Finite element methods in scientific computing

Wolfgang Bangerth, Colorado State University

Lecture 3.93:

The ideas behind the finite element method

Part 4: Finding an approximation

Two fundamental questions

Question 1: What is a good way to *approximate* functions that requires only finitely much data/computation?

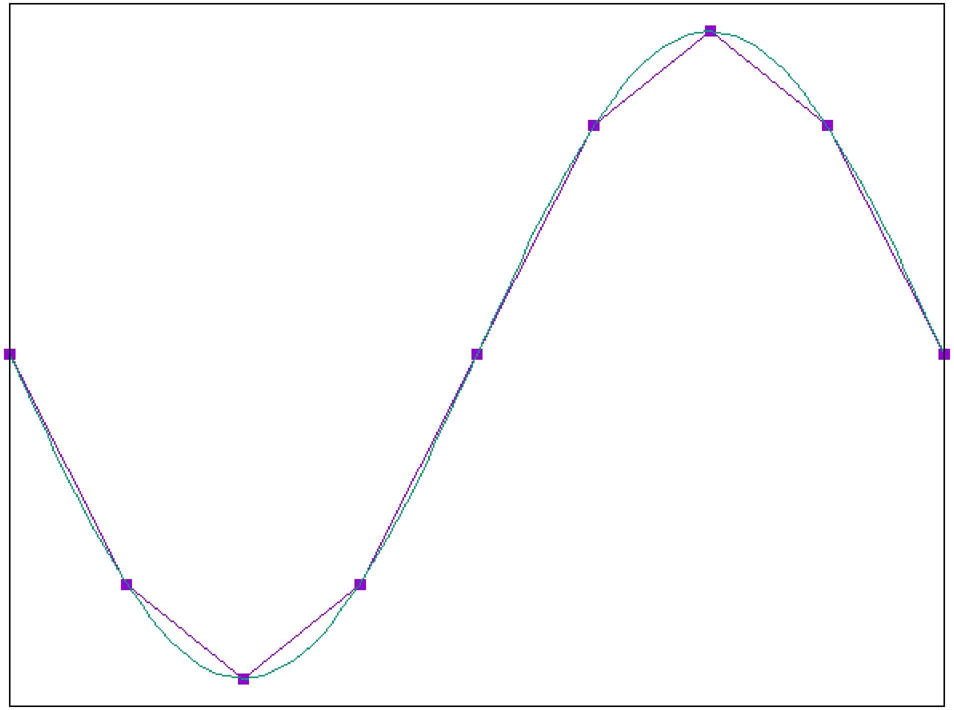
Question 2: How do we *find an approximation of the solution* of a PDE *without knowing the solution itself*?

How to find the approximation

Let us assume the following situation for now:

