

Exercice 1 : Méthode d'Euler et équation de diffusion [★]

Introduction On s'intéresse ici à une barre métallique cylindrique dont la surface latérale est calorifugée. Les deux extrémités gauche et droite de la barre sont en contact avec de la glace fondante de température $T_g = 0^\circ\text{C}$, qui joue le rôle de thermostat. L'objectif de cet exercice est de résoudre numériquement l'équation de la diffusion thermique dans cette situation simple. On appelle désormais $L = 1\text{ m}$ la longueur de la barre. On rappelle que l'équation de diffusion thermique s'écrit

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2},$$

où κ désigne la diffusivité thermique. On considérera un matériau de diffusivité $\kappa = 10^{-4}\text{ m}^2\cdot\text{s}^{-1}$.

Nous allons dans la suite déterminer la température $T(x, t)$, en tout point x de la barre, et à tout instant t de l'expérience. Pour cela, nous allons découper la barre en tronçons de longueur $dx = 1\text{ cm}$. L'étude sera menée sur une durée de $\Delta t = 30\text{ min}$, avec un pas temporel de $dt = 0,4\text{ s}$. La fonction $T(x, t)$ sera donc approximée par un tableau $T[i, j]$ de telle sorte que :

$$T[i, j] = T(x = jdx, t = idt).$$

Autrement dit, chaque ligne i du tableau correspond à l'instant $t = idt$ de l'expérience (la ligne $i=0$ correspondant à l'instant initial), et chaque colonne j du tableau représente la position $x = jdx$ (la colonne $j=0$ correspondant au bord gauche).

Le tableau utilisé est de type `numpy.array`. On aura de plus besoin du module `matplotlib.pyplot` pour effectuer les représentations graphiques du champ de température. On importe pour cela les modules suivants.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
```

Initialisation

- 1) On nommera désormais les variables entières `Nt` et `Nx` correspondant respectivement au nombre de lignes et de colonnes du tableau `T`.

Compléter les lignes suivantes pour définir `Nx` et `Nt` sur Python, et créer le tableau `T` en le remplissant de zéros pour l'instant. Créer également les tableaux `x` et `t` correspondant respectivement aux positions x et aux instants t pour lesquels la température est évaluée.

```
1 dx = 1e-2
2 dt = 4e-1
3 L = 1
4 Deltat = 30*60
```

✓ Solution

```
1 Nx = int(L/dx) + 1
2 Nt = int(Deltat/dt) + 1
3
4 T = np.zeros((Nt, Nx))
5
6 x = np.array([j*dx for j in range(Nx)])
7 t = np.array([i*dt for i in range(Nt)])
```

- 2) On suppose que le profil de température initial a pour expression

$$T(x, t = 0) = T_0 \sin(2\pi x/L),$$

avec $T_0 = 20^\circ\text{C}$. Remplir la première ligne du tableau T pour spécifier les conditions initiales de l'expérience.

✓ **Solution**

```
1 T0 = 20
2 T[0,:] = T0*np.sin(2*np.pi*x/L)
```

- 3) Compléter le code suivant pour tracer l'évolution de la température initiale au sein de la barre.

```
1 plt.close()
2 plt.figure()
3 plt.grid()
4 plt.xlabel("x (m)")
5 plt.ylabel("T (°C)")
6 plt.title("Profil de température initial")
```

✓ **Solution**

```
1 plt.plot(x,T[0,:])
2 plt.show()
```

- 4) Remplir le tableau T de manière à imposer, à tout instant, les conditions aux limites aux deux extrémités de la barre, $T(x = 0, t) = 0$ et $T(x = L, t) = 0$.

✓ **Solution**

```
1 for i in range(1,Nt): # l'instant initial a déjà été traité
2     T[i,0] = 0
3     T[i,-1] = 0
```

Discretisation de l'équation de diffusion Il s'agit donc désormais de résoudre l'équation :

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2}.$$

- 5) En utilisant l'approximation de la dérivée, exprimer $\frac{\partial T}{\partial t}(t = idt, x = jdx)$ en fonction des éléments de matrice $T[i, j]$ et $T[i+1, j]$.

✓ **Solution**

$$\frac{\partial T}{\partial t}(t = idt, x = jdx) \simeq \frac{T[i+1, j] - T[i, j]}{dt}$$

- 6) Soit une fonction f . En écrivant la formule de Taylor à l'ordre 2 pour $f(x+dx)$ et $f(x-dx)$, démontrer que :

$$f''(x) \simeq \frac{f(x+dx) + f(x-dx) - 2f(x)}{dx^2}.$$

✓ **Solution**

Les développements de Taylor à l'ordre 2 de $f(x + dx)$ et $f(x - dx)$ sont respectivement :

$$f(x + dx) \simeq f(x) + f'(x)dx + \frac{f''(x)}{2}dx^2$$

et

$$f(x - dx) \simeq f(x) - f'(x)dx + \frac{f''(x)}{2}dx^2.$$

En sommant ces deux expressions, on obtient :

$$f(x + dx) + f(x - dx) \simeq 2f(x) + f''(x)dx^2.$$

En isolant $f''(x)$, on obtient bien la relation souhaitée.

- 7) En déduire l'approximation de la grandeur $\frac{\partial^2 T}{\partial x^2}(t = idt, x = jdx)$ en fonction des éléments de matrice $T[i, j]$, $T[i, j+1]$ et $T[i, j-1]$.

✓ **Solution**

$$\frac{\partial^2 T}{\partial x^2}(t = idt, x = jdx) \simeq \frac{T[i, j+1] + T[i, j-1] - 2T[i, j]}{dx^2}.$$

- 8) En déduire que la résolution de l'équation de diffusion thermique dans la barre se ramène au schéma numérique :

$$T[i+1, j] = T[i, j] + \frac{\kappa dt}{dx^2} (T[i, j+1] + T[i, j-1] - 2T[i, j]).$$

Résolution On peut montrer qu'un tel schéma numérique converge si

$$\frac{\kappa dt}{dx^2} < \frac{1}{2}.$$

Avec les valeurs de dx et de dt choisies, cette condition est respectée pour la valeur choisie de la diffusivité thermique $\kappa = 10^{-4} \text{ m}^2 \cdot \text{s}^{-1}$.

- 9) Implémenter ce schéma numérique en réalisant une boucle sur i et éventuellement sur j pour remplir le tableau T dans son intégralité.

✓ **Solution**

```
1 kappa = 1e-4
2 for i in range(0, Nt-1): # on remplit à l'indice i+1
3     for j in range(1, Nx-1): # les conditions aux limites sont déjà fixées
4         T[i+1, j] = T[i, j] + kappa*dt/(dx**2)*(T[i, j+1] + T[i, j-1] - 2*T[i, j])
```

- 10) On souhaite tracer le profil de température aux différents instants définis dans la liste ci-dessous. Compléter le code ci-dessous de sorte à tracer sur un même graphique légendé l'évolution du profil de température dans le temps.

```
1 plt.close()
2 plt.figure()
3 plt.grid()
4 plt.xlabel("x (m)")
5 plt.ylabel("T (°C)")
6 plt.title("Profil de température dans la barre à différents instants")
7
8 instants = [0, 1, 3, 6, 9, 12, 15, 30] # instants choisis en minutes
```

✓ **Solution**

```
1  for t in instants: # tracé du profil du température aux instants choisis
2      i = int(t*60/dt)
3      plt.plot(x,T[i,:],label="t = "+str(t)+" min")
4
5  plt.legend()
6  plt.show()
```