Résolution numérique de l'équation de la diffusion thermique

Situation du problème:

On considère une barre solide de longueur L, bon conducteur thermique de préférence, de coefficient de diffusion thermique D. On suppose le problème à 1D. La barre est initialement "préparée" dans un état de température:

$$T(x, t < 0) = T(x)$$

A l'instant $t \geq 0$, les extrémités de la barre sont mises en contact avec deux sources de température identiques T_{extr} donc:

$$T(0, \forall t) = T(L, \forall t) = T_{extr}$$

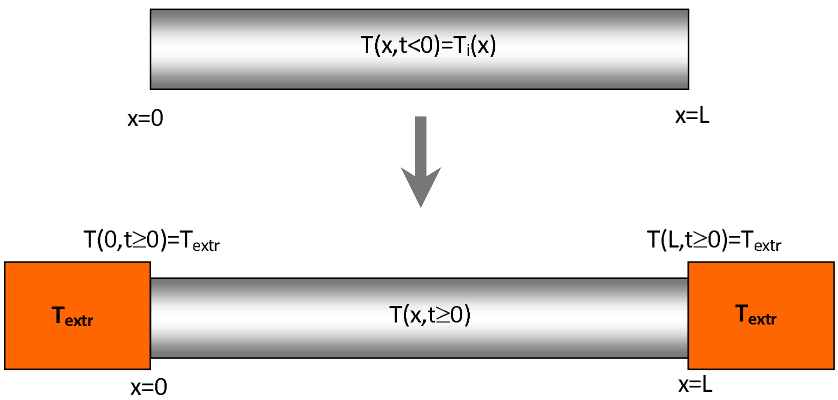


Figure 1: Choc thermique sur une barre de longueur L

La température $T(x, t)$ de la barre est solution de l'équation de la diffusion thermique 1D:

$$\boxed{\frac{\partial}{\partial t} T(x, t) = D \frac{\partial^2}{\partial x^2} T(x, t)} \tag{1}$$

Principe de la résolution numérique:

On recherche une solution numérique à ce problème par la classique **méthode des différences finies**. Supposons que nous cherchions l'évolution de $T(x, t)$ sur une durée totale τ.

La barre est spatialement discrétisée en n_x tronçons de longueur égale $\Delta x = \dfrac{L}{n_x}$. Ainsi, l'abscisse discrète x_m est:

$$x_m = m \cdot \Delta x \qquad \text{avec } m \in [0, n_x]$$

De même, la durée totale d'évolution est discrétisé en n_t intervalles de durée $\Delta t = \dfrac{\tau}{n_t}$. Ainsi, l'instant "discret" t_n est:

$$t_n = n \cdot \Delta t \qquad \text{avec } n \in [0, n_t]$$

On peut par conséquent poser la température discrétisée: $\boxed{T_{m,n} = T(m \cdot \delta x, n \cdot \Delta t)}$

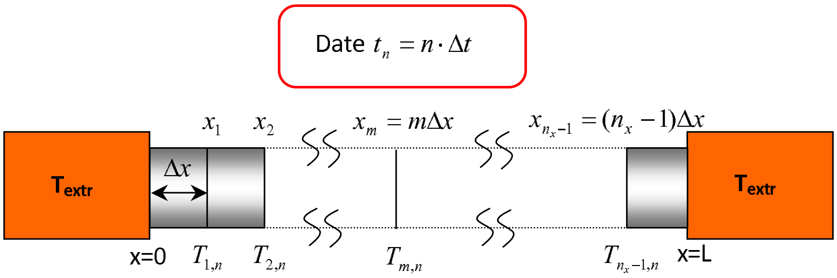


Figure 2: Discrétisation de la température de la barre $T_{m,n}$

En développant la température au 1er ordre discret sur le temps au voisinage de t_n à l'abscisse x_m, il vient:

$$T_{m,n+1} = T_{m,n} + \left.\frac{\partial T(x,t)}{\partial t}\right|_{x_m} \cdot \Delta t + o(\Delta t)$$

ce qui permet de dégager la dérivée première temporelle "discrète" à t_n en x_m:

$$\boxed{\left.\frac{\partial T(x,t)}{\partial t}\right|_{x_m} \simeq \frac{T_{m,n+1} - T_{m,n}}{\Delta t}}$$

