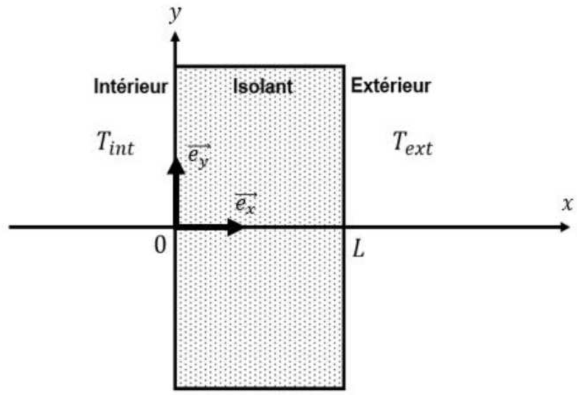
Modélisation: résolution de l'équation de diffusion

On s'intéresse aux transferts thermiques dans un isolant d'épaisseur L, de masse volumique ρ et de capacité thermique massique c_p.

Dans la situation étudiée, la température intérieure est $T(t, x = 0) = T_{int} = 20^0C$ et la température extérieure est $T(t, x = L) = T_{ext} = 5^0C$. On supposera ces températures constantes et uniformes sur toute la surface de la paroi. L'isolant est initialement à la température $T(t = 0, x > 0) = T_{ext} = 5^0C$.

On souhaite étudier l'évolution de la température dans le mur au cours du temps.



1. L'équation de diffusion thermique s'écrit $\dfrac{\partial T}{\partial t} = \dfrac{\lambda}{\rho c_p}\Delta T$.

1.a. La simplifier dans le cas où $T = T(x, t)$.

1.b. Exprimer le temps τ donnant un ordre de grandeur de la durée du régime transitoire.

1.c. Etablir l'expression de la température dans l'isolant en fonction de x, T_{int}, T_{ext} et L en régime permanent. Représenter la courbe donnant $T(x)$.

Etude numérique du régime transitoire

On cherche à résoudre numériquement l'équation aux dérivées partielles :

$$\frac{\partial T}{\partial t} = k_{th}\left(\frac{\partial^2 T}{\partial x^2}\right) \text{ où } k_{th} \text{ est une constante.}$$

2 Exprimer la diffusivité thermique k_{th} en fonction de la conductivité thermique $\lambda_{isolant}$, de la masse volumique ρ et de la chaleur spécifique massique à pression constante c_p.

On discrétise l'intervalle $[0, L]$, représentant l'épaisseur de l'isolant, en $N\_X + 1$ points régulièrement espacés d'un pas spatial dx (**figure 3**). On souhaite déterminer la température en chacun de ces points.

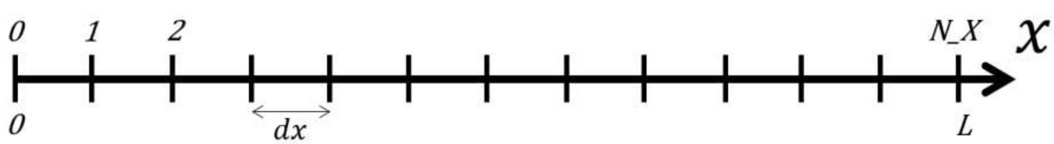


Figure 3 - Discrétisation de l'isolant selon $\boldsymbol{x}$

3 Donner le nombre d'intervalles spatiaux dans l'intervalle $[0, L]$. Donner l'expression de dx en fonction des données du problème. En déduire l'abscisse x_i du (i)-ème point.

4 À l'aide de la formule de Taylor-Young, **équation (1)**, exprimer :
- **a.** $T(t + dt, x)$, au premier ordre par rapport à t, dt étant l'incrément temporel ;
- **b.** $T(t, x - dx)$, au second ordre par rapport à x ;
- **c.** $T(t, x + dx)$, au second ordre par rapport à x.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + o((x-a)^2) \qquad \textbf{(1)}$$

5 . En déduire une expression de $\dfrac{\partial^2 T(t,x)}{\partial x^2}$ en fonction de dx, $T(t, x)$, $T(t, x - dx)$ et $T(t, x + dx)$.

