Dr Björn Stinner Term 2, 2021/2022

MA250 Introduction to PDEs
Assignment 1

Please submit your answers to all questions via Moodle by

Friday 28 January 2022 (week 3), 12 noon (UK time).

Some but possibly not all questions will be marked for credit.

In this assignment we consider problems of the form

$$\begin{aligned} \partial_t u(x,t) + v(u(x,t),x,t)\partial_x u(x,t) &= 0, && (x,t) \in \mathbb{R} \times (0,\infty), \\ u(x,0) &= \Phi(x), && x \in \mathbb{R}. \end{aligned} \tag{1}$$

1. In this question, $v(u,x,t) = ax(t+2)^{-2}$ for some $a > 0$ is independent of u and

$$\Phi(x) = \begin{cases} -\frac{8}{3}\left|x - \frac{1}{2}\right| + 1, & x \in (\frac{1}{8}, \frac{7}{8}), \\ 0, & \text{otherwise.} \end{cases}$$

   (a) Using the method of characteristics,
   find the solution $u(x,t)$ for any point $(x,t) \in \mathbb{R} \times (0,\infty)$ in terms of a.

   (b) Assuming that $a = 6\log(5)$ (natural logarithm),
   specifically compute the value of u in $(x,t) = (\frac{5}{2}, 1)$.

2. We study a simple model for sedimentation[1] governed by the PDE

$$\partial_t u(x,t) + \frac{d}{dx} q(u(x,t)) = 0, \quad (x,t) \in \mathbb{R} \times (0,\infty),$$

   for the particle density $u(x,t)$. For the so-called flux q we assume that

$$q(u) = u(1-u)(1-\beta u) \quad \text{with some number } \beta \in [0,1].$$

   (a) Write the PDE in the form (1) by finding a suitable velocity function $v(u,x,t)$.

   (b) Assume that there is a smooth solution $u(x,t)$ for $\beta = 0$ and $\Phi(x_0) = 1 - \frac{1}{3}x_0$.
   Find the characteristics and an explicit solution formula for any given point (x,t).
   Specifically also compute $u(\frac{5}{2}, 1)$.

   (c) Let now $\beta = \frac{3}{4}$ and consider the initial data given by

$$\Phi(x_0) = \begin{cases} 1, & \text{if } x_0 \le 2, \\ 0, & \text{otherwise.} \end{cases}$$

   Show that there is no characteristic through the point $(\frac{5}{2}, 1)$ that starts in any point $x_0 \in \mathbb{R}$ at time $t = 0$.

PTO

[1] The particles 'drop' in the direction of the x axis.

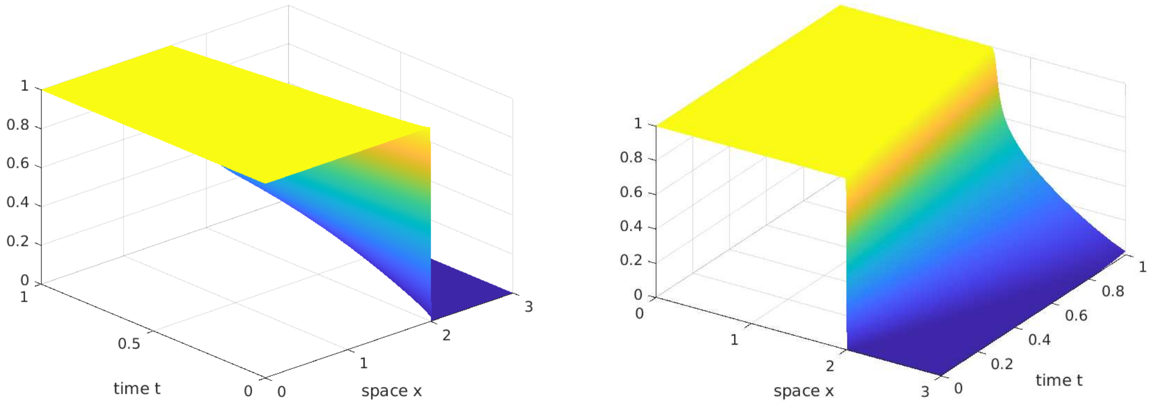

Figure 1: Physically sensible, so-called 'entropy' solution to Question 2(c).

3. Retrieve one of the files (depending on what you prefer to work with)

$$\texttt{upwind.mlx} \text{ (Matlab)} \quad \text{ or } \quad \texttt{upwind.py} \text{ (Python)}.$$

Each file contains code for a numerical solver of (1) on $[0,3] \times [0,1]$ (it also implements some spatial boundary conditions but which we ignore here).

Given a number $N \in \mathbb{N}$, a mesh is defined with a spatial step size of $h = 1/2N)$ and a time step size of $\delta = 0.96 * h/maxvel$ where $maxvel$ has to be greater or equal to the maximal value of v. The mesh points are given by $x_j = jh$, $j = 0, \ldots, 6N$, and $t^{(m)} = m\delta$, $m = 0, \ldots, no\_steps$, where $no\_steps \in \mathbb{N}$ is the largest number such that $no\_steps * \delta \leq 1$. In these mesh points, the function `CL1D_upwind` computes an approximation $U(m+1, j+1) \approx u(x_j, t^{(m)})$ (Matlab) or $U(m,j) \approx u(x_j, t^{(m)})$ (Python) if functions `icfct` for Φ and `velfct` for v are provided. The central code of the function `CL1D_upwind` is an implementation of a finite difference method. It is of first order, i.e., when N is doubled then the error roughly halves[2].

(a) In the code, implement the correct velocity $v(u,x,t)$ from Question 1 by amending the function `velfct`. Submit your code for this function.
The main routine (at the top in Matlab, at the bottom in Python) computes the approximation `unum` of u in the mesh point $(x_{\bar{j}}, t^{(\bar{m})})$ that is closest to $(\frac{5}{2}, 1)$. Use it to compute (and report on) the *relative error*

$$\Big| \frac{u(\frac{5}{2},1) - U(x_{\bar{j}}, t^{(\bar{m})})}{u(\frac{5}{2},1)} \Big|$$

where $u(\frac{5}{2},1)$ is the value that you have computed in Question 1.

(b) The computation in the previous part is done with the discretisation parameter $N = 60$. Increase N until the relative error is $\leq 7\%$.
What is the smallest number N for which this is the case?

(c) (*This part is not for credit*)
Amend your code so that is accounts for the data (v, β, and Φ) in Question 2(c).
Run the code ($N = 60$ should be sufficient) and reproduce Figure 1, which could be a nice picture to finish your submission file.

[2] For illustration purposes this is sufficient, but in real applications one will want to resort to more efficient higher-order schemes.