TP1 Analyse numérique (B1-TP1)

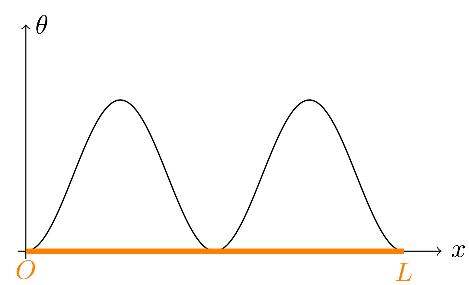
Ibrahim ALAME

28/02/2024

1 Problème de la chaleur

Soit OL une barre chauffée initialement à la température $\theta(x) = \sin^2(2\pi \frac{x}{L})$, $0 \leq x \leq L$. On cherche à refroidir la barre en appliquant une température 0 aux extrémités O et L. Déterminer l'évolution de la température $u(x,t)$ en tout point x de la barre et à chaque instant $t \geq 0$:

$$\begin{cases} \frac{\partial u}{\partial t} - \kappa \frac{\partial^2 u}{\partial x^2} = 0 & \forall (x,t) \in [0,L] \times \mathbb{R}^+ \\ u(x,0) = \theta(x) & \forall x \in [0,L] \\ u(0,t) = u(L,t) = 0 & \forall t \in \mathbb{R}^+ \end{cases}$$



On prendra $\kappa = 0.1$, $L = 1$ et on choisira un pas de temps $\tau = 0.0005$

```
# Initialisation des constantes
L=1; kappa = 0.1
h=0.01; tau= 0.0005
c = kappa * tau / h**2
```

et on discrétisera l'intervalle $[0, L]$ en N sous-intervalles avec un pas $h = 0.01$ ($N + 1$ points de discrétisation).

```
# Coordonnées en espace
N = int(L / h)
X = np.linspace(0, L, N+1)
```

L'animation de l'évolution de la température est à afficher dans une fenêtre 800×600 de la librairie python `tkinter` :

```python
#=============  PROGRAMME PRINCIPAL  ==============
fen1 = Tk() # Création de la fenêtre principale
fen1.title("Equation de la chaleur")
W=800; H=600
# Définir une zone graphique W × H
can1 = Canvas(fen1,bg = 'dark gray', width=W, height=H)
can1.pack() # Positionne can1 dans la fenêtre fen1
DF() # Fonction de calcul en Diff Finies et affichage graphique
fen1.mainloop() # Boucle principale de la fenêtre
```

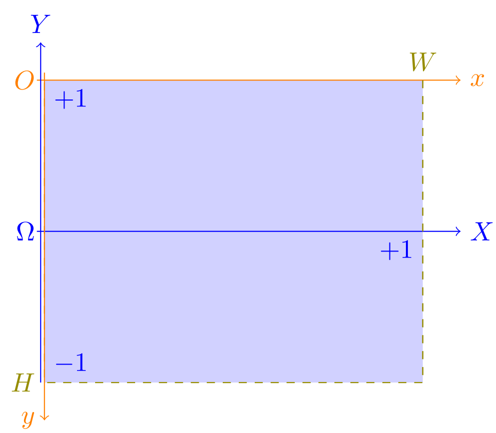
où `DF()` est la fonction qui calcule et affiche la température de la barre à l'instant $t_n = n\tau$ à partir du schéma de la méthode explicite :

$$\frac{u_i^n - u_i^{n-1}}{\tau} - \kappa \frac{u_{i-1}^{n-1} - 2u_i^{n-1} + u_{i+1}^{n-1}}{h^2} = 0$$

```python
def DF():
    global U0 # on désigne par U0[i] la solution à l'instant précédent  u(x_i, t_{n-1}) = u_i^{n-1}
    can1.delete('all') # on efface la courbe de l'instant t-1
    U=np.zeros(N+1)# U[i] est la solution à l'instant actuel  u(x_i, t_n) = u_i^n
    for i in range(1,N):
        U[i]=...
    # courbe à l'instant t; segment entre deux points consécutifs: create_line(x_{i-1}, y_{i-1}, x_i, y_i)
    for i in range(1,N+1):
        can1.create_line(X[i-1]*W,(1-U[i-1])*H,X[i]*W,(1-U[i])*H,fill='blue', width=5)
    U0 = U.copy()
    fen1.after(10, DF) # réexécute la fonction DF() à l'instant suivant après 10ms
```

