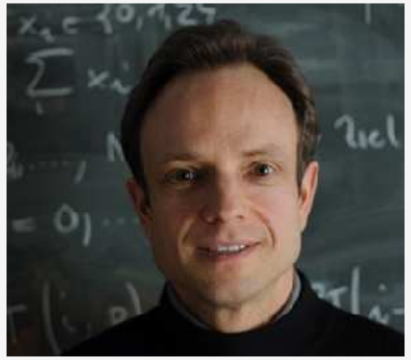
ETH Lecture 401-0674-00L Numerical Methods for Partial Differential Equations

Course Video

Section 2.3: Case Study: Linear FEM for Two-Point Boundary Value Problems

Prof. R. Hiptmair, SAM, ETH Zurich

Date: February 20, 2019



Prerequisites.

- Simple calculus in one dimension
- Numerical quadrature by means of composite quadrature rules

Dependency. Depends on [Lecture → Section 1.3], [Lecture → Section 1.5.1], [Lecture → Section 2.2]

Note: Possible minor *mismatch of video and tablet notes*!

[Corrections and updates can be incorporated into tablet notes only]

II. Finite Element Methods (FEM)

2.3. Case Study: Linear FEM for Two-Point BVPs

A 2-pt Dirichlet BVP:

$$u'' = f \quad \text{in } \Omega :=]a,b[\,, \quad u(a) = u(b) = 0$$

LVP: $\underbrace{u \in H^1_0(]a,b[)}_{\in V_0}$: $\underbrace{\int_a^b \frac{du}{dx}(x)\frac{dv}{dx}(x)\,\mathrm{d}x}_{=:a(u,v)} = \underbrace{\int_a^b f(x)v(x)\,\mathrm{d}x}_{=:\ell(v)} \quad \forall v \in \underbrace{H^1_0(]a,b[)}_{\in V_0}$. (2.3.0.1)

$f \in L^2(\Omega)$ is given in procedural form: $[f \in C^0([a,b])]$

C++11 code 2.1.2.6: Example: a functor class definition

```
template<typename ReturnType, typename PointCoordinates>
class Function {
  using value_type = ReturnType;
  using arg_type = PointCoordinates;
  Function(void);
  // evaluation operator
  value_type operator ()(const arg_type &x) const;
  .....
};
```

(2) 2.3.1. GD Step I: Choice of discrete trial & test space

Approximation in the FEM

Idea underlying the finite element method:

approximate u by means of continuous, *piecewise polynomial* functions.

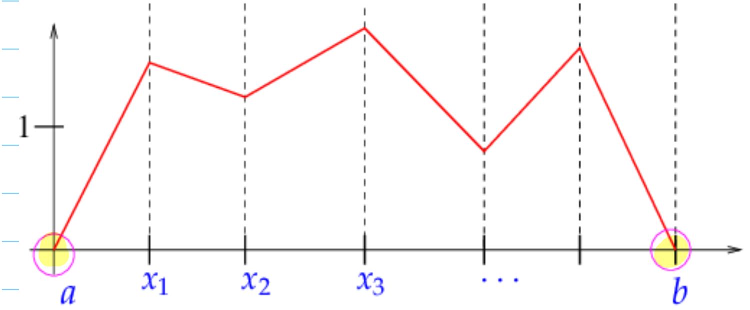
needs a partition of $[a,b]$

$x_i \hat{=}$ node ; $a =: x_0 < x_1 < \dots < x_{M-1} < x_M := b$

▷ mesh $\mathcal{M} = \{]x_{i-1}, x_i[\,;\, i = 1, \dots, M \}$

cell

Simplest option: p.w. linear



⇑ a function $\in \mathcal{S}^0_{1,0}(\mathcal{M})$

The simplest space of continuous, $\mathcal{M}$-piecewise polynomial functions in $H^1_0(]a,b[)$:

$$V_{0,h} = \mathcal{S}^0_{1,0}(\mathcal{M}) := \left\{ \begin{array}{l} v \in C^0([a,b]):\ v_{|[x_{i-1},x_i]} \text{ linear}, \\ i = 1,\dots,M,\ v(a) = v(b) = 0 \end{array} \right\}$$

