Magistère de mathématiques

TP Bonus

Tracer une solution approchée de

$$\begin{cases} -\partial_t^2 u(t,x) + \partial_x(a(x)\partial_x u(t,x)) = f(x), & t \geq 0, x \in]0,1[\\ u(0,x) = \sin(x) & x \in]0,1[\\ u(t,0) = u(t,1) = 0 & t \geq 0 \end{cases}$$

Pour cela, on utilisera le schéma aux différences finies suivant

$$\begin{cases} -\dfrac{U_j^{n+1} - 2U_j^n + U_j^{n-1}}{\Delta t} + \dfrac{1}{\Delta x}\left(a(x_{j+1/2})\dfrac{U_{j+1}^n - U_j^n}{\Delta x} - a(x_{j-1/2})\dfrac{U_j^n - U_{j-1}^n}{\Delta x}\right) = f(x_j) \\ U_0^n = U_{J+1}^n = 0 \\ U_j^{-1} = 0, U_j^0 = \sin(x_j) \end{cases}$$

où U_j^n approche la valeur de $u(t^n, x_j)$ avec $t^n = n\Delta t$, $x_j = j\Delta x$, $\Delta x = \frac{1}{J+1}$ et $\Delta t > 0$.

On tracera la solution en fonction du temps en adaptant le code suivant :

```
from math import *
import numpy as np
import matplotlib.pyplot as plt

T = np.linspace(0,10,300)
X = np.linspace(0,10,400)

def f(t,x):
   return cos(x-t)

for t in T:
   Y = np.vectorize(f)(t,X)
   if t == 0:
      line, = plt.plot(X, Y)
   else:
      line.set_ydata(Y)
   plt.pause(0.01) # pause avec duree en secondes
plt.close()

plt.show()
```