Algorithms
A1. Introduction to finite element method

Aleš Janka

office Math 0.107
ales.janka@unifr.ch
http://perso.unifr.ch/ales.janka/mechanics

September 29, 2010, Université de Fribourg

Basic ideas behind the Finite Element Method (FEM)

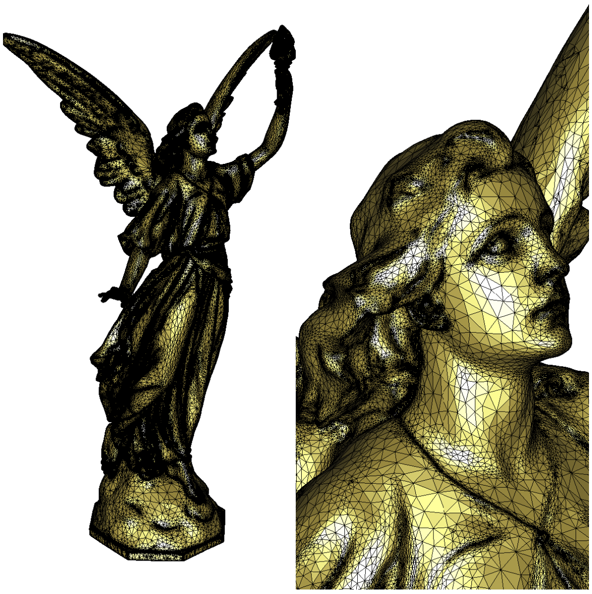
Main task: find an easy way how to represent general functions $u : \Omega \to \mathbb{R}$ (scalar fields) on a computer. The method has to:

- work well even on complicated computational domains $\Omega \subset \mathbb{R}^d$,
- represent well a large class of functions u (continuous, discontinuous, singular at a point),
- be simple and easy to use (clear data-representation, fast),
- provide possibility of adaptation (improving precision of the result).

Main problem: The space $C^0(\Omega)$ of all continuous functions on Ω is infinite-dimensional. How do we represent such functions on a computer with finite RAM memory in finite time?

1. Basic ideas: complex computational domains Ω

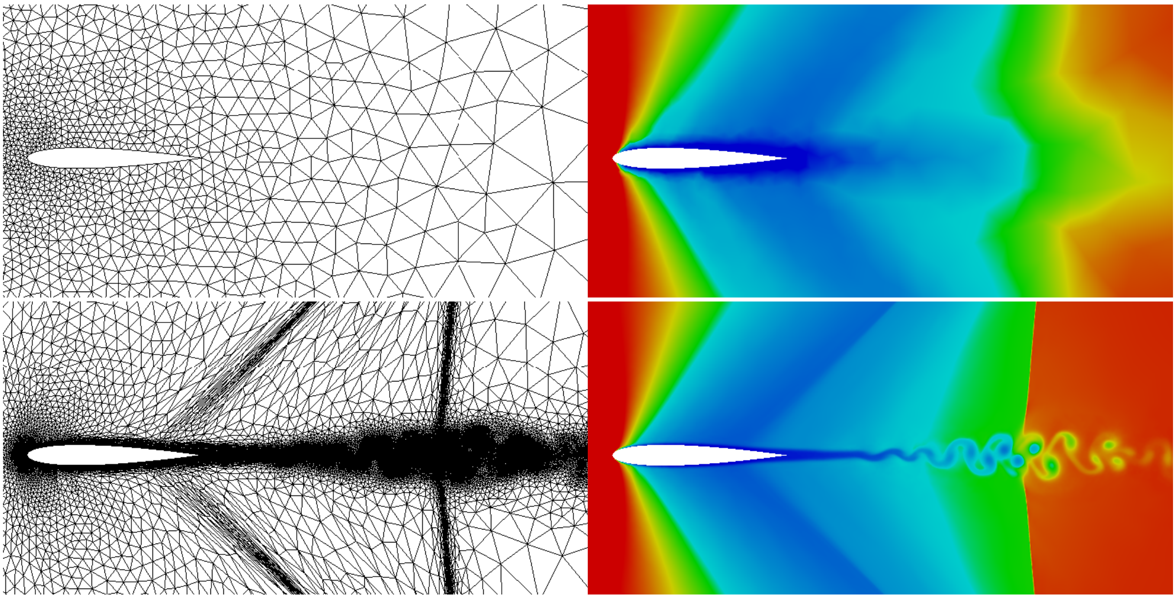
Approximate Ω by a union of non-overlapping simple (finite) elements



