

La Ville : Les Îlots de chaleur Urbain

Comment limiter les
phénomènes des ICU
par la couleur ?

L'influence de la
couleur sur la
température

Baraldi Enzo
14 730



Objectif du TIPE

1. Caractériser l'impact de différentes couleurs sur la température surfacique du matériau à l'aide d'une caméra thermique
2. Établir un lien direct entre l'albédo et la luminance
3. Établir une corrélation entre l'albédo et la température surfacique

Sommaire

I : introduction au problème

II : mesure colorimétrique

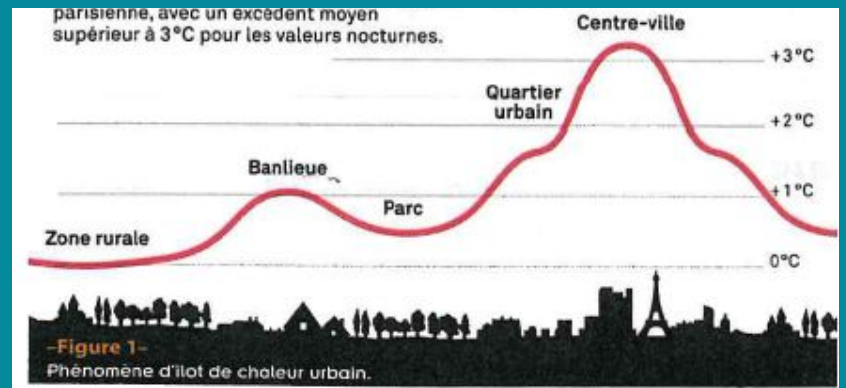
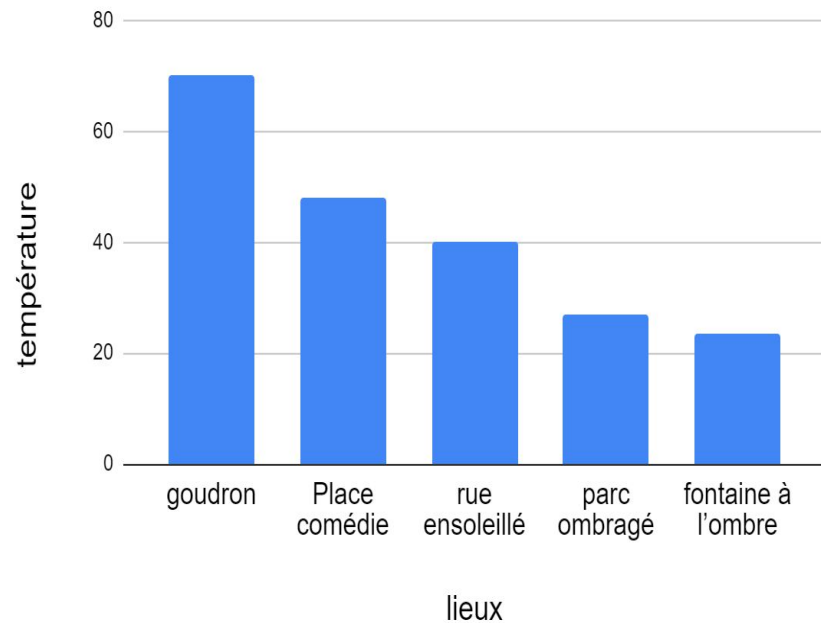
III : mesure d'albédo

