PW #1: advection-diffusion equations

The aim of these first exercises consists in solving partial differential equations (PDEs) of advection-diffusion type:

$$\partial_t u + \boldsymbol{a} \cdot \nabla u - \varepsilon \Delta u = f, \tag{1}$$

where the unknown is $u : [0, \mathscr{T}] \times \Omega \longrightarrow \mathbb{R}^d$, $d \in \{1, 2\}$. Ω denotes a **smooth bounded domain** in $\mathbb{R}^d$, $\boldsymbol{a}$ is a **given vector field**, ε is a **positive number** and f is a **given source term** assumed to be continuous.

*This work is divided into three parts: the two first parts deal with the 1D case (depending on whether the time variable is taking into account or not) handled by means of a **Finite-Difference Method** (FDM), while the last part concerns the resolution in 2D of* (1) *with a **Finite-Volume Method** (FVM).*

1 The 1D steady case

We first consider the problem where u does not depend on the time variable t, which leads to introduce the 1D elliptic equation:

$$u + a\partial_x u - \varepsilon \partial^2_{xx} u = f. \tag{2a}$$

Here, $a \in \mathbb{R}$ and $\Omega = (0, 1)$. PDE (2a) is supplemented with the following boundary conditions:

$$\begin{cases} u(0) = \alpha, \\ u(1) = \beta, \end{cases} \tag{2b}$$

where $(\alpha, \beta) \in \mathbb{R}^2$. The data set is thus $(a, \varepsilon, f, \alpha, \beta)$.
We discretize Ω with a homogeneous cartesian grid $(x_i)_{0 \le i \le N_x}$, with:

$$N_x \in \mathbb{Z}^*_+, \ \Delta x = \frac{1}{N_x} \ \text{ and } \ x_i = i\Delta x, \ i \in \{0, \ldots, N_x\}.$$

$U_{\Delta x} = (u_0, \ldots, u_{N_x})$ denotes the vector of discrete unknowns, where u_i is assumed to approach the value of the exact solution u at node x_i.
The problem consists in choosing the way to discretize differential operators (∂_x and ∂^2_{xx}). *We propose to compare the following discrete formulas:*

$$\partial_x u(x_i) \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x} \ \text{ and } \ \partial_x u(x_i) \approx \begin{cases} \dfrac{u_i - u_{i-1}}{\Delta x}, & \text{if } a > 0, \\ \dfrac{u_{i+1} - u_i}{\Delta x}, & \text{if } a < 0, \end{cases} \tag{3a}$$

for the first order derivative. The first formula is referred to as "centered" FDM. As for the second order derivative ∂^2_{xx}, we use:

$$-\partial^2_{xx} u(x_i) \approx \frac{-u_{i-1} + 2u_i - u_{i+1}}{\Delta x^2}. \tag{3b}$$

Exercise 1

1. Implement (with SCILAB) the resolution of Syst. (2) using formulas (3). Explain first how to take BC (2b) into account.

2. Compare the two provided schemes in (3a) by means of **convergence rate graphs**, which represent the evolution of L^2 (or L^∞) errors with respect to the mesh size Δx. You may use the following data sets:

   - $\big(a = 0, \varepsilon = 1, f(x) = -12x^2 + 12x - 2, \alpha = 0, \beta = 0\big)$;
   - $\big(a = -\frac{8}{3}, \varepsilon = 1, f(x) = 0, \alpha = e^2, \beta = e^{-1}\big)$;
   - $\big(a = \frac{1}{25}, \varepsilon = \frac{1}{2000}, f(x) = \frac{-1}{e^{100}-1}, \alpha = 0, \beta = 1\big)$.

   The last example may allow to emphasize the numerical difficulties encountered when tackling problems with solutions varying in a narrow area of the domain.

3. With the latter data set, show the solutions for $N_x \in \{20, 30, 40, 50\}$. What do you remark? Try to account for this phenomenon. To do so, rewrite the DFM scheme as a linear system $\mathbf{A}U_{\Delta x} = F$.

