

# Utilisation d'isolants biosourcés

- ▶ Isolation : Une des solutions face au changement climatique
- ▶ Choix de l'isolant (des isolants non polluants)
- ▶ Intérieur /extérieur : quel type d'isolation choisir ?

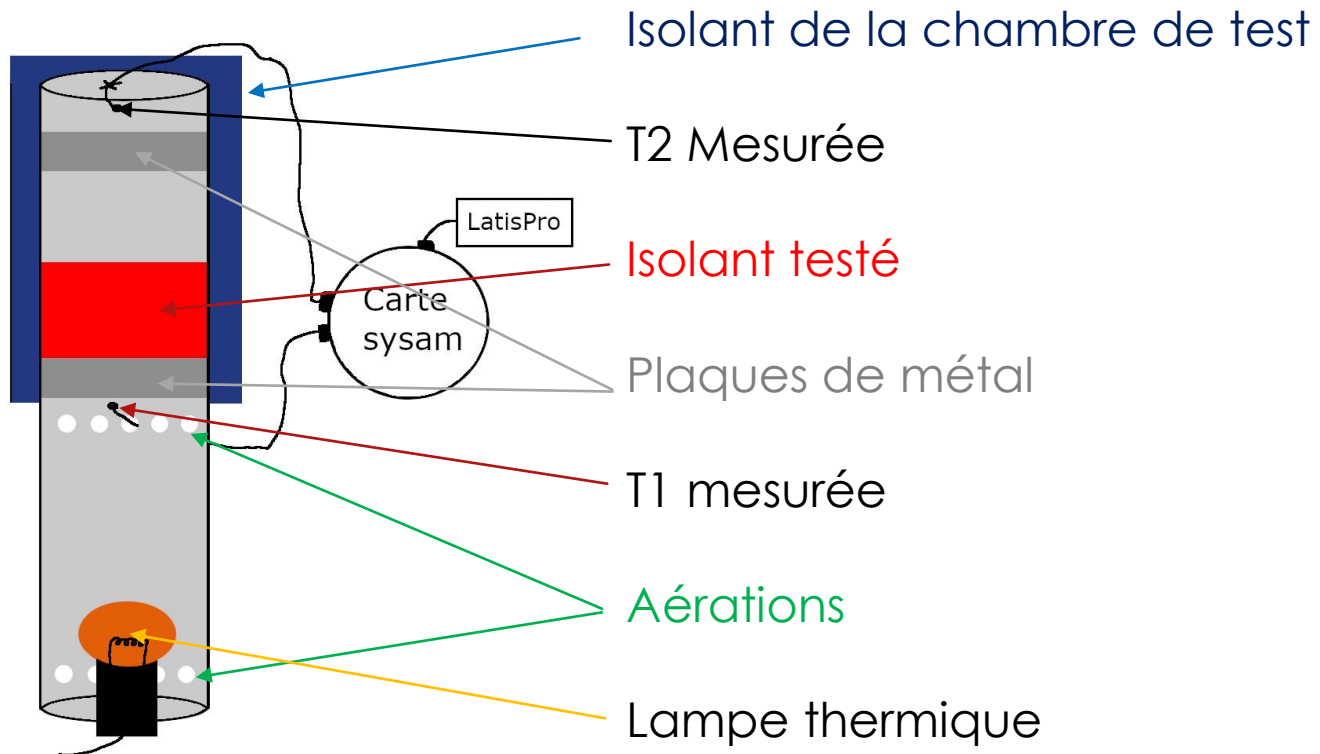
Peut-on envisager des isolants biosourcés pour une isolation extérieure ?