IV : mesure de températures

V : Conclusion

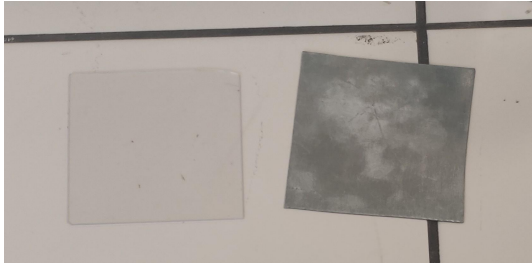
I : Introduction aux ICU

température de différents lieux



Principe du projet

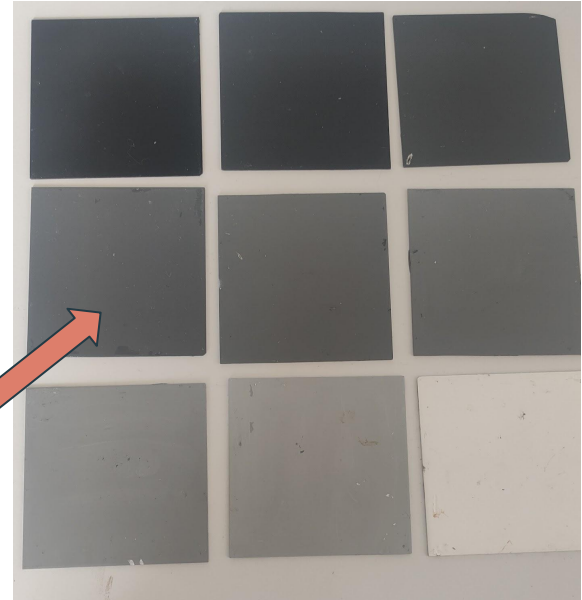
Échantillon d'acier et de plexiglass



Peinture noire/blanche



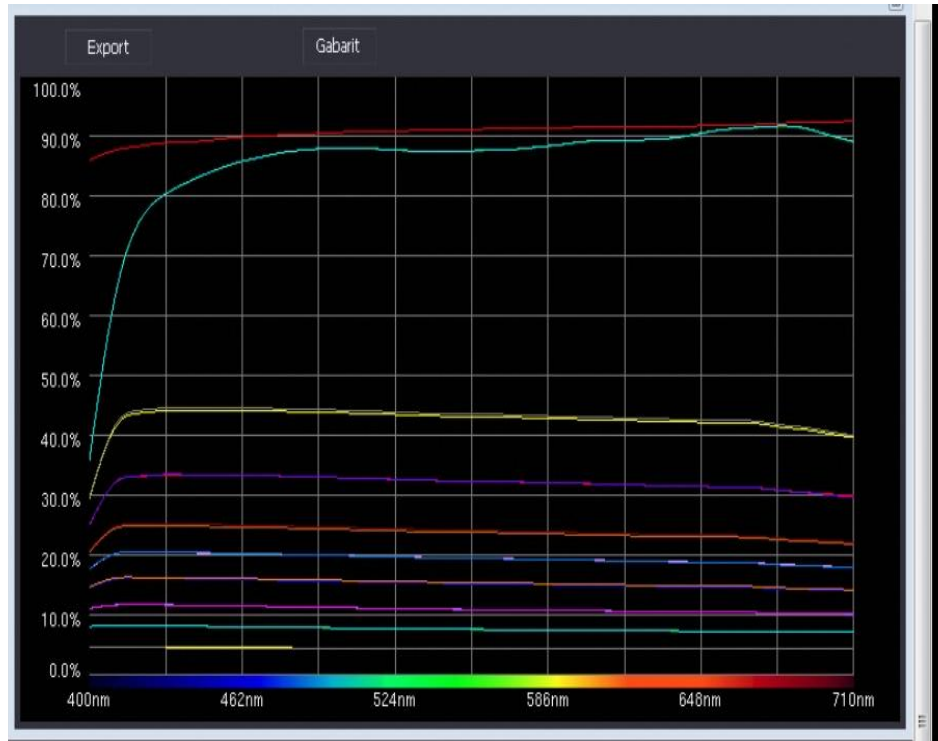
Echantillon



Echantillon gris 2

II Mesure colorimétrique

La transmittance en % fonction
de la longueur d'onde



spectralon
(PTFE)


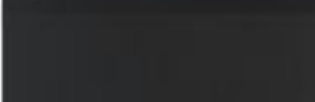
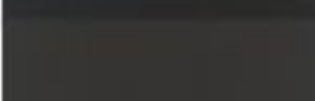
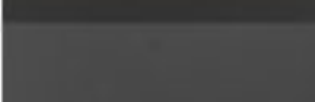







corps noir :
cavité noir

Spectromètre



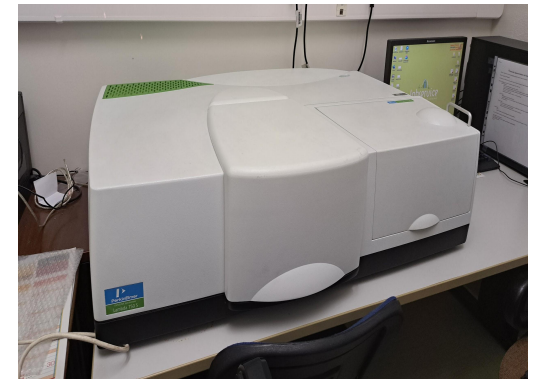
Mesure de Luminosité L^* et de luminance Y

		L^* (luminosité)	Y luminance
noir 1		25,45	4,49
noir 2		33,42	7,73
gris 1		39,67	11,05
gris 2		46,13	15,56
gris 3		51,44	19,62
gris 4		56,31	23,94
gris 5		63,76	32,53
gris 6		71,76	43,72
blanc		95,14	88

III Mesure d'albédo



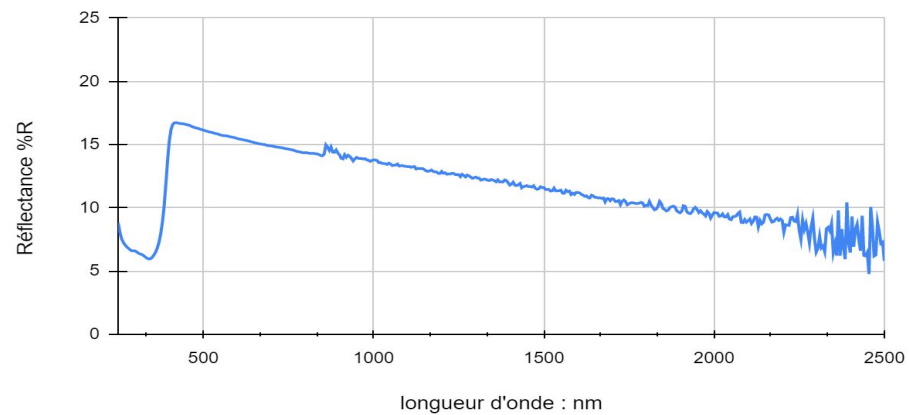
spectromètre



Spectralon (PTFE)



spectre de réflectance du gris 2



Formule du calcul de
l'albédo à partir du
spectre de réflectance

$$A = \frac{\int_{\lambda_a}^{\lambda_b} P(\lambda) \times S(\lambda) d\lambda}{\int_{\lambda_a}^{\lambda_b} S(\lambda) d\lambda}$$