La température à l'abscisse x_i à une date t_n sera notée : T_i^n. notations: x_i+1=x_i+dx et t_n+1=t_n+dt

Que représentent T_i^(n+1)? T_i+1^n? T_i-1^n?

6 En reformulant le résultat des **questions** Q4 et Q5, déterminer une relation entre :

a. $T_i^{n+1}, T_i^n, \dfrac{\partial T(t,x)}{\partial t}$ et dt ;

b. $T_{i+1}^n, T_{i-1}^n, T_i^n, \dfrac{\partial^2 T(t,x)}{\partial x^2}$ et dx.

7 Déduire des questions précédentes et de l'équation de diffusion , la relation

$$T_i^{n+1} = dt.k_{th}\left(\frac{T_{i+1}^n + T_{i-1}^n - 2T_i^n}{(dx)^2}\right) + T_i^n. \qquad \textbf{(2)}$$

Pour comprendre la suite de l'épreuve: le code propose d'utiliser pour la résolution un tableau 2D nommé Temp tel que $Temp[i,n] = T(t_n, x_i)$.
Rappeler les conditions aux limites et préciser la position (ligne, colonne?) dans le tableau Temp des termes associés.
Rappeler les conditions initiales et préciser la position (ligne, colonne?) dans le tableau Temp des termes associés.
Repérer sur le tableau le terme $Temp[i, n+1]$ et repérer les termes nécessaires pour son calcul par récurrence (relation (2)). Conclure

$$\begin{array}{ccc}
 & \begin{array}{ccccc} n=0 & n=1 & n=2 & n & j=Nt \end{array} & \\
\begin{array}{c} i=0 \\ i=1 \\ i=2 \\ \\ i \\ \\ i=Nx \end{array} & \left[\begin{array}{ccccc} & & & & \\ \\ \\ \\ \\ \\ \\ \end{array}\right] & \begin{array}{c} x= \\ x= \\ x= \\ \\ \\ \\ x= \end{array} \\
 & \begin{array}{ccccc} t= & t= & t= & & t= \end{array} &
\end{array}$$

Le code de l'**algorithme 1** permet de déterminer les valeurs de température aux points de discrétisation. Dans les questions suivantes, on cherchera à compléter les instructions manquantes.

8 Donner l'`Instruction 1` permettant de définir la diffusivité thermique k_{th}.

9 L'**équation (2)** est-elle valable pour toute valeur de $i \in \{0 \ldots N\_X\}$?

10 Définir les incréments de temps et d'espace en précisant les `Instruction 2.1` et `Instruction 2.2`. N_t intervalles seront réalisés dans l'intervalle de temps $[0; t\_max]$.

11 Déduire des CI et CL des données de l'énoncé les `Instruction 3.1`, `Instruction 3.2`, `Instruction 3.3` et `Instruction 3.4`.

12 À partir de la relation (2), compléter `Instructions 4.1`, `Instructions 4.2` et `Instructions 4.3`.

On donne en **figure 4** le profil de température dans le composite à plusieurs instants.

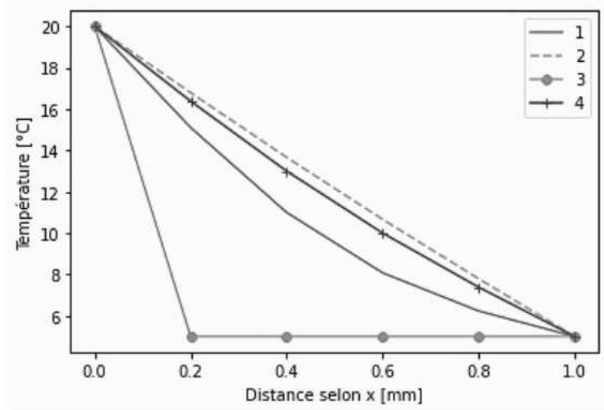


Figure 4 - Évolution de la température dans le composite à plusieurs instants

13 Associer à chaque courbe de la **figure 4** les instants de la liste suivante :
$t = [0\ s, 6\,000\ s, 12\,000\ s, 18\,000\ s]$.

