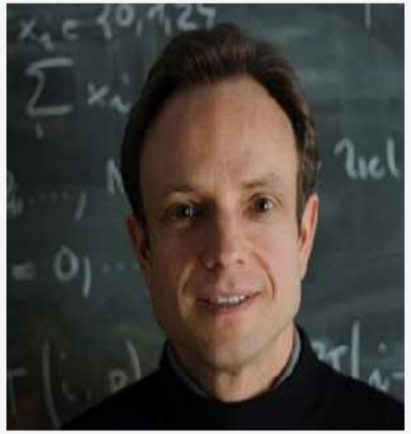
ETH Lecture 401-0674-00L Numerical Methods for Partial Differential Equations

Course Video

Section 3.8: Validation and Debugging of Finite Element Codes

Prof. R. Hiptmair, SAM, ETH Zurich

Date: March 21, 2019



Dependency. [Lecture → Section 3.1] and [Lecture → Section 3.5]

III FEM: Convergence & Accuracy

3.8 Validation & Debugging of FE Codes

Code based on $\mathcal{S}_p^0(\mathcal{M})$, p fixed, for solving

$$u \in H^1(\Omega),\ u = g \text{ on } \Gamma_D: \quad \int_\Omega \boldsymbol{\alpha}(\boldsymbol{x})\, \mathbf{grad}\, u \cdot \mathbf{grad}\, v + \gamma(\boldsymbol{x}) u\, v \,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_R} \lambda(\boldsymbol{x}) u\, v \,\mathrm{d}S(\boldsymbol{x}) = \int_\Omega f\, v \,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_N} h\, v \,\mathrm{d}S(\boldsymbol{x}) \quad \forall v \in H^1_{\Gamma_D}(\Omega), \quad (3.8.0.2)$$

● passed through function

For testing we will take for granted the availability of sequences of meshes $\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \ldots$, which satisfy (see § 3.3.5.17 for related requirements)

1) that the meshwidth decreases geometrically: $h_k = q h_{k-1}$ for some $0 < q < 1$, where h_k is the meshwidth of $\mathcal{M}_k$.
2) that all cells of $\mathcal{M}_k$ have about the same size h_k. This feature is called quasi-uniformity.
3) that the shape regularity measure (→ Def. 3.3.2.20) all meshes stays below a common bound, a property called uniform shape regularity.

[LF++ Mesh Hierarchy]

2

(A) Observe "asymptotic" (alg.) cvg : (§3.8.0.4)

From theory we know sharp bounds :

$\| u - u_h \| \approx C N_K^{-\alpha}$, $N_K = \dim S_p^0(\mathcal{M}_K)$

$$N_k \approx \kappa N_{k-1} \quad \text{for some} \quad \kappa > 1 \quad \Rightarrow \quad N_k \approx \kappa^k N_0 \,. \tag{3.8.0.6}$$

$$\|u_k - u_{k-1}\| \leq \|u_k - u\| + \|u - u_{k-1}\| \approx CN_0\left(\kappa^{-k\alpha} + \kappa^{-(k-1)\alpha}\right) \approx C' N_k^{-\alpha} \,, \tag{3.8.0.7}$$

↑ ?

$$\big|\|u_k\| - \|u_{k-1}\|\big| \leq \|u_k - u_{k-1}\| \overset{(3.8.0.7)}{\approx} CN_k^{-\alpha} \,. \tag{3.8.0.8}$$

(B) Method of manufactured solutions (§3.8.0.9)

- Pick polygonal $\Omega \subset \mathbb{R}^d$ ($\to$ no bd. approx.)
- Use coefficients w/ simple analytic formulas
- Pick $u \in C^\infty(\bar{\Omega})$ $\to$ set f, g, h accordingly
  ↳ simple analytic formula
- CODE : Solve "manufactured" BVP

CODE : Compute $\| u - u_K \|$ ["overkill quad."]

- Estimate rate of alg. cvg.
  Matches theoretical predictions ?

Mismatch : Simplify : $\Gamma_D = \partial\Omega$, $g = 0$
$\lambda = 0$, $\alpha = 1$

Beware of polynomial exact solutions $u \in \mathcal{P}_p$! (Why?) On the other hand, if the above test fails for non-polynomial u, the next step should be to probe $u \in \mathcal{P}_p$ (Why?).

[If $u \in S_p(\mathcal{M}) \Rightarrow u = u_h$]

(C) Direct testing H^1-continuous (bi-) linear forms: (§3.8.0.10)

Goal: Validate Galerkin matrix B for a symmetric bilinear form $b(\cdot,\cdot)$ on $S_p^0(\mathcal{M})$

- Use a simple polygonal / circular Ω
- Choose $w \in C^\infty(\bar{\Omega})$: analytic formula
- Compute $b(w,w)$ [exactly by computer algebra]
- CODE : Compute $b(I_p w, I_p w)$ on $\mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_K$
  ↑ nodal interpolant $\in S_p^0(\mathcal{M})$

$b(I_p w, I_p w) = \vec{\nu}^T B \vec{\nu}$, $\vec{\nu} \hat{=}$ coefficient vector for $I_p w$

