

SAE 105

16/12/25

# TP2

## BALLON SONDE

FATET GAETAN

MARESIA ENZO

HOUNTONDJI LIONEL

# Introduction

Chaque année, un ballon sonde est lâché par l'IUT afin de collecter des données atmosphériques à l'aide de différents capteurs (altimètre, GPS, thermomètres interne et externe, etc.). Lors du lâcher d'avril 2024, ces capteurs ont enregistré plusieurs paramètres physiques tout au long de l'ascension et de la descente du ballon. Notre objectif dans ce TP est d'analyser ces données afin d'étudier l'évolution des températures interne et externe en fonction de l'altitude, et de déterminer les altitudes minimale et maximale ainsi que les températures minimale et maximale mesurées.

## Exercice 1 : Problème ouvert

L'intégralité du traitement est réalisée en Python. Le rapport présente la problématique, la démarche suivie, les graphiques obtenus et une conclusion. L'ensemble des fichiers du projet est géré via GitHub comme demandé dans le cahier des charges.

## A- Températures en fonction de l'altitude

### 1-Etude des températures extérieures en fonction de l'altitude :

Dans notre code python, nous avons :

- Importé notre fichier csv dans la liste data
- Supprimé la première ligne
- Importé ensuite la colonne 3 de la liste data dans la liste Altitude
- Ainsi que la colonne 6 de la liste data dans la liste tempext

```
import csv
import matplotlib.pyplot as plt
data = []
with open("Donnees.csv", newline= '') as csvfile :
    reader=csv.reader(csvfile,delimiter=',')
    for row in reader:
        data.append(row)
#print(data)
del(data[0])

#Ici, on importe notre fichier csv dans la liste data à laquelle
#on supprime la première ligne

alt = []
for i in range(len(data)):
    alt.append(float(data[i][3]))
#print(alt)

#On importe donc la colonne 3 de la liste data dans la liste altitude

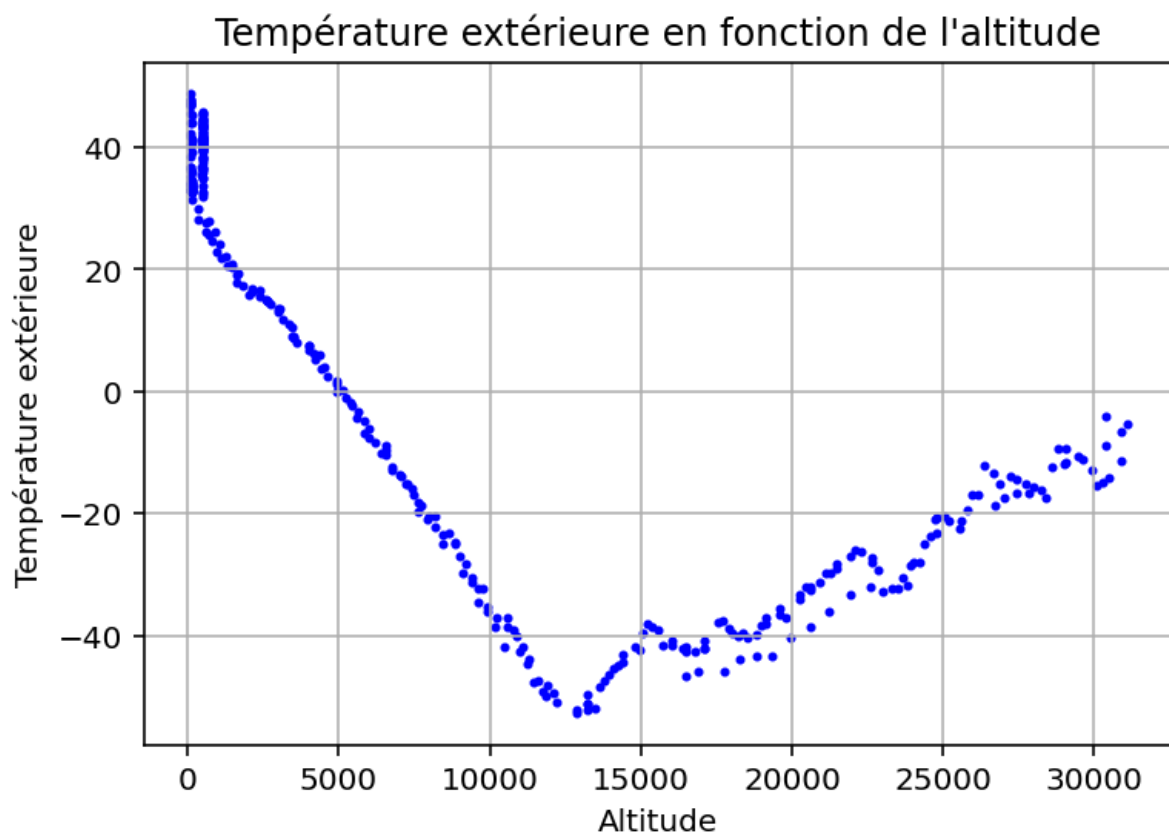
tempext = []
for i in range(len(data)):
    tempext.append(float(data[i][6]))
#print(tempext)

#Ainsi, nous pouvons choisir de garder la colonne 6 (ici des floats) représentant
#la température extérieure.

plt.scatter(alt,tempext,s=4)
plt.xlabel("Altitude")
plt.ylabel('Température extérieure')
plt.title("Température extérieure en fonction de l'altitude")
plt.show()
```