Mesure de l'albédo pour
chaque échantillon

noir 1

noir 2

gris 1

gris 2

gris 3

gris 4

gris 5

gris 6

blanc

4,65

7,25

10,42

14,12

17,47

21,57

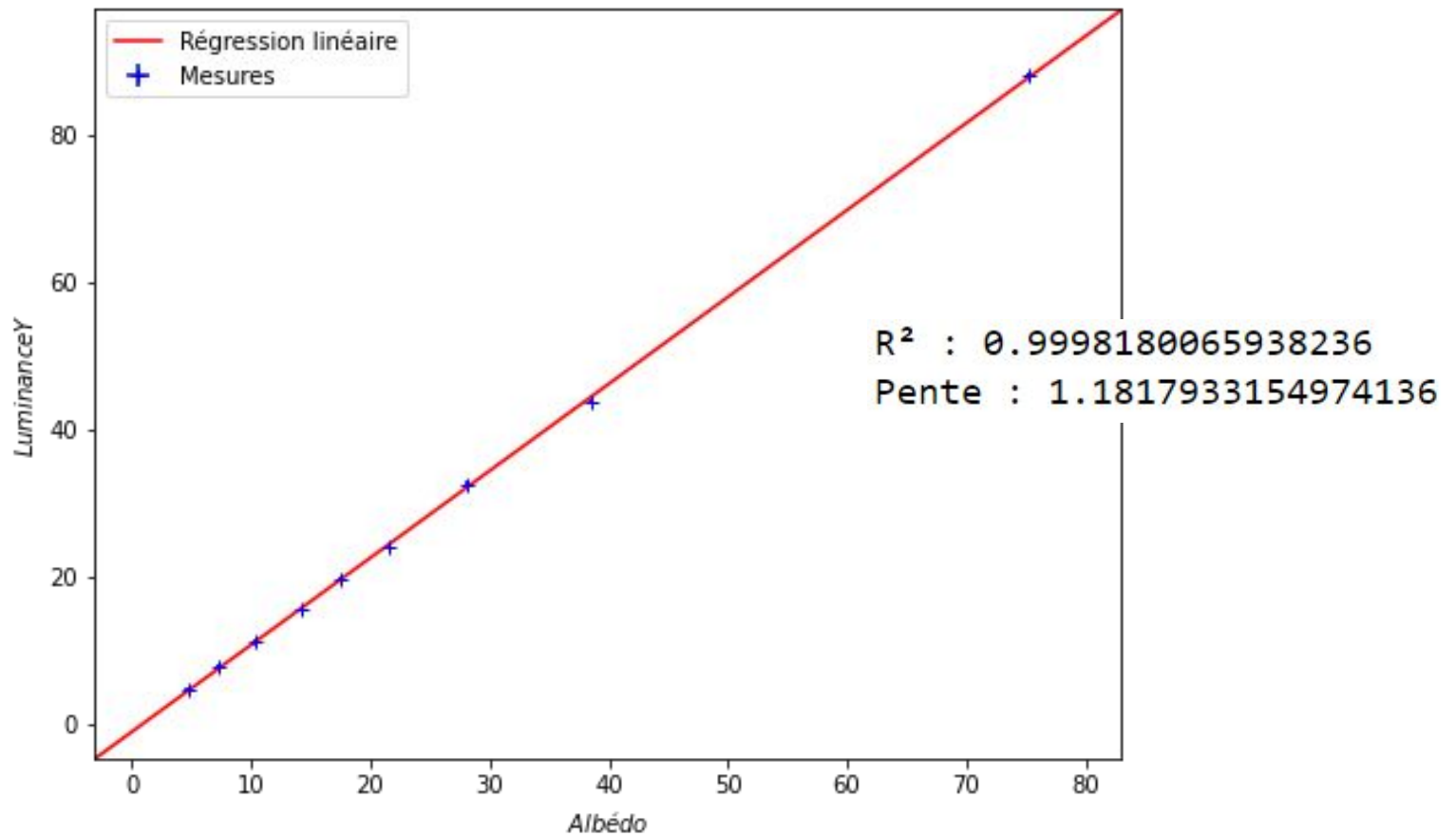
28,2

38,54

75,19

Lien entre l'albédo et la luminance Y

Graphique de l'albédo en fonction de la luminance



IV : Mesures de températures



$$T_{corr} = \frac{T_{mes} - T_{amb}}{\varepsilon}$$

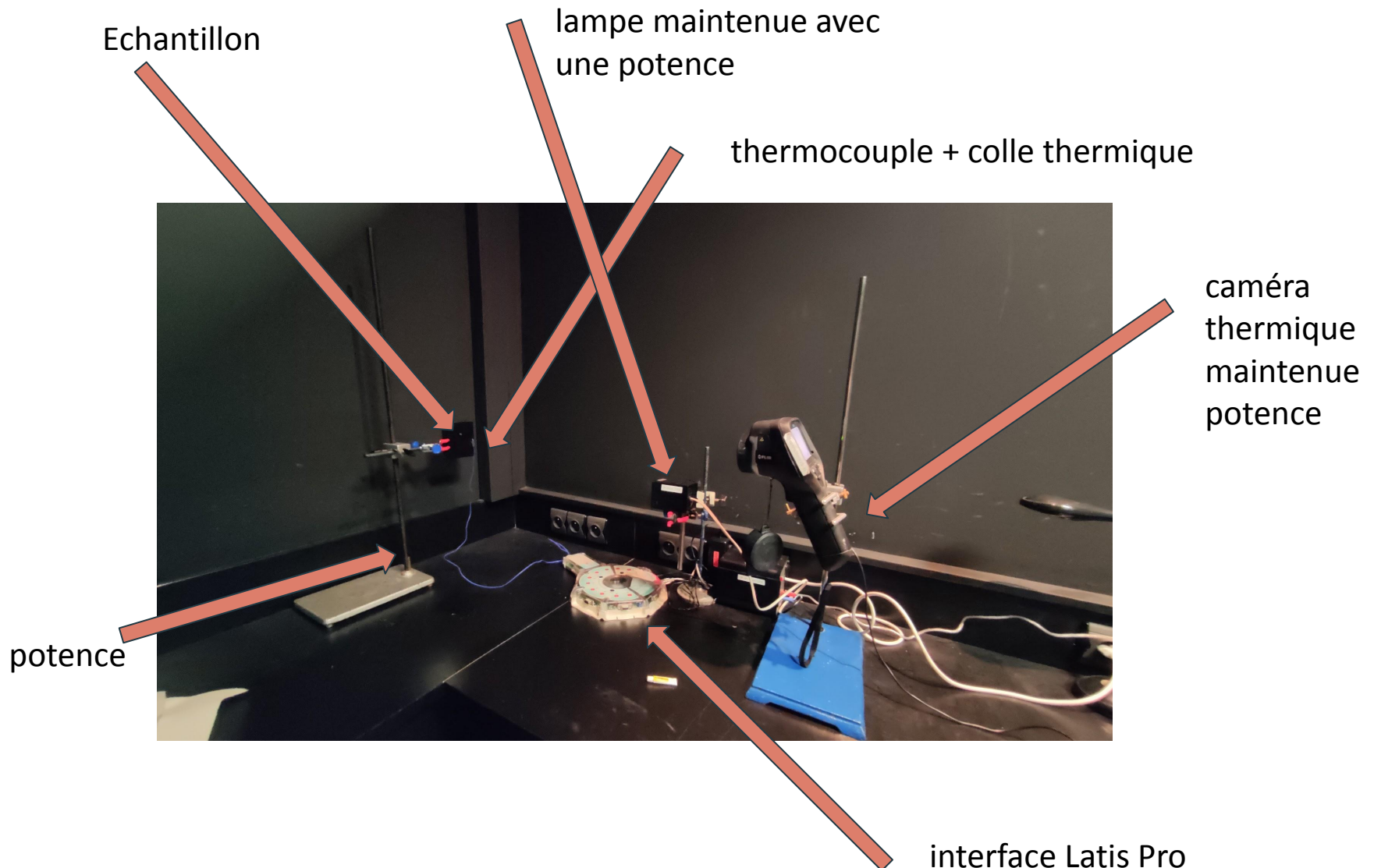
T_{cor} : température corrigée

T_{mes} : température mesurée

T_{amb} : température ambiante

ε : émissivité

Première expérience pour mesurer l'émissivité



Mesure de l'émissivité : Deuxième expérience

Emissivité :

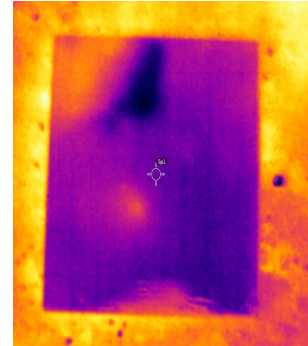
	Emissivité directionnelle	Emissivité hémisphérique
noir	0.960	0.943
gris1	0.955	0.937
gris2	0.957	0.938
gris3	0.957	0.939
gris4	0.957	0.939
gris5	0.951	0.931
gris6	0.950	0.929
gris7	0.943	0.920
blanc	0.950	0.930