Transformation géométrique entre le repère informatique (Oxy) et le repère mathématique (ΩXY)

$$\begin{cases} x = X \cdot W \\ y = (1 - Y) \cdot H/2 \end{cases}$$



2 Propagation des ondes

On considère une corde de masse linéïque ρ fixée en ses deux extrémités O et L et tendue avec une grande force F. On pose $c = \sqrt{\frac{F}{\rho}}$. A l'instant $t = 0$, on déplace verticalement chaque point d'abscisse x d'une distance $f(x)$ à une vitesse initiale $g(x)$. En petites perturbations, le déplacement $u(x,t)$ en tout point x de la corde et à chaque instant $t \geq 0$ est solution du problème des ondes suivant :

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 & \forall (x,t) \in [0,L] \times \mathbb{R}^+ \\ u(x,0) = f(x) & \forall x \in [0,L] \\ \frac{\partial u}{\partial t}(x,0) = g(x) & \forall x \in [0,L] \\ u(0,t) = u(L,t) = 0 & \forall t \in \mathbb{R}^+ \end{cases}$$

Résoudre numériquement ce problème par la méthode des différences finies :

$$\frac{u_i^n - 2u_i^{n-1} + u_i^{n-2}}{\tau^2} - c^2 \frac{u_{i+1}^{n-1} - 2u_i^{n-1} + u_{i-1}^{n-1}}{h^2} = 0$$

en prenant les données numériques suivantes :

$$L = 1 \text{ m}, \quad c = 3 \text{ m/s}, \quad f(x) = e^{-500(x-L/2)^2}, \quad g(x) = 0,$$

afficher le résultat en animation graphique dans une fenêtre `tkinter` de hauteur $H = 800$px et de largeur $W = 800$px.

3 Équation de diffusion

On considère l'équation d'advection suivante :

$$\begin{cases} \dfrac{\partial u}{\partial t} + V \dfrac{\partial u}{\partial x} = 0, \quad 0 \leq x \leq 1, \quad t \in \mathbb{R}^+ \\ u(x+1,t) = u(x,t) \\ u(x,0) = u_0(x) = \cos(\pi x)^8 \end{cases}$$

1. Pour résoudre ce problème, programmer le *schéma explicite centré*, suivants :

$$\frac{u_i^n - u_i^{n-1}}{\tau} + V \frac{u_{i+1}^{n-1} - u_{i-1}^{n-1}}{2h} = 0$$

2. *Schéma de Lax-Wendroff* On fait un développement limité à l'ordre 2 en temps de $u(x,t)$:

$$u(x_i, t_{n+1}) = u(x_i, t_n) + \tau \frac{\partial u}{\partial t}(x_i, t_n) + \frac{\tau^2}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t_n) + o(\tau^2)$$

Nous allons passer les dérivées partielles par rapport au temps à des dérivées partielles par rapport à l'espace :

$$\frac{\partial u}{\partial t} = -V \frac{\partial u}{\partial x}$$

et

$$\frac{\partial^2 u}{\partial t^2} = -V \frac{\partial}{\partial t}\left[\frac{\partial u}{\partial x}\right] = -V \frac{\partial}{\partial x}\left[\frac{\partial u}{\partial t}\right] = +V^2 \frac{\partial^2 u}{\partial x^2}$$

Finalement

$$u(x_i, t_{n+1}) = u(x_i, t_n) - \tau V \frac{\partial u}{\partial x}(x_i, t_n) + \frac{\tau^2 V^2}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t_n) + o(\tau^2)$$

D'où le schéma de Lax-Wendroff :

$$\frac{u_i^n - u_i^{n-1}}{\tau} + V \frac{u_{i+1}^{n-1} - u_{i-1}^{n-1}}{2h} - \frac{\tau V^2}{2} \frac{u_{i+1}^{n-1} - 2u_i^{n-1} + u_{i-1}^{n-1}}{h^2} = 0$$

Programmer ce schéma et comparer avec la méthode explicite centrée.