

TP5 Thermodynamique

Partie 1 : Rayonnement

BERREDO DE LA COLINA Lucas

MARTIN Lola

■ L'emissivité

Nous allons determiner l'emmissivité d'un corps noir et gris.
Pour cela nous avons :

$$\epsilon = \frac{C}{\tau S_1 \sigma 4 T_f^3}$$

■ Échange de chaleur

Nous allons déterminer l'échange de chaleur de nos deux corps
 h représente l'ensemble des échanges de chaleur
(conduction,rayonnement,convection)

$$h = \frac{C}{\tau' S_1} - \frac{\sigma 4 T_f^3}{\frac{1}{\epsilon} + \frac{1}{\epsilon_{air}} - 1}$$

EXPLICATION DE L'EXPERIENCE

La variation de Temperature obéit à l'équation différentielle :

$$C \frac{dT}{dt} = \epsilon S \sigma (T_f^4 - T(t)^4)$$

■ Résolution exacte

$$t = 2\tau \left[\left(\arctan \left(\exp \left(2 \tanh^{-1} \frac{T(t)}{T_f} \right) \right) - \arctan \left(\exp \left(2 \tanh^{-1} \frac{T_i}{T_f} \right) \right) \right) + \left(\tanh^{-1} \frac{T(t)}{T_f} - \tanh^{-1} \frac{T_i}{T_f} \right) \right]$$

■ Résolution approchée

$$T = T_f + (T_i - T_f) \left(1 - \exp\left(\frac{-t}{\tau}\right) \right)$$

- Mise sous vide
Évite les échange de chaleur $h = 0$
- Pression ambiante
Permet de déterminer l'échange de chaleur

- Deux échantillons (gris, noir)
- Deux chambres :
 - ▶ Four (200°C)
 - ▶ Refroidissement à l'eau
- Elles peuvent être mises sous vide
- Mesures de temperature analogiques (100 points, 15 min)

EXPERIENCES

1. Corps gris, **chauffage**, vide
2. Corps gris, **refroidissement**, vide
3. Corps gris, **chauffage**, sans vide
4. Corps gris, **refroidissement**, sans vide
5. Corps noir, **chauffage**, vide
6. Corps noir, **chauffage**, sans vide

1. tp5-gris-vide-chauff.csv
2. tp5-gris-vide-refroid.csv
3. tp5-gris-air-chauff.csv
4. tp5-gris-air-refroid.csv
5. tp5-noir-vide-chauff.csv

Temps	Thermocouple	EA0
0	43,2356657	0,209796296
9	50,75927386	0,214779578
18	57,27973427	0,214779578
27	64,80334243	0,219762859
36	71,32380283	0,219762859
45	77,84426324	0,22474614
54	84,36472365	0,219762859
63	90,88518405	0,22474614
72	96,90407058	0,219762859
81	102,9229571	0,22474614
90	108,4402698	0,219762859
99	113,9575824	0,214779578

APPROXIMATION GRAPHIQUE 1ER ORDRE

Il ne faut que vérifier les valeur initiales, finales et approcher τ de façon qu'on trouve des courbes proches

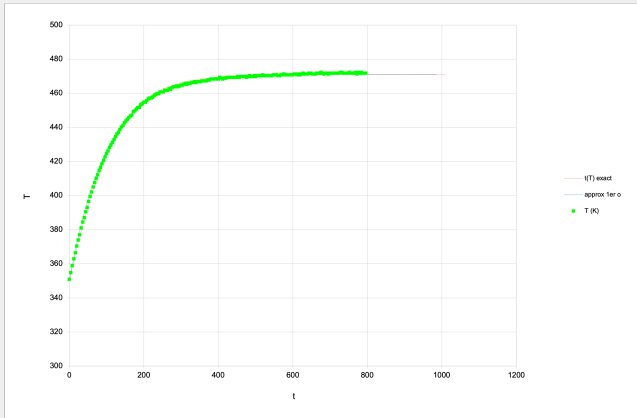


Figure – Example representation graphique : Vert : points expérimentaux, Bleu : courbe théorique

APPROXIMATION GRAPHIQUE

Nous obtenons les prochains valeurs :

1. Corps gris, **chauffage**, vide $\tau = 100$
2. Corps gris, **refroidissement**, vide $\tau = 69$
3. Corps gris, **chauffage**, sans vide $\tau = 117$
4. Corps gris, **refroidissement**, sans vide $\tau = 68$
5. Corps noir, **chauffage**, vide $\tau = 105$
6. Corps noir, **chauffage**, sans vide $\tau = 99$

APPROXIMATION GRAPHIQUE 2ÈME ORDRE

- | | |
|---|--------------|
| 1. Corps gris, chauffage, vide | $\tau = 82$ |
| 2. Corps gris, refroidissement, vide | $\tau = 85$ |
| 3. Corps gris, chauffage, sans vide | $\tau = 100$ |
| 4. Corps gris, refroidissement, sans vide | $\tau = 84$ |
| 5. Corps noir, chauffage, vide | $\tau = 111$ |
| 6. Corps noir, chauffage, sans vide | $\tau = 89$ |

