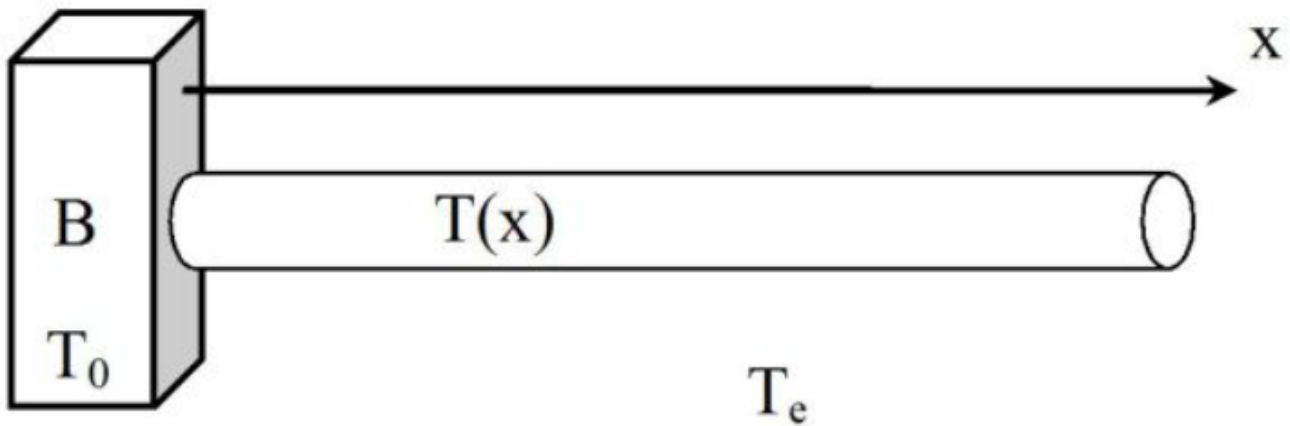


Ailette de refroidissement

I- Introduction :

On considère une ailette de refroidissement de longueur $L = 10\text{cm}$, de rayon $R = 1\text{cm}$ constitué d'un matériau solide de conductivité $\lambda = 100\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$, de capacité thermique massique $c = 100\text{J} \cdot \text{K}^{-1} \cdot \text{kg}^{-1}$ et de masse volumique $\rho = 10^4\text{kg} \cdot \text{m}^{-3}$.

En $x = 0$, la tige est à la température $T(0) = 60^\circ\text{C}$. On note $T(x)$ le champ des températures (supposé uniforme sur chaque section droite). On note $T_e = 20^\circ\text{C}$ la température extérieure.



II- Présentation

Les transferts conducto-convectifs et les échanges thermiques de type radiatifs de la tige sont intégrés dans un seul terme, noté P_S , traduisant la puissance thermique surfacique dissipée avec l'extérieur :

$$P_S = k (T(x) - T_e) \text{ où } k = 100\text{W} \cdot \text{K}^{-1} \cdot \text{m}^{-2}$$

- 1) Obtenir l'équation différentielle vérifiée par $T(x)$.
- 2) La solution $T(x)$ est du type $T(x) = Ae^{-\frac{x}{\delta}} + Be^{\frac{x}{\delta}} + C$ où A, B et C sont des constantes. Donner l'expression et la valeur de δ . Que vaut C ?
- 3) Proposer deux égalités permettant d'accéder à A et B.
- 4) Trouver l'expression $T(L)$ en résolvant le système ci-dessus à l'aide de python. On donne ci-dessous des éléments sur la fonction solve du module sympy.
- 5) Obtenir le profil $T(x)$. Commenter.

Entrée [12]:

```

1 from sympy import *
2 import numpy as np
3 import matplotlib.pyplot as plt
4 A = symbols("A")
5 B = symbols("B")
6 def sol(A,B,f,g):
7     dico = solve([f,g],[A,B]) #solve résoud le système d'équation f(A,B)=0 et
8     #la famille de solution obtenue se retrouve dans un dictionnaire dont les
9     print(dico)
10    print("A="+str(dico[A]))
11    print("B="+str(dico[B]))

```

On effectue un bilan enthalpique en régime stationnaire sur une tranche de longueur dx :

$$\left(\frac{d^2 T}{dx^2} \right)_x - \frac{T(x)}{\delta^2} = -\frac{T_{ext}}{\delta^2}$$

Avec $\delta^2 = \frac{\lambda * R}{2k}$

La solution particulière est T_{ext} et la solution de l'équation sans second membre est :

$$Ae^{-\frac{x}{\delta}} + Be^{\frac{x}{\delta}} \text{ donc } T(x) = Ae^{-\frac{x}{\delta}} + Be^{\frac{x}{\delta}} + T_{ext}$$

On peut utiliser les relations de continuité de la température et du flux :

C'est donc ce système qu'il faut résoudre avec Python :

$$T(0) = A + B + T_{ext}$$

$$-\lambda \frac{dT}{dx} \Big|_L = k(T(L) - T_{ext})$$

Entrée [13]:

```

1 from sympy import *
2 import numpy as np
3 import matplotlib.pyplot as plt
4 k = 100.
5 R=0.01
6 L = 0.10
7 Te = 20.
8 Ti = 60.
9 lam = 100.
10 rho=10**4
11 c=100
12 d = ((lam*R)/(2*k))**0.5
13 print ("d="+str(d))
14
15 A = symbols("A")
16 B = symbols("B")
17 f=A+B*Te-Ti
18 g=k*(A*np.exp(-L/d)+B*np.exp(L/d))+lam/d*(B*np.exp(L/d)-A*np.exp(-L/d))
19 sol(A,B,f,g)

```

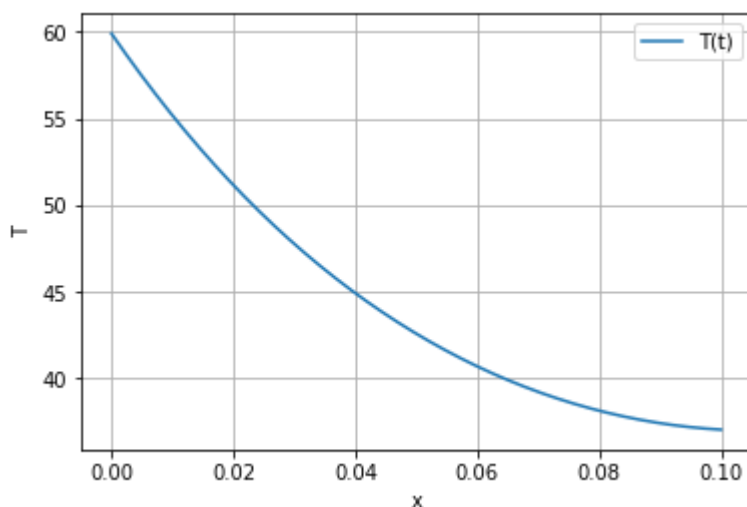
d=0.07071067811865475
{B: 1.95183130261749, A: 38.0481686973825}
A=38.0481686973825
B=1.95183130261749

Entrée [14]:

```

1 x = np.linspace(0, 0.10, 1000)
2 T = 38*np.exp(-x/d)+1.9*np.exp(x/d)+Te
3 plt.plot(x,T,label="T(t) ")
4 plt.xlabel("x")
5 plt.ylabel("T")
6 plt.legend()
7 plt.grid()
8 plt.show()

```



On a :

- une température au bout de la tige qui n'est pas la température extérieure
- on a une tige qui évacue des calories sur toute sa longueur

Entrée []:

1	
---	--