De même en développant la température au 2sd ordre discret sur l'espace au voisinage de x_{m+1} et x_{m-1} à la date t_n, il vient:

$$T_{m+1,n} = T_{m,n} + \left.\frac{\partial T(x,t)}{\partial x}\right|_{t_n} \cdot \Delta x + \frac{1}{2}\left.\frac{\partial^2 T(x,t)}{\partial x^2}\right|_{t_n} \cdot (\Delta x)^2 + o(\Delta x^2) \tag{2}$$

$$T_{m-1,n} = T_{m,n} - \left.\frac{\partial T(x,t)}{\partial x}\right|_{t_n} \cdot \Delta x + \frac{1}{2}\left.\frac{\partial^2 T(x,t)}{\partial x^2}\right|_{t_n} \cdot (\Delta x)^2 + o(\Delta x^2) \tag{3}$$

En sommant les équations 2 et 3, on obtient finalement l'expression de la dérivée seconde spatiale «discrète» en x_m à la date t_n:

$$\boxed{\left.\frac{\partial^2 T(x,t)}{\partial x^2}\right|_{t_n} \simeq \frac{T_{m+1,n} - 2T_{m,n} + T_{m+1,n}}{\Delta x^2}}$$

Ainsi, l'équation de la diffusion 1D 1 discrétisée s'écrit:

$$\frac{T_{m,n+1} - T_{m,n}}{\Delta t} = \frac{D}{\Delta x^2} \cdot \left[T_{m+1,n} - 2T_{m,n} + T_{m-1,n}\right] \tag{4}$$

d'où l'on tire finalement la relation de récurrence permettant d'obtenir la température en x_m à l'instant t_{n+1} en fonction des températures $T_{m,n}$, $T_{m+1,n}$, $T_{m-1,n}$ calculées à l'instant antérieur t_n:

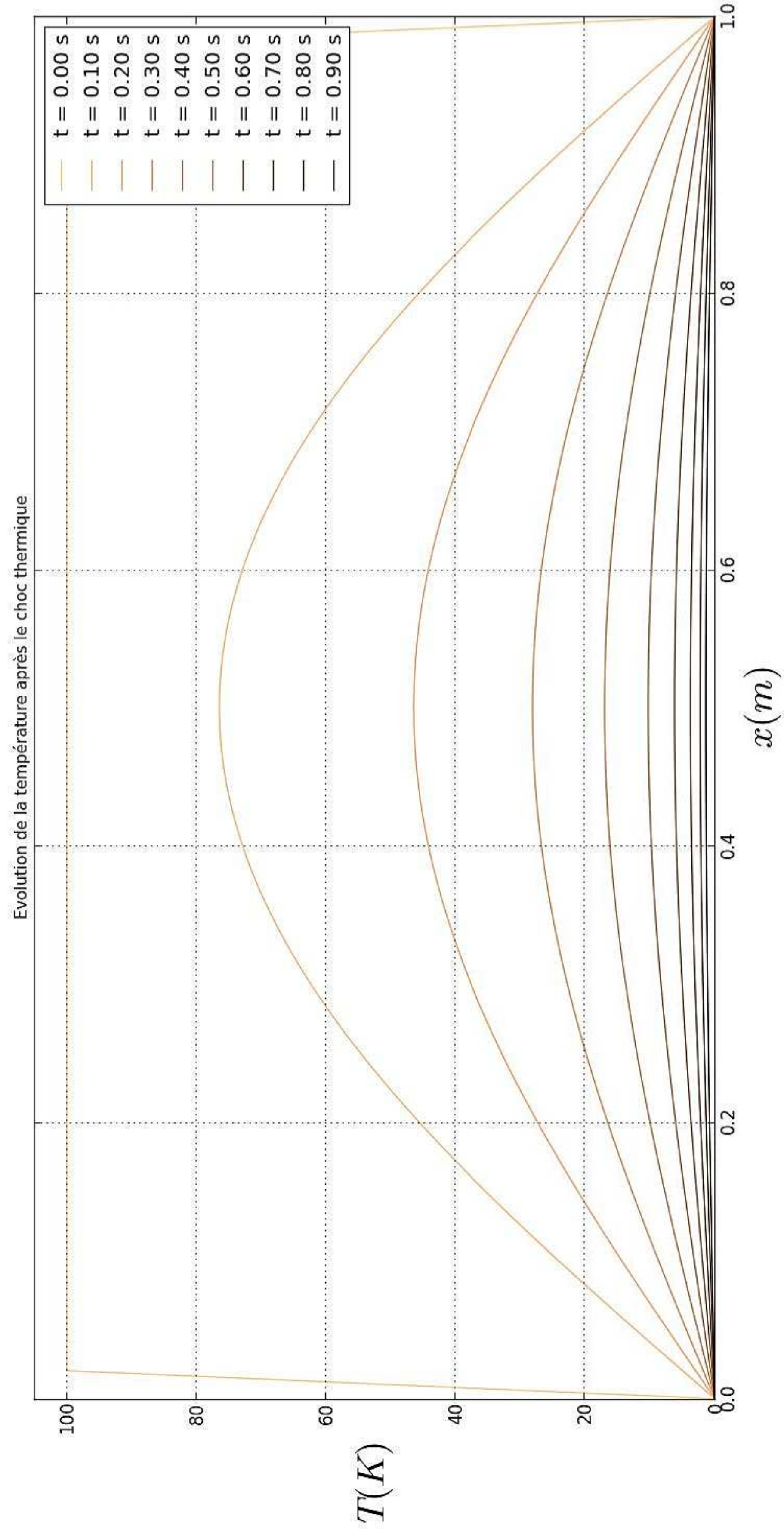
$$\boxed{T_{m,n+1} = T_{m,n} + \frac{D\Delta t}{\Delta x^2} \cdot \left[T_{m+1,n} - 2T_{m,n} + T_{m-1,n}\right]} \tag{5}$$

