Approximations and representations of functions in the Finite Element Method

Martin Berggren

October 3, 2014

Approximation of functions

**Aim**: to approximate a function defined in a domain $\Omega$ (with possibly a complicated shape) such that it can be represented in the computer with a finite number of floats

- Finite volume method: uses **cell averages** to approximate the function. Leads to piecewise **constant** functions
- Such functions are sometimes (but rarely) used also for FEM
- FEM often require approximations that are **continuous** functions
- Common strategy: use continuous functions that are **linear** on each element of a **triangulation** of the computational domain

Triangulation

- Divide the domain $\Omega$ into nonoverlapping triangles. Max diameter of any triangle: $h$.
- $N$ vertices located at points $x_i, i = 1, \ldots, N$.
- In a valid triangulation, each triangle should contain nodes only at vertices. No “hanging nodes”.

Piecewise polynomials

- Assume that $u$ is a function defined on $\Omega$
- 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 $\Omega$ and **linear** on each triangle
Vector of nodal values

- $u_h$ is uniquely defined by its values at the vertices $x_i$,
  $i = 1, \ldots, N$.
- In the computer, store these values in a $N$-vector $u$:

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

- Thus $u_i = u_h(x_i)$

Note: Distinguish between the vector $u$ (left) and the function $u_h$ (right)!

Basis functions

The function $u_h$ can be recreated from the vector of nodal values $u = (u_1, \ldots, u_N)^T$ through the use of the nodal basis functions $\phi_i$,
$i = 1, \ldots, N$:

\[
  u_h(x) = \sum_{i=1}^{N} u_i \phi_i(x)
\]

The “hat” or “tent” basis function $\phi_i(x)$ is continuous and piecewise linear, and satisfies, for each $i, j = 1, \ldots, N$,

\[
  \phi_i(x_j) = \begin{cases} 
  1 & \text{if } j = i, \\
  0 & \text{if } j \neq i 
  \end{cases}
\]

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

\[
  u^{(n)}_h(x) = \sum_{i=1}^{n} u_i \phi_i(x)
\]
Partial sums

\[
 u_h(x) = \sum_{i=1}^{N} u_i \phi_i(x)
\]

\[
 u_h^{(3)}(x) = \sum_{i=1}^{3} u_i \phi_i(x)
\]

\[
 u_h^{(4)}(x) = \sum_{i=1}^{4} u_i \phi_i(x)
\]
Partial sums

\[ u_h(x) = \sum_{i=1}^{N} u_i \phi_i(x) \]

\[ u_h^{(7)}(x) = \sum_{i=1}^{7} u_i \phi_i(x) \]

\[ u_h(x) = \sum_{i=1}^{N} u_i \phi_i(x) \]

\[ u_h^{(8)}(x) = \sum_{i=1}^{8} u_i \phi_i(x) \]

\[ u_h(x) = \sum_{i=1}^{N} u_i \phi_i(x) \]

\[ u_h^{(10)}(x) = \sum_{i=1}^{10} u_i \phi_i(x) \]
Partial sums

\[ u_h(x) = \sum_{i=1}^{N} u_i \phi_i(x) \]

\[ u_h^{(1)}(x) = \sum_{i=1}^{11} u_i \phi_i(x) \]

Model problem

We will consider the following boundary-value problem:

\[-\Delta u = f \quad \text{in } \Omega \]
\[ u = 0 \quad \text{on } \Gamma_D \]
\[ \frac{\partial u}{\partial n} = g \quad \text{on } \Gamma_N \]

The finite-element discretization will lead to a linear system of equations

\[ Au = b \]

for the vector \( u \) of nodal values in a finite-element function \( u_h \) that approximates the solution \( u \) of problem (1).

- Matrix \( A \) involves the basis functions \( \phi_i \), and the right-hand side vector \( b \) involves functions \( f \) and \( g \).
- The function \( u_h \) is not twice differentiable. We cannot form \(-\Delta u_h\)!