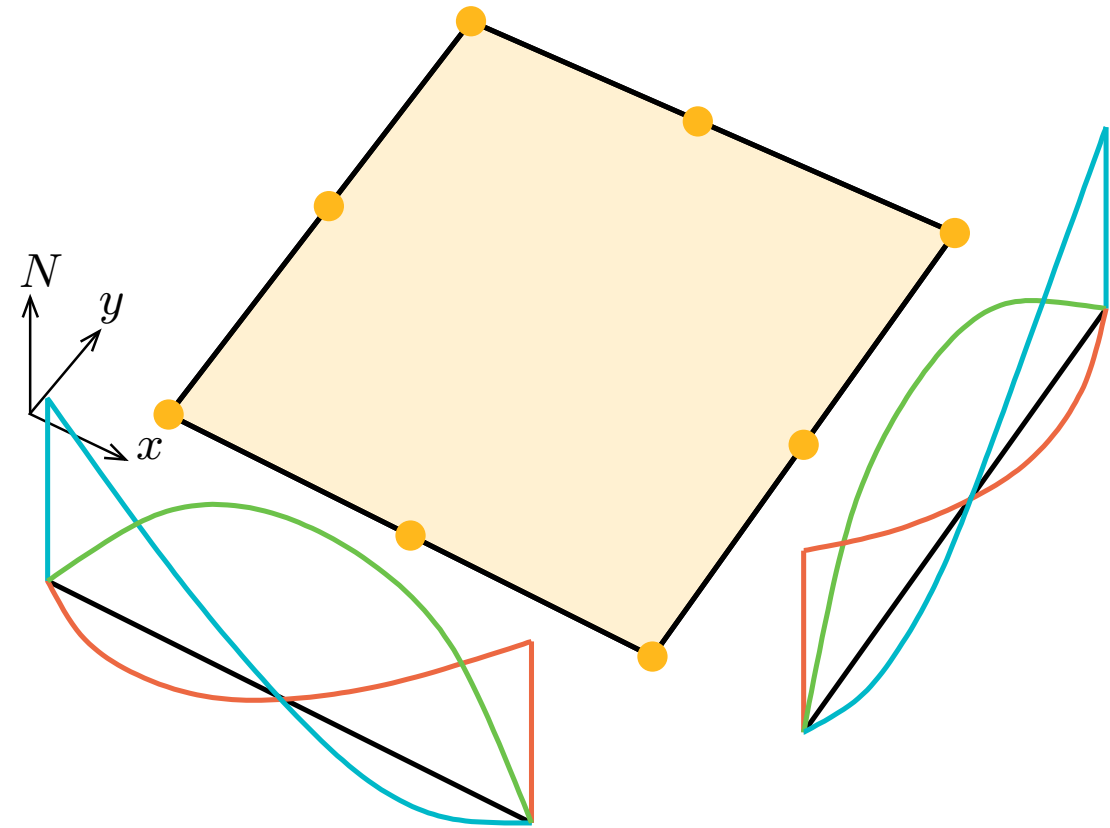


$$N_i = a_i + b_i x + c_i y + d_i xy$$

2D shape functions on a quadrilateral elements



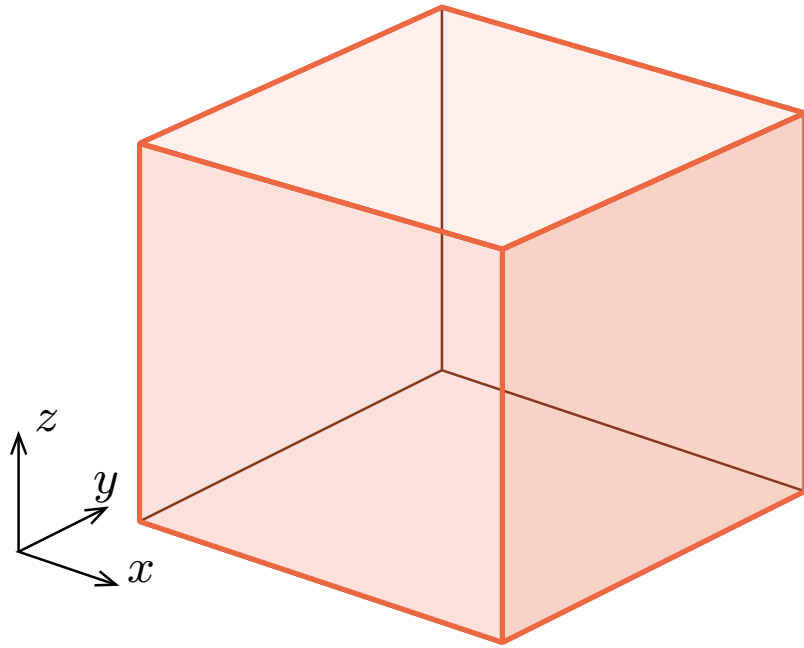
Quad-8 element



$$N_i = a_i + b_i x + c_i y + d_i x^2 + e_i xy + f_i y^2 + g_i x^2 y + h_i xy^2$$

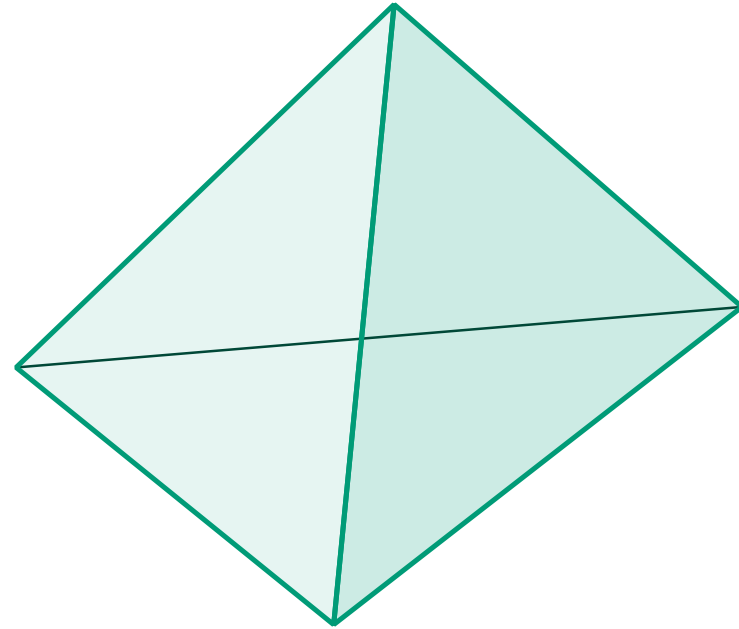
3D elements

Hexahedral element



$$N_i = a_i + b_i x + c_i y + d_i z + e_i xy + f_i xz + g_i yz + h_i xyz$$

Tetrahedral element



$$N_i = a_i + b_i x + c_i y + d_i z$$

Problem definition in 2D

The Poisson equation:

$$-\nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f \quad \text{on } \Omega$$

