

## <sup>26</sup>Résolution d'une équation aux dérivées partielles par la méthode d'Euler

### Exemple de l'équation de la chaleur en 1D non stationnaire

#### Idée de base :

L'idée de base est de discréteriser à la fois l'espace et le temps, et de calculer les dérivées partielles par rapport au temps ou par rapport à l'espace de façon approchée.

#### Formule de récurrence pour l'équation de la chaleur :

Rappels : En 1D, sans terme de création,  $\frac{\partial T}{\partial t} = \frac{\lambda}{\mu c} \frac{\partial^2 T}{\partial x^2}$

- On se fixe un pas temporel  $\Delta t$  et un pas spatial  $\Delta x$  (voir comment, plus loin).
- Pour représenter la température  $T(x, t)$ , on cherche à construire un tableau  $T[i, j]$ .
- Avec la méthode d'Euler, la dérivée temporelle d'ordre 1 est exprimée de façon approchée :

$$\frac{\partial T}{\partial t} \quad \leftarrow \rightarrow \quad \frac{T[i, j + 1] - T[i, j]}{\Delta t}$$

- De même, on peut donner une expression approchée de  $\frac{\partial T}{\partial x}$ , puis de  $\frac{\partial^2 T}{\partial x^2}$ .  
→ Donner ces expressions.
- En déduire la formule de récurrence pour l'équation de la chaleur, c'est-à-dire  $T[i, j+1]$  en fonction de  $T[i, j]$ ,  $T[i+1, j]$ ,  $T[i-1, j]$ ,  $\lambda$ ,  $\mu$ ,  $c$ ,  $dx$ ,  $dt$ .

#### Choix du pas spatial et du pas temporel :

- On commence par choisir un pas spatial  $\Delta x$ . Pour cela, on divise la longueur  $L$  du domaine spatial étudié en un nombre  $N_x - 1$  d'intervalles  $\Delta x$ , avec  $N_x$  suffisamment grand (par exemple 101) pour que la courbe qui donnera  $T$  en fonction de  $x$ , à  $t$  fixé, soit bien lisse. Il y a alors  $N_x$  valeurs de  $x$  dans le domaine étudié, y compris 0 et  $L$ .
- Ensuite, on choisit  $\Delta t$  : **il faut que  $\Delta t$  soit un peu plus petit** (par exemple cinq ou dix fois plus petit) **que le temps caractéristique avec lequel la « chaleur » diffuse sur une distance  $\Delta x$** .
- Ensuite, on choisit le nombre total d'instants :  $N_t$ , de la façon suivante :
  - > En régime transitoire, le nombre  $N_t$  d'instants utilisés pour calculer la durée totale  $T_{tot}$  (c'est-à-dire  $N_t \cdot \Delta t$ ) de l'étude, doit être assez élevé pour pouvoir observer l'installation du régime stationnaire (donc la fin du régime transitoire).
  - > En régime temporel périodique établi, de période  $T_{per}$ , le nombre  $N_t$  d'instants utilisés pour calculer la durée totale  $T_{tot}$  (c'est-à-dire  $N_t \cdot \Delta t$ ) doit être tel qu'on puisse observer plusieurs périodes, typiquement 3 ou 4.

#### Conditions aux limites possibles (du point de vue spatial) :

- Cas d'une température maintenue constante aux deux extrémités,  $T_1$  en  $x = 0$ , et  $T_2$  en  $x = L$  :  
→ Ecrire ce qu'imposent ces conditions pour certains  $T[i, j]$ .
- Cas d'une isolation thermique parfaite aux deux bouts, c'est-à-dire en  $x = 0$  et en  $x = L$  :  
→ Ecrire ce qu'imposent ces conditions pour certains  $T[i, j]$ .

**Première application : température au sein d'un barre de cuivre de longueur L=0,15 m, soumise à des températures constantes aux deux bouts.**

- On part d'une situation initiale avec la barre à une température uniforme de 20°C.
- On impose brutalement, à partir de l'instant  $t = 0$ , une température de 40°C en  $x = 0$ , tout en maintenant une température de 20°C en  $x = L$ .

