TP 1: jeudi 20/09/2018
Equation de transport à coefficients constants

---

1 Numerical analysis

We focus on the numerical analysis of several schemes thanks to the open source language Python. See Internet for details about it. If you are a beginner in Python, take time to study the following notions in Python: indentation, matrix, vector, the numerical resolution of linear system, ...
We consider $(0,1)$ as the space domain and impose periodic boundary conditions: $\bar{u}(x+1,t) = \bar{u}(x,t)$. The periodicity condition will be taken into account in the numerical schemes by letting $J \in \mathbb{N}^*$, $\Delta x = 1/J$, $x_j = j\Delta x$, $u_j^n = u_{j+J}^n$, for $j = 0,\dots,J$ and $n \in \mathbb{N}$.

We will test two different (also periodic) initial conditions on $(0,1)$:

$$u_0^1(x) = \sin(2\pi x)\,, \qquad \text{and} \qquad u_0^2(x) = \begin{cases} 0\,, & \text{if } 0 \le x < \frac{1}{4} \text{ or } \frac{3}{4} \le x < 1\,, \\ 1\,, & \text{if } \frac{1}{4} \le x < \frac{3}{4}\,. \end{cases}$$

As mentioned in chapter 2, section 2.2.1, the analytical solution is given by $\bar{u}(x,t) = u_0(x-at)$. We will take $a = 1$ in the numerical examples.

Q1. Implement numerically the Lax-Wendroff scheme

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} + a\,\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + \frac{a^2\Delta t}{2}\,\frac{2u_j^n - u_{j-1}^n - u_{j+1}^n}{\Delta x^2} \;=\; 0, \tag{1}$$

the centered explicit scheme

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} + a\,\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \;=\; 0, \tag{2}$$

and the implicit scheme

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} + a\,\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} \;=\; 0. \tag{3}$$

If you are not used to Python, you can download a file implementing these schemes at `http://www.ljll.math.upmc.fr/~despres/BD_fichiers/TP1.py`. By playing with the data, illustrate graphically the stability behavior of each scheme.

Q2. By mimicking the other schemes, implement the *upwind* scheme

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} + a\,\frac{u_j^n - u_{j-1}^n}{\Delta x} \;=\; 0 \tag{4}$$

and check its stability and convergence behaviors.

Q3. By keeping the CFL constant, perform suitable numerical tests to highlight the numerical convergence order of the schemes in term of J for the first initial condition $u_0^1(x)$.

Q4. How is the convergence modified for the second initial condition $u_0^2(x)$?

Q5. Implement the other schemes mentioned in the previous lesson and study their numerical behavior.

Q6. Test a scheme which implements a Monte-Carlo technique

$$u_j^{n+1} = \begin{cases} u_{j-1}^n\,, & 0 \le \text{random}(0,1) < \nu, \\ u_j^n\,, & \nu \le \text{random}(0,1) \le 1. \end{cases}$$

In this case random$(0,1)$ is recomputed for all j. Compare to the Glimm scheme where random$(0,1)$ is the same for all j.

2 Numerical modeling

We consider a population of cells (or human beings) modeled with the age-structured model

$$\begin{cases} \partial_t n + \partial_a n = -d(n), & t > 0, \quad a > 0, \\ n(0,a) = n_0(a) \ge 0, & a > 0, \\ n(t,0) = 0, & t > 0. \end{cases}$$

The age variable is $a \ge 0$. The death parameter is $d(n) \ge 0$. Notice that there is no birth due to the condition $n(t,0) = 0$.

Q1. For $d(n) = -\sigma n$ ($\sigma > 0$), write the analytical solution.

Q2. By modifying the upwind scheme, implement the zero birth condition and the death parameter to simulate this model.

Q3. Check that the population never vanishes for $n_0 \neq 0$ (the extinction paradox).

Q4. Now we change the death parameter to $d(n) = -\sigma\sqrt{n_+}$ and $n_+ = \max(n,0)$. Determine the analytical solution and show that it solves the extinction paradox.

Q5. Implement the new death parameter numerically.

Q6. One can think of adding a birth term

$$n(t,0) = \int_0^\infty b(a)n(t,a)da, \quad t > 0,$$

where $b(a) \ge 0$ is a birth coefficient (which depends of the age a). Implement a reasonable birth coefficient, and discuss the competition between birth and death with numerical simulations.