2 Time-dependent 1D case

The steady case being implemented, we go back to Eq. (1) with $d = 1$, that is:

$$\partial_t u + a\partial_x u - \varepsilon\partial^2_{xx} u = f, \tag{4a}$$

supplemented with initial and boundary conditions:

$$\begin{cases} u(0,x) = u_0(x), \\ u(t,0) = \alpha, \\ u(t,1) = \beta. \end{cases} \tag{4b}$$

As previously, α and β are data in $\mathbb{R}$ and u_0 is a given function. We introduce a time discretization of the interval $[0, \mathscr{T}]$ with $\mathscr{T} > 0$:

$$t^n = n\Delta t,\ n \in \{0, \ldots, N_t\},\ N_t \in \mathbb{Z}^*_+ \ \text{and}\ \Delta t = \frac{\mathscr{T}}{N_t},$$

so that the discrete solution is denoted by $U^n_{\Delta x} = (u^n_0, \ldots, u^n_{N_x})$.
This PDE requires to choose a discrete time operator to discretize ∂_t. We present in the sequel several methods:

- (Euler "forward") We discretize the time derivative with a first order approximation:

$$\partial_t u(t^n, x_i) \approx \frac{u^{n+1}_i - u^n_i}{\Delta t}. \tag{5a}$$

- (Euler "backward") We still use a first order approximation:

$$\partial_t u(t^n, x_i) \approx \frac{u_i^n - u_i^{n-1}}{\Delta t}. \tag{5b}$$

- ("leapfrog") We can also use a two-step formula:

$$\partial_t u(t^n, x_i) \approx \frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t}. \tag{5c}$$

- ("method of characteristics") Noting $\tilde{u}(t,x) = u(t, x+at)$ and differentiating $\tilde{u}$, we easily come to the conclusion that $\partial_t \tilde{u}(t,x) = [\partial_t u + a\partial_x u](t, x+at)$, $\partial^2_{xx}\tilde{u}(t,x) = \partial^2_{xx} u(t, x+at)$ and $\tilde{u}(0,x) = u_0(x)$. Thus, $\tilde{u}$ is a solution to:

$$\begin{cases} \partial_t \tilde{u}(t,x) - \varepsilon \partial^2_{xx}\tilde{u}(t,x) = f(t, x+at), \\ \tilde{u}(0,x) = u_0(x), \end{cases} \tag{5d}$$

  which is the heat equation. This method thus consists in discretizing Eq. (5d) then interpolating the solution to recover $u(t,x) = \tilde{u}(t, x-at)$. Unfortunately, an explicit discretization of the heat equation suffers a severe stability condition, that is:

$$\frac{2\varepsilon \Delta t}{\Delta x^2} \le 1.$$

  That is why we prefer implicit schemes, such as the θ-scheme (which is unconditionally stable for $\theta \ge \frac{1}{2}$):

$$\frac{\tilde{u}_i^{n+1} - \tilde{u}_i^n}{\Delta t} = \theta\varepsilon \frac{\tilde{u}_{i-1}^{n+1} - 2\tilde{u}_i^{n+1} + \tilde{u}_{i+1}^{n+1}}{\Delta x^2} + (1-\theta)\varepsilon \frac{\tilde{u}_{i-1}^{n} - 2\tilde{u}_i^{n} + \tilde{u}_{i+1}^{n}}{\Delta x^2} + \theta \hat{f}_i^{n+1} + (1-\theta)\hat{f}_i^n,$$

  where $\hat{f}_i^n = f(t^n, x_i + at^n)$.

Exercise 2

1. Implement successively the resolution of Syst. (4) based on (5a) then on (5b). Show how the previous code may be re-utilized.

2. Adapt your code to unsteady boundary conditions $u(t,0) = \alpha(t)$ and $u(t,1) = \beta(t)$.

3. Let us consider the case $a = 1$, $\varepsilon = 1$, $\alpha(t) = (1+t)^2$ and $\beta(t) = 2 + t^2$, for which the exact solution is $u(t,x) = 1 + (x-t)^2 + 2t$. Carry out simulations for different time steps and mesh sizes.

4. For stability reasons, Δt and Δx must check the inequality $2\varepsilon\Delta t \le \Delta x^2$. Implement the time resolution based on (5c). What to conclude about stability?

3 Steady 2D case

We now consider the following 2D problem:

$$\mathrm{div}\big(u(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x})\big) - \varepsilon\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), \text{ in } \Omega, \tag{6a}$$

where u is the scalar unknown, $\boldsymbol{a}$ a vector field, $\varepsilon \in \mathbb{R}$ and $\Omega = (0,1)\times(0,1)$. PDE (6a) is supplemented with the following boundary conditions:

$$u = 0, \text{ on } \partial\Omega. \tag{6b}$$

We consider that a mesh $\mathcal{T}$ of the domain Ω is given. Such meshes and the description of the storage format are available in the file `maillages2D.sci`. A point $\boldsymbol{x}_K$ of the domain is associated to each control volume $K \in \mathcal{T}$ and we assume the standard orthogonality condition: $\overrightarrow{\boldsymbol{x}_K\boldsymbol{x}_L}\perp\sigma$ for all edges $\sigma = K|L$ of the mesh.