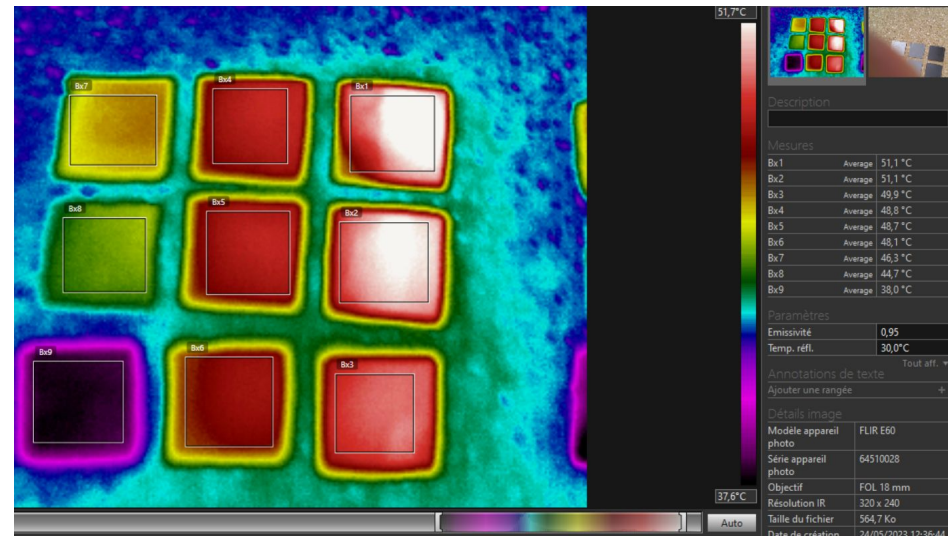
emissiomètre

Mesures de températures des échantillons

Nos échantillons disposés
au soleil



Mesure de
température d'un
échantillon à la
caméra thermique



Mesure de température sur
flir-Tools de chaque échantillon

Tableau des mesures de températures à différent instant

Tm1	Tm2	Tm3	Tm4	Tm5	Tm6	Tm7	Tm8	Tp9
27,3	52,6	41,4	45,2	51,1	60	61	58	69,7
27	52,5	41,4	45,2	51,1	59,5	60,5	57,9	67,7
26,7	51,2	40,9	44	49,9	57,5	57,9	56,1	66,1
26,7	50,5	40,8	43,4	48,8	56,3	57	55	64,5
26,8	49,5	40,6	43,7	48,7	56,3	56,5	54,9	62,3
26,5	48	40	43	48,1	55,3	55,5	54,1	61,1
26	46,5	39,4	41,8	46,3	53	53,7	51,6	57,2