- Works for general surfaces and volumes
- Simple data-representation: list of vertices and elements (2 matrices)
- Can be refined when needed, to capture geometry or physics

1. Basic ideas: complex computational domains Ω

Approximate Ω by a union of non-overlapping simple (finite) elements



- Works for general surfaces and volumes
- Simple data-representation: list of vertices and elements (2 matrices)
- Can be refined when needed, to capture geometry or physics

2. Basic ideas: clear data-representation

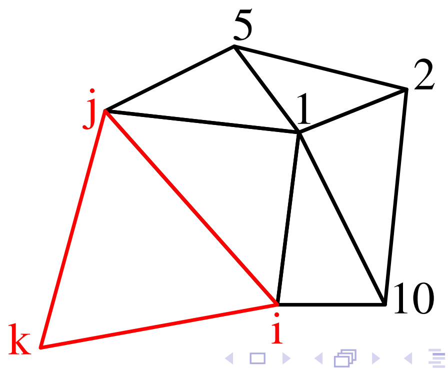
List of vertex coordinates and list of vertices in each element

Nodal coordinates:
coordinates of mesh vertices

$$XYZ = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_3 \\ \vdots & \vdots & \vdots \\ x_i & y_i & z_i \\ \vdots & \vdots & \vdots \\ x_j & y_j & z_j \\ \vdots & \vdots & \vdots \\ x_k & y_k & z_k \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{pmatrix}$$

Element connectivity:
node indices in each element

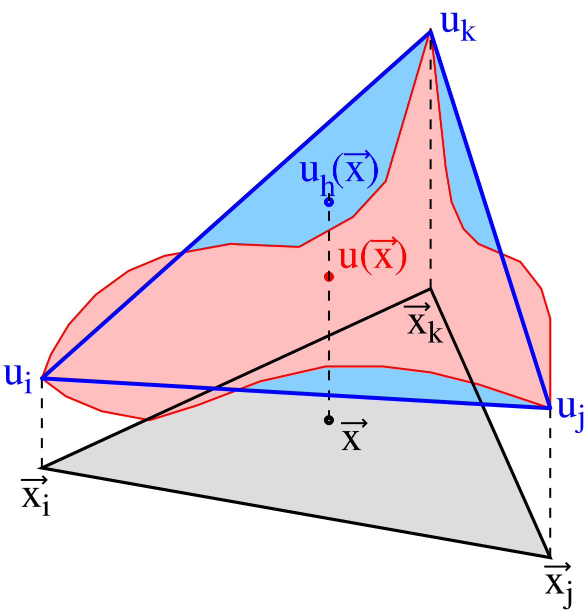
$$Elm = \begin{pmatrix} 1 & 10 & 2 \\ \vdots & \vdots & \vdots \\ i & j & k \\ \vdots & \vdots & \vdots \\ 5 & 1 & 2 \end{pmatrix}$$



3. Basic ideas: representing a scalar function $u : \Omega \longrightarrow \mathbb{R}$

Interpolation of $u(\mathbf{x})$ on one element: $u(\mathbf{x}) \approx u_h(\mathbf{x})$

Lagrange element $P1$



Type of $u_h(\mathbf{x})$: polynomial of low degree
The simplest = linear $u_h(\mathbf{x})$:

$$u_h(\mathbf{x}) = a + b \cdot x + c \cdot y$$

Interpolation conditions:

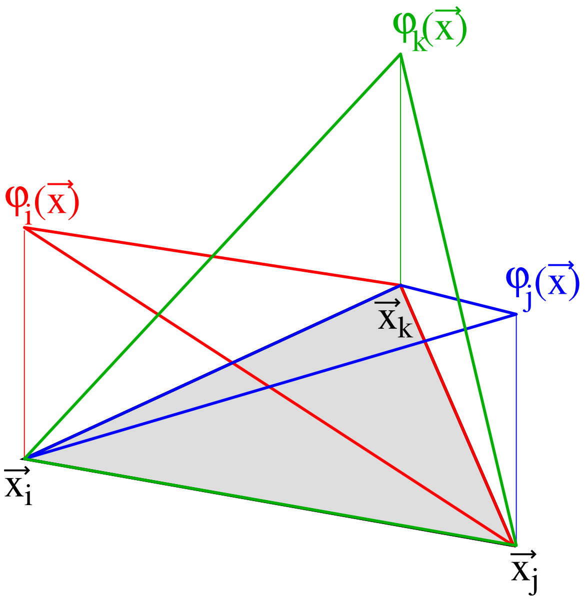
$$\begin{aligned} u_h(\mathbf{x}_i) &= u(\mathbf{x}_i) = u_i \\ u_h(\mathbf{x}_j) &= u(\mathbf{x}_j) = u_j \\ u_h(\mathbf{x}_k) &= u(\mathbf{x}_k) = u_k \end{aligned}$$

