Résolution numérique de l'équation de la chaleur

Pour résoudre une équation aux dérivées partielles d'ordre 2, on utilise les approximations des dérivées partielles premières et secondes suivantes :

$$\frac{\partial F}{\partial x_i}(x_i,\ldots) \approx \frac{F(x_i+dx_i,\ldots)-F(x_i,\ldots)}{dx_i} \approx \frac{F(x_i,\ldots)-F(x_i-dx_i,\ldots)}{dx_i}$$
$$\frac{\partial^2 F}{\partial x_i^2}(x_i,\ldots) \approx \frac{F(x_i+dx_i,\ldots)+F(x_i-dx_i,\ldots)-2F(x_i,\ldots)}{2dx_i}$$

que l'on injecte dans l'EDP pour en déduire un schéma de résolution.

Exemple. *Diffusion de la chaleur le long d'une barre métallique.*

Une barre métallique de longueur l a un coefficient de diffusion K. L'une des extrémités de la barre est reliée à une source de chaleur de température T_1 et l'autre à une source de chaleur de température T_2 avec $T_1 < T_2$. Soit $T(x,t)$ la température de la barre au point d'abscisse x au temps t.

L'équation de la chaleur est l'équation aux dérivées partielles d'ordre 2 :

$$\boxed{\frac{\partial T}{\partial t}(x,t) = K\frac{\partial^2 T}{\partial x^2}(x,t)}.$$

On se donne $K = 10^{-5}\,\mathrm{m^2.s^{-1}}$, $l = 1\,\mathrm{m}$, $T_1 = 20\,°\mathrm{C}$ et $T_2 = 100\,°\mathrm{C}$. Initialement la température est de T_1 en tout point de la barre. On souhaite tracer l'évolution de la température le long de la barre pendant 5 heures.

On applique les approximations des dérivées première et seconde :

$$\frac{\partial T}{\partial t}(x,t) \approx \frac{T(x,t+dt)-T(x,t)}{dt}$$
$$\frac{\partial^2 T}{\partial x^2}(x,t) \approx \frac{T(x+dx,t)+T(x-dx,t)-2T(x,t)}{dx^2}$$

pour exprimer $T(x,t+dt)$ (température au point x au temps $t+dt$) en fonction de $T(x,t)$, $T(x+dx,t)$, $T(x-dx,t)$ (températures aux points $x-dx$, x, $x+dx$ au temps t) et des pas de discrétisation dx et dt.

L'équation de la chaleur se discrétise avec pour pas spatial dx et pas temporel dt en :

$$\frac{T(x,t+dt)-T(x,t)}{dt} \approx K\frac{T(x+dx,t)+T(x-dx,t)-2T(x,t)}{(dx)^2}$$
$$\boxed{T(x,t+dt) \approx T(x,t) + \frac{Kdt}{(dx)^2}\left(T(x+dx,t)+T(x-dx,t)-2T(x,t)\right)}.$$

C'est le schéma de résolution qui va permettre la résolution.

Initialement la température est de T_1 en tout point de la barre. Au temps $t+dt$ la température vaut T_1 au premier point, T_2 au dernier point, et s'obtient en tout autre point x de la barre grâce au schéma ci-dessus à partir des températures au temps t aux positions x, $x-dx$ et $x+dx$. Le tableau des températures au temps $t+dt$ se calcule à partir du tableau au temps t.

• Pour un tracé 2D

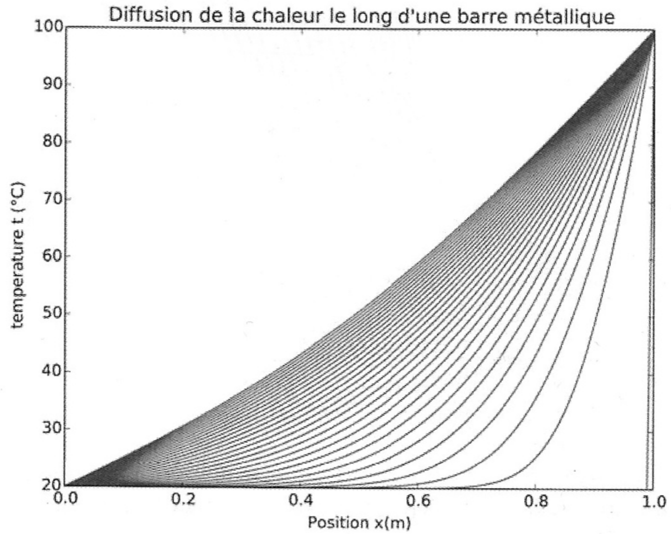
À un instant t la température aux différentes positions le long de la barre (subdivision régulière de pas dx de la barre) est contenue dans un tableau unidimensionnel.