Ainsi on a pu obtenir la courbe suivante :

Sur cette courbe en fait nous pouvions remarquer un aller-retour de la température. Ceci se traduit par le fait qu'au-delà de 30.000 m, la courbe de la température rebrousse chemin (le deuxième tracé en pointillé en dessous) autour des mêmes valeurs qu'au départ. En effet, nous avons pu remarquer sur le fichier.csv que le ballon montait dans les airs pour redescendre une fois le ballon éclaté, c'est pourquoi nous avons ici 2 courbes. Cependant, étant donné que cette courbe représente la température extérieure, alors nous pouvons remarquer que ces valeurs ne diffèrent pas grandement pendant la montée de pendant la descente. Ainsi, une fois le ballon éclaté, les instances des relevés ne changent pas mais le ballon redescend plus vite grâce à la gravité, nous montrant ici que la deuxième courbe comprend beaucoup moins de points.

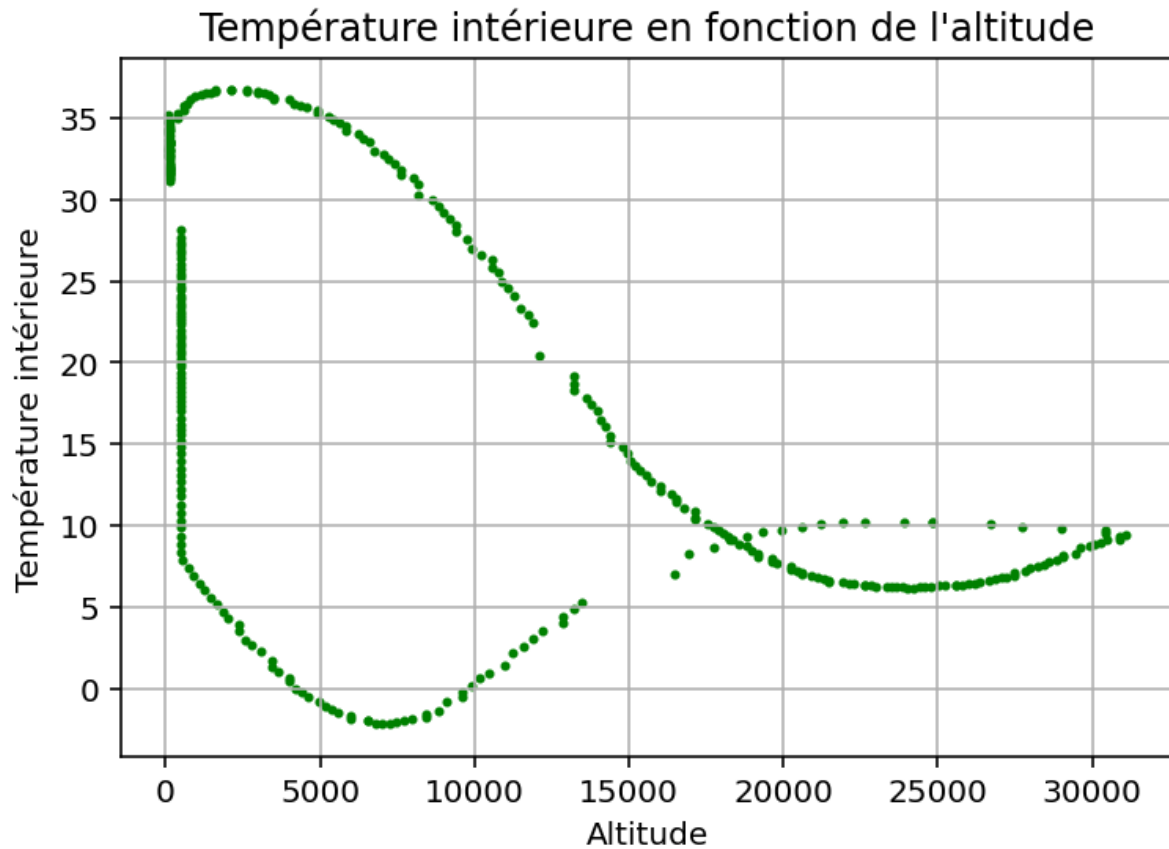


## 2-Etude des températures intérieures en fonction de l'altitude

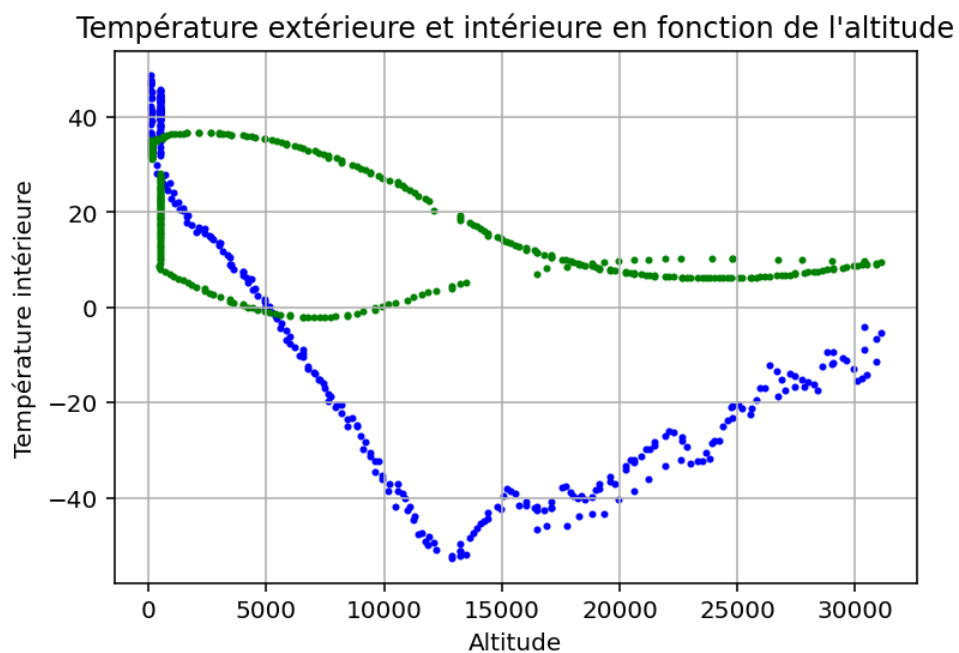
