Institute for Analysis and Scientific Computing
TU Wien
Philip L. Lederer

Numerical methods for partial differential equations
Exercise 7 – 5. May 2020

Example 7.1

Let $\Omega \subset \mathbb{R}^d$ be a convex domain with Lipschitz boundary and $V = H^1(\Omega)$, $f \in L^2(\Omega)$. Denote by $\mathcal{T}_h$ an admissible triangulation of Ω and let V_h be the Lagrange finite element space of order k. Consider the following problem:

$$\text{Find } u \in V \text{ s.t. } \langle \nabla u, \nabla v \rangle_{L^2(\Omega)} + \langle u, v \rangle_{L^2(\Omega)} = f(v), \quad \forall\, v \in V$$

The corresponding FE approximation is denoted by $u_h \in V_h$. For a function $\tilde{u} \in H^1$ we define the residual $R(\tilde{u}) \in H^{-1}$ by

$$R(\tilde{u}) := \Delta \tilde{u} - \tilde{u} + f,$$

with

$$R(\tilde{u})(v) = \langle R(\tilde{u}), v \rangle_{H_0^1} = \int_\Omega -\nabla \tilde{u} \cdot \nabla v - \tilde{u} v + f \cdot v \,\mathrm{d}x \quad \forall v \in H_0^1(\Omega).$$

1. Show that $R(u_h)(v_h) = 0$ holds for all $v_h \in V_h$.
2. Derive the localised representation of $R(u_h)(v)$, $v \in V$, in the following manner:

$$R(u_h)(v) = \sum_{T \in \mathcal{T}_h} \left[\int_T \cdots (v - v_h) \,\mathrm{d}x + \sum_{F \in \partial T \setminus \partial\Omega} \int_F \cdots (v - v_h) \,\mathrm{d}s + \sum_{F \in \partial T \cap \partial\Omega} \int_F \cdots (v - v_h) \,\mathrm{d}s \right]$$

3. Show that
$$\|u - u_h\|_{H^1(\Omega)} \leq c \|R(u_h)\|_{V^*} \quad \text{with constant } c = 1.$$
(This will show that the residual error estimator is reliable with constant 1).

Example 7.2 (Goal driven error estimates:)

In the lecture we only introduced error estimators with respect to the error in $\|\cdot\|_V$. Some applications require to compute certain values (such as point values, average values, line integrals, fluxes through surfaces, ...). These values are described by linear functionals $b : V \to \mathbb{R}$, where $V = H_0^1(\Omega)$. We want to design a method such that the error in this goal, i.e.,

$$b(u) - b(u_h)$$

is small. To this end let $f : V \to \mathbb{R}$ be the (linar and continuous) right hand and

$$A(u, v) := \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x \quad \forall u, v \in V.$$

Using a finite element space $V_h \subset V$, we consider the following variational problems:

$$
\begin{array}{lllll}
\text{(a)} & \text{Find} & u \in V & \text{s.t.} & A(u,v) = f(v), & \forall v \in V \\
\text{(b)} & \text{Find} & u_h \in V_h & \text{s.t.} & A(u_h,v_h) = f(v_h), & \forall v_h \in V_h \\
\text{(c)} & \text{Find} & w \in V & \text{s.t.} & A(v,w) = b(v), & \forall v \in V \\
\text{(d)} & \text{Find} & w_h \in V_h & \text{s.t.} & A(v_h,w_h) = b(v_h), & \forall v_h \in V_h
\end{array}
$$

So the primal problem (continuous + discrete) is considered with f whereas the dual problem (continuous + discrete) is considered with b. Show that

$$
|b(u) - b(u_h)| \leq \eta^1(u_h)\eta^2(w_h),
$$

where η^1 and η^2 are reliable error estimator for problem (a) and (c), respectively.

Remark: A good heuristic is the following (unfortunately, not correct) estimate

$$
b(u - u_h) \leq \sum_T \eta_T^1(u_h)\, \eta_T^2(w_h),
$$

where η_T^1 and η_T^2 are the local contributions of the the estimators η^1 and η^2. The last step would require a local reliability estimate. But, this is not true.

