

PDE python exercise 7

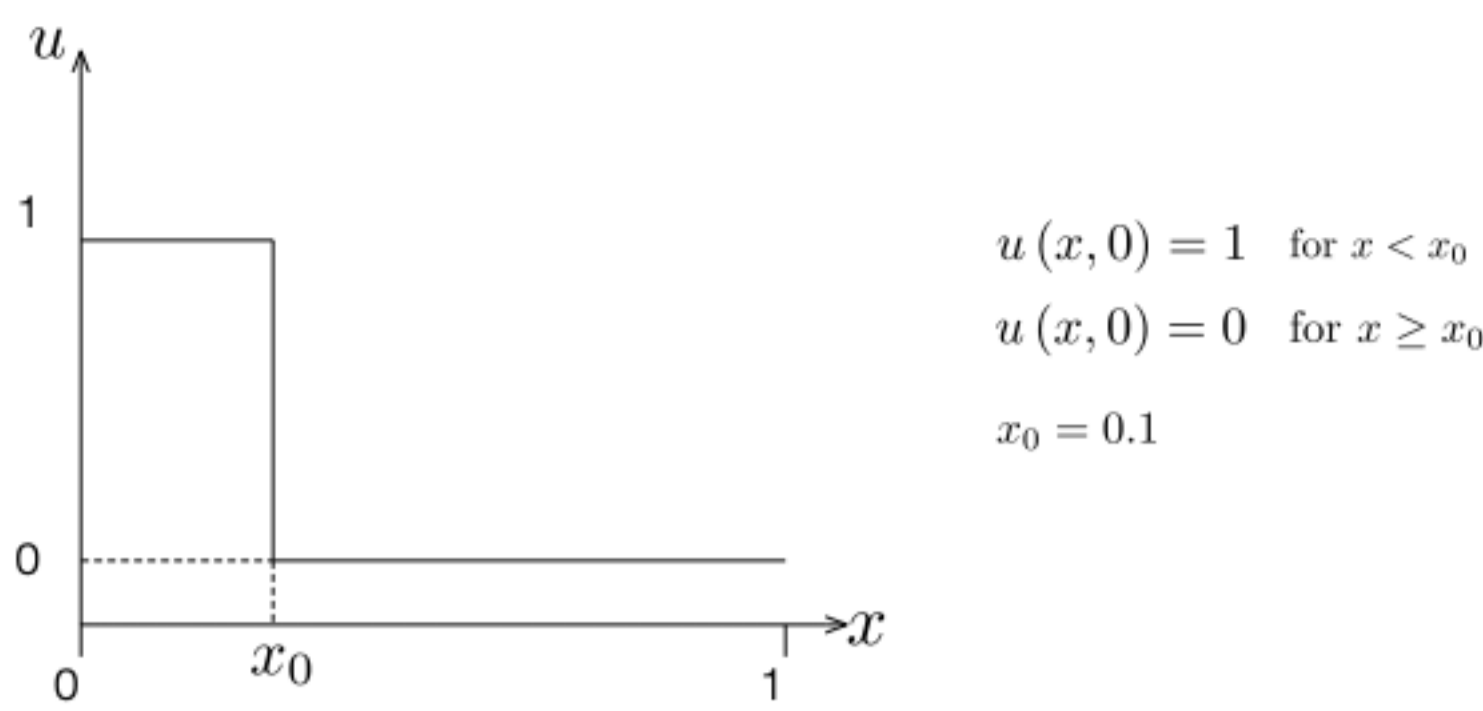
In this exercise we are interested to solve the advection equation defined as:

$$\frac{\partial u}{\partial t} + a_0 \frac{\partial u}{\partial x} = 0$$

where a_0 is a positive constant.

Here are therefore interested by how a pressure wave u is moving through a compressible fluid enclosed into a cylindrical pipe. To this end, we are considering a one dimensional problem (along x) which we will observe through time t .

Our cylindrical pipe is defined to be of length 1 (see figure below) and we are looking at the pressure wave travelling for a time of 0.5. The initial conditions for this PDE are defined below where we give an initial shape to the pressure wave. Here an additional boundary condition is present where the pressure in the fluid u must be equal to 1 when $x = 0$. The wave is assumed to travel with a constant velocity a_0 equal to 1.