Ici en revanche nous avons plutôt fait l'étude la température intérieure en fonction de l'altitude. En effet, l'utilisation de matplotlib est nécessaire pour tracer des graphiques et des courbes relatives aux valeurs comprises dans nos listes de températures ou d'altitude.

```
tempint = []  
for i in range(len(data)):  
    tempint.append(float(data[i][5]))  
  
#print(tempint)  
  
#Nous utilisons ici, la colonne 5 représentant la température intérieure.  
  
plt.scatter(alt,tempext,s=4)  
plt.xlabel("Altitude")  
plt.ylabel('Température extérieure')  
plt.title("Température extérieure en fonction de l'altitude")  
plt.grid()  
plt.show()  
  
plt.scatter(alt,tempint,s=4)  
plt.xlabel("Altitude")  
plt.ylabel('Température intérieure')  
plt.title("Température intérieure en fonction de l'altitude")  
plt.grid()  
plt.show()
```

Sur la courbe ci-dessous, nous pouvons remarquer un aller-retour de la température. Ceci se traduit par le fait qu'au-delà de 30.000 m, la courbe de la température rebrousse chemin (le deuxième tracé en pointillé en dessous, de 10° puis remonte jusqu'en dessus de 35°). Nous pouvons également notifier que la température intérieure du ballon ne descend jamais en dessous de 0°C.