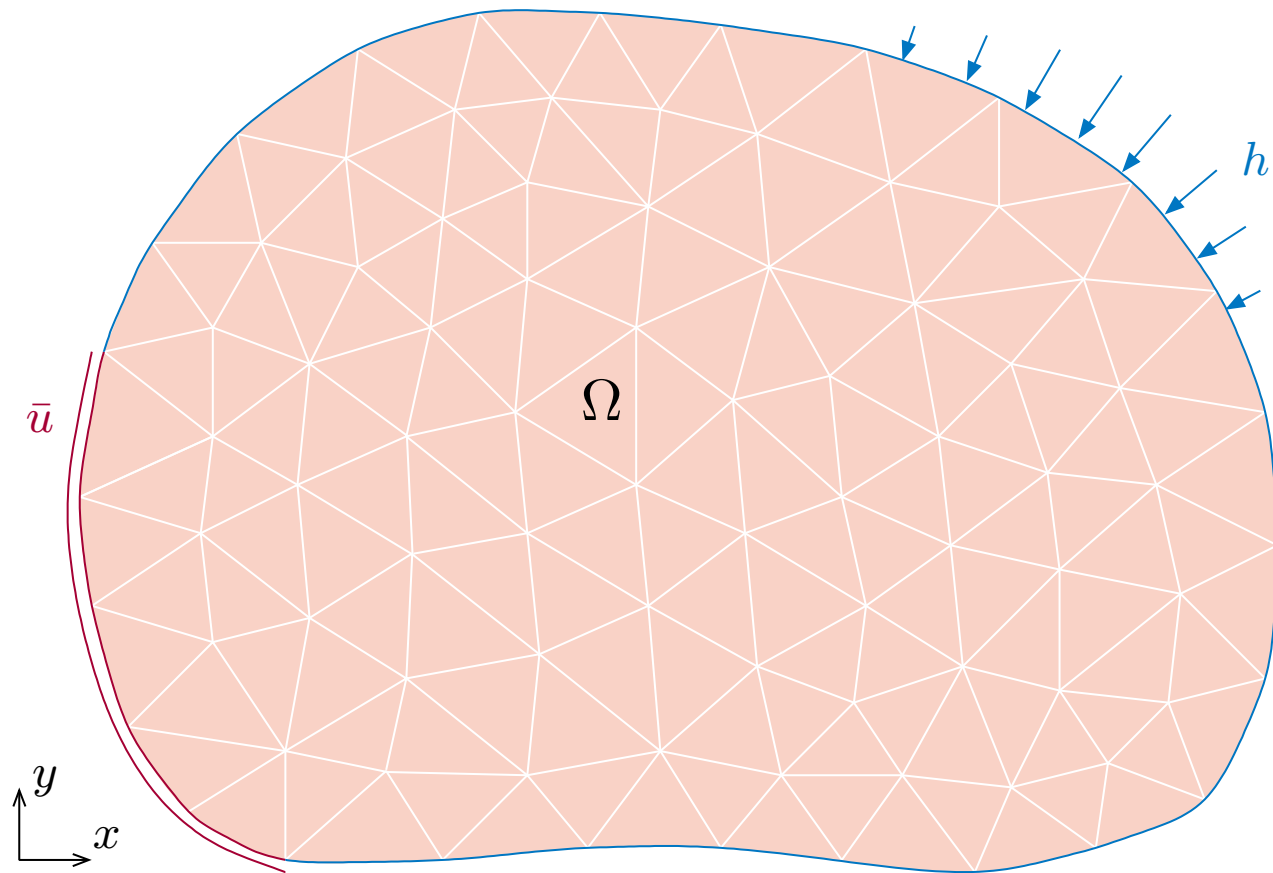
With boundary conditions

$$\begin{aligned} u &= \bar{u}(x, y) \quad \text{on } \Gamma_D \\ \nu \nabla u \cdot \mathbf{n} &= h(x, y) \quad \text{on } \Gamma_N \end{aligned}$$

Aim: discretize into a system of equations

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

Where $\mathbf{u}$ contains approximate values for $u(x, y)$ at the nodes of a finite element mesh