- ▶ I – Conception / réalisation :
  - ▶ Choix des isolants
  - ▶ Dispositif expérimental choisi
  - ▶ Simulation des conditions extérieures
  - ▶ Approche théorique
  
- ▶ II – Expérimentation et résultats :
  - ▶ Protocole expérimental
  - ▶ Comparaison des isolants choisis
  - ▶ Effet de l'humidité

# Choix des isolants

	Polystyrène	Fibres de bois	Fibres végétales (Thermasoft)	Coton Recyclé
	Isolant Synthétique	Isolant biosourcé		
Résistance thermique par $m^2$ . (K/W)	1,2	1,05	1,15	1,15
Impact écologique	- - -	+ +	+ + +	+ + +
Perméabilité	+ + +	- - -	- - -	- - -
Classement européen de réaction au feu	E Inflammable et toxique	De A à E Voir cave	F	E

# Dispositif expérimental choisi



# Simulation des conditions extérieures

7

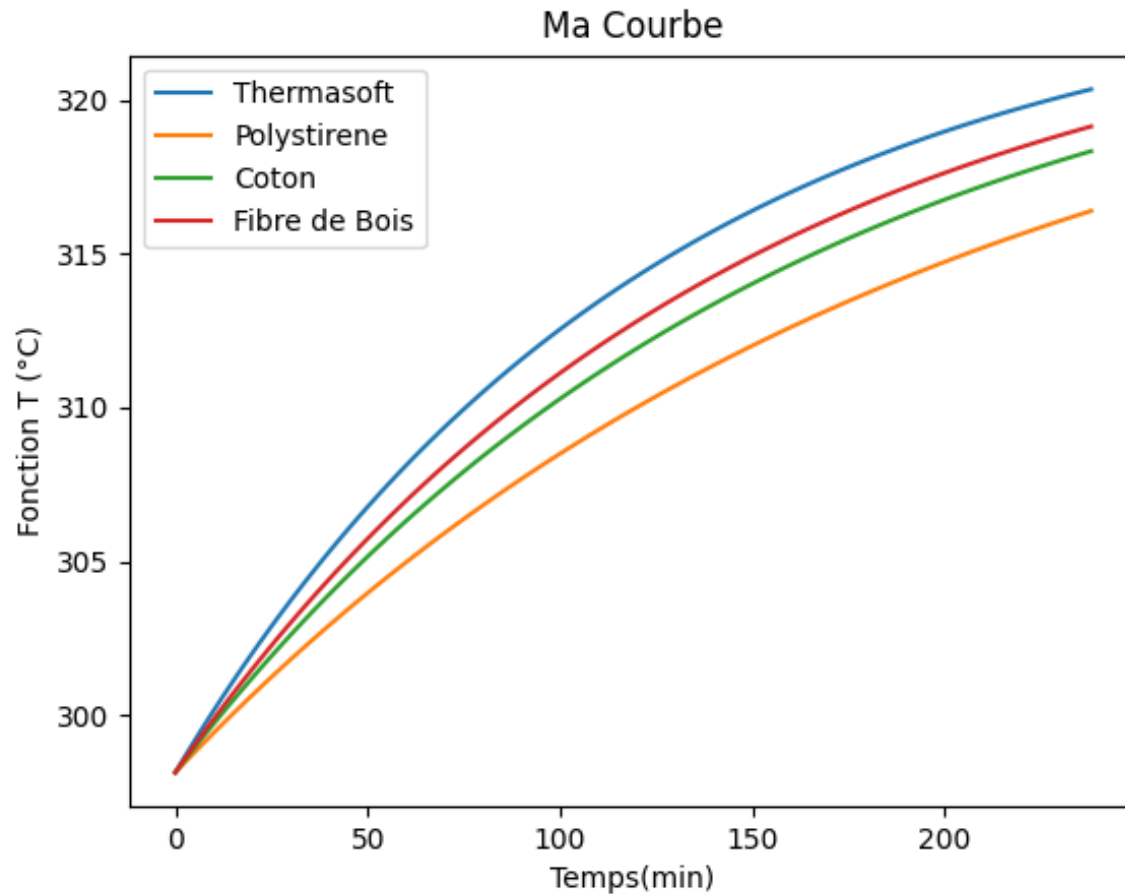
- ▶ Rayonnement
  - ▶ Bardage pour protéger des rayonnements



