Approximations and representations of functions in the Finite Element Method

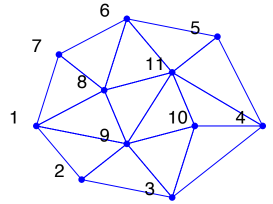
Martin Berggren

3 oktober 2011

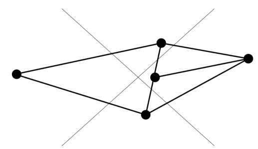
Triangulation

Aim: to approximate a a function defined in a domain Ω (with possibly a complicated shape) such than it can be represented in the computer with a finite number of floats

- **Triangulate** the domain Ω. Max diameter of any triangle: h. N vertices located at points $\boldsymbol{x}_i$, $i = 1, \dots, N$.
- In a valid triangulation, each triangle should contain nodes only at vertices. No "hanging nodes".



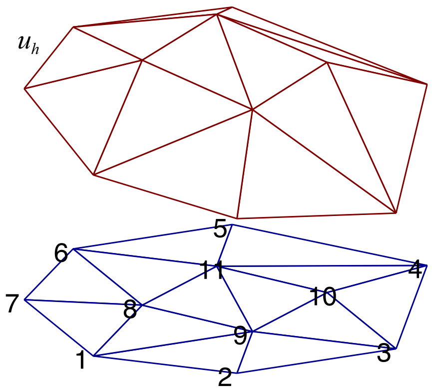
A valid triangulation with $N = 11$ nodes



Not a valid triangulation. Contains "hanging nodes".

Piecewise polynomials

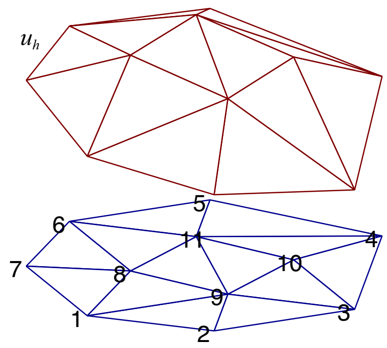
- Assume that u is a function defined on Ω
- In FEM, u is approximated with a function u_h that is glued together from simple functions on the triangles, typically polynomials
- Easiest example: u_h is **continuous** on Ω and **linear** on each triangle



Vector of nodal values

- u_h is uniquely defined by its values at the vertices $\boldsymbol{x}_i$, $i = 1, \dots N$.
- In the computer, store these values in a N-vector $\mathbf{u}$:

$$\mathbf{u} = \begin{pmatrix} 10.5 \\ 9.00 \\ 7.90 \\ 10.5 \\ 12.7 \\ 12.9 \\ 12.3 \\ 14.3 \\ 12.9 \\ 13.4 \\ 14.7 \end{pmatrix}$$



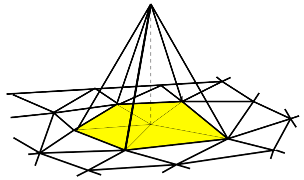
- Thus $u_i = u_h(\boldsymbol{x}_i)$

Note: Distinguish between the *vector* $\mathbf{u}$ (left) and the *function* u_h (right) !

Basis functions

The function u_h can be recreated from the vector of nodal values $\mathbf{u} = (u_1, \dots, u_N)^T$ through the use of the **nodal basis functions** ϕ_i, $i = 1, \dots, N$:

$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x})$$



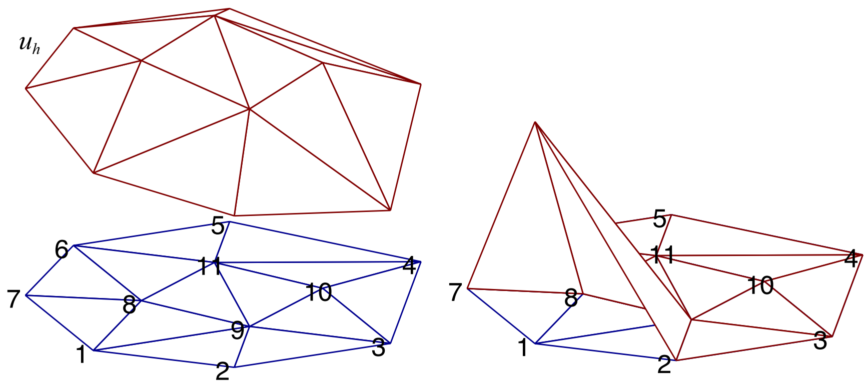
The "hat" or "tent" basis function $\phi_i(\boldsymbol{x})$ is continuous and piecewise linear, and satisfies, for each $i, j = 1, \dots, N$,