Discretizing the solution in 2D

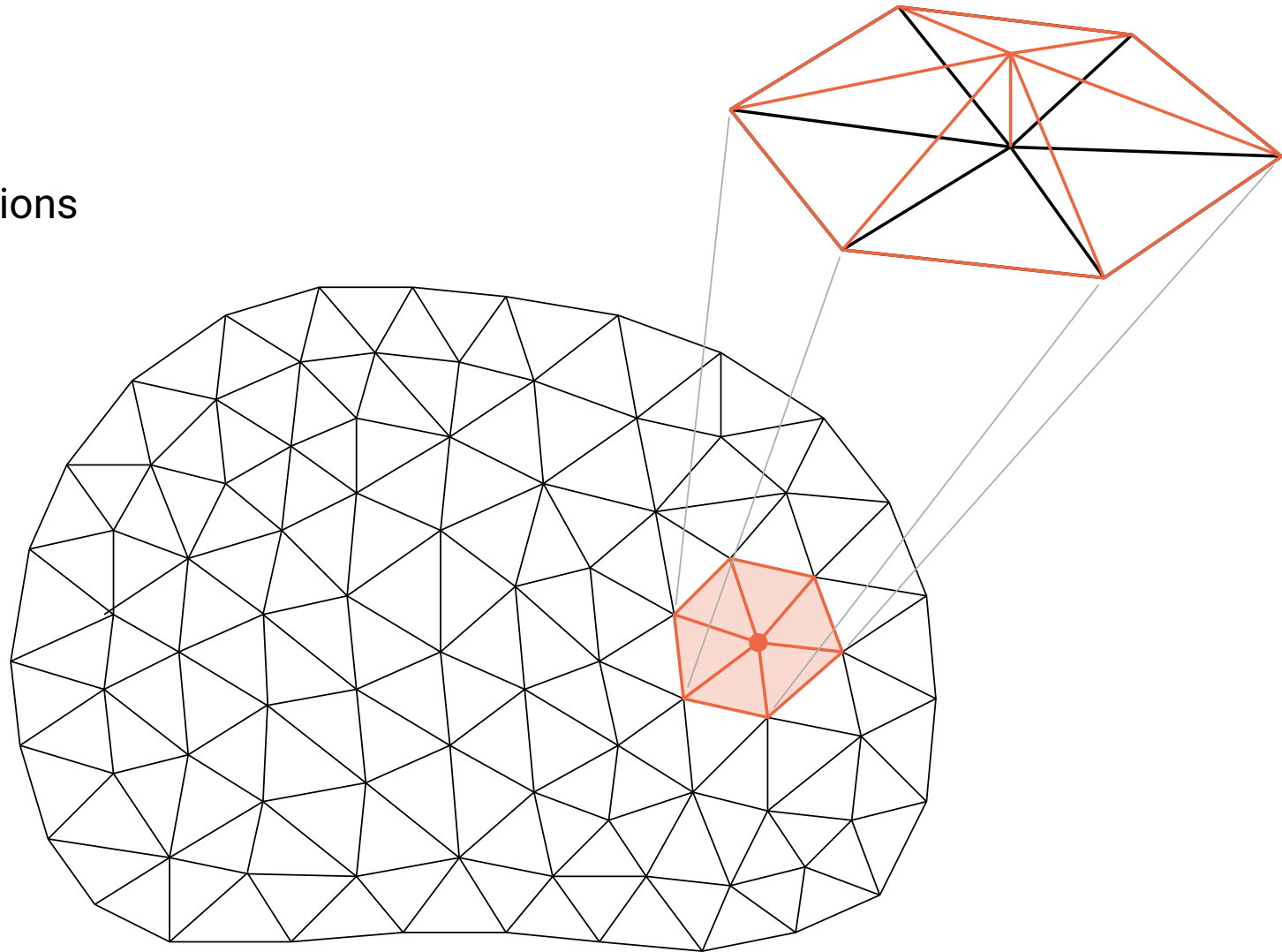
The Poisson equation:

$$-\nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f$$

Approximate u as u^h with 2D shape functions

$$u^h(x, y) = \sum_i N_i(x, y) u_i = \mathbf{N} \mathbf{u}$$

- $\mathbf{u}$ contains nodal values
- $\mathbf{N}$ defines the interpolation
→ Find $\mathbf{u}$ such that $u^h \approx u$



Weak form equation in 2D

Weighted residual formulation:

$$-\nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f \quad \Leftrightarrow \quad - \int_{\Omega} w \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) d\Omega = \int_{\Omega} w f d\Omega \quad \forall \quad w$$

Integration by parts (with divergence theorem):

$$\int_{\Omega} w \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) d\Omega = - \int_{\Omega} \nu \nabla w \cdot \nabla u d\Omega + \int_{\Gamma} w \nu \nabla u \cdot \mathbf{n} d\Gamma \quad \forall \quad w$$

Substitution:

$$\int_{\Omega} \nu \nabla w \cdot \nabla u d\Omega - \int_{\Gamma} w \nu \nabla u \cdot \mathbf{n} d\Gamma = \int_{\Omega} w f d\Omega \quad \forall \quad w$$

With boundary conditions ($w = 0$ on Γ_D and $\nu \nabla u \cdot \mathbf{n} = h$ on Γ_N):

$$\int_{\Omega} \nu \nabla w \cdot \nabla u d\Omega = \int_{\Omega} w f d\Omega + \int_{\Gamma_N} w h \quad \forall \quad w$$

Discretized form

Weak form equation

$$\int_{\Omega} \nu \nabla w \cdot \nabla u \, d\Omega = \int_{\Omega} w f \, d\Omega + \int_{\Gamma_N} w h \, d\Gamma \quad \forall \quad w$$