- We are in 1d
- We want to solve

$$-\frac{d^2}{dx^2}u(x) = f(x)$$

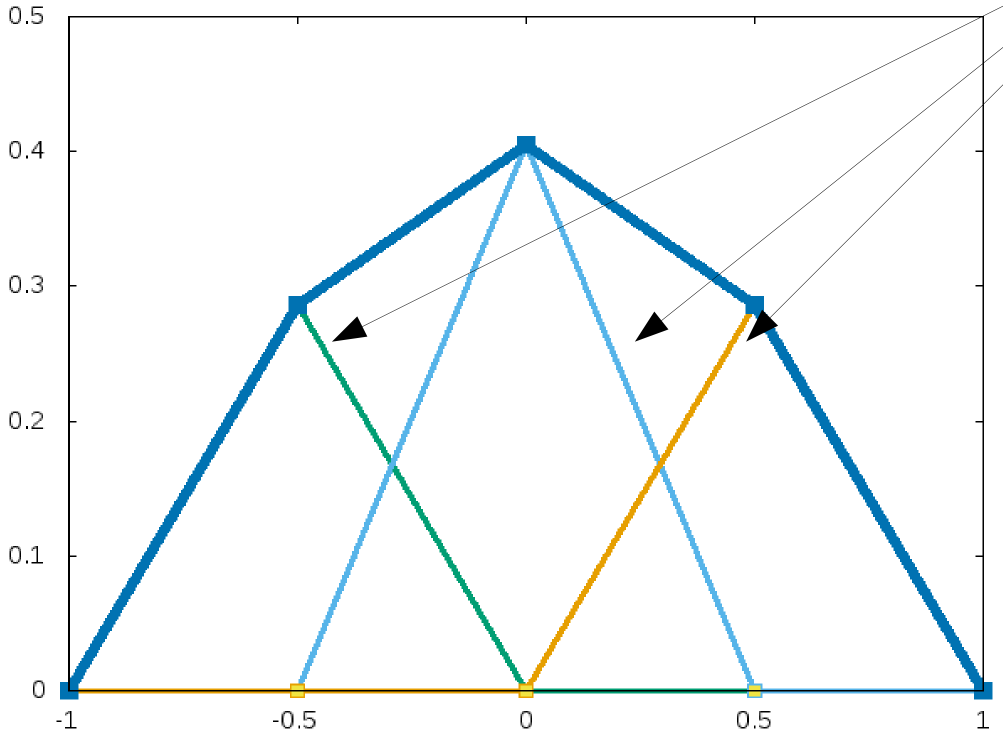


- We seek a piecewise linear approximation of the solution *u(x)*
- We will call the approximation $u_h(x)$

How to find the approximation

A little bit of mathematical abstraction:
Every piecewise linear function can be written in the form