$$\phi_i(\boldsymbol{x}_j) = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{if } j \neq i \end{cases}$$

To verify this expansion, we consider, for $n = 1, \dots, N$, the **partial sums**

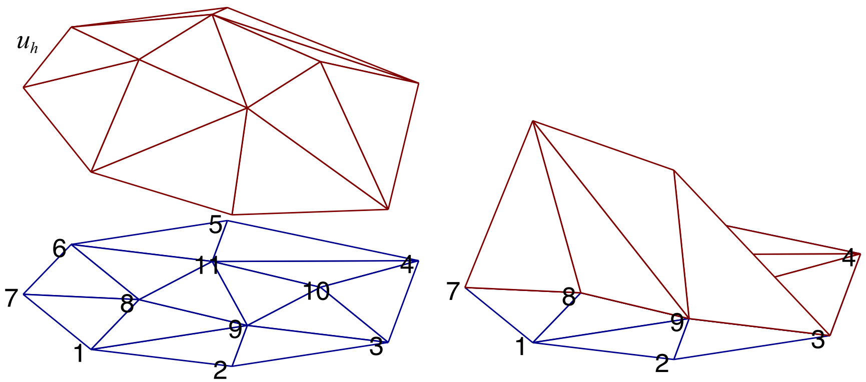
$$u_h^{(n)}(\boldsymbol{x}) = \sum_{i=1}^{n} u_i \phi_i(\boldsymbol{x})$$

Partial sums



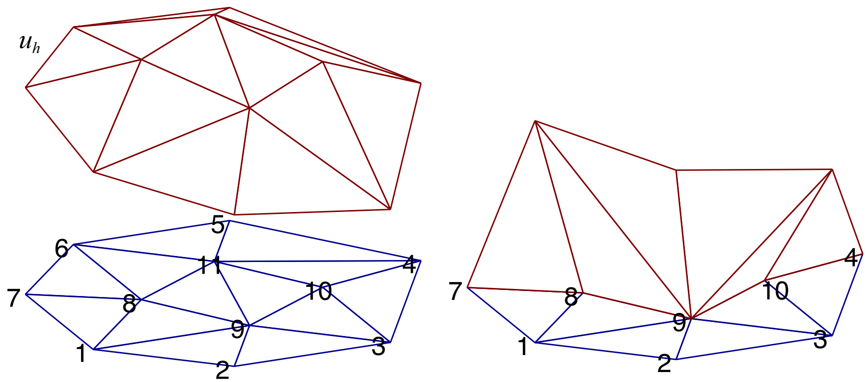
$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x}) \qquad u_h^{(1)}(\boldsymbol{x}) = \sum_{i=1}^{1} u_i \phi_i(\boldsymbol{x})$$

Partial sums



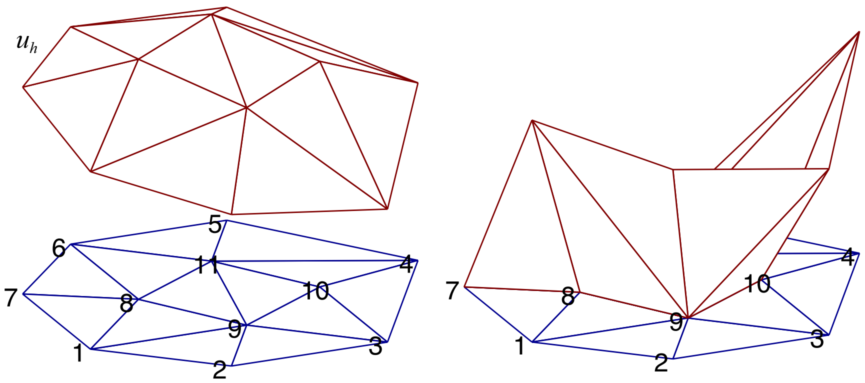
$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x}) \qquad u_h^{(2)}(\boldsymbol{x}) = \sum_{i=1}^{2} u_i \phi_i(\boldsymbol{x})$$