Introduce discretization:

$$u \leftarrow u^h = \mathbf{N}\mathbf{u}, \quad w \leftarrow w^h = \mathbf{N}\mathbf{w}, \quad \mathbf{N} = [N_1 \quad N_2 \quad \cdots \quad N_n]$$

$$\nabla u \leftarrow \nabla u^h = \mathbf{B}\mathbf{u}, \quad \nabla w \leftarrow \nabla w^h = \mathbf{B}\mathbf{w}, \quad \mathbf{B} = \nabla \mathbf{N} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \cdots & \frac{\partial N_n}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \cdots & \frac{\partial N_n}{\partial y} \end{bmatrix}$$

Substitution gives:

$$\int_{\Omega} \mathbf{B}\mathbf{w} \nu \mathbf{B}\mathbf{u} \, d\Omega = \int_{\Omega} \mathbf{N}\mathbf{w} f \, d\Omega + \int_{\Gamma_N} \mathbf{N}\mathbf{w} h \, d\Gamma \quad \forall \quad \mathbf{w} \quad \Rightarrow \quad \int_{\Omega} \mathbf{B}^T \nu \mathbf{B} \, d\Omega \mathbf{u} = \int_{\Omega} \mathbf{N}^T f \, d\Omega + \int_{\Gamma_N} \mathbf{N}^T h \, d\Gamma$$