- ▶ Pluie
  - ▶ Exemple d'un mois de pluie en automne en Ile et Vilaine :  
Moyenne : 3 mL
  - ▶ Exemple d'un mois de pluie au printemps en Ile et Vilaine :  
Moyenne : 2 mL

# Approche théorique

8

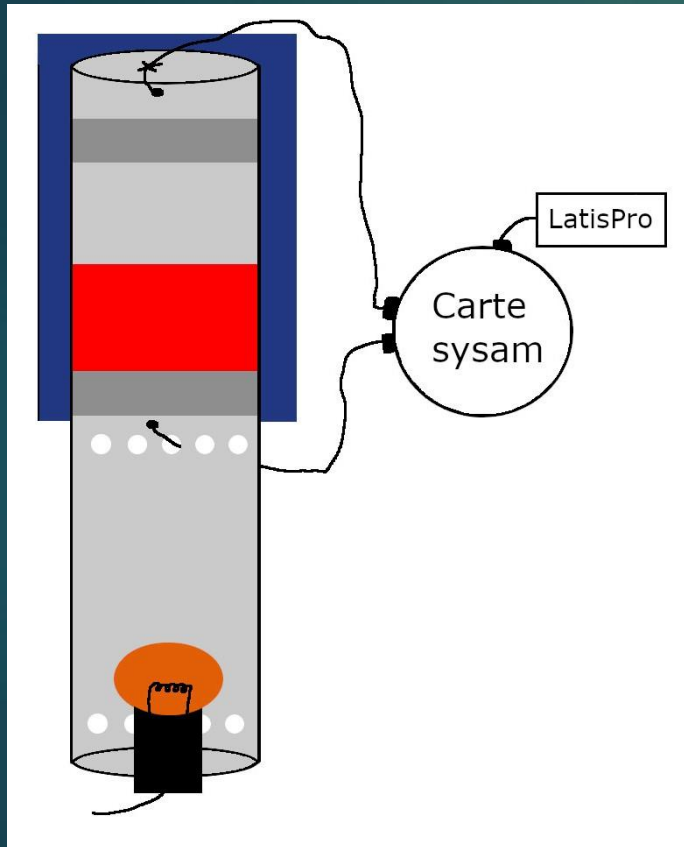


$$T_2(t) = T_1 + (T_{\text{init}} - T_{\text{ext}})e^{-t/R_{\text{th}}C}$$



# Protocole expérimental

9



## Experiences :

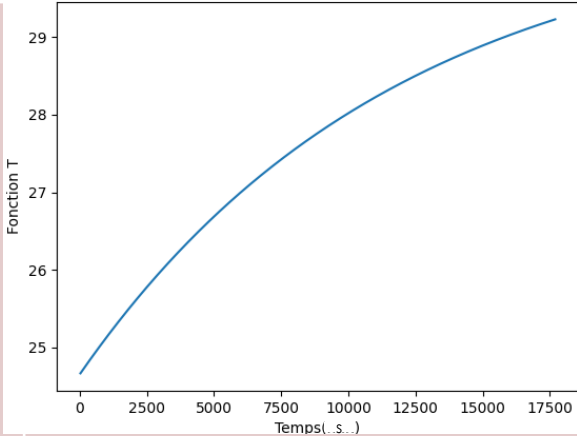
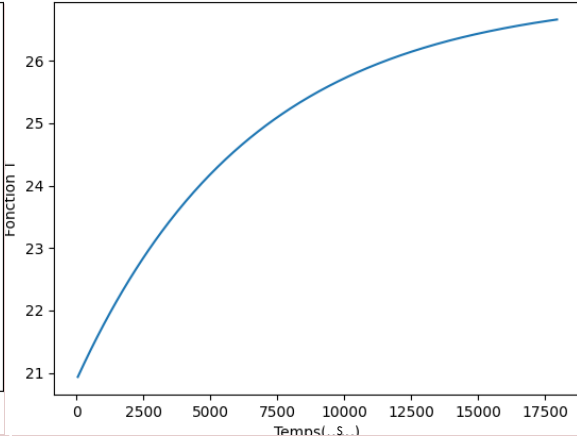
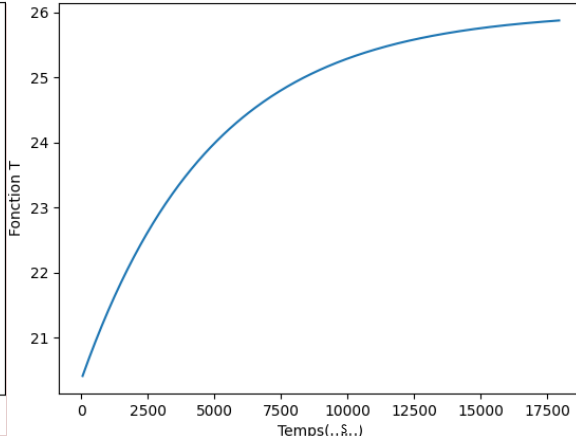
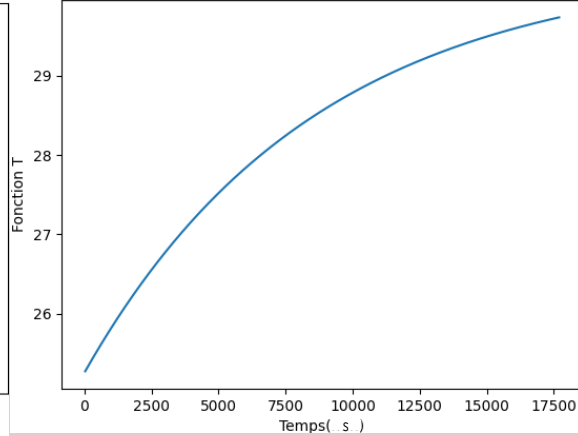
- Sur Latispro : Mesurer T1 et T2
- Lancer la prise des mesures de température sur Latispro et allumer la lampe