On met à jour la température le long de la barre des temps $t = 0$ à $t_{\max}$ par pas dt de 1 seconde. Toutes les 600 secondes (=10 minutes) on trace la courbe des températures le long de la barre.

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

T1 = 20                    # T° ambiante
T2 = 100                   # Source chaleur
K = 1e-5                   # Coefficient diffusion thermique
tmax = 5*3600              # 5 heures
Nx = 100                   # Nbre points sur la barre
x = np.linspace(0,1,Nx)
T = T1*np.ones(Nx)         # T° initiale
dx=1/(Nx-1)                # Pas spatial
```

```
dt=1                          # Pas temporel
c= K*dt/dx**2
for i in range(tmax):
    t = np.empty(Nx)
    for k in range(1,Nx-1):
        t[k] = T[k]+c*(T[k+1]+T[k-1]-2*T[k])
    t[0] = T1
    t[Nx-1] = T2
    T = t
    if i%600 == 0:            # Tracé toutes les 10 minutes
        plt.plot(x,T,color='blue')
plt.xlabel('Position x(m)')
plt.ylabel('temperature t (°C)')
plt.title("Diffusion de la chaleur le long d'une barre\
                                          métallique")
plt.show()
```



Exercice pour aller plus loin…

4) ■ Résolution de l'équation de la chaleur par les méthodes des différences finies et de Cranck-Nicholson.

On considère une barre métallique d'épaisseur négligeable, placée dans le vide, de longueur L et de coefficient de diffusion calorifique K. Au temps $t \geqslant 0$ sur une section de la barre à l'abscisse $x \in [0, L]$ tous les points sont à même température $T(x, t)$.

Initialement au temps $t = 0$, la température est donnée en tout point de la barre par $T(x, 0) = f(x)$. Nous étudions l'évolution de la température en tout point de la barre lorsque ses deux extrémités sont maintenues à température constante nulle. On démontre que la fonction $(x, t) \longmapsto T(x, t)$ est l'unique solution de :

$$\begin{cases} \dfrac{\partial}{\partial t} T(x,t) = K \cdot \dfrac{\partial^2}{\partial x^2} T(x,t) & & \text{(Équation de la chaleur)} \\ T(x,0) = f(x) & \forall\, x \in [0, L] & \text{(Température initiale)} \\ T(0,t) = T(L,t) = 0 & \forall\, t \geqslant 0 & \text{(Conditions aux limites)} \end{cases}$$

Le but de l'exercice est de tracer l'évolution de la température le long de la barre au cours de la première seconde par la méthode des différences finies.

On prendra pour valeurs numériques :

$$\forall\, x \in]0, L[,\ f(x) = 1\,°\text{C et } f(0) = f(L) = 0\,°\text{C}.$$
$$\text{et } L = 2\,\text{m}\,;\ K = 1\,\text{m}^2\cdot\text{s}^{-1}$$

en définissant en début de programme les 2 variables globales :