The finite volume scheme is obtained by integrating the equation (6a) on a control volume K. We propose the following discrete formulas (for inner cells):

- Diffusion term:

$$\int_K \Delta u(\boldsymbol{x})d\boldsymbol{x} = \sum_{\sigma\in\mathcal{E}_K}\int_\sigma \nabla u(\gamma)\cdot\boldsymbol{n}_{K\sigma}d\gamma \approx \sum_{\sigma\in\mathcal{E}_K,\sigma=K|L}\tau_{KL}(u_K-u_L), \tag{7a}$$

  where $\mathcal{E}_K$ stands for the set of edges of the control volume K, $\boldsymbol{n}_{K\sigma}$ is the outward unit normal vector to σ and τ_{KL} is defined by $\tau_{KL} = \dfrac{|\sigma|}{\|\overrightarrow{\boldsymbol{x}_K\boldsymbol{x}_L}\|}$. This scheme is classically called **VF4** (the stencil contains 4 points when the mesh is triangular) or **TPFA** (two points flux approximation).

- Advective term:

$$\int_K \mathrm{div}\big(u(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x})\big)d\boldsymbol{x} = \sum_{\sigma\in\mathcal{E}_K}\int_\sigma u(\gamma)\boldsymbol{a}(\gamma)\cdot\boldsymbol{n}_{K\sigma}d\gamma \approx \sum_{\sigma\in\mathcal{E}_K,\sigma=K|L} V_{K\sigma}u_\sigma, \tag{7b}$$

  where $V_{K\sigma} = |\sigma|\boldsymbol{a}(\boldsymbol{x}_\sigma)\cdot\boldsymbol{n}_{K\sigma}$, $\boldsymbol{x}_\sigma$ being the center of the edge σ and $u_\sigma = \begin{cases} u_K, & \text{if } V_{K\sigma} > 0 \\ u_L, & \text{if } V_{K\sigma} < 0\end{cases}$. This scheme is called the **upwind scheme**.

Exercise 3

1. Implement (with SCILAB) the resolution of Syst. (6) using formulas (7) which must be adapted for boundary cells.
2. Represent the evolution of L^2 and H^1 errors with respect to the mesh size Δx. Some test cases and analytical solutions are provided in the file `donnees2D.sci`.

4 Time dependent 2D case

Finally, using the same notations as in the previous part, we consider the unsteady 2D problem :

$$\partial_t u(t,\boldsymbol{x}) + \boldsymbol{a}(\boldsymbol{x}) \cdot \nabla u(t,\boldsymbol{x}) - \varepsilon \Delta u(t,\boldsymbol{x}) = 0. \tag{8a}$$

PDE (8a) is supplemented with the following initial and boundary conditions:

$$\begin{cases} u(0,\boldsymbol{x}) = u_0(\boldsymbol{x}),\ \boldsymbol{x} \in \Omega, \\ u = 0,\ \text{on}\ \partial\Omega, \end{cases} \tag{8b}$$

where u_0 is a given function.
We choose to write the time discretization of Eq. (8a) as follows:

$$\begin{cases} \dfrac{u^{n+1} - u^n}{\Delta t} + \boldsymbol{a}(\boldsymbol{x}) \cdot \nabla u^{n+1}(\boldsymbol{x}) - \varepsilon \Delta u^{n+1}(\boldsymbol{x}) = 0, \\ u^0(\boldsymbol{x}) = u_0(\boldsymbol{x}). \end{cases} \tag{9}$$

The finite volume scheme is then obtained as in the previous section.

Exercise 4

1. Implement (with SCILAB) the resolution of Syst. (8). The previous code may be re-utilized.

2. Test your programm using the following set of data:

$$\begin{cases} u_0(\boldsymbol{x}) = e^{-500\left[(\boldsymbol{x}_1-0{,}25)^2+(\boldsymbol{x}_2-0{,}5)^2\right]}, \\ \boldsymbol{a}(\boldsymbol{x}) = \big(-10\sin(\pi\boldsymbol{x}_1)\cos(\pi\boldsymbol{x}_2), 10\cos(\pi\boldsymbol{x}_1)\sin(\pi\boldsymbol{x}_2)\big), \\ \varepsilon = 0{,}01. \end{cases}$$

Directions You may deliver your SCILAB codes as well as relevant outputs electronically on January 19, 2011.

sebastian.minjeaud@inria.fr ; yohan.penel@inria.fr