- Durée des prises de données : 5h
- Modéliser les courbes obtenues

## Exploitation :

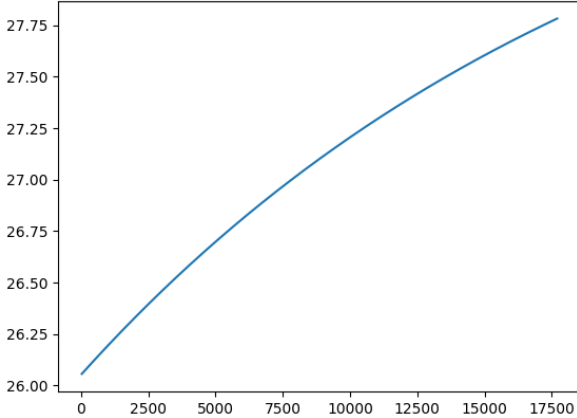
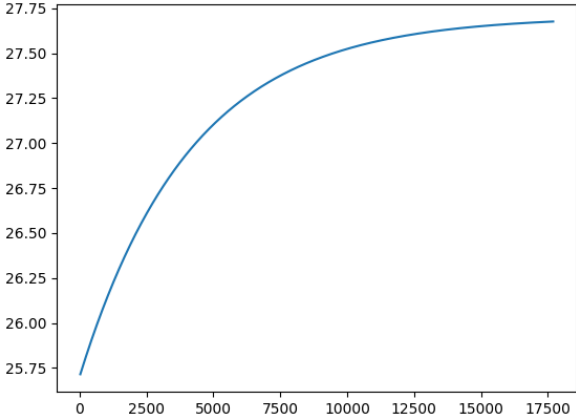
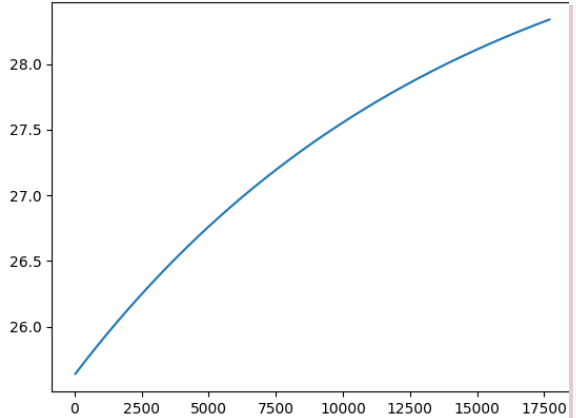
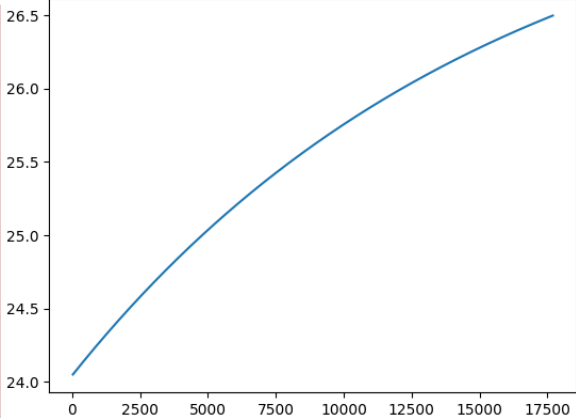
- A  $t = 1h$  : T1 constant,  $T2 = T_{initial}$
- A l'aide de la courbe obtenue, on obtient la résistance expérimentale de l'isolant étudié
- Comparer les résistances expérimentales aux résistances théoriques pour valider la fonction théorique.
- Etudier les résistances obtenues avec ou sans humidité pour connaître son effet sur l'isolant.