➤ $N := \dim V_{0,h} = M - 1$

$V_{0,h} \subset C^1_{pw}([a,b]) \subset H^1(]a,b[)$

Notation: $\mathcal{S}^0_{1,0}(\mathcal{M})$

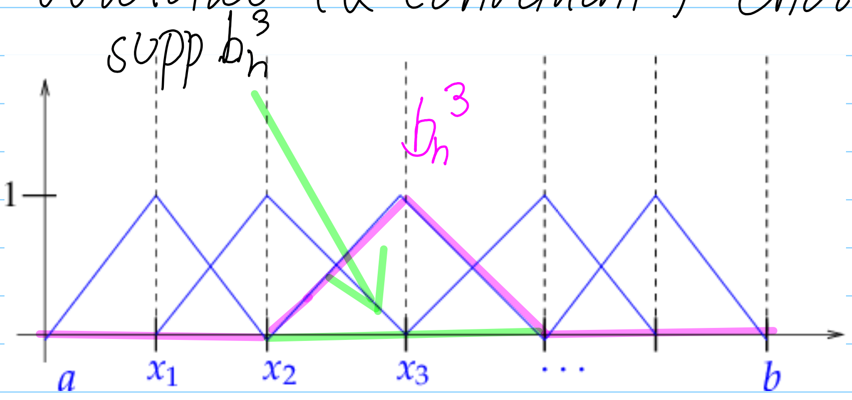
$\mathcal{S}$ for "scalar"; C^0-continuous; degree 1 polynomials; the partition; $= 0$ in endpoints

Remark: $\mathcal{S}^0_{1,0}(\mathcal{M}) \not\subset C^1([a,b])$

↳ can be used in DVP, since smoothness requirements for v.f. relaxed.

2.3.2 GD Step II: Choice of basis

Natural (& convenient) choice: tent functions



defined by cardinal basis property w.r.t. $\{x_i\}$

$$b_h^j(x_i) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

$$u_h \in \mathcal{S}^0_{1,0}(\mathcal{M}) \Rightarrow u_h(x) = \sum_{i=1}^{N} u_h(x_i)\, b_h^i(x)$$

$$N := \dim \mathcal{S}^0_{1,0}(\mathcal{M}) = M - 1$$

③ For $v_h \in \mathcal{S}^0_{1,0}(\mathcal{M}) \Rightarrow \frac{dv_h}{dx}$ is $\mathcal{M}$-p.w. constant

[ok for v.f., because we integrate]

(*) $$\frac{db_h^j}{dx}(x) = \begin{cases} 1/h_j & , \; x_{j-1} < x \le x_j \\ -1/h_{j+1} & , \; x_j < x \le x_{j+1} \\ 0 & , \; \text{elsewhere} \end{cases}$$

$h_j := |x_j - x_{j-1}|$ ≙ cell size

2.3.3. Formulas for Galerkin Matrices & r.h.s Vectors

$$u_h = \sum_{\ell=1}^{N} \mu_\ell b_h^\ell \quad \hat{=} \text{ Galerkin solution}$$

Plug into LVP

$$u \in H_0^1(]a,b[): \quad \underbrace{\int_a^b \frac{du}{dx}(x)\frac{dv}{dx}(x)\,dx}_{=:\mathsf{a}(u,v)} = \underbrace{\int_a^b f(x)v(x)\,dx}_{=:\ell(v)} \quad \forall v \in H_0^1(]a,b[)\,. \tag{2.3.0.1}$$

$$\underbrace{\int_a^b \sum_{l=1}^{N} \mu_l \frac{db_h^l}{dx}(x)\frac{db_h^k}{dx}(x)\,dx}_{=\mathsf{a}(u_h,b_h^k)} = \underbrace{\int_a^b f(x)b_h^k(x)\,dx}_{=\ell(b_h^k)}\,, \quad k = 1,\dots,N\,.$$