25,8	45,1	39,3	41,3	44,7	50,3	50,7	49,1	54,2
25,6	36,3	37,2	37,2	38	39,5	40,1	39,2	40,2

```
import numpy as np

# Exemple de données d'échantillon
L = np.array([26.7,51.2,40.9,44,49.9,57.5,57.9,56.1,66.1])

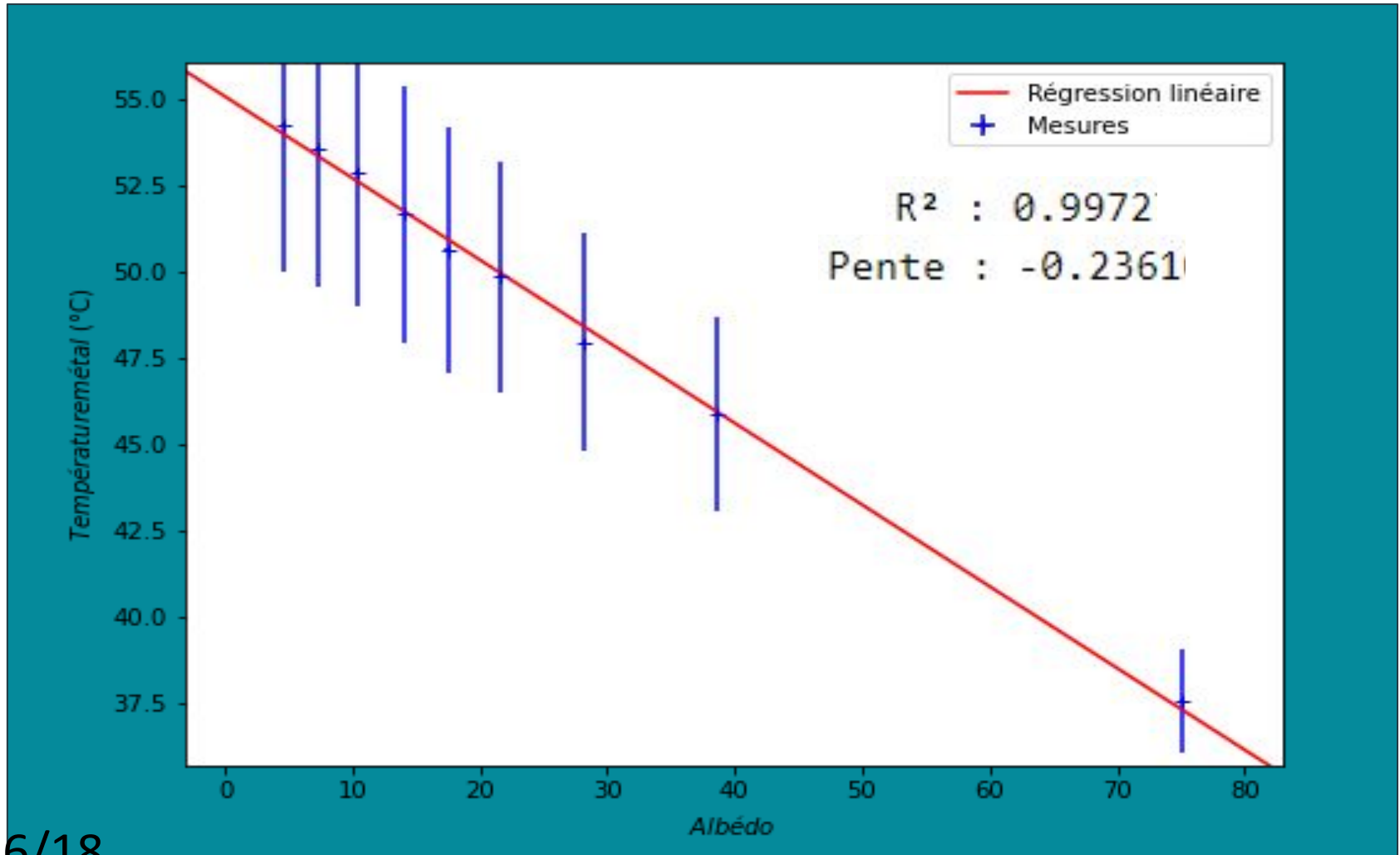
#Calcul de la moyenne de la liste
def moyenne(L):
    S=sum(L)
    moy=S/len(L)
    return moy

# Calcul de l'écart type des échantillons
ec = np.std(L, ddof=1)

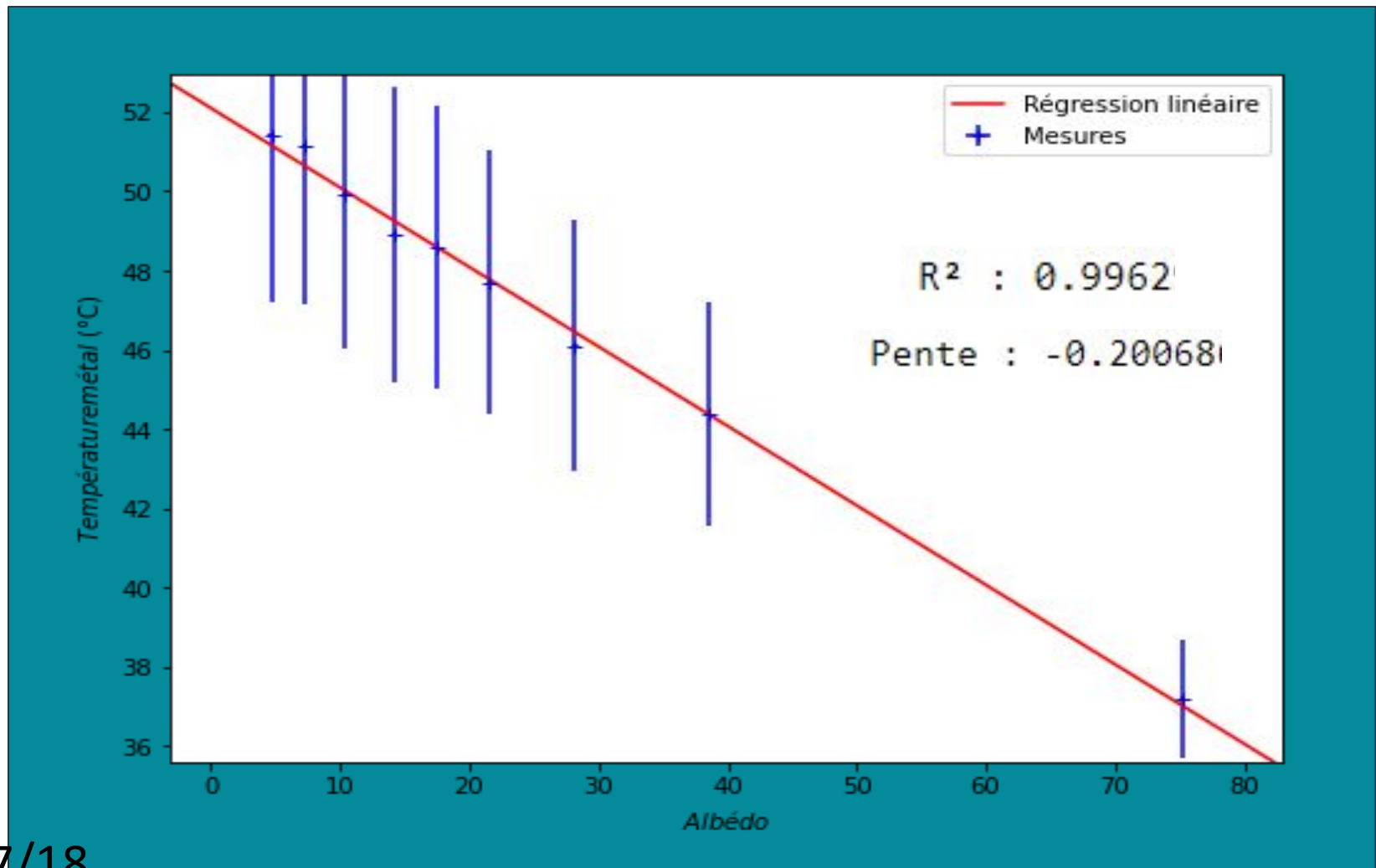
# Calcul de l'incertitude de type A
un = ec / np.sqrt(len(L))

# Affichage des résultats
print("la moyenne est :", moyenne(L))
print(f"Écart type : ", ec)
print(f"Incertitude de type A :", un)
```