In this exercise you have to:

- Express the upwind scheme for the given advection equation
- Make a function which takes in the number of grid points along the x axis and the number of timesteps and return the pressure as a matrix (i.e. as function of time and space). The initial conditions for the pressure wave, its velocity as well as the CFL number have to be computed inside the function.
- Run the simulation for 100 points along the x axis and 100 timesteps. Plot the distribution of the pressure along the pipe for timesteps number 10, 50 and 99. Can you comment on what you can observe?
- Run two more simulations (with the same grid size) but this time with 200 and 51 timesteps (i.e. `time = np.linspace(0.0,0.5,200)` and `time = np.linspace(0.0,0.5,51)`). Plot the distribution of the pressure along the pipe at the end of the simulation. Can you comment on what you can observe?
- Run a final simulation with 50 timesteps and compare it to the one carried out with 51 timesteps. What can you conclude?

Question 1

The upwind scheme makes use of a backward difference for the space derivative:

$$\frac{\partial u}{\partial t} \Big|_j^n \approx \frac{u_j^{n+1} - u_j^n}{\Delta t}$$

and a forward derivative for the time dependency.

$$\frac{\partial u}{\partial x} \Big|_j^n \approx \frac{u_j^n - u_{j-1}^n}{\Delta x}$$

This leads to the numerical scheme:

$$u_j^{n+1} = (1 - C) u_j^n + C u_{j-1}^n$$

where C is the CFL number defined as:

$$C = a_0 \frac{\Delta t}{\Delta x}$$

First, we import our mandatory python modules.

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt
```

```
In [ ]: def compute_u(Nx,Nt):
# We define the velocity of the pressure wave
a0 = 1.0
# We define the space and time arrays
x = np.linspace(0.0,1.0,Nx)
t = np.linspace(0.0,0.5,Nt)
# We compute the timestep and grid space
delta_x = x[1]-x[0]
delta_t = t[1]-t[0]
# We initialize the matrix for the pressure
u = np.zeros([Nt,Nx])
# We compute the CFL number
C = a0*delta_t/delta_x
print('CFL = ',C)
# We impose the initial conditions
u[0,np.where(x < 0.1)] = 1.0
u[0,np.where(x >= 0.1)] = 0.0
# We impose the boundary conditions
u[:,0] = 1.0
# We compute our solution
for n in range(1,Nt):
    for j in range(1,Nx):
        u[n,j] = (1.0-C)*u[n-1,j]+C*u[n-1,j-1]
return u
```

Question 3

We run the simulation for 100 grid points and 100 timesteps and plot the results in a dedicated cell.

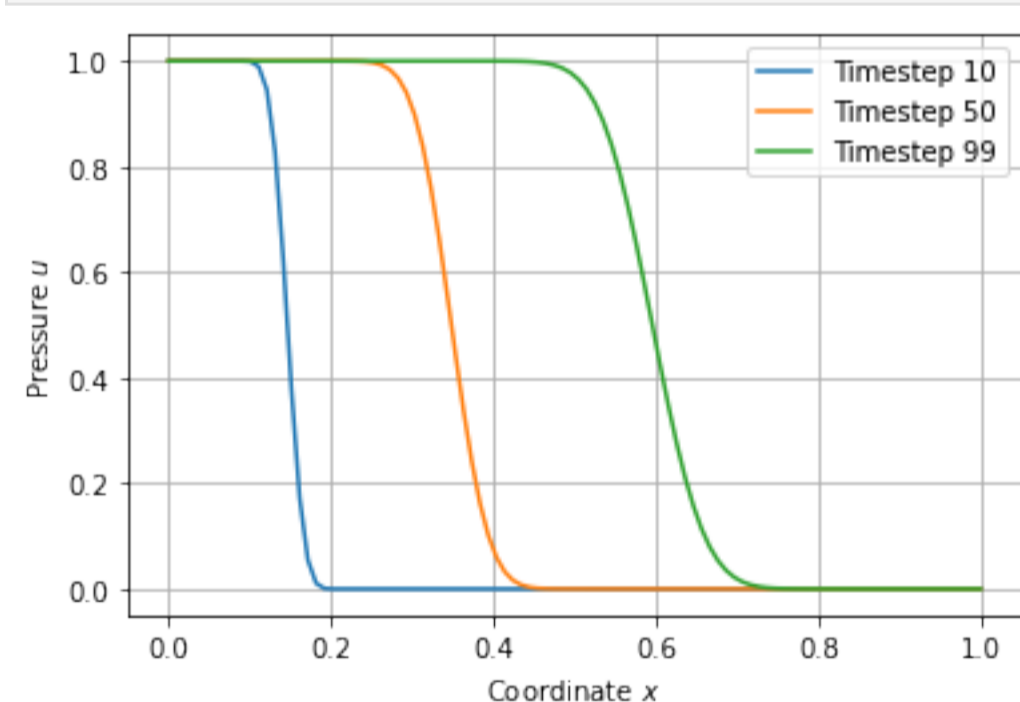
We can observe that the pressure wave is moving along the x axis as expected for such PDE. When we increase the time, we can see that the sharp step from the initial conditions is somehow *lost* or *smoothed*