piecewise derivatives ⇕

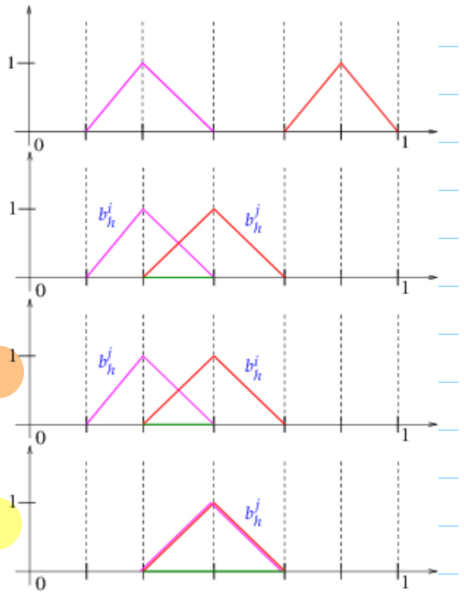
$$\sum_{l=1}^{N} \underbrace{\left(\int_a^b \frac{db_h^l}{dx}(x)\frac{db_h^k}{dx}(x)\,dx\right)}_{=(\mathbf{A})_{k\ell}} \mu_l = \underbrace{\int_a^b f(x)b_h^k(x)\,dx}_{:=\varphi_k}\,, \quad k = 1,\dots,N\,.$$

⇕

Straightforward from (*)

non-overlapping supports →

$$\int_a^b \frac{db_h^j}{dx}(x)\frac{db_h^i}{dx}(x)\,dx = \begin{cases} 0 & , \text{ if } |i-j| \ge 2 \\ -\frac{1}{h_{i+1}} & , \text{ if } j = i+1 \\ -\frac{1}{h_i} & , \text{ if } j = i-1 \\ \frac{1}{h_i} + \frac{1}{h_{i+1}} & , \text{ if } 1 \le i = j \le M-1 \end{cases}$$



$$\mathbf{A} = \begin{bmatrix} \frac{1}{h_1}+\frac{1}{h_2} & -\frac{1}{h_2} & & 0 & & 0 \\ -\frac{1}{h_2} & \frac{1}{h_2}+\frac{1}{h_3} & -\frac{1}{h_3} & & & \\ 0 & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & 0 \\ & & & \ddots & \ddots & -\frac{1}{h_{M-1}} \\ 0 & & & 0 & -\frac{1}{h_{M-1}} & \frac{1}{h_{M-1}}+\frac{1}{h_M} \end{bmatrix} \in \mathbb{R}^{N,N}, \quad N := M-1\,. \tag{2.3.3.2}$$

S.p.d. tridiagonal matrix

4

Special case: equidistant mesh $x_j = a + \frac{b-a}{M} j$

$$\frac{1}{h}\begin{bmatrix} 2 & -1 & 0 & & & & 0 \\ -1 & 2 & -1 & & & & \\ 0 & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & 0 \\ & & & & -1 & 2 & -1 \\ 0 & & & & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \end{bmatrix} = h \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}. \tag{2.3.3.4}$$

1D Poisson matrix

Right-hand-side vector:

$$\varphi_k = \int_a^b f(x)\, b_h^k(x)\, dx$$

If only point evaluations of f are possible, then the only option is the computation of right hand side vector $\vec{\varphi} \in \mathbb{R}^N$ by **numerical quadrature** !

Integrals are approximated by weighted sums of point values of the integrands!

Replace the integral with an m-point quadrature formula/quadrature rule on $[a,b]$, $m \in \mathbb{N}$ $\to$:

$$\int_a^b \psi(t)\,\mathrm{d}t \approx Q_m(\psi) := \sum_{j=1}^{m} \omega_j^m \, \psi(\zeta_j^m) . \tag{2.3.3.7}$$

ω_j^m: quadrature weights , ζ_j^m: quadrature nodes $\in [a,b]$.

$\Longrightarrow$ only p.w. smooth (polynomial) FE functions suggests:

$\triangleright$ use M-based composite quadrature rule

$$\int_a^b \psi(t)\,dt = \sum_{i=1}^{M} \int_{x_{i-1}}^{x_i} \psi(t)\,dt$$

Apply local quadrature rule here!