# Comparaison des isolants choisis sans eau

10

	Polystyrène	Fibres de bois	Fibres végétales et recyclée	Coton
Courbes				
Rth (W/K)	72,03	63,13	61,24	72,72
Rth * (W/K)	72,03 ± 0,73	63,13 ± 0,64	61,24 ± 0,62	72,72 ± 0,73

# Comparaison des isolants choisis avec eau

	Polystyrène	Fibres de bois	Fibres végétales et recyclés	Coton
Courbes				
	Temps en s	Temps en s	Temps en s	Temps en s
Rth (W/K)	214,93	269,92	135,00	136,04
Rth * (W/K)	214,93 ± 2,15	269,92 ± 2,7	135,00 ± 1,36	136,04 ± 1,37

# Conclusion

- ▶ La résistance thermique est plus élevée pour tous les isolants avec de l'eau alors que c'est une baisse qui était attendue.
- ▶ Pour chacune des expériences il y a toujours un isolant biosourcé qui est équivalent ou plus performant que le polystyrène.
- ▶ Je n'ai pas pu tester l'effet à long terme de l'eau sur les isolants.
- ▶ On peut s'interroger sur les résultats que l'on obtiendrait en faisant la combinaison de plusieurs isolants biosourcés

# Annexe 1 :

## Tracer les courbes théoriques

```
def KtoC(T):
    return T-273.15
```

```
# heure en minutes
def HtoMin(n) :
    return n*60
```

```
#Fonction qui calcule la valeur de C selon Text
def C(Text) :
    return (65*10**(-3)*(7.85*10**(-3)))/(Text*0.4)
```

```
# Fonction qui renvoie le tableau T via Tinit,T1,lbd et n = nombre de mesures souhaité
def T(n,Tinit,T1,Text,lbd) :
    result = []
    for k in range(n) :
        result.append(KtoC(T1+(Tinit-T1)*math.exp(-k*(Rth(lbd)*C(Text)))))
    return result
```

```
# Fonction renvoyant le Rth selon la permittivité thermique de l'isolant (appelé ici "lbd" aka lambda)
def Rth(lbd) :
    li = 45*10**(-3)
    S = 7.85*10**(-3)
    h = 10
    lp = 2*10**(-3)
    lbdp = 235
    rp = lp/(lbdp*S)
    rcc = 1/(h*S)

    return 2*rp+3*rcc+(li/(lbd*S))
```

```
# Fonction qui renvoie la courbe T(t,Tinit,T1,lbd) en fonction de t, ou t = tableau contenant tout les temps lors desquels une mesure est effectuée
def TracerCourbe(t,Tinit,Text,T1,lbd):
    fig, ax = plt.subplots()
    ax.set_title("Ma Courbe")
    ax.set_xlabel("Temps(min)")
    ax.set_ylabel("Fonction T")
    ax.plot(t, T(len(t),Tinit,T1,Text,lbd))
    plt.show()
```