$$u_h(x) = \sum_{j=1}^{N} \underbrace{U_j \varphi_j(x)}$$

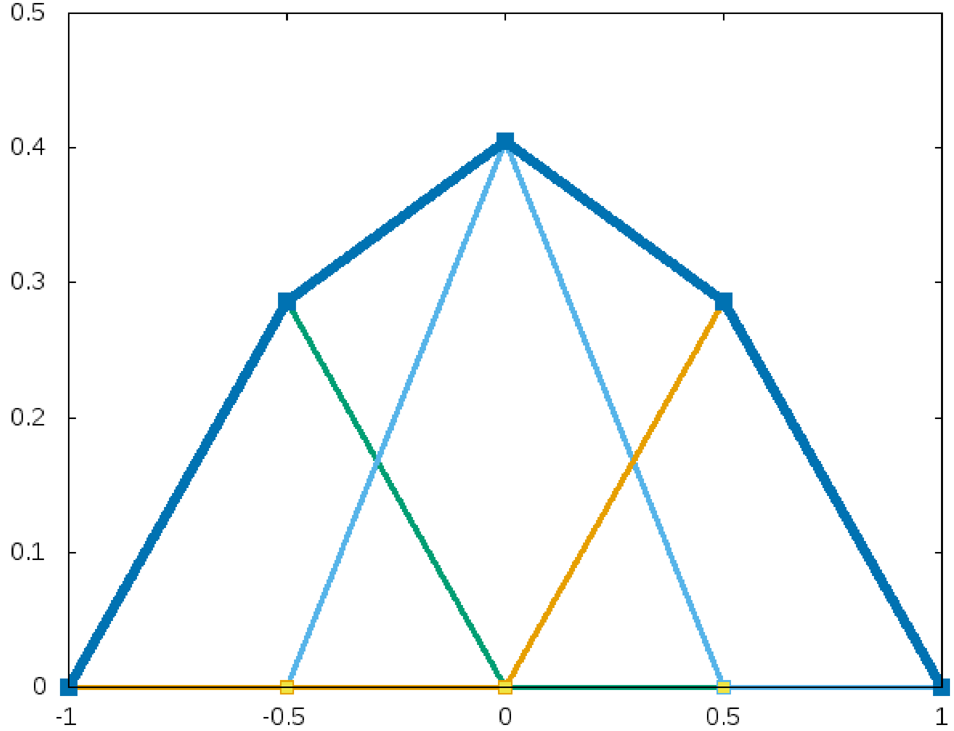


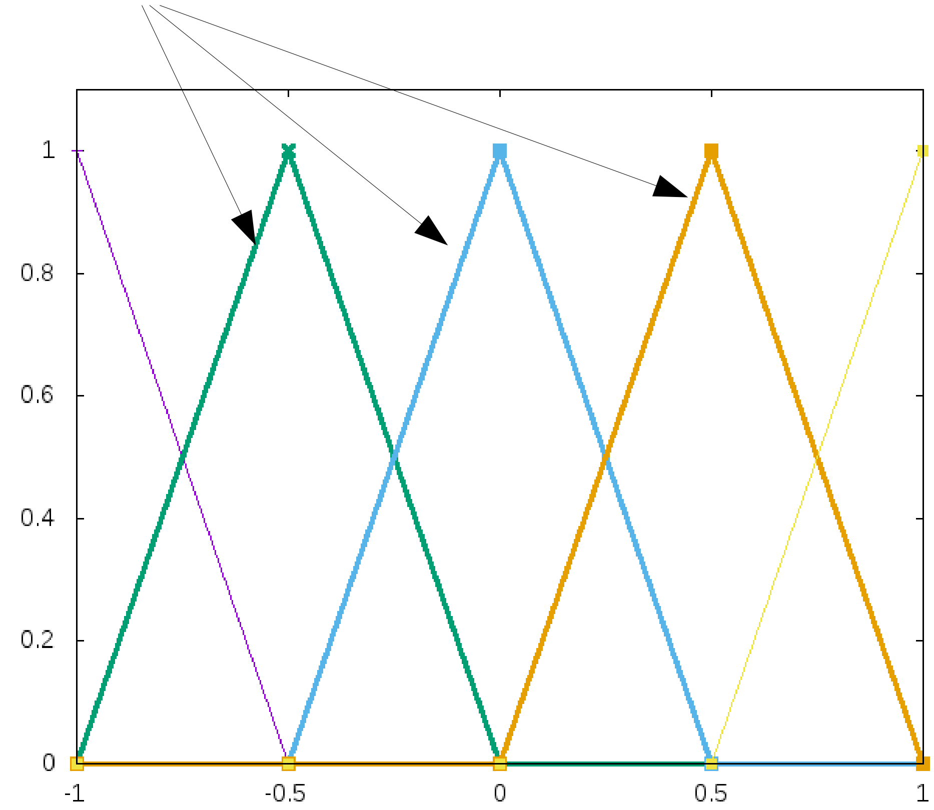
$$\begin{aligned} u_h(x) &= \text{dark blue} \\ &= \text{green} + \text{light blue} + \text{orange} \end{aligned}$$

How to find the approximation

A little bit of mathematical abstraction:
Every piecewise linear function can be written in the form

$$u_h(x) = \sum_{j=1}^{N} U_j \varphi_j(x)$$





How to find the approximation

A little bit of mathematical abstraction:
Every piecewise linear function can be written in the form

$$u_h(x) = \sum_{j=1}^{N} U_j \varphi_j(x)$$

In other words: To know $u_h(x)$, we only need to know the (finitely many) coefficientcs U_j.

Two fundamental questions

Question 2: How do we *find an approximation of the solution* of a PDE *without knowing the solution itself*?

Equivalently: How to find the coefficients U_j that define the approximation u_h?

Answer: We need to use the PDE!

How to find the approximation

Idea 1:
Take the form

$$u_h(x) = \sum_{j=1}^{N} U_j \varphi_j(x)$$

and put it into the differential equation:

$$-\frac{d^2}{dx^2} u_h(x) = f(x)$$

This does not work:

- Second derivatives $\frac{d^2}{dx^2} u_h(x)$ are zero on each interval
- Second derivatives are not defined at the "node points"

→ $\frac{d^2}{dx^2} u_h(x)$ can not equal *-f(x)*

How to find the approximation

Idea 2: Use the mathematical theory of "weak solutions"

Starting point for this theory: When we say that we want two functions *g(x)* and *h(x)* to be equal,

$$g = h$$

what do we actually mean by that?

- That they are equal for every *x*?
- That they are equal for almost every *x*?
- ...?

The problem arises because we only know how to compare *numbers*, but we now need to compare functions!

Weak solutions

A solution: Turn the equation

$$g = h$$

into an (infinite) number of comparisons of numbers.

Here: We say that g equals h if

$$\begin{aligned} F_1[g] &= F_1[h] \\ F_2[g] &= F_2[h] \\ F_3[g] &= F_3[h] \\ &\dots \end{aligned}$$

for an infinite number of appropriate "functionals" $F_i[.]$.

(Functional: Something that takes a function as argument and returns a number.)

Weak solutions

Example 1: We say that *g* equals *h* on the interval (0,1) if

$$\begin{aligned}
\int_0^1 g(x)\,dx &= \int_0^1 h(x)\,dx \\
\int_0^1 x\,g(x)\,dx &= \int_0^1 x\,h(x)\,dx \\
\int_0^1 x^2 g(x)\,dx &= \int_0^1 x^2 h(x)\,dx \\
&\dots
\end{aligned}$$

Weak solutions

Example 2: We say that g equals h on the interval (0,1) if

$$\begin{aligned}
\int_0^1 g(x)\,dx &= \int_0^1 h(x)\,dx \\
\int_0^1 \sin(\pi x) g(x)\,dx &= \int_0^1 \sin(\pi x) h(x)\,dx \\
\int_0^1 \cos(\pi x) g(x)\,dx &= \int_0^1 \cos(\pi x) h(x)\,dx \\
\int_0^1 \sin(2\pi x) g(x)\,dx &= \int_0^1 \sin(2\pi x) h(x)\,dx \\
\int_0^1 \cos(2\pi x) g(x)\,dx &= \int_0^1 \cos(2\pi x) h(x)\,dx \\
&\dots
\end{aligned}$$

Weak solutions

In general: We say that g equals h if

$$\begin{aligned} F_1[g] &= F_1[h] \\ F_2[g] &= F_2[h] \\ F_3[g] &= F_3[h] \\ &\dots \end{aligned}$$

for an infinite number of appropriate "functionals" $F_i[.]$.