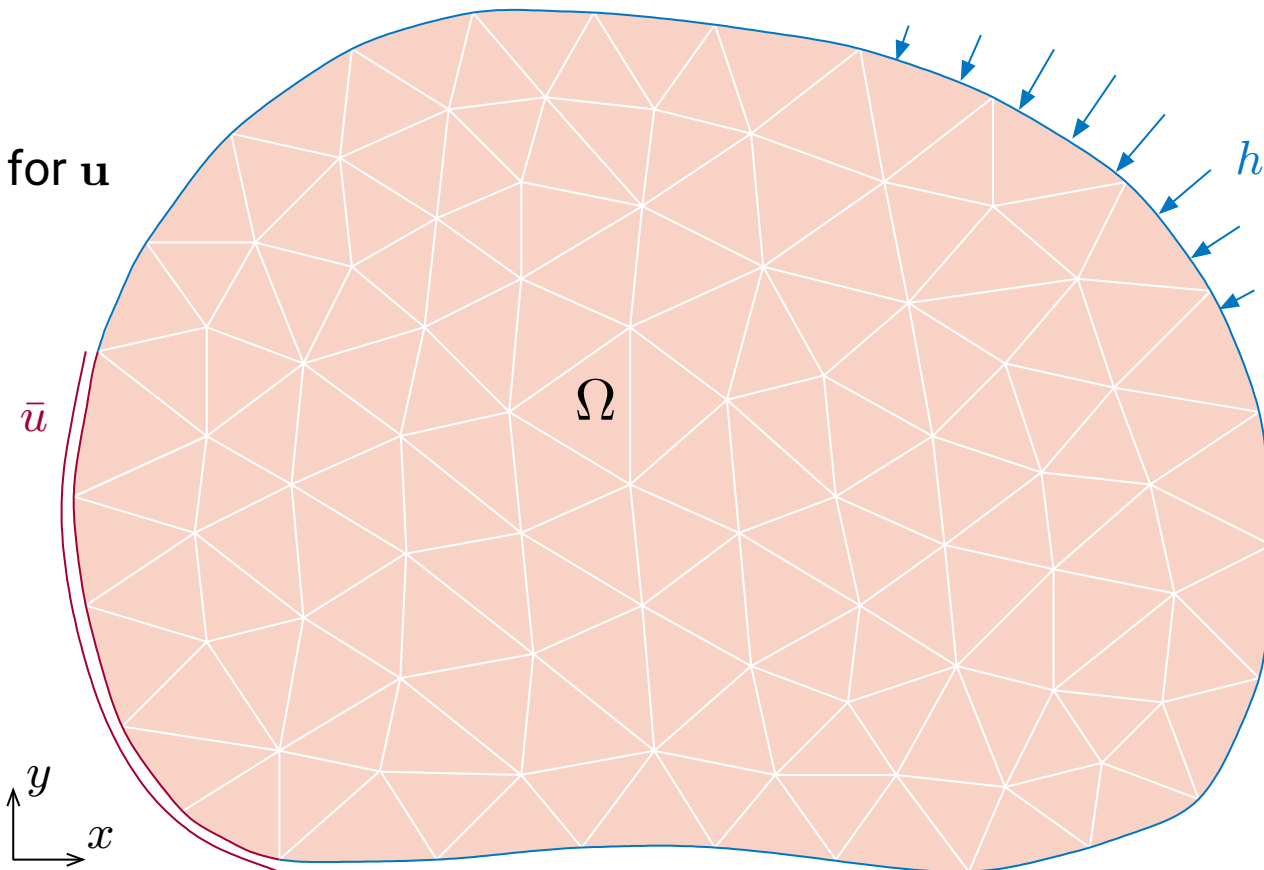
Finding the approximate solution

Discretized form:

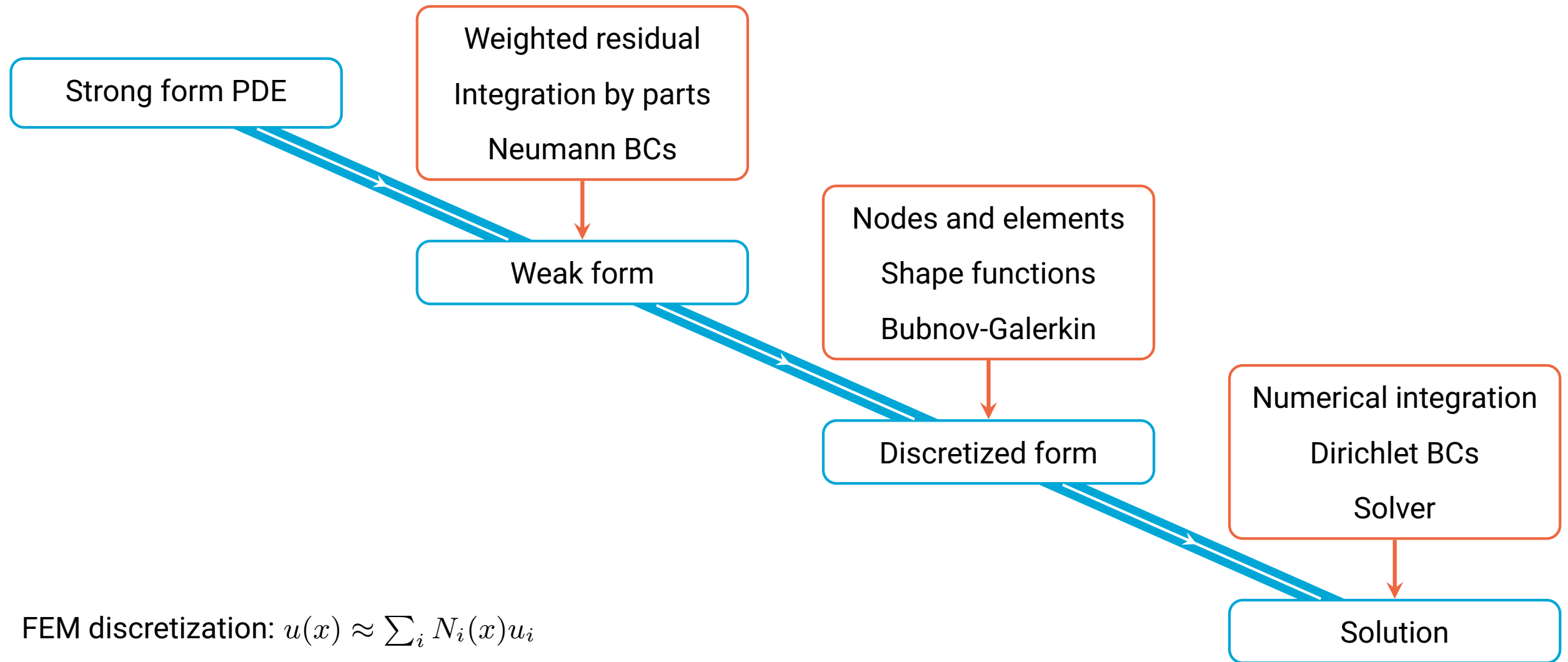
$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad \text{with} \quad \mathbf{K} = \int_{\Omega} \mathbf{B}^T \nu \mathbf{B} \, d\Omega \quad \text{and} \quad \mathbf{f} = \int_{\Omega} \mathbf{N}^T f \, d\Omega + \int_{\Gamma_N} \mathbf{N}^T h \, d\Gamma$$

