Assignment 06

Software atelier: PDE Lab

April 4, 2020

Deadline: April 9th
Everybody has to hand in his or her own assignment sheet.
Code is part of the assignment and has to be submitted. Submissions in Latex are not mandatory but highly encouraged.

Important

Refer to the code developed in class.

Exercise 1 - FD for problems with non constant coefficients (5 Pts)

Consider a uniform grid on the interval $(0, 1)$ with m elements $K_i = (x_i, x_{i+1})$ of size h and $m + 1$ nodes. Moreover, consider the following diffusion operator

$$Du = -(au')' \tag{1}$$

where the coefficient a is element-wise constant and takes value a_i over the interval K_i.
The discretization of (1) with a centered FD method reads

$$D_h u_i = \frac{1}{h^2}\big(- a_{i-1}u_{i-1} + (a_{i-1} + a_i)u_i - a_i u_{i+1}\big). \tag{2}$$

Observe that setting $a_{i-1} = a_i = 1$ we obtain the standard stencil for the second derivative.

1. Write the stiffness matrix arising from the discretization (2) assuming that for the Neumann boundary conditions are imposed by means of the ghost-node technique.

   Not mandatory: bonus points if the assembly is performed on non uniform grids following the formula

$$D_h u_i = \frac{2}{h_{i-1} + h_i}\Big(- \frac{a_{i-1}}{h_{i-1}}u_{i-1} + (\frac{a_{i-1}}{h_{i-1}} + \frac{a_i}{h_i})u_i - \frac{a_i}{h_i}u_{i+1}\Big). \tag{3}$$

2. Write two Matlab functions for the assembly of such a matrix using the standard loops (`V0`) and the vectorized version (`V2`). The two functions should take the grid and the array $a = [a_1; a_2; \dots a_m]$ as input arguments.

Not mandatory: bonus points if the assembly is performed on a non-uniform grid following the formula (3).

Consider the differential problem

$$\begin{cases} -(au')' + \sigma u &= 1 \quad \text{on } (0,1) \\ u(0) &= 0 \\ u(1) &= 0 \end{cases} \tag{4}$$

with $\sigma = 10$ and

$$a(x) = \begin{cases} 1 & \text{if } x < 0.25 \text{ or } x > 0.75 \\ 100 & \text{if } 0.25 < x < 0.75 \end{cases} .$$

3. Write the finite difference discretization of this problem with $m = 4$.
4. Report the plot of the solution with $m = 4000$. Define this solution as u_{ref} over a grid τ_{ref}.
5. Report the L^∞ error for $m = 8, 16, 32, 64, 128, 256, 512$ and evaluate the order convergence. To evaluate the error, interpolate the solution u_m over the grid τ_{ref} using the Matlab command `interp1`.

Exercise 2 - FD for diffusion problems in 2D (5 Pts)

Consider the Laplace equation

$$-\Delta u \;=\; 1 \quad \text{in } \Omega = (0,1)^2. \tag{5}$$

Define the side with $y = 0$ as Γ_b, the side with $x = 1$ as Γ_r, the side with $y = 1$ as Γ_t, the side with $x = 0$ as Γ_ℓ and consider the finite difference method as seen during class.

1. Consider a purely Neumann problem with the conditions $\nabla u \cdot \boldsymbol{n} = J_k \quad \text{on } \Gamma_k$, where $k = \{b, r, t, \ell\}$.
   (a) write the finite difference method for the four nodes at the corners of the domain,
   (b) modify the matlab code seen in class, in order to implement non-homogeneous Neumann boundary conditions,
   (c) report the solution for the problem with the following boundary conditions

$$\begin{aligned} u &= 0 \quad \text{on } \Gamma_b \text{ and } \Gamma_r \\ \nabla u \cdot \boldsymbol{n} &= 1 \quad \text{on } \Gamma_t \\ \nabla u \cdot \boldsymbol{n} &= 2 \quad \text{on } \Gamma_\ell \end{aligned} \tag{6}$$

   and setting $m = 20$.
2. Consider a problem with the Dirichlet conditions which are coordinate dependent, e.g. $u(x,0) = g_b(x)$ on Γ_b.

(a) Modify the Matlab code seen in class, in order to implement such boundary conditions. The class `BoundaryConditions2D` should store four functions (or function handlers) and the SetBC method of the Diffusion problem class has to be modified accordingly.

(b) report the solution for the problem with the following boundary conditions

$$\begin{aligned} u &= 0 && \text{on } \Gamma_\ell \text{ and } \Gamma_r \\ u &= x(1-x) && \text{on } \Gamma_b \text{ and } \Gamma_t \end{aligned} \tag{7}$$

and setting $m = 20$.