Here: What is the "appropriate" set of "functionals" $F_i[.]$ depends on what kinds of functions g, h we consider.

Weak solutions

For differential equations: We say a function *u(x)* is a “weak solution” of the PDE if

$$F_1\left[-\frac{d^2}{dx^2}u\right] = F_1[f]$$

$$F_2\left[-\frac{d^2}{dx^2}u\right] = F_2[f]$$

$$F_3\left[-\frac{d^2}{dx^2}u\right] = F_3[f]$$

$$\ldots$$

where we choose $F_k[g]=\int_\Omega \varphi_k(x)g(x)\,dx$ for an infinite set of functions $\varphi_k(x)$

Weak solutions

Put differently: We say a function *u(x)* is a "weak solution" of the PDE if the equation

$$\int_\Omega \varphi(x)\left[-\frac{d^2}{dx^2}u(x)\right]dx = \int_\Omega \varphi(x) f(x)\,dx$$

holds "for all *test functions* functions $\varphi(x)$ ".

In mathematical notation:

$$\int_\Omega \varphi(x)\left[-\frac{d^2}{dx^2}u(x)\right]dx = \int_\Omega \varphi(x) f(x)\,dx \qquad \forall \varphi$$

Weak solutions

We would like to treat solution and test functions the same:

We can achieve this by integrating by parts:

$$\int_\Omega \varphi(x)\left[-\frac{d^2}{dx^2}u(x)\right]dx = \int_\Omega \left[\frac{d}{dx}\varphi(x)\right]\left[\frac{d}{dx}u(x)\right]dx + \text{boundary terms}$$

For now, we will ignore boundary terms.

In mathematical notation: *u(x)* is a solution if

$$\int_\Omega \left[\frac{d}{dx}\varphi(x)\right]\left[\frac{d}{dx}u(x)\right]dx = \int_\Omega \varphi(x) f(x)\,dx \qquad \forall \varphi$$

A side note

Remark: The following two problems are equivalent:

$$\int_\Omega \left[\frac{d}{dx}\varphi_1(x)\right]\left[\frac{d}{dx}u(x)\right]dx = \int_\Omega \varphi_1(x) f(x)\, dx$$

$$\int_\Omega \left[\frac{d}{dx}\varphi_2(x)\right]\left[\frac{d}{dx}u(x)\right]dx = \int_\Omega \varphi_2(x) f(x)\, dx$$

$$\dots$$

and

$$\int_\Omega \left[\frac{d}{dx}\varphi(x)\right]\left[\frac{d}{dx}u(x)\right]dx = \int_\Omega \varphi(x) f(x)\, dx \qquad \forall \varphi$$

This is because “every function” can be expressed as

$$\varphi(x) = \sum_{k=1}^{\infty} c_k \varphi_k(x)$$

How to find the approximation

Idea 2: Use the mathematical theory of "weak solutions"

We know that the exact solution satisfies the equality

$$\int_\Omega \left[\frac{d}{dx}\varphi(x)\right]\left[\frac{d}{dx}u(x)\right]dx = \int_\Omega \varphi(x) f(x)\, dx \qquad \forall \varphi$$

So we could try to find an approximate solution u_h that satisfies

$$\int_\Omega \left[\frac{d}{dx}\varphi(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx = \int_\Omega \varphi(x) f(x)\, dx \qquad \forall \varphi$$

How to find the approximation

Idea 2: Use the mathematical theory of "weak solutions" to find an approximate solution:

We seek

$$u_h(x) = \sum_{j=1}^{N} U_j \varphi_j(x)$$

so that

$$\int_\Omega \left[\frac{d}{dx}\varphi(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx = \int_\Omega \varphi(x) f(x)\,dx \qquad \forall \varphi$$

Pro: Only first derivatives on u_h

Con: Only N unknowns U_j, but *infinitely many equations*!

How to find the approximation

Idea 3: Restrict the set of "test functions" in the "weak formulation" to find an approximate solution:

We seek

$$u_h(x) = \sum_{j=1}^{N} U_j \varphi_j(x)$$

so that

$$\int_\Omega \left[\frac{d}{dx}\varphi_h(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx = \int_\Omega \varphi_h(x) f(x)\, dx \qquad \forall \varphi_h = \sum_{k=1}^{N} c_k \varphi_k$$

This is equivalent to *N* equations for *N* unknowns!