→ Réaliser un programme python pour déterminer  $T(x, t)$  pour  $x \in [0, L]$  et  $t \geq 0$ , en résolvant l'équation de la chaleur. On donne  $\lambda = 390 \text{ W.K}^{-1}.\text{m}^{-1}$ ,  $\mu = 8960 \text{ kg.m}^{-3}$ ,  $c = 385 \text{ J.K}^{-1}.\text{kg}^{-1}$ .

→ Avec python, tracer  $T(x, t)$  en fonction de  $x$  pour  $x \in [0, L]$ , à différents instants choisis.

→ Eventuellement, avec python, construire un film (avec l'instruction `animate`, cf remarque en page 3) donnant  $T(x, t)$  en fonction de  $x$  pour  $x \in [0, L]$ , sur une certaine durée, à choisir en fixant la valeur de `Nt`.

**Seconde application : température au sein d'un barre de cuivre de longueur L=0,15 m, calorifugée aux deux bouts, avec initialement une répartition de température en triangle, avec 20°C aux deux bouts, et 40°C au milieu.**

→ Réaliser un programme python pour déterminer  $T(x, t)$  pour  $x \in [0, L]$  et  $t \geq 0$ .

→ Avec python, tracer  $T(x, t)$  en fonction de  $x$  pour  $x \in [0, L]$ , à différents instants choisis.

→ Eventuellement, avec python, construire un film donnant  $T(x, t)$  en fonction de  $x$  pour  $x \in [0, L]$ , sur une certaine durée, à choisir en fixant la valeur de `Nt`.

**Troisième application : effet de peau thermique annuel dans un sous-sol argileux, de profondeur L=20 m (onde thermique qui se propage en s'atténuant).**

- On part d'une situation initiale avec le sous-sol à une température uniforme de 15°C.
- On impose pour  $t > 0$ , une température de surface fluctuant sinusoïdalement (avec une période d'un an) autour de 15°C, avec une amplitude de 5°C, tout en maintenant une température de 15°C en  $x = L$ .

→ Réaliser un programme python pour déterminer  $T(x, t)$  pour  $x \in [0, L]$  et  $t \geq 0$ .

→ Avec python, tracer  $T(x, t)$  en fonction de  $x$  pour  $x \in [0, L]$ , à différents instants choisis.

→ Eventuellement, construire un film donnant  $T(x, t)$  en fonction de  $x$  pour  $x \in [0, L]$ , sur une certaine durée, correspondant à 3 ou 4 périodes temporelles.

$\lambda = 2,2 \text{ W.K}^{-1}.\text{m}^{-1}$ ,  $\mu \times c = 2,2 \cdot 10^6 \text{ S.I.}$  Période temporelle : 1 an.

**Remarque : pour construire une animation à partir d'une fonction mise sous la forme d'un tableau  $f[i, j]$  :**

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 from matplotlib import animation
4
5 #exemple de l'animation d'une ODE
6 Nx=101 ; Nt=301 ; Lamb=10 ; Periode=0.01 ; L=3*Lamb # constantes
7 dx=L/(Nx) #pas spatial
8 dt=Periode/(Nt) #pas temporel
9 x=np.linspace(0,L,Nx)
10
11 f=np.zeros((Nx,Nt)) #création du tableau de la fonction f
12
13 for j in range(Nt):
14     for i in range(Nx):
15         f[i,j]=1.5*np.sin(2*np.pi*(j*dt/Periode-i*dx/Lamb))
16
17 fig = plt.figure() # initialise la figure
18 line, = plt.plot([],[])
19 plt.xlim(0,L)
20 plt.ylim(-2,2)
21 # fonction à définir quand blit=True
22 def init():
23     line.set_data([],[])
24     return line,
25
26 def animate(j):
27     y = f[:,j]
28     line.set_data(x, y)
29     return line,
30
31 ani = animation.FuncAnimation(fig, animate, init_func=init, frames=300, blit=True, interval=1)
32 plt.grid(color='r',linestyle='--')
33 plt.show()
```