Partial sums



$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x}) \qquad u_h^{(3)}(\boldsymbol{x}) = \sum_{i=1}^{3} u_i \phi_i(\boldsymbol{x})$$

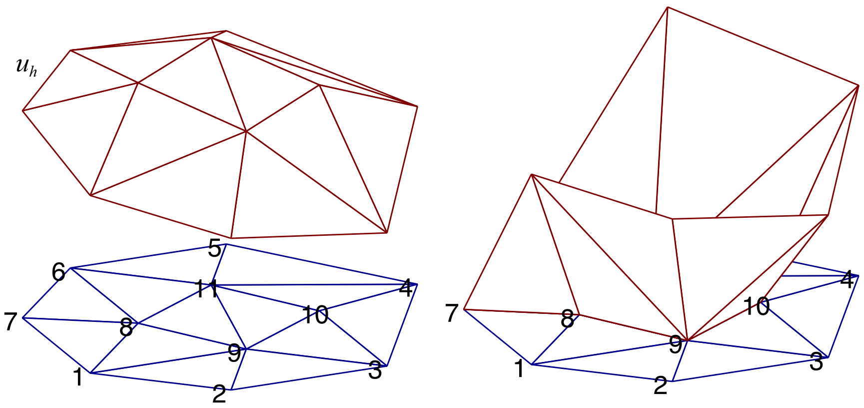
Partial sums



$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x})$$

$$u_h^{(4)}(\boldsymbol{x}) = \sum_{i=1}^{4} u_i \phi_i(\boldsymbol{x})$$

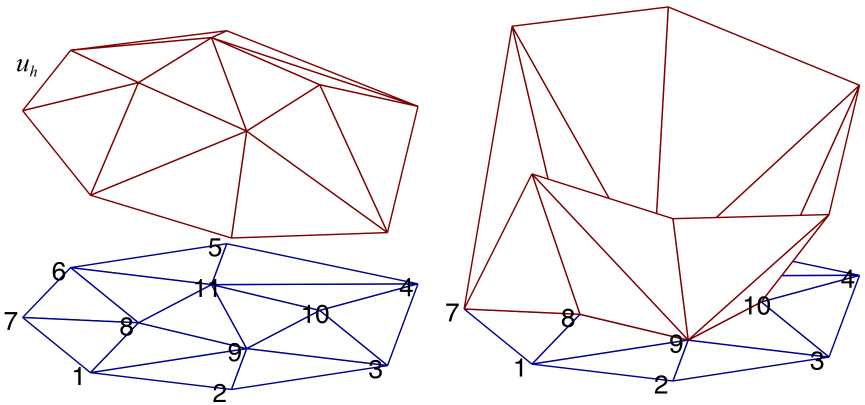
Partial sums



$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x})$$

$$u_h^{(5)}(\boldsymbol{x}) = \sum_{i=1}^{5} u_i \phi_i(\boldsymbol{x})$$

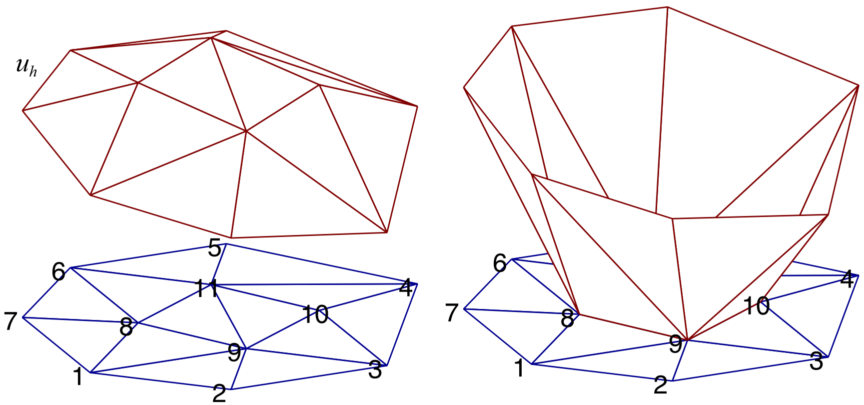