Température des échantillons de plexiglass par rapport à l'albédo



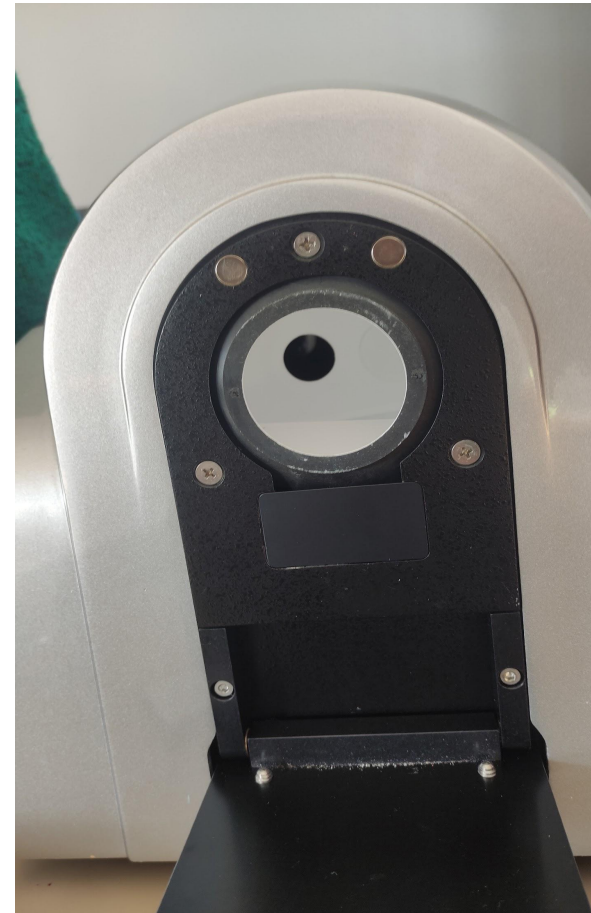
Température des échantillons d'acier par rapport à l'albédo



V : Conclusion

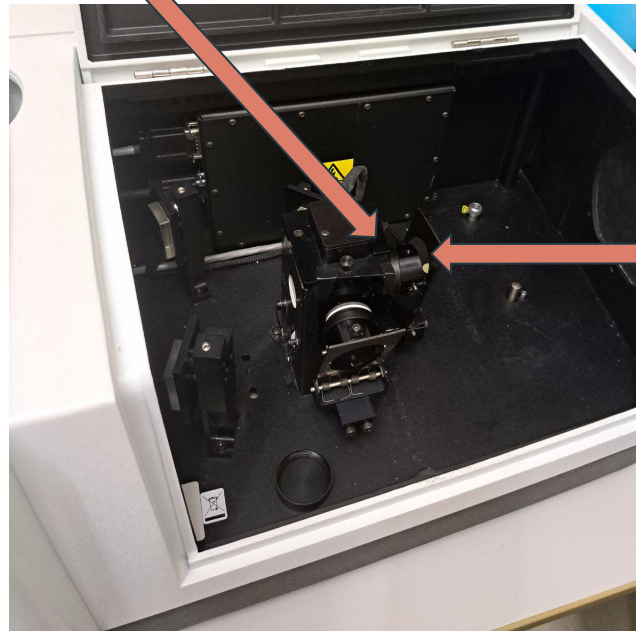


Annexes : le spectromètre 1



Annexes le spectromètre 2

Capteur CCD dans
l'infrarouge



Capteur
CCD dans
le visible

Annexes : l'émmisiomètre

A:Echantillon

B:cylindre adiabatique permettant de réémettre le rayon IR et de le maintenir à une température T constante ($\text{emm}=0.063$)