# Annexe 2 : (suite)

```
def TracerCourbeComp(t, repet):
    fig, ax = plt.subplots()
    ax.set_title("Ma Courbe")
    ax.set_xlabel("Temps(min)")
    ax.set_ylabel("Fonction T (°C)")
    #ax.plot(t, T(len(t), TInit, T1, lbd), label='Que')
    for i in range(repet):

        Tinit = input("Tinit (en Kelvin svp) : ")
        MTI = float(Tinit)

        lbd2 = input(" Lambda : ")
        Mlbd = float(lbd2)

        Text = input(" Text (en Kelvin svp): ")
        Text2 = float(Text)

        T1 = input(" T1 (en Kelvin svp): ")
        T12 = float(T1)
        nom = input("Nom de l'isolant: \n")
        print("_____")
        ax.plot(t, T(len(t), MTI, T12, Text2, Mlbd), label=nom)
    plt.legend()
    plt.show()
```

# creer un tableau qui avance par palier de 2 au lieu de 1

```
def Tab2a2(n) : # n = taille du tableau
    result = []
    i=0
    while i<(n/2) :
        result.append(i*2)
        i+=1
    return result
```

# // \_\_\_\_\_ 5 \_\_\_\_\_ 1

```
def Tab5a5(n) :
    result = []
    i = 0
    while i<n/5 :
        result.append(i*5)
        i+=1
    return result
```

```
Mode = input("Mode Courbes Comparées? Y/N: \n")
```

```
if Mode == "N":
    t = input("Temps(en heures) : ")
    temps = HtoMin(int(t))
    print(temps)

    taille = temps

    DeuxADeux = Tab2a2(taille)

    print("_____")

    CinqACinq = Tab5a5(taille)

    print("_____")

    MyLambda = input(" Lambda : ")
    print(Rth(float(MyLambda)))

    MTI = input(" TInitial (en Kelvin svp): ")

    Text = input(" Text (en Kelvin svp): ")

    T1 = input(" T1 (en Kelvin svp): ")
    print("_____")
    TracerCourbe(DeuxADeux, float(MTI), float(Text), float(T1), float(MyLambda))
```

```
else :
    t = input("Temps(en heures) : \n")
    temps = HtoMin(int(t))
    print(temps)

    print("_____")

    taille = temps

    print("_____")

    DeuxADeux = Tab2a2(taille)
    #print(DeuxADeux)

    print("_____")

    N = input("Combien de courbes aller vous comparer: \n")

    TracerCourbeComp(DeuxADeux, int(N))
```

# Annexe 3 :

## Afficher les courbes CSV avec python

```
1  import fileinput
2
3  # Input file path
4  input_file = "CourbesCSV\Courbes_exp_Polystirene.csv"
5
6  # Modify the CSV file in-place
7  for line in fileinput.input(input_file, inplace=True):
8      modified_line = line.replace(',', ' ')
9      print(modified_line, end='')
10
11 print("CSV file processing complete.")
```