Example: 1D composite trapezoidal rule

$$\int_a^b \psi(t)\,\mathrm{d}t \approx \sum_{l=1}^{M} \tfrac{1}{2} h_l \left(\psi(x_{l-1}) + \psi(x_l)\right) . \tag{2.3.3.8}$$

$\Downarrow$

$$\blacktriangleright \quad \varphi_k := (\vec{\varphi})_k = \int_a^b f(x)\, b_h^k(x)\,\mathrm{d}x \approx \tfrac{1}{2}(h_k + h_{k+1}) f(x_k), \quad 1 \le k \le N . \tag{2.3.3.9}$$

Use $b_h^j(x_i) = \delta_{ij}$

5 Another use case of numerical quadrature

→ Treatment of coefficient function :

$$u \in H_0^1(]a,b[): \int_a^b \sigma(x) \frac{du}{dx}(x) \frac{dv}{dx}(x)\,\mathrm{d}x = \int_a^b f(x)v(x)\,\mathrm{d}x \quad \forall v \in H_0^1(\Omega)]a,b[\,.$$

Assumption 2.3.3.11. Smoothness requirement for stiffness coefficient

σ is piecewise continuous, $\sigma \in C^0_{\mathrm{pw}}([a,b])$, with jumps *only* at grid nodes x_j

$$\int_a^b \sigma(x) \sum_{l=1}^N \mu_l \frac{db_h^l}{dx}(x) \frac{db_h^k}{dx}(x)\,\mathrm{d}x = \int_a^b f(x) b_h^k(x)\,\mathrm{d}x\,, \quad k = 1,\dots,N\,. \qquad (2.3.3.12)$$

p.w. constant

Num. quad. : Composite midpoint rule

$$\int_a^b \psi(x)\,\mathrm{d}x \approx \sum_{j=1}^M h_j \psi(m_j)\,, \quad m_j := \tfrac{1}{2}(x_j + x_{j-1})\,. \qquad (2.3.3.13)$$

$$\sum_{\ell=1}^N \underbrace{\left(\sum_{j=1}^M h_j \sigma(m_j) \frac{db_h^\ell}{dx}(m_j) \frac{db_h^k}{dx}(m_j)\right)}_{=(\mathbf{A})_{k,\ell}} \mu_\ell = \underbrace{\tfrac{1}{2}(h_{k+1} + h_k) f(x_k)}_{=:(\vec{\varphi})_k}\,, \quad k = 1,\dots,N\,,$$

just a point evaluation

$$\Updownarrow$$

$$\mathbf{A}\vec{\mu} = \vec{\varphi}\,.$$

▷ For equidistant mesh with cell size $h > 0$:

$$\frac{1}{h}\begin{bmatrix} \sigma_1+\sigma_2 & -\sigma_2 & 0 & & & 0 \\ -\sigma_2 & \sigma_2+\sigma_3 & -\sigma_3 & & & \\ 0 & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & 0 \\ & & & -\sigma_{M-2} & \sigma_{M-2}+\sigma_{M-1} & -\sigma_{M-1} \\ 0 & & & 0 & -\sigma_{M-1} & \sigma_{M-1}+\sigma_M \end{bmatrix} \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \end{bmatrix} = h \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}, \qquad (2.3.3.14)$$

with $\sigma_j := \sigma(m_j)$, $j = 1,\dots,m$.

Review questions 2.3.3.17 *

[to be answered in closed-book setting]

A:

The piecewise linear functions used for the finite element Galerkin discretization of a second-order elliptic boundary value problem on $]a,b[$ are not differentiable on $]a,b[$ in general. Explain why they can be used nevertheless.

B:

We consider a **graded mesh** $\mathcal{M}_\alpha$ of the interval $\Omega := [0,1]$ with node set

$$\mathcal{V}(\mathcal{M}) := \{x_j := (j/n)^\alpha,\ j = 0,\dots,n\}\ ,\quad n \in \mathbb{N}\ ,\quad \alpha > 0\ . \tag{2.3.3.18}$$

What is the smallest and largest cell size of $\mathcal{M}_\alpha$?