Error of interpolation in L_2 norm:

$$\|u - u_h\|_{2,\Omega} = \left(\int_\Omega [u(\mathbf{x}) - u_h(\mathbf{x})]^2 \, dx \right)^{\frac{1}{2}}$$

Rewrite $u_h(\mathbf{x})$ in a nodal basis $\langle \varphi_i(\mathbf{x}), \varphi_j(\mathbf{x}), \varphi_k(\mathbf{x}) \rangle$:

$$u_h(\mathbf{x}) = u_i\, \varphi_i(\mathbf{x}) + u_j\, \varphi_j(\mathbf{x}) + u_k\, \varphi_k(\mathbf{x})$$



with linear $\varphi_\ell(\mathbf{x})$ such that

$$\varphi_\ell(\mathbf{x}_m) = \begin{cases} 1, & \text{if } \ell = m \\ 0, & \text{otherwise} \end{cases}$$

4. Calculation of nodal basis functions on each element

Linear basis functions: $\varphi_\ell(\mathbf{x}) = a_\ell + b_\ell \cdot x + c_\ell \cdot y$, for $\ell = i, j, k$.

How to calculate the 3×3 constants a_ℓ, b_ℓ, c_ℓ:

$$\begin{aligned} \varphi_\ell(\mathbf{x}_i) &= a_\ell + b_\ell \cdot x_i + c_\ell \cdot y_i = \delta_\ell^i \\ \varphi_\ell(\mathbf{x}_j) &= a_\ell + b_\ell \cdot x_j + c_\ell \cdot y_j = \delta_\ell^j \\ \varphi_\ell(\mathbf{x}_k) &= a_\ell + b_\ell \cdot x_k + c_\ell \cdot y_k = \delta_\ell^k \end{aligned}$$

Equations for a_ℓ, b_ℓ, c_ℓ in matrix form:

$$\begin{pmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{pmatrix} \cdot \begin{pmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \\ c_i & c_j & c_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \\ c_i & c_j & c_k \end{pmatrix} = \begin{pmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

5. Spatial derivatives of $u_h(\mathbf{x})$ on each element

$u_h(\mathbf{x})$ in the nodal basis:

$$u_h(\mathbf{x}) = u_i\,\varphi_i(\mathbf{x}) + u_j\,\varphi_j(\mathbf{x}) + u_k\,\varphi_k(\mathbf{x})$$

Linear basis functions: $\varphi_\ell(\mathbf{x}) = a_\ell + b_\ell \cdot x + c_\ell \cdot y$, for $\ell = i, j, k$.

Spatial derivatives of the scalar field $u_h(\mathbf{x})$ on one element T:

$$\left.\frac{\partial u_h}{\partial x}\right|_T = u_i \frac{\partial \varphi_i}{\partial x} + u_j \frac{\partial \varphi_j}{\partial x} + u_k \frac{\partial \varphi_k}{\partial x} = u_i \cdot b_i + u_j \cdot b_j + u_k \cdot b_k$$

$$\left.\frac{\partial u_h}{\partial y}\right|_T = u_i \frac{\partial \varphi_i}{\partial y} + u_j \frac{\partial \varphi_j}{\partial y} + u_k \frac{\partial \varphi_k}{\partial y} = u_i \cdot c_i + u_j \cdot c_j + u_k \cdot c_k$$

Matrix form:

$$\begin{pmatrix} \left.\frac{\partial u_h}{\partial x}\right|_T \\ \left.\frac{\partial u_h}{\partial y}\right|_T \end{pmatrix} = \begin{pmatrix} b_i & b_j & b_k \\ c_i & c_j & c_k \end{pmatrix} \cdot \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix}$$