Partial sums



$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x})$$

$$u_h^{(6)}(\boldsymbol{x}) = \sum_{i=1}^{6} u_i \phi_i(\boldsymbol{x})$$

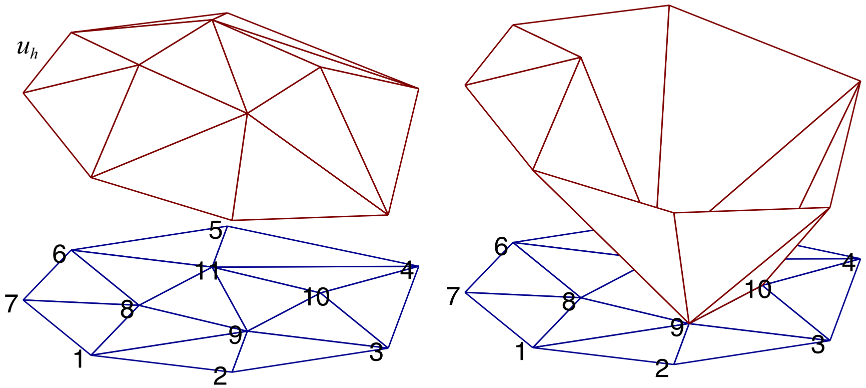
Partial sums



$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x})$$

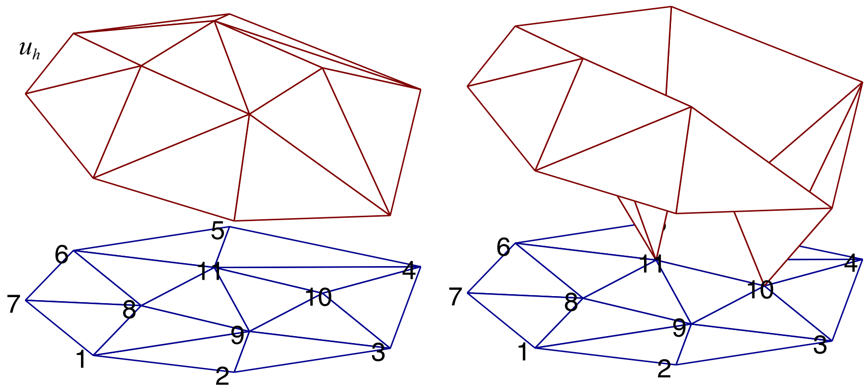
$$u_h^{(7)}(\boldsymbol{x}) = \sum_{i=1}^{7} u_i \phi_i(\boldsymbol{x})$$

Partial sums



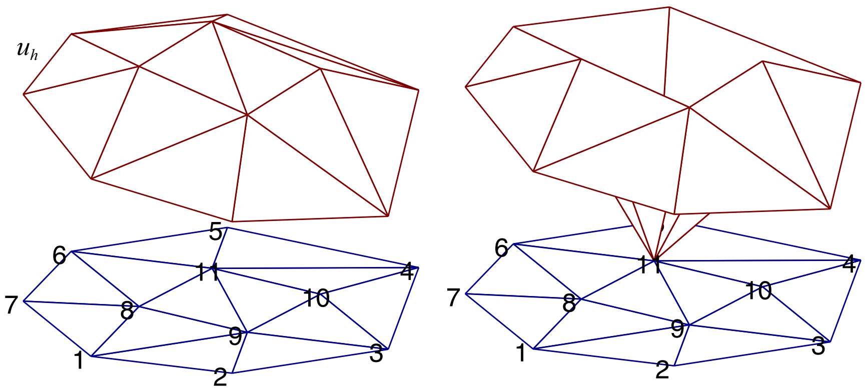
$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x}) \qquad u_h^{(8)}(\boldsymbol{x}) = \sum_{i=1}^{8} u_i \phi_i(\boldsymbol{x})$$