Example 7.3

Implement an error estimator based on the above (heuristic) bound, thus use the product $\eta_T^1(u_h, f)\, \eta_T^2(w_h, b)$ as an error estimator (and for the marking process) for the following problem: $\Omega = (0,1)^2$, $A(u,v) = \int_\Omega \nabla u \nabla v$ and $f(v) = \int_{[0.2,0.3]\times[0.45,0.55]} 100v$, and the functionals

1. $b_1(u) = 100 \int_{[0.7,0.8]\times[0.45,0.55]} u$
2. $b_2(u) = 10 \int_{0.75\times[0.45,0.55]} u$
3. $b_3(u) = u(0.75, 0.5)$

Present convergence plots for the error in the goal functional b_i. Compare your results using the above estimator and using an error estimator for the primal error $\|u - u_h\|_V$ (use the estimator from the last exercise). You can use the script adaptive.py as starting point.

Example 7.4 (Mass lumping for boundary layers)

We want to solve the problem: Find u such that

$$
\begin{aligned}
-\varepsilon \Delta u + u &= 1 \qquad \text{in } \Omega, \\
u &= 0 \quad \text{on } \partial\Omega.
\end{aligned}
$$

For $\varepsilon > 0$, the variational framework that we developed in the lecture shows, that the (weak) solution is in the space $H_0^1(\Omega)$, and the second order differential operator "allowed" us to enforce boundary conditions. For the case $\varepsilon = 0$, we are not allowed to ask for boundary conditions anymore (the weak formulation of the above problem is well defined in $L^2(\Omega)$), and the solution of the above problem is $u = 1$. For the case where $\varepsilon \to 0$ the solution converges to the constant 1, but since we have $u = 0$ on $\partial\Omega$ there appears a so called boundary layer with thickness ε. A sketch in one dimension is given in the left plot

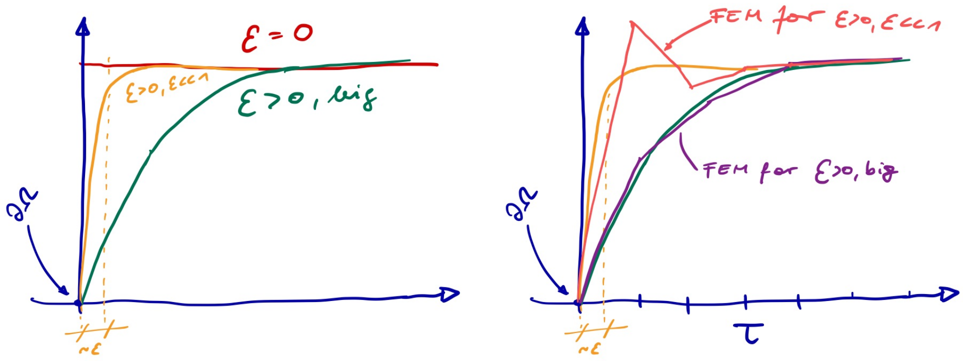

Figure 1: boundary layer for vanishing ε

of Figure 1. In the discrete setting, a boundary layer can only be resolved in a proper way, if the mesh size is of order ε, otherwise the method creates oscillations as the FEM tries to approximate the big gradient of the solution in the boundary layer, see right plot of Figure 1.

1. Derive the weak formulation of the problem

2. Let $\Omega = (0,1)^2$ and use a structured mesh (use `bndlayer.py` as starting point). Use NGSolve to approximate the solution with linear finite elements and $\varepsilon = 10^{-i}$ with $i = 0, \ldots, 6$. Evaluate the solutions (and plot it in one figure) along the line $L = \{(x, 0.5) : 0 \leq x \leq 1\}$. For this define a one-dimensional grid on L (for example with numpy) and evaluate the solution at those points via `gfu(mesh(x,y))` and plot the resulting values (matplotlib).

3. We want to apply mass lumping for this method. To this end define a new quadrature rule in NGSolve using

   ```
   IR = IntegrationRule(pnts=[...],wights=[...])
   ```

   where `pnts` and `weights` are lists with the quadrature points and the corresponding weights on the reference element $\hat{T}$ with the vertices $\{(0,0), (0,1), (1,0)\}$. To use this integration rule for the bilinear forms change the integration symbol to

   ```
   ... * dx(intrules = {TRIG: IR}).
   ```

   Define `IR` that it is exact for linear polynomials (as in the lecture). Then

   - Define a bilinear form for the massmatrix M where

     $$M_{ij} = \int_\Omega \varphi_i \varphi_j \, \mathrm{d}x,$$

     and check if the corresponding mass-lumped matrix M_h is diagonal.

   - Apply mass lumping to the bilinear form of the above problem and repeat the calculations (including the plots) from point 2. What to you observe?