Solving the FE equations finally requires:

- Numerical integration of $\mathbf{K}$ and $\mathbf{f}$
- Constraining $u_i = \bar{u}$ for nodes on Γ_D
- Solving the constrained system of equations for $\mathbf{u}$



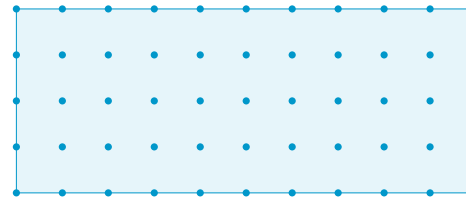
Take home message



One more Finite _____ Method

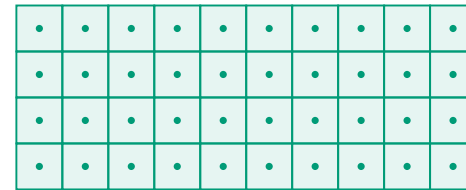
You have seen Finite Difference Method in week 2.1

- Easiest to implement and understand
- Super efficient for some problems
- Simple geometries and structured grids



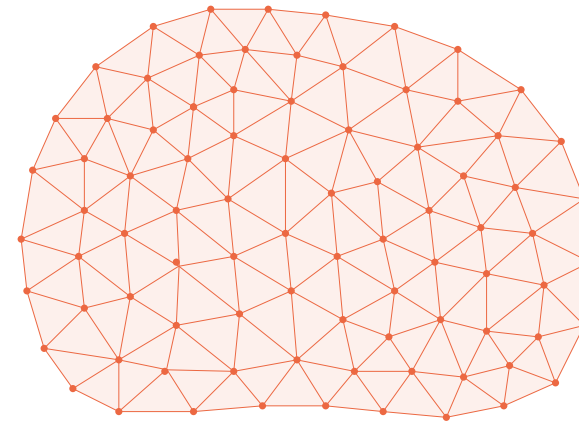
And may have heard of the Finite Volume Method

- Mostly for problems involving flow
- Local conservation is guaranteed



How the finite element method relates to these:

- Originally but not exclusively for solid mechanics
- Straightforward handling of boundary conditions
- Native support for unstructured meshes
- Higher order accuracy with higher order shape functions
- Many other cool possibilities from the choice of shape function



Program for this week

Before Wednesday: Self study

- Book: Up to section 2.4 (FE implementation)

Wednesday: Supported bar problem

- Derive weak form
- Extend python implementation from book

Before Friday: Self study

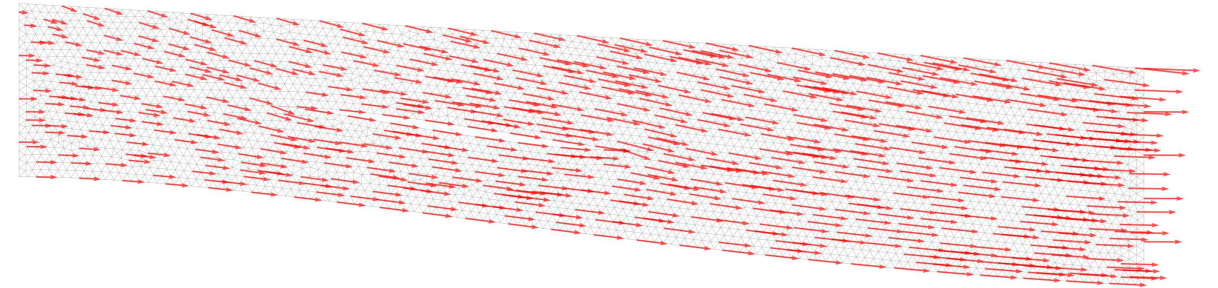
- Book: Up to section 2.7 (Poisson 2D)
- Book: Programming with large matrices
- Programming assignment: sparse vs dense

Friday: Advection/Diffusion/Reaction equation

- 2D on non-trivial geometry
- Transient problem with FEM

Happy FEM week!

Groundwater Velocity Field



Contaminant concentration for time step 1840