14 Le régime permanent est-il atteint ? Justifier.

Code pyhton à compléter:

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

# Données du problème
Lambda = 0.037
Cp = 1500
Rho = 1.325
L = 1       #Epaisseur de l'isolant
t_max = 20000    #Temps de fin d'intégration en secondes
N_t = 100     #Nombre d'intervalles dans le temps
N_X = 5      #Nombre d'intervalles dans l'espace
T_int = 20
T_ext = 5
K = [Instruction 1]      #Diffusivité thermique

#Discrétisation de l'espace et du temps
dx = [Instruction 2.1]
dt = [Instruction 2.2]
Temp = np.zeros((N_t+1, N_X+1))
```

```
#Initialisation de la température
#Conditions initiales
Temp[0,0]=[Instruction 3.1]

for i in range(1,N_X+1):
    [Instruction 3.2]

#Conditions aux limites
for n in range(1,N_t+1):
    [Instruction 3.3]
    [Instruction 3.4]


#Calcul des températures aux différents instants
for n in [Instruction 4.1]:
    for i in [Instruction 4.2]:
        [Instruction 4.3]
```

Annexes pour la syntaxe python:

- **np.linspace(start, stop, N point) :**
  - description : renvoie un nombre d'échantillons espacés uniformément, calculés sur l'intervalle [start, stop]
  - argument d'entrée : début, fin et nombre d'échantillons dans l'intervalle
  - argument de sortie : un tableau

Commande	Résultat
`np.linspace(1, 4, 5)`	`[1., 1.75, 2.5, 3.25, 4.]`

- **np.zeros(i) :**
  - description : renvoie un tableau de taille i rempli de zéros.
  - argument d'entrée : un scalaire
  - argument de sortie : un tableau

Commande	Résultat
`np.zeros(5)`	`[0, 0, 0, 0, 0]`

- **A[i,j] :**
  - description : retourne l'élément (i + 1, j + 1) de la matrice A. Pour accéder à l'intégralité de la ligne i + 1 de la matrice A, on écrit A[i, :]. De même, pour obtenir toute la colonne j + 1 de la matrice A, on utilise la syntaxe A[: , j]
  - argument d'entrée : une liste contenant les coordonnées de l'élément dans le tableau A
  - argument de sortie : l'élément (i + 1, j + 1) de la matrice A

Commande	Résultat
`A = np.array([[1, 2, 1],[4, 6, 3], [1, 3, 8]])` `A[1, 2]`	3

Cette bibliothèque permet de tracer des graphiques. Dans les exemples ci-dessous, la bibliothèque `matplotlib.pyplot` a préalablement été importée à l'aide de la commande : `import matplotlib.pyplot as plt.`

- description : fonction permettant de tracer un graphique de n points dont les abscisses sont contenues dans le vecteur x et les ordonnées dans le vecteur y. Cette fonction doit être suivie de la fonction `plt.show()` pour que le graphique soit affiché
- argument d'entrée : un vecteur d'abscisses x (tableau de n éléments) et un vecteur d'ordonnées y (tableau de n éléments). La chaîne de caractères 'SC' précise le style et la couleur de la courbe tracée. Des valeurs possibles pour ces deux critères sont :

Valeurs possibles pour S (style) :

Description	Ligne continue	Ligne traitillée	Marqueur rond	Marqueur plus
Symbole S	-	- -	o	+

Valeurs possibles pour C (couleur) :

Description	bleu	rouge	vert	noir
Symbole C	b	r	g	k

- argument de sortie : un graphique

```
x= np.linspace(3,25,5)
y=sin(x)
plt.plot(x,y,'-b') # tracé d'une ligne bleue continue
plt.title('titre_graphique') # titre du graphe
plt.xlabel('x') # titre de l'axe des abscisses
plt.ylabel('y') # titre de l'axe des ordonnées
plt.show()
```