```
In [ ]: u = compute_u(100,100)
```

CFL = 0.5

```
In [ ]: x = np.linspace(0.0,1.0,100)
```

```
#
plt.plot(x,U[10,:], label='Timestep 10')
plt.plot(x,U[50,:], label='Timestep 50')
plt.plot(x,U[99,:], label='Timestep 99')
plt.xlabel(r'Coordinate $x$')
plt.ylabel(r'Pressure $u$')
plt.legend()
plt.grid()
```



Question 4

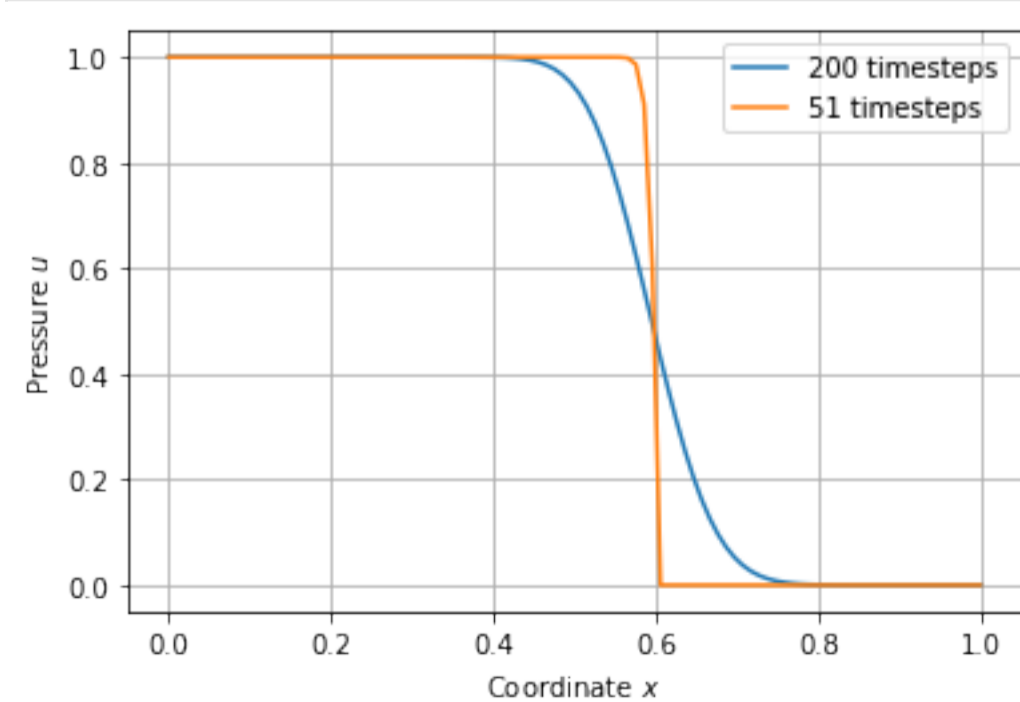
We now carry out two more simulations with the same grid size but different number of timesteps (200 and 51). We plot the resulting pressure distribution at the end of the simulation.

We can observe that the simulation with 51 timesteps is much closer to the initial pressure wave that the one with 200 timesteps. If we print out the CFL number C we can see that the 51 timesteps simulation yields a value close to 1.

```
In [ ]: U_2 = compute_u(100,200)
U_3 = compute_u(100,51)
```

CFL = 0.24874371859296482
CFL = 0.99

```
In [ ]: plt.plot(x,U_2[-1,:], label='200 timesteps')
plt.plot(x,U_3[-1,:], label='51 timesteps')
plt.xlabel(r'Coordinate $x$')
plt.ylabel(r'Pressure $u$')
plt.legend()
plt.grid()
```



```
In [ ]: U_4 = compute_u(100,50)
```

CFL = 1.010204081632653

Question 5

We finally run a simulation with 50 timesteps and compare it to the previous computation with 51 timesteps.

We see that a small difference of one timestep leads us to an unstable scheme. When looking closer to the CFL number we end up with a value of 1.01 which only slightly larger than 1 but enough to trigger an unstable results.

```
In [ ]: plt.plot(x,U_3[-1,:], label='51 timesteps')
plt.plot(x,U_4[-1,:], label='50 timesteps')
plt.xlabel(r'Coordinate $x$')
plt.ylabel(r'Pressure $u$')
plt.legend()
plt.grid()
```