# Annexe 4 :

## Afficher les courbes CSV avec python

```
Mode = input("Mode Courbes Comparées? Y/N: \n")

if Mode == "N":
    MonTabCsv = input("Chemin du fichier(.csv) de votre Tableau: ")
    with open(MonTabCsv,'r') as f:
        obj = csv.DictReader(f)
        ligne1=0
        count=0
        tps=[]
        Thermo=[]
        for i in obj:
            count+=1
            if ligne1 == 1 : # sauter la 1ere ligne
                rowli = i["Temps;Thermocouple"].split(';')
                tps.append(float(rowli[0]))
                Thermo.append(float(rowli[1]))
            ligne1 = 1
            print("_____")
        TracerCourbe(tps,Thermo)
else :
    print("_____")

    N = input("Combien de courbes aller vous comparer: \n")
    TracerCourbeComp(int(N))
```

```
# Fonction qui renvoie la courbe T(t,Th) en fonction de t, ou t = tableau contenant tout les temps lors desquels une mesure est effectuée
def TracerCourbe(t,Th):
    fig, ax = plt.subplots()
    ax.set_title("Ma Courbe")
    ax.set_xlabel("Temps(min)")
    ax.set_ylabel("Fonction T")
    ax.plot(t,Th)
    plt.show()
```

```
def TracerCourbeComp(repet):
    fig, ax = plt.subplots()
    ax.set_title("Ma Courbe")
    ax.set_xlabel("Temps(min)")
    ax.set_ylabel("Fonction T (°C)")
    #ax.plot(t, T(len(t),TInit,T1,lbd), label='Que')
    for i in range(repet):
        MonTabCsv = input("Chemin du fichier(.csv) de votre Tableau: ")
        with open(MonTabCsv,'r') as f:
            obj = csv.DictReader(f)
            ligne1=0
            tps=[]
            Thermo=[]
            for i in obj:
                print(i)
                if ligne1 == 1 : # sauter la 1ere ligne
                    rowli = i["Temps;Thermocouple"].split(";")
                    tps.append(float(rowli[0]))
                    Thermo.append(float(rowli[1]))
                    valueE = (rowli[0],rowli[1])
                ligne1 = 1
            nom = input("Nom de l'isolant: \n")
            print("_____")
            ax.plot(tps,Thermo,label=nom)
    plt.legend()
    plt.show()
```



# Annexe 5 : Pluviomètre Ile et vilaine

16

► Eau tombée du 6 octobre au 6 novembre 2022 : Automne Moyenne 2,75 ml

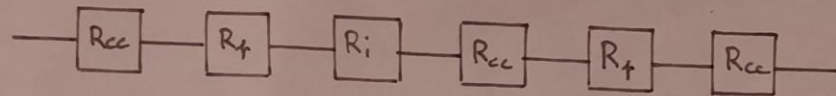
Date	Octobre															Novembre						31 jours
	6	14	15	16	17	18	19	20	21	22	23	24	25	27	30	1	2	3	4	5	6	
Quantité d'eau en mm pour 1m²	1	5	11	0,5	2	1	0,5	26	0,5	2	0,5	1	0,5	1	4	1	5	6	16	1	15	100,5

► Eau tombée du 24 avril au 15 mai : Printemps Moyenne 2,06 ml

Date	Avril				Mai											31 jours
	24	26	27	29	2	3	6	7	8	9	10	11	12	13	15	
Quantité d'eau en mm pour 1m²	5	9	2	0,5	1	2	6	1	2	7	14	0,5	8	3	3	64

# Annexe 6 : Calculs détaillés

## ► Par analogie avec l'électronique



$$R_j = \frac{P_j}{\lambda_j S} \quad \text{et} \quad R_{cc} = \frac{1}{hS}$$

$$R_{th} = 2R_f + 3R_{cc} + R_i$$

On applique le 1<sup>er</sup> principe à la partie supérieure de la chambre de Test (2)

$$\frac{dT_2}{dt} + \frac{1}{R_{th}C} T_2 = \frac{T_1}{R_{th}C} \rightarrow \tau \quad \text{ou} \quad \tau = R_{th}C$$

$$C = \frac{P_{ext} V}{RT} \times \frac{R}{\gamma - 1}$$

$$T_{2H} = A e^{-t/\tau} \quad \text{solution de l'équation homogène}$$

$$T_{2p} = T_1 \quad \text{solution de l'équation particulière}$$

$$\Rightarrow T_2 = T_1 + A e^{-t/\tau}$$

CI:

$$\text{à } t=0, T_2 = T_i = T_1 + A e^0$$

$$\text{Donc } A = T_i - T_1$$

$$T_2 = T_1 + (T_i - T_1) e^{-t/\tau}$$