3

$$\underbrace{\mathsf{b}(w,w) - \vec{v}_k^\top \mathbf{B}_k \vec{v}_k}_{\text{known}} = \mathsf{b}(w,w) - \mathsf{b}(\mathsf{I}_k w, \mathsf{I}_k w) \overset{\text{bilinearity \& symmetry}}{=} \mathsf{b}(w + \mathsf{I}_k w, w - \mathsf{I}_k w)$$

$$\overset{(3.8.0.12)}{\leq} C_c \|w + \mathsf{I}_k w\|_{H^1(\Omega)} \|w - \mathsf{I}_k w\|_{H^1(\Omega)}$$

continuity

$$\leq C_c \|w\|_{H^1(\Omega)} \left(1 + C h_k^p \|w\|_{H^{p+1}(\Omega)}\right) \left(C h_k^p \|w\|_{H^{p+1}(\Omega)}\right) = O(h_k^p)\ .$$

known asymptotic behavior

④ Review questions 3.8.0.13 :

A :

What is the strong (PDE) for of the boundary value problem, whose weak form reads

$$\begin{array}{l} u \in H^1(\Omega) \\ u = g \text{ on } \Gamma_D \end{array}: \int_\Omega \boldsymbol{\alpha}(\boldsymbol{x})\,\mathbf{grad}\,u \cdot \mathbf{grad}\,v + \gamma(\boldsymbol{x}) u\,v\,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_R} \lambda(\boldsymbol{x}) u\,v\,\mathrm{d}S(\boldsymbol{x})$$
$$= \int_\Omega f\,v\,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_N} h\,v\,\mathrm{d}S(\boldsymbol{x}) \quad \forall v \in H^1_{\Gamma_D}(\Omega)\,, \quad (3.8.0.2)$$

with the Sobolev space

$$H^1_{\Gamma_D}(\Omega) := \left\{ v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D \right\}, \quad (3.8.0.3)$$

based on a partition $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_N \cup \Gamma_R$.

B :

We consider (3.8.0.2) for $\boldsymbol{\alpha} \equiv \mathbf{I}$, $\gamma \equiv 1$, $\lambda \equiv 1$, and on the unit disk domain $\Omega := \{\boldsymbol{x} \in \mathbb{R}^2 : \|\boldsymbol{x}\| < 1\}$ and with $\Gamma_R := \partial\Omega$.

(i) Is it possible to choose the data f and h such that $u(\boldsymbol{x}) = \cos(\pi/2\|\boldsymbol{x}\|)$ will be the exact solution of the variational problem. If not, suggests a modification that makes it possible.

(ii) Possibly under the modification found in [(i)], determine those functions f and h that will yield that exact solution $u(\boldsymbol{x}) = \cos(\pi/2\|\boldsymbol{x}\|)$.

C :

In connection with the method of manufactured solutions you have seen the warning

> ⚠ Beware of polynomial exact solutions $u \in \mathcal{P}_p$! (Why?) On the other hand, if the above test fails for non-polynomial u, the next step should be to probe $u \in \mathcal{P}_p$ (Why?).

Try to answer the "Whys".

D :

You have just finished the implementation of the LehrFEM++-based C++ function

```
template<typename FUNC_ALPHA, typename FUNC_GAMMA, typename
   FUNC_BETA>
  Eigen::SparseMatrix<double> compGalerkinMatrix(
     const lf::assemble::DofHandler &lfe_dofh,
     FUNC_ALPHA&& alpha, FUNC_GAMMA&& gamma, FUNC_BETA&& beta);
```

$$u \in H^1(\Omega): \int_\Omega \alpha(\boldsymbol{x})\,\mathbf{grad}\,u(\boldsymbol{x}) \cdot \mathbf{grad}\,v(\boldsymbol{x}) + \gamma(\boldsymbol{x})\,u(\boldsymbol{x})\,v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} \beta(\boldsymbol{x})\,u(\boldsymbol{x})\,v(\boldsymbol{x})\,\mathrm{d}S(\boldsymbol{x}) =$$
$$\int_\Omega f(\boldsymbol{x})\,v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \forall v \in H^1(\Omega)\,,$$

where $\alpha, \gamma : \Omega \to \mathbb{R}$, $\beta : \partial\Omega \to \mathbb{R}$ are bounded coefficient functions, and $f \in L^2(\Omega)$. The use of the standard nodal basis consisting of "tent functions" is assumed.

The argument `lfe_dofh` passes the loca-to-global index mapping for $\mathcal{S}^0_1(\mathcal{M})$ and the mesh $\mathcal{M}$, while `alpha`, `gamma`, and `beta` are functors for the coefficient functions $\boldsymbol{x} \mapsto \alpha(\boldsymbol{x})$, $\boldsymbol{x} \mapsto \gamma(\boldsymbol{x})$, and $\boldsymbol{x} \mapsto \beta(\boldsymbol{x})$.

Unfortunately, your implementation does not work properly. Sketch a debugging strategy based on the policy of direct testing of bilinear forms.

> ⚠ This list of review questions may not be complete. Additional review questions may be provided in the lecture document.

6