```
L = 2
K = 1
```

On aura aussi importé les modules numpy et matplotlib.pyplot par les commandes :

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

A. Résolution par la méthode des différences finies

Une résolution numérique s'obtient par la méthode des différences finies en discrétisant les intervalles spatial $[0,L]$ et temporel $[0,T]$, c'est-à-dire en posant :

- $(x_n)_{0\leqslant n\leqslant N}$ une subdivision régulière de $[0,L]$ de pas Δx : $\forall n\in[\![0,N-1]\!]$, $x_{n+1}-x_n=\Delta x$;
- $(t_m)_{0\leqslant m\leqslant M}$ une subdivision régulière de $[0,T]$ de pas Δt : $\forall n\in[\![0,M-1]\!]$, $t_{m+1}-t_m=\Delta t$,

et en appliquant les approximations des dérivées partielles premières et secondes :

$$\forall n\in[\![0,N]\!],\ \forall m\in[\![0,M-1]\!],\quad \frac{\partial}{\partial t}T(x_n,t_m)\approx\frac{T(x_n,t_{m+1})-T(x_n,t_m)}{\Delta t}$$

$$\forall n\in[\![1,N-1]\!],\ \forall m\in[\![0,M]\!],\quad \frac{\partial^2}{\partial x^2}T(x_n,t_m)\approx\frac{T(x_{n+1},t_m)-2T(x_n,t_m)+T(x_{n-1},t_m)}{(\Delta x)^2}$$

pour obtenir un schéma de résolution.

1. Appliquer les approximations ci-dessus pour exprimer $T(x_n,t_{m+1})$ en fonction de $T(x_n,t_m)$, $T(x_{n+1},t_m)$, $T(x_{n-1},t_m)$, Δx et Δt. On précisera pour quelles valeurs de n et m cette expression est valide.
2. On prend $N=100$, $\Delta t=10^{-4}$ s. Donner le code permettant à partir du tableau T des valeurs $T(x_n,t_m)$ pour $n\in[\![0,N]\!]$ (températures le long de la barre au temps t_m), d'obtenir le tableau T1 des valeurs $T(x_n,t_{m+1})$ pour $n\in[\![0,N]\!]$ (températures le long de la barre au temps $t_m+\Delta t$).
3. En déduire le code permettant d'effectuer le tracé des courbes de température le long de la barre toutes les 5.10^{-2} secondes durant la première seconde.

B. Résolution par la méthode de Crank-Nicholson

La méthode de Crank-Nicholson diffère de la méthode des différences finies en approchant $\frac{\partial^2}{\partial x^2}T(x_n,t_m)$ par la moyenne des approximations des dérivées partielles secondes au temps t_m et t_{m+1}.

Schémas de Crank-Nicholson :

$$\frac{\partial^2}{\partial x^2}T(x_n,t_{m+1})\approx\frac{T(x_{n+1},t_{m+1})-2T(x_n,t_{m+1})+T(x_{n-1},t_{m+1})}{2\Delta x^2}+\frac{T(x_{n+1},t_m)-2T(x_n,t_m)+T(x_{n-1},t_m)}{2\Delta x^2}$$

$$\frac{\partial}{\partial t}T(x_n,t_m)\approx\frac{T(x_n,t_{m+1})-T(x_n,t_m)}{\Delta t}.$$

En notant $T(x_n,t_m)=T_n^m$ on obtient selon ce schéma la version discrétisée de l'équation de la chaleur :

$$\frac{T_n^{m+1}-T_n^m}{K\Delta t}=\frac{T_{n+1}^{m+1}-2T_n^{m+1}+T_{n-1}^{m+1}+T_{n+1}^m-2T_n^m+T_{n-1}^m}{2(\Delta x)^2}.$$

Ce qui donne en notant : $\boxed{a=\frac{K\Delta t}{2(\Delta x)^2}}$ l'équation suivante exprimant la température au temps t_{m+1} exprimé en fonction de la température au temps t_m :

$$\boxed{-aT_{n-1}^{m+1}+(1+2a)T_n^{m+1}-aT_{n+1}^{m+1}=aT_{n-1}^m+(1-2a)T_n^m+aT_{n+1}^m}.$$

La température le long de la barre (en (x_n)) étant stockée dans un vecteur, cela va aboutir à un système linéaire que l'on va résoudre de proche en proche.

4. Écrire une fonction `mat(N,a,b)` qui retourne une matrice carrée de type $(N+1)$ de la forme :

$$M(N,a,b)=\begin{pmatrix}1&0&0&0&0\\a&b&a&0&0\\0&a&b&a&0\\0&0&a&b&a\\0&0&0&0&1\end{pmatrix}\quad(\text{ici } N=4)$$

On pourra partir d'une matrice contenant des zéros obtenue par l'instruction :
`M = np.zeros((n+1,n+1))`

5. On note T^m le vecteur $(T_0^m,T_1^m,\cdots,T_N^m)$. T^m étant connu T^{m+1} est alors solution d'un système linéaire de la forme :
$$M(N,a_g,b_g)\times T^{m+1}=M(N,a_d,b_d)\times T^m.$$
Exprimer a_g, b_g et a_d, b_d en fonction de la constante a.
6. À l'aide de la fonction `solve()` de `numpy.linalg` en déduire le code permettant, connaissant T0$=T^m$, K, Dx$=\Delta x$, et Dt$=\Delta t$, d'en déduire T1$=T^{m+1}$.
On rappelle que `solve(A,B)` retourne le vecteur solution X d'un système de Cramer de la forme $AX=B$.
7. Effectuer le tracé de la température le long de la barre toutes les 5.10^{-2} secondes durant la première minute. On prendra pour valeurs numériques :
$$N=100 \text{ et } \Delta t=10^{-4}.$$

Calculs et codes seront exprimés sous forme littérale ; on pourra ainsi changer ultérieurement les valeurs de K et L sans avoir à changer tout le code.
3. Tracer une courbe sur 500.

1. L'équation devient :

$$\frac{T(x_n,t_{m+1})-T(x_n,t_m)}{\Delta t}=K\cdot\frac{T(x_{n+1},t_m)-2T(x_n,t_m)+T(x_{n-1},t_m)}{(\Delta x)^2}$$

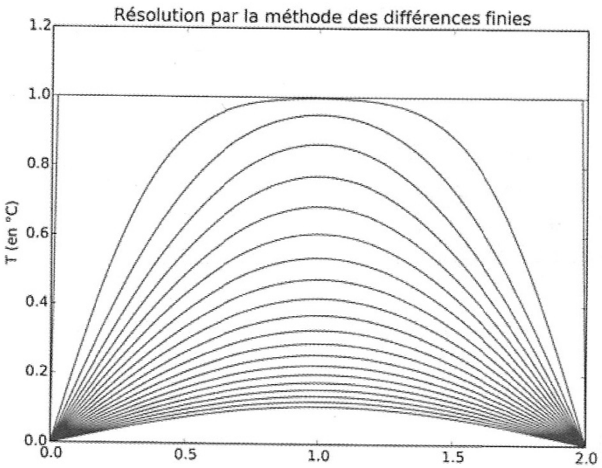
$$\Longrightarrow \quad \boxed{T(x_n,t_{m+1}) = T(x_n,t_n) + \frac{K\cdot\Delta t}{(\Delta x)^2}\left(T(x_{n+1},t_m) - 2T(x_n,t_m) + T(x_{n-1},t_m)\right)}.$$

L'expression est valable pour tout $n \in [\![1, N-1]\!]$ et $m \in [\![0, M-1]\!]$.

2. Obtention du tableau T1 au temps $t + dt$ à partir du tableau T au temps t.