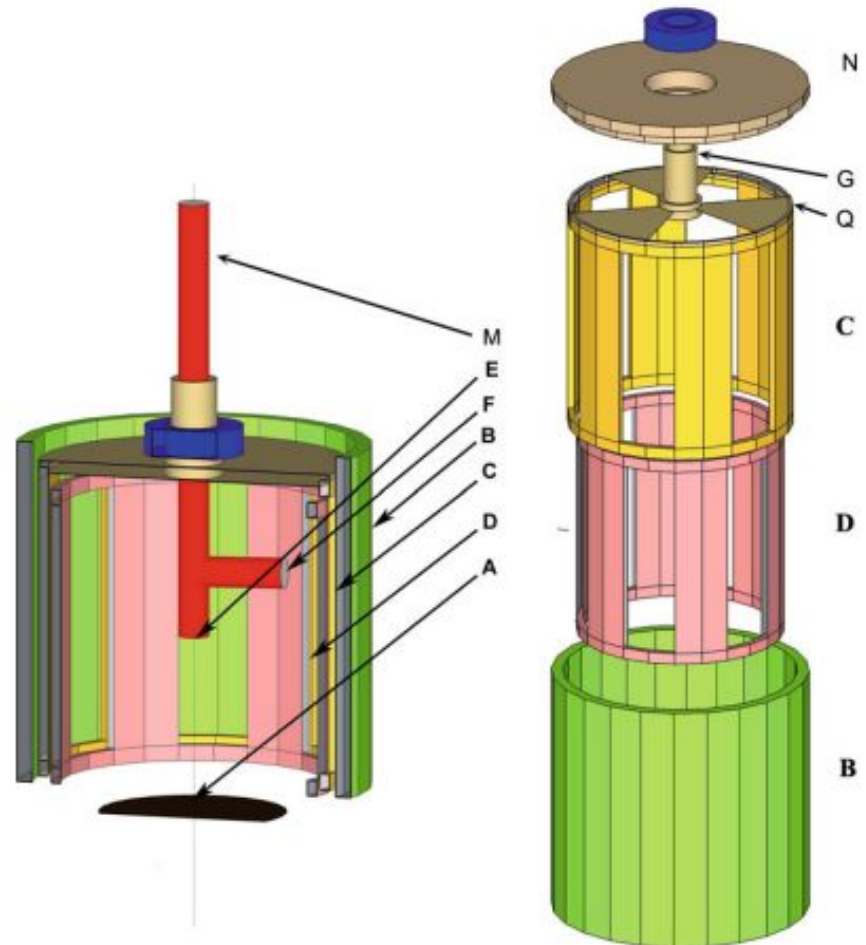
C et D : Écran fixe et mobile qui module le flux

E: Détecteur infrarouge (échantillon)

E1 (1 à 40 μm) **E2** (8 à 14 μm)

F: Détecteur infrarouge (surface B)

M: tube amenant au calculateur



Annexes : Code python Régression linéaire avec Monte-Carlo

```
import numpy as np
import matplotlib.pyplot as plt
%pylab inline
x=np.array([4.65,7.25]) #nom matériaux albédo
y=np.array([54.2,53.56]) #température en °C
dy = [4.2,4.0] #incertitude température
dx = 0.2
N=10000 #nbr itération
#liste des pentes
ta=[]
#liste des ordonnées
tb=[]
for i in range(0,N):
    l = len(x)
    mx=np.random.uniform(x-dx,x+dx,l)
    my=np.random.uniform(y-dy,y+dy,l)
    p=np.polyfit(mx,my,1)
    ta.append(p[0])
    tb.append(p[1])
    #pente et ordonnées moyennes
a = np.mean(ta) #calcul la moyenne
b = np.mean(tb)
#incertitude sur la pente et l'ordonnée
u_a = np.std(ta) #calcule l'écart type
u_b = np.std(tb)
```

```
#Pour le tracé
yfit = a*x + b
n=len(x)
pas=abs(max(x)-min(x))/n
x1=min(x)-pas
x2=max(x)+pas
y1=min(y)-abs(a*pas)
y2=max(y)+abs(a*pas)
xfit=np.linspace(x1,x2,n)

plt.figure(figsize=(8,6))
plt.xlabel(r'$Albédo$ ')
plt.ylabel(r'$Température métal$ (°C)')
plt.axis([x1,x2,y1,y2])
plt.plot(xfit, b + a*xfit, 'r', label='Régression linéaire')
plt.errorbar(x,y,xerr=dx,yerr=dy,fmt='b+',zorder=2,label='Mesures')
plt.legend()
plt.show()
```

Annexes : Code python Propagation d'incertitude

```
import numpy as np

def f(x):
    # Fonction à intégrer
    return x # Exemple : intégrale de  $x^2$  dans l'intervalle  $[a, b]$ 

def monte_carlo_integration(f, a, b, n_samples):
    # Calcul de l'intégrale par méthode de Monte-Carlo
    x_samples = np.random.uniform(a, b, n_samples)
    y_samples = f(x_samples)
    integral = np.mean(y_samples) * (b - a)
    uncertainty = np.std(y_samples) * (b - a) / np.sqrt(n_samples)
    return integral, uncertainty

# Paramètres d'entrée
a = 250 # Limite inférieure de l'intégrale
b = 2500 # Limite supérieure de l'intégrale
n_samples = 10000 # Nombre d'échantillons à générer

# Calcul de l'intégrale et de l'incertitude
integral, uncertainty = monte_carlo_integration(f, a, b, n_samples)

# Affichage des résultats
print("Intégrale :", integral)
print("Incertitude :", uncertainty)
```