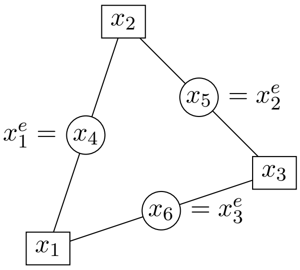
Institute for Analysis and Scientific Computing
TU Wien
Philip L. Lederer

Numerical methods for partial differential equations

Exercise 6 – 28 April 2020

Example 6.1

Let $\mathcal{T}$ be an admissible, quasi-uniform triangulation of a bounded polygonal domain $\Omega \subset \mathbb{R}^2$. Let I_T be the interpolation operator to the Lagrange finite element $(T, \mathcal{P}^2(T), \Psi_T)$ where $\Psi_T(v) := v(x_i)$ where x_i are the vertices and edge midpoints of the triangle T.



We define the quadrature rule

$$Q_T(f) := \frac{|T|}{3} \left(f(x_1^e) + f(x_2^e) + f(x_3^e) \right)$$

where $x_i^e,\ i = 1, 2, 3$ are the edge midpoints of the triangle. Show that $Q_T(v) = \int_T v \,\mathrm{d}x$ for all $v \in \mathcal{P}^2(T)$.

Hint: Show the result first on $\hat{T}$.

Example 6.2

We continue with the setting from the previous example. Let $Q_h(v) := \sum_{T \in \mathcal{T}} Q_T(v)$ and $v \in H^3(\Omega)$. Show that

$$\left| \int_\Omega v \,\mathrm{d}x - Q_h(v) \right| \preceq h^3 |v|_{H^3(\Omega)}$$

Hint: Try to involve the interpolation estimates from Theorem 69 (with quasi uniformity).

Example 6.3

Let $\Omega := (0, 1)$ and $u \in H_0^1(\Omega)$ be the weak solution of

$$\begin{aligned} -u'' &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{auf } \partial\Omega \end{aligned}$$

to a given $f \in L^2(\Omega)$. For $n \in \mathbb{N}$ let

$$\mathcal{T}_h := \{[x_{i-1}, x_i] \mid 0 = x_0 < \ldots < x_n = 1, i = 1, \ldots, n\}$$

be an admissible triangulation of Ω, $V_h^k := \{v \in C^0(\Omega) \mid v_{|T} \in \mathcal{P}^k(T) \quad \text{for} \quad T \in \mathcal{T}_h\}$ und $V_{h,0}^k = V_h^k \cap \{u \in C^0 | u|_{\partial\Omega} = 0\}$. Let $u_h \in V_{h,0}^k$ be the Galerkin approximation to u, $e_h := u - u_h$ and $I_h^L : H_0^1 \to V_{h,0}^k$ the Lagrange interpolator.
Prove that

$$\|e_h'\|_{L^2(\Omega)}^2 \quad \leq \quad \sum_{T \in \mathcal{T}_h} \left(f + u_h'', e_h - I_h e_h\right)_{L^2(T)}$$

and conclude with the help of interpolation estimates the a posteriori estimate

$$|u - u_h|_{H^1(\Omega)}^2 \quad \leq \quad c \sum_{T \in \mathcal{T}_h} h_T^2 \|f + u_h''\|_{L^2(T)}^2,$$

where the constant $c > 0$ does not depend on u.
Hint:

- First, prove that for $v_h \in V_{h,0}^k$ there holds: $\|e_h'\|_{L^2(\Omega)}^2 = (e_h', e_h' - v_h')_{L^2(\Omega)}$.
- Since $d = 1$, there holds the estimate $||v - I_h v||_{L^2(T)} \leq c h_T |v|_{H^1}$.