This is called the *Galerkin Method*.

How to find the approximation

Equivalently, the "Galerkin method" reads:

Find

$$u_h(x) = \sum_{j=1}^{N} U_j \varphi_j(x)$$

so that

$$\int_\Omega \left[\frac{d}{dx}\varphi_1(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx = \int_\Omega \varphi_1(x) f(x)\, dx$$

$$\int_\Omega \left[\frac{d}{dx}\varphi_2(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx = \int_\Omega \varphi_2(x) f(x)\, dx$$

$$\int_\Omega \left[\frac{d}{dx}\varphi_3(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx = \int_\Omega \varphi_3(x) f(x)\, dx$$

$$\ldots$$

A word on notation

We typically use the following abbreviated notation:

$$(g,h)_\Omega := \int_\Omega g(x)\,h(x)\,dx$$

We can then re-write the problem

$$\int_\Omega \left[\frac{d}{dx}\varphi_h(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx = \int_\Omega \varphi_h(x) f(x)\,dx \qquad \forall \varphi_h = \sum_{k=1}^N c_k \varphi_k$$

as follows:

$$\left(\frac{d}{dx}\varphi_h, \frac{d}{dx}u_h\right) = (\varphi_h, f)_\Omega \qquad \forall \varphi_h = \sum_{k=1}^N c_k \varphi_k$$

A word on notation

Similarly:

$$\begin{aligned}
\int_\Omega \left[\frac{d}{dx}\varphi_1(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx &= \int_\Omega \varphi_1(x) f(x)\,dx \\
\int_\Omega \left[\frac{d}{dx}\varphi_2(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx &= \int_\Omega \varphi_2(x) f(x)\,dx \\
\int_\Omega \left[\frac{d}{dx}\varphi_3(x)\right]\left[\frac{d}{dx}u_h(x)\right]dx &= \int_\Omega \varphi_3(x) f(x)\,dx \\
&\dots
\end{aligned}$$

is the same as:

$$\begin{aligned}
\left(\frac{d}{dx}\varphi_1(x), \frac{d}{dx}u_h(x)\right) &= \left(\varphi_1(x), f(x)\right) \\
\left(\frac{d}{dx}\varphi_2(x), \frac{d}{dx}u_h(x)\right) &= \left(\varphi_2(x), f(x)\right) \\
\left(\frac{d}{dx}\varphi_3(x), \frac{d}{dx}u_h(x)\right) &= \left(\varphi_3(x), f(x)\right) \\
&\dots
\end{aligned}$$

How to find the approximation

Similarly, in higher dimensions this look as follows:

Start with

$$-\Delta u(\vec{x}) = f(\vec{x})$$

Multiply by a test function, integrate:

$$\int_\Omega \varphi(\vec{x})\left[-\Delta u(\vec{x})\right] dx = \int_\Omega \varphi(\vec{x}) f(\vec{x})\, dx$$

Then integrate by parts on the left hand side:

$$\int_\Omega \varphi(\vec{x})\left[-\Delta u(\vec{x})\right] dx = \int_\Omega \left[\nabla \varphi(\vec{x})\right]\cdot\left[\nabla u(\vec{x})\right] dx + \text{boundary terms}$$

How to find the approximation

Similarly, in higher dimensions this look as follows:

We then seek

$$u_h(\vec{x}) = \sum_{j=1}^{N} U_j \varphi_j(\vec{x})$$

so that

$$\int_\Omega [\nabla \varphi_h(\vec{x})] \cdot [\nabla u_h(\vec{x})] dx = \int_\Omega \varphi_h(\vec{x}) f(\vec{x}) dx \qquad \forall \varphi_h = \sum_{k=1}^{N} c_k \varphi_k$$

Or, in shorthand notation:

$$\boxed{(\nabla \varphi_h, \nabla u_h) = (\varphi_h, f) \qquad \forall \varphi_h = \sum_{k=1}^{N} c_k \varphi_k}$$

More questions

For this method to be useful, we need to ask more questions:

Question 3: Is the approximation u_h so defined “close” to the exact solution u?

Question 4: Does u_h “converge” towards u in some useful sense?

Question 5: What is the computational effort to reach a certain accuracy? Optimality?

These are all non-trivial mathematical questions left for later lectures.

Finite element methods in scientific computing

Wolfgang Bangerth, Colorado State University