Partial sums



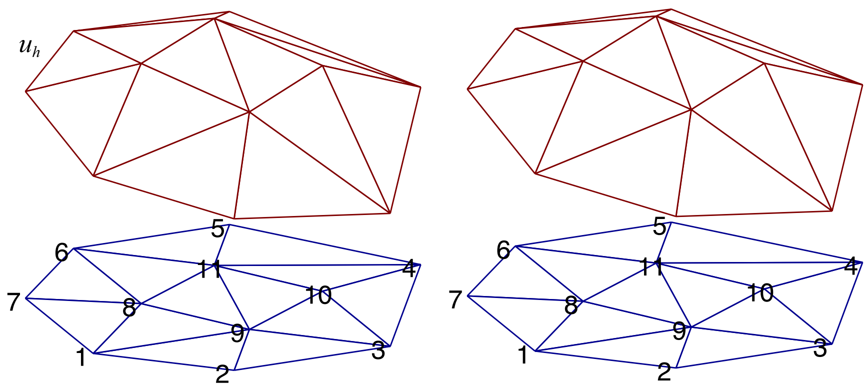
$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x}) \qquad u_h^{(9)}(\boldsymbol{x}) = \sum_{i=1}^{9} u_i \phi_i(\boldsymbol{x})$$

Partial sums



$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x}) \qquad u_h^{(10)}(\boldsymbol{x}) = \sum_{i=1}^{10} u_i \phi_i(\boldsymbol{x})$$

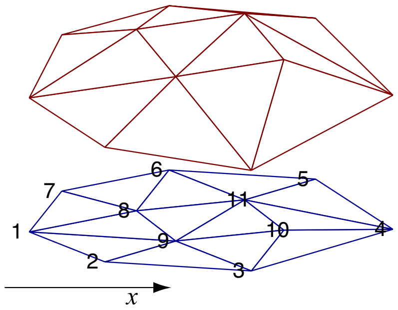
Partial sums



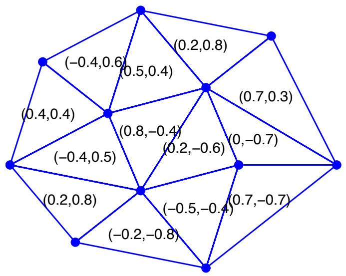
$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \phi_i(\boldsymbol{x}) \qquad u_h^{(11)}(\boldsymbol{x}) = \sum_{i=1}^{11} u_i \phi_i(\boldsymbol{x})$$

Derivatives

- The function u_h is **once** (but not twice) differentiable everywhere (except at the triangle edges).
- The gradient $\nabla u_h = \left(\frac{\partial u_h}{\partial x}, \frac{\partial u_h}{\partial y} \right)$ is **discontinuous** and **piecewise constant**



The function u_h



The piecewise-constant values of ∇u_h

Model problem

We will consider the following boundary-value problem:

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega \\ u &= 0 && \text{on } \Gamma_D \\ \frac{\partial u}{\partial n} &= g && \text{on } \Gamma_N \end{aligned} \qquad (1)$$

- The finite-element discretization will lead to a linear system of equations
$$\mathbf{A}\mathbf{u} = \mathbf{b}$$
for the vector $\mathbf{u}$ of nodal values in a finite-element function u_h that approximates the solution u of problem (1).
- Matrix $\mathbf{A}$ involves the basis functions ϕ_i, and the right-hand side vector $\mathbf{b}$ involves functions f and g
- The function u_h is **not** twice differentiable. We cannot form $-\Delta u_h$!