<u>Réalisation:</u> (comme toujours en Python!!!)

Script Python Résolution de l'équation de diffusion thermique 1D

```python
# -*- coding: utf-8 -*-
#from math import *
import numpy as np
import matplotlib.pyplot as plt

#DonnÃ©es numÃ©riques
L=1.0           #Longueur de la barre
tau=1.0           #Duree totale evolution
nx=100          #Nombre de troncons
nt=10000        #Nombre intervalles de temps
Deltax=L/nx     #Longueur tronÃ§on
Deltat=tau/nt     #Intervalle elementaire de temps
D=0.5           #Coefficient de diffusion thermique
TB=100.0        #Temperature initiale de la barre
Textr=0.0       #Temperature des extremites

#Construction de axe des abscisses (variable espace x)
x=np.linspace(0.0,L,nx)

#Construction CI,CL (2 exemples pour la CI)
T=[Textr]+(nx-2)*[TB]+[Textr] #Echelon de temperature
#T=Textr+(TB-Textr)*np.sin(np.pi*x/L) # Arche sinusoidale temperature

#Construction du tableau vierge des accroissements de temperature
accroissT=np.zeros(nx)

#Corps de la rÃ©solution
for n in range(nt):           #boucle evolution du temps pas Ã  pas
    for m in range(1,nx-1): #boucle de calcul de accroissement de temperature pour chaque abscisse
        accroissT[m]=((Deltat*D)/(Deltax**2))*(T[m-1]-2*T[m]+T[m+1])
    for m in range(1,nx-1): #boucle de calcul de T instant suivant
        T[m] += accroissT[m]

#Trace tous les 1000 intervalles de temps
    if (n%1000 == 0):
        plotlabel = "t␣=␣%1.2f␣s" %(n * Deltat)
        plt.plot(x,T, label=plotlabel,color = plt.get_cmap('copper')(1-float(n)/nt))

#Trace evolution temperature
plt.grid()
plt.xlabel(r'$x(m)$', fontsize=26)
plt.ylabel(r'$T(K)$', fontsize=26, rotation=0)
plt.title(u"Evolution␣de␣la␣tempÃ©rature␣aprÃ¨s␣le␣choc␣thermique",size=11)
plt.axis([0,L,0,105])
plt.legend()
plt.show()
```

Listing 1: Sources_Python/Resolution_equation_diffusion_thermique_1D.py


Evolution de la température après le choc thermique
t = 0.00 s
t = 0.10 s
t = 0.20 s
t = 0.30 s
t = 0.40 s
t = 0.50 s
t = 0.60 s
t = 0.70 s
t = 0.80 s
t = 0.90 s
$T(K)$
$x(m)$
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
1.0