```
N = 100
dt = 1e-4
dx = L/N
T1 = np.empty(N+1)
T1[0] = T1[N] = 0
for x in range(1,N):
    T1[x] = T[x] + K*dt/(dx**2) *(T[x+1]-2*T[x]+T[x-1])
```

3. Résolution par la méthode des différences finies.




```
N=100
X = np.linspace(0,L,N+1)
plt.figure(1)
T = np.ones(N+1)          # Température initiale
T[0] = T[N] = 0
dt = 1e-4                 # Pas temporel
dx = L/N                  # Pas spatial
for t in range(10001):        # Toutes les 0,0001s pendant 1s
    T1 = np.empty(N+1)
    T1[0] = T1[N] = 0         # Conditions aux limites
    for x in range(1,N):      # Températures au temps t+dt
        T1[x] = T[x] + K*dt/(dx**2) *(T[x+1]-2*T[x]+T[x-1])
    if t % 500 == 0:          # Tracé toutes les 0,05s
        plt.plot(X,T,color='black')
    T = T1
plt.axis([0,L,0,1.2])
plt.title("Résolution par la méthode des différences finies")
plt.ylabel("T (en °C)")
```

4. Fonction mat :

```
def mat(n,a,b):
    M = np.zeros((n+1,n+1))
    M[0,0], M[n,n] = 1,1
    for i in range(1,n):
        M[i,i-1], M[i,i], M[i,i+1] = a, b, a
    return M
```

5. Expressions de a_g, b_g, a_d et b_d en fonction de a :

$$a_g = -a, \quad b_g = 1+2a, \quad a_d = a, \quad b_d = 1-2a$$

6. Obtention de T1= T^{m+1} :

```
a = K*Dt/2/Dx**2
Mg = mat(N,-a,1+2*a)
Md = mat(N,a,1-2*a)
b = np.dot(Md,T0)
T1 = np.linalg.solve(Mg,b)
```

7. Tracé de la température toutes les 0.05 s durant la première minute :

```
N = 100
T0 = np.ones(N+1)
T0[0] = T0[N] = 0
Dt = 1e-4
Dx = L/N
a = K*Dt/2/Dx**2
%
plt.figure(2)   # Ouverture d'un graphique
X = np.linspace(0,L,N+1)
Mg = mat(N,-a,1+2*a)
Md = mat(N,a,1-2*a)
T=T0
%
compteur = 1
plt.plot(X,T,color='black')
for t in np.arange(Dt,1+Dt,Dt):
   b = np.dot(Md,T)
   T = np.linalg.solve(Mg,b)
   if compteur % 500 == 0:
```

```
        plt.plot(X,T)
    compteur += 1
plt.title("Résolution par la méthode de Crank-Nicholson")
plt.ylabel('T (en °C)')
plt.show()
```

On obtient le graphique suivant :