Numerical Approximation of PDEs

Spring Semester 2025

Lecturer: Prof. Annalisa Buffa Assistant: Mohamed Ben Abdelouahab

Session 08: April 17, 2025

Exercise 1. [Implementation of SUPG scheme (Exercise 7.1 continued)]
Let Ω be a bounded domain in $\mathbb{R}^2$. We consider the following advection-diffusion-reaction problem for $u : \Omega \to \mathbb{R}$:

$$\begin{aligned} -\Delta u + \operatorname{div}(\boldsymbol{\beta} u) + u &= f && \text{in } \Omega \\ u &= \phi && \text{on } \Gamma_D \\ \nabla u \cdot \mathbf{n} &= u\boldsymbol{\beta} \cdot \mathbf{n} && \text{on } \Gamma_N \end{aligned} \;, \tag{1}$$

where $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\Gamma_D \cap \Gamma_N = \emptyset$ is a partition of the boundary and where and $\boldsymbol{\beta} = \boldsymbol{\beta}(\mathbf{x}) \in \mathbb{R}^2$ is the flow vector field and $f \in L^2(\Omega)$ the source term.
We assume that $\Omega = [0,1]^2$, $\boldsymbol{\beta} = (-10^3, -10^3)^T$ and $\Gamma_D = \partial\Omega$. We also define ϕ as:

$$\begin{cases} \phi = 1 & \text{on } \Gamma = \{(x,y) \mid 0 \le x \le 1 \text{ and } y = 0, \text{ or } x = 0 \text{ and } 0 \le y \le 1\} \\ \phi = 0 & \text{on } \Gamma_D \setminus \Gamma. \end{cases}$$

We take $f = 10$. To acquire a consistent scheme in this case, we implement the SUPG stabilisation as discussed in class. You will be able to re-use a lot from the implementation of exercise 1 in session 7. The `integrate_template.py` script contains a function creating an iterator for assembling the SUPG right hand side offset

$$+ \sum_{K \in \tau_h} \frac{\gamma h_K}{\|\boldsymbol{\beta}\|_{L^\infty(K)}} \int_K f \boldsymbol{\beta} \cdot \nabla \phi_i$$

Finalise the script and test the SUPG solution for $f = 10$ against the non-stabilised one for the set of parameters $h \in \{0.1, 0.05, 0.025\}$ and $\gamma \in \{0.1, 1, 5\}$.

Exercise 2. Let $T > 0$ and $\Omega \subseteq \mathbb{R}^n$. Prove the Cauchy-Schwarz inequality for the space $L^2(0,T;L^2(\Omega))$:

$$\int_0^T \int_\Omega |uv| dx dt \le \|u\|_{L^2(0,T;L^2(\Omega))} \|v\|_{L^2(0,T;L^2(\Omega))}.$$

Exercise 3. Consider the heat equation over a bounded Lipschitz domain $\Omega \subseteq \mathbb{R}^n$ with homogeneous Neumann boundary conditions:

$$\begin{aligned} -\partial_t u + \operatorname{div}(\nabla u) &= 0 && \text{over } \Omega \\ \nabla u \cdot \mathbf{n} &= 0 && \text{along } \partial\Omega \\ u &= u_0 && \text{at } t = 0, \end{aligned}$$

where $u_0 : \Omega \to \mathbb{R}$ is the initial heat distribution.

- Show that the integral

$$t \mapsto \int_\Omega u(x,t)dx$$

stays constant at all times $t > 0$. Give a physical interpretation of that observation.

- In the case that $\Omega = (a,b)$ is a one-dimensional interval, show that the constant heat distribution is the only stationary heat distribution.

Exercise 4. Let $\Omega = [0,1]^2$ and $T = 10$. Consider the heat equation

$$\begin{aligned} \partial_t u - \operatorname{div}\nabla u &= f \text{ over } \Omega \times (0,T) \\ u_{|\Gamma_D} &= 0 \text{ along } \partial\Omega \\ u(x,0) &= \frac{1}{2}\sin(\pi x_1)\sin(\pi x_2) \text{ at } t = 0. \end{aligned}$$

Here,

$$f(x_1,x_2) = \sin(2\pi t + \pi/2) * \left((1 - \frac{t}{T}) f_0(x_1,x_2) + \frac{t}{T} * f_1(x_1,x_2) \right),$$

$$f_0(x_1,x_2) = 50\sin(2\pi x_1)\sin(2\pi x_2), \quad f_1(x_1,x_2) = 100\sin(4\pi x_1)\sin(4\pi x_2).$$

Given a triangulation $\mathcal{T}$ and the corresponding piecewise linear finite element space V_h, we consider the semidiscrete problem as discussed in the lecture:

$$M\partial_t U + AU = F$$

- What are the matrices A and M and the right-hand side F in that context?

- Isolate the time derivative $\partial_t U$ and state the explicit and implicit Euler method with some time step size $\Delta t > 0$. Discuss what is the major computational difficulty when using those methods.

- On Moodle, you will find a template `parabolic_template.py` which implements, quite generally, the θ-method, of which explicit (or forward) Euler is just a special case. Finalise the `thetamethod` by filling in the blanks. Then, use the `euler_template.py` script to test for which combinations of the mesh size h and timestep Δt the two methods work. The script will produce an animation that you can use to verify that your approach works.