Puis, nous nous sommes demandés ce que ces deux courbes pouvaient représenter dans un même graphique. Nous pouvons voir ici que comme dit précédemment, la deuxième courbe, ici en verte ne descend pas en dessous de 0°C. C'est pourquoi le relevé semble aussi différent entre les deux courbes.



## B- Relevé des valeurs maximales et minimale

Dans le code suivant, nous avons choisi de garder le code précédent tout en ajoutant un calcul des valeurs maximales et minimales dans leurs boucles. En effet, nous avons pu remarquer que les calculs se faisaient également pour toutes les valeurs dans le fichier (donc pour toute la longueur de la liste data). C'est pourquoi nous avons juste eu à incorporer les comparaisons ainsi que les variables que nous voulions étudier.

```
import csv
import matplotlib.pyplot as plt
data = []
with open("Donnees.csv", newline= '') as csvfile :
    reader=csv.reader(csvfile,delimiter=',')
    for row in reader:
        data.append(row)
#print(data)
del(data[0])

#Ici, on importe notre fichier csv dans la liste data à laquelle
#on supprime la première ligne

alt = []
maxalt = 0
minalt = 1000000
for i in range(len(data)):
    alt.append(float(data[i][3]))
    #On importe donc la colonne 3 de la liste data dans la liste altitude
    if alt[i]>maxalt:
        maxalt = alt[i]
    if alt[i]<minalt:
        minalt = alt[i]
    #Puis nous calculons les valeurs minimales et maximales de l'altitude
print("L'altitude maximale mesurée est ", maxalt)
print("L'altitude minimale mesurée est ", minalt)
#print(alt)
```

```
tempext = []
maxext = 0
minext = 1000000
for i in range(len(data)):
    tempext.append(float(data[i][6]))
    #Ainsi, nous pouvons choisir de garder la colonne 6 (ici des floats) représentant
    #la température extérieure.
    if tempext[i]>maxext:
        maxext = tempext[i]
    if tempext[i]<minext:
        minext = tempext[i]
    #Puis nous calculons les valeurs minimales et maximales de la température extérieure
print("La température extérieure maximale mesurée est ", maxext)
print("La température extérieure minimale mesurée est ", minext)
#print(tempext)

tempint = []
maxint = 0
minint = 1000000
for i in range(len(data)):
    tempint.append(float(data[i][5]))
    #Nous utilisons ici, la colonne 5 représentant la température intérieure.
    if tempint[i]>maxint:
        maxint = tempint[i]
    if tempint[i]<minint:
        minint = tempint[i]
    #Puis nous calculons les valeurs minimales et maximales de la température intérieure
print("La température intérieure maximale mesurée est ", maxint)
print("La température intérieure minimale mesurée est ", minint)
#print(tempint)
```

```
L'altitude maximale mesurée est 31087.7
L'altitude minimale mesurée est 103.6
La température extérieure maximale mesurée est 48.7
La température extérieure minimale mesurée est -52.8
La température intérieure maximale mesurée est 36.7
La température intérieure minimale mesurée est -2.2
```

Conclusion :

En somme nous avons tout d'abord fais une analyse de l'évolution de la température (interne et externe) du ballon en fonction de l'altitude. Puis nous avons déterminé les valeurs maximales et minimales des températures et des altitudes. Nous avons pu remarquer que nos valeurs concordaient de façon plutôt logique sur les valeurs de températures en fonction de l'altitude du ballon. Une fois après avoir comparé les valeurs de l'intérieur du ballon à celles de l'extérieur du ballon, nous avons pu remarquer que globalement, la température intérieure était supérieure car elle ne descendait jamais en dessous de 0°C pour toujours garder les capteurs fonctionnels.