APPROXIMATION NUMÉRIQUE AVEC PYTHON

Comme nous avons la résolution pour τ , nous pouvons donner ça vers un `curve_fit` dans Python.

```
def theoretical_model(t, tau):  
    term1 = np.arctan(np.exp(2) * arccoth(T_kelvin_data / T_f))  
    term2 = np.arctan(np.exp(2) * arccoth(T_i / T_f))  
    term3 = arccoth(T_kelvin_data / T_f)  
    term4 = arccoth(T_i / T_f)  
    return 2 * tau * (term1 - term2 + term3 - term4)  
  
# Use curve_fit to find the optimal tau  
popt, pcov = curve_fit(theoretical_model, t_data, T_kelvin_data, p0=[1.0])  
optimal_tau = pop[0]  
print(optimal_tau)
```

Figure – Exemple refroidissement. Il y a aussi un fichier pour chauffage.

APPROXIMATION NUMÉRIQUE AVEC PYTHON

Résultats :

- | | |
|---|------------------------|
| 1. Corps gris, chauffage, vide | $\tau = 96,0399\dots$ |
| 2. Corps gris, refroidissement, vide | $\tau = 86,1429\dots$ |
| 3. Corps gris, chauffage, sans vide | $\tau = 99,2634\dots$ |
| 4. Corps gris, refroidissement, sans vide | $\tau = 84,5618\dots$ |
| 5. Corps noir, chauffage, vide | $\tau = 111,8591\dots$ |
| 6. Corps noir, chauffage, sans vide | $\tau = 90,5110\dots$ |

COMPARAISON DES RÉSULTATS

Expérience	1er ordre	2ème ordre	Python
1	100	82	96,0399
2	69	85	86,1429
3	117	100	99,2634
4	68	84	84,5618
5	105	111	111,8591
6	99	89	90,5110

Table – Valeurs du paramètre τ

- Les valeurs générés par Python suivent avec le moindre erreur possible la courbe.
- L'approximation à premier ordre a un erreur autour 20% pour chaque expérience.
- L'approximation à deuxième ordre est beaucoup plus proche (autour ± 1 point).

On trouve les résultats suivant

- $\epsilon_{gris} = 0,54$

- $h_{gris} = 11,27 W.m.K^{-1}$

- $\epsilon_{noir} = 0,76$

- $h_{noir} = 15,13 W.m.K^{-1}$

TP5 Thermodynamique

Partie 2 : Loi de Stefan

BERREDO DE LA COLINA Lucas

MARTIN Lola

Bien que nous ayons travaillé avec l'équipement et observé des résultats avec Mme Quilliet, nous n'avons pas enregistré de résultats numériques.

■ Loi de Stefan

$$M = \sigma T^4$$

Avec M la densité de puissance ($W.m^{-2}$)

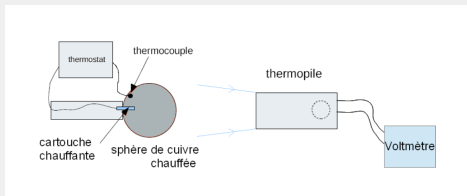


Figure – Shémas de l'expérience

- Deux parties :
 - ▶ Côté emmissive - Boule à cuivre “corps noir”
 - ▶ Côté receptive - Thermopile CA2 (filtre en option) et multimètre
- Emmisivité fixé - mesure du puissance avec le multimètre
- Il faut attendre après chaque changement vers la stabilisation
- Mesures a plusieurs distances (0,3 ; 0,4 ; 0,8 ; 1,2m) et temperatures (20, 60, 90, 120 °C)

APPROXIMATION DES RÉSULTATS

```
distances <- c(0.3, 0.4, 0.6, 0.8, 1)      # Distances emitter-thermopile (m)
T_ext <- 293.15                             # External temperature (K)
temperatures <- c(293.15, 333.15, 363.15, 395.15) # Emmitter temperatures (K)
isFiltered <- TRUE                          # Is the thermopile filtered? (Bool)
times <- 1:50                               # Measurement points (default: 1:50 - once every second for 50 seconds)
```

Figure – Paramètres à choisir

```
T_func <- function(t, PK, KC, T_ext) {
|   PK * (1-exp(-KC*t)) + T_ext
}
```

Figure – Fonction pour l'évolution temporelle théorique

APPROXIMATION DES RÉSULTATS

```
noisy_func_name <- paste0("NoisyDeltaV_d", d, "_T", T)
f <- function(x) {
  original_val <- original_f(x)
  noise <- rnorm(n=1, mean=1, sd=0.05)*original_val
  return(noise)
}
assign(noisy_func_name, f)
```

Figure – Ajout du bruit