C:

Let $\{b_h^1,\dots,b_h^{n-1}\}$ stand for the tent function basis of $\mathcal{S}_{1,0}^0(\mathcal{M}_2)$ on the graded mesh $\mathcal{M}_2$ of $\Omega :=]0,1[$, whose nodes are given by (2.3.3.18) for $\alpha = 2$. These basis functions satisfy $b_h^j(x_i) = \delta_{i,j}$, $i,j = 1,\dots,n-1$. Compute the norms $\|b_h^i\|_{L^2(\Omega)}$ and $|b_h^i|_{H^1(\Omega)}$.

D:

On the graded mesh $\mathcal{M}_2$ as given by (2.3.3.18) for $\alpha = 2$ we consider the finite element space $\mathcal{S}_{1,0}^0(\mathcal{M})$ equipped with the tent function basis $\{b_h^1,\dots,b_h^{n-1}\}$, $b_h^j(x_i) = \delta_{i,j}$, $i,j = 1,\dots,n-1$. Compute the vector $\vec{\varphi} \in \mathbb{R}^{n-1}$ obtained by the Galerkin discretization of the right-hand side functional $\ell(v) := \int_0^1 v(x)\,\mathrm{d}x$.

E:

We conduct the Galerkin finite-element discretization of the generic second-order elliptic two-point Dirichlet boundary value problem

$$u \in H_0^1(]0,1[):\quad \int_a^b \sigma(x)\frac{du}{dx}(x)\frac{dv}{dx}(x)\,\mathrm{d}x = \int_a^b f(x)v(x)\,\mathrm{d}x \quad \forall v \in H_0^1(]0,1[)\ ,$$

with uniformly positive coefficient σ, using the trial and test space $\mathcal{S}_{1,0}^0(\mathcal{M}_\alpha)$ on the graded mesh according to (2.3.3.18) and with the standard tent function basis. Give sharp upper and lower bounds for the number of non-zero entries of the Galerkin matrix.

F:

What is numerical quadrature and how is it applied in the context of the Galerkin discretization of the right-hand side linear functional

$$v \mapsto \int_0^1 f(x)v(x)\,\mathrm{d}x\ ,\quad f \in C^0([0,1])\ .$$

* Additional review questions may be given in lecture document.

7

G:

The use of the composite trapezoidal quadrature rule on a mesh $\mathcal{M}$ of $]a,b[$ with nodes $a = x_0 < x_1 < \cdots < x_{M-1} < x_M := b$ amounts to the approximation

$$\int_a^b \phi(x)\,\mathrm{d}x \approx \tfrac{1}{2}h_1\phi(x_0) + \sum_{j=1}^{M-1} \phi(x_j)\tfrac{1}{2}(h_j + h_{j+1}) + \tfrac{1}{2}h_M\phi(x_M)\ .$$

Explain the difficulty encountered when applying this quadrature rule for the computation of entries of the Galerkin matrix for the linear variational problem

$$u \in H_0^1(]a,b[):\quad \int_a^b \sigma(x)\frac{du}{dx}(x)\frac{dv}{dx}(x)\,\mathrm{d}x = \int_a^b f(x)v(x)\,\mathrm{d}x \quad \forall v \in H_0^1(]a,b[)\ ,$$

discretized based on $\mathcal{S}_{1,0}^0(\mathcal{M})$ and its tent function basis.

H:

We consider the bilinear form

$$\mathsf{b}(u,v) := \int_0^1 u(x)\,\frac{dv}{dx}(x)\,\mathrm{d}x\ ,\quad u \in L^2(]0,1[)\ ,\quad v \in H^1(]0,1[)\ .$$

We study its Galerkin discretization based on $\mathcal{S}_{1,0}^0(\mathcal{M})$ on an equidistant mesh of $]0,1[$ with M cells, using the standard tent function basis. Compute the resulting Galerkin matrix.