Sheet 9

Discussion of the sheet: Tue., 23.05.2023

---

1. (Prager-Synge theorem) Let $\Omega \subset \mathbb{R}^d$. Consider the solution $u \in H_0^1(\Omega)$ of the Poisson equation

$$-\Delta u = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial\Omega.$$

Let σ be a vector field satisfying $\operatorname{div}\sigma = -f$. Show that for any $v \in H_0^1(\Omega)$ there holds

$$(\nabla(u - v), \nabla u - \sigma)_{L^2(\Omega)} = 0$$

and consequently

$$\|\nabla(u - v)\|_{L^2(\Omega)}^2 + \|\nabla u - \sigma\|_{L^2(\Omega)}^2 = \|\nabla v - \sigma\|_{L^2(\Omega)}^2.$$

Note that the idea stems from a rewriting of the Poisson problem as $\nabla u = \sigma$, $\operatorname{div}\sigma = -f$, where the new variable σ can be seen as the flux of u. Prager-Synge theorem can then be used to define an error estimator by choosing $v = u_h$ and constructing σ as a FEM approximation of the flux (i.e. post-process ∇u_h to σ_h so that $\nabla \cdot \sigma_h = f$).

2. Go to gitlab.tuwien.ac.at/asc/praetorius/mooafem and download the MATLAB package MooAFEM (an object orientated adaptive FEM code). Run 'setup' in your Matlab to install it.

Go to the examples folder and run the codes 'adaptiveFEM.m' and 'convergenceRates.m' and explain: what is the PDE? on which domain? what kind of FEM space is used? how are the adaptive steps (error estimator, marking, refinement) implemented? where does most of the refinement take place? what do we see from the convergence rates?

3. Consider the advection equation. Determine the numerical domain of dependence for the forward time/backward space method. Argue that the CFL-condition is $\frac{k}{h} \leq 1$.

4. Write a program to solve the heat equation

$$\partial_t u - \partial_{xx} u = f \quad \text{in } (0,1) \times (0,T)$$
$$u(0,t) = u(1,t) = 0 \quad \forall t \in (0,T)$$
$$u(x,0) = u_0(x)$$

by using a semi-discretization by classical FEM in space and discretize the resulting ODE with the implicit Euler method. Take piecewise linear Lagrangian finite elements of order 1 on an uniform mesh of mesh.width h in space and take uniform time-steps of length k.

Test your program with the exact solution $u(x,t) = e^{-t}x(1-x)$ at $T = 1$ using $k = ch$ with different constants $c > 0$. For the error you can take the error measure $\sqrt{h}\sqrt{\sum_j |u(x_j,T) - u_j^T|^2}$, where you should see convergence of order h.

5. Modify your program from the previous example by using the explicit Euler method instead of the implicit Euler method. Take two choices of time steps k:

$$k = \frac{2.01}{\lambda_{\text{max}}}, \qquad k = \frac{1.99}{\lambda_{\text{max}}},$$

where λ_{max} is the largest eigenvalue of $M^{-1}A$ (can e.g. be computed in Matlab using `eigs`). Plot your computed solutions, the error versus h, and λ_{max} versus h.