Example 6.4

On $\Omega = (-1, 1)^2$ we want to solve the stationary diffusion problem

$$\begin{cases} -\operatorname{div}(\alpha \nabla u) = f & \text{in } \Omega, \\ \qquad\qquad u = u_D & \text{on } \partial\Omega. \end{cases}$$

α and f are piecewise constants:

$$\alpha = \begin{cases} 1 & \text{if } \|x\|_2 < R, \\ 2\ln(R) & \text{if } \|x\|_2 \geq R, \end{cases} \qquad f = \begin{cases} 1 & \text{if } \|x\|_2 < R, \\ 0 & \text{if } \|x\|_2 \geq R, \end{cases}$$

with $R = 1/2$. We choose u_D so that the solution is

$$u = \begin{cases} \dfrac{1}{8} - \dfrac{\|x\|_2^2}{4} & \text{if } \|x\|_2 < R \\ \dfrac{\ln(x^2 + y^2)}{32\ln(R)} & \text{if } \|x\|_2 \geq R. \end{cases}$$

We are interested in the error $\|\alpha^{\frac{1}{2}} \nabla(u - u_h)\|_{L^2(\Omega)} = \sqrt{A(u - u_h, u - u_h)}$ of a numerical solution to the PDE.
In `simple_adaptive.py` the FEM solution to this problem with piecewise cubic polynomials is shown using an adaptive algorithm where the error estimator is simply $\eta_T = \|\alpha^{\frac{1}{2}} \nabla(u - u_h)\|_{L^2(T)}$ and $\eta_h = \left(\sum_{T \in \mathcal{T}} \eta_T^2\right)^{\frac{1}{2}}$. Note that we can only do this because we know u. In the script adaptive refinements are carried out until $\dim V_h > 10000$. In every step the elements with the largest error contributions which add up to roughly 10% of the total error are marked for refinement.

1. Run the script and compare the accuracy (in the L^2 norm) that is obtained using uniform refinements and adaptive refinements. To use uniform refinements remove the lines

```
for el in mesh.Elements():
    mesh.SetRefinementFlag(el, marks[el.nr])
```

What do you observe (where is the refeinement located)? Try to explain this!

2. Next, we consider an error estimator that does not rely on a known solution. For this we again use the Sobolev space

$$H(\mathrm{div}, \Omega) := \{\sigma \in [L^2(\Omega)]^d : \mathrm{div}(\sigma) \in L^2(\Omega)\},$$

where $\mathrm{div}(\sigma)$ is the weak-divergence. We can consider the gradient recovery (GR) error estimator. The idea is as follows: Let Σ_h be an $H(\mathrm{div}, \Omega)$-conforming finite element space of degree $k-1$ (here $k-1=2$).

We interpolate the flux $\alpha \nabla u_h$ into this space, $\sigma_h = I_h^{\Sigma}(\alpha \nabla u_h) \in \Sigma_h$ to obtain two approximations of the flux:

- $\alpha \nabla u_h \notin \Sigma_h$ from u_h the discrete solution of the diffusion problem
- $\sigma_h \in \Sigma_h$ from the interpolation into Σ_h

The difference between these two approximations is the indicator for the error.

$$\eta_T^{GR} = \|\alpha^{-\frac{1}{2}}(\alpha \nabla u_h - \sigma_h)\|_{L^2(T)}.$$

Implement this estimator based on the snippet below

```
# finite element space and gridfunction to represent the heatflux:
space_flux = HDiv(mesh, order=2)
gf_flux = GridFunction(space_flux, "flux")
...
def CalcError():
    space_flux.Update()
    gf_flux.Update()

    # interpolate finite element flux into H(div) space:
    gf_flux.Set (flux)

    # Gradient-recovery error estimator
    err = 1/alpha*(flux-gf_flux)*(flux-gf_flux)
    elerr = Integrate (err, mesh, VOL, element_wise=True)
    ...
    print ("estimated error = ", sqrt(sumerr))
```

and compare the performance of the estimators. Based on this experiment is the estimator efficient and reliable? If yes, give a rough estimate of the constants.