

# Étude d'une centrale thermique solaire

Réponses aux questions

4 mai 2025

## Partie I : Étude du récepteur solaire

### 1.1

Les différentes raisons expliquant l'efficacité de 70 % entre l'énergie solaire reçue par le récepteur et l'énergie solaire incidente sur les héliostats sont :

- Réflexions imparfaites et diffusion par les miroirs ;
- Pertes atmosphériques (absorption et diffusion) ;
- Mauvais alignement ou suivi solaire imparfait ;
- Salissures ou vieillissement des héliostats.

### 1.2

La puissance reçue par le récepteur depuis les miroirs est :

$$\begin{aligned} P_{\text{récepteur}} &= S_{\text{hel}} \cdot E_s \cdot \eta_m \\ &= 75\,000 \text{ m}^2 \cdot 1000 \text{ W m}^{-2} \cdot 0,70 \\ &= \boxed{52,5 \text{ MW}} \end{aligned}$$

### 2.1

On a :

$$\epsilon(T) = \alpha(T) = \int_0^{+\infty} \epsilon_\lambda(\lambda) f_\lambda(T, \lambda) d\lambda$$

(Pour conclure, il faut la data fournie en annexe mais ya pas d'annexe...)

### 2.2

Le facteur de forme  $F_{e \rightarrow r}$  de l'environnement vers le récepteur vaut trivialement 1, car l'intégralité du flux du rectangle rouge est capté. Par réciprocité :

$$F_{r \rightarrow e} = \frac{S_e}{S_r} = \boxed{0,65}$$

## 2.3

On a :

$$Ra = Gr \cdot Pr \quad \text{avec} \quad \Delta T = 254 \text{ K}, \quad L = 10 \text{ m}, \quad T_m = 420 \text{ K}, \quad \beta = \frac{1}{T}$$

Aux conditions données :

$$Pr = 0,7, \quad \mu = 2,4 \times 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}, \quad \rho = 0,83 \text{ kg m}^{-3}, \quad \lambda = 0,035 \text{ W m}^{-1} \text{ K}^{-1}$$

Ainsi  $Ra > 10^9$  et on utilise la corrélation :

$$Nu_L = 0,13 \cdot (Gr_L \cdot Pr)^{0,33}$$

Soit

$$\begin{aligned} h &= 0,13 \cdot \frac{\lambda}{L} \cdot (Gr_L \cdot Pr)^{0,33} \\ &= \boxed{7,04 \text{ W m}^{-2} \text{ K}^{-1}} \end{aligned}$$

## 3.1

La puissance absorbée après  $n$  réflexions est (démonstration par récurrence triviale) :

$$\alpha S_{\text{hel}} \eta_m E_s \cdot (\rho F_{r \rightarrow r})^n = \alpha S_{\text{hel}} \eta_m E_s \cdot [(1 - \alpha)(1 - F_{r \rightarrow e})]^n$$

En sommant sur  $n$ , on obtient :

$$\begin{aligned} P_{\text{abs}} &= \frac{\alpha}{1 - \alpha \cdot F_{r \rightarrow e}} \cdot S_{\text{hel}} \cdot \eta_m \cdot E_s \\ &= \frac{0,96}{1 - 0,96 \cdot 0,65} \cdot 52,5 \times 10^6 \text{ W} \\ &= \boxed{133,89 \text{ MW}} \end{aligned}$$

## 3.2

Avec l'analogie électrique :

$$\phi_{r \rightarrow e}^{\text{nette}} \cdot R = M_e^0 - M_r^0 = \sigma (T_a^4 - T_r^4)$$

où :

$$R = \frac{1 - \epsilon}{\epsilon S} + \frac{1}{S F_{r \rightarrow e}}$$

D'où :

$$\phi_{r \rightarrow e}^{\text{nette}} = \frac{\sigma S (T_a^4 - T_r^4)}{\frac{1 - \epsilon}{\epsilon} + \frac{1}{F_{r \rightarrow e}}}$$

### 3.3

On a :

$$\dot{m} = \frac{66\,000 \text{ kg h}^{-1}}{3600 \text{ s h}^{-1}} = 18,33 \text{ kg s}^{-1}, \quad \Delta h = 2689 \text{ kJ kg}^{-1} = 2,689 \times 10^6 \text{ J kg}^{-1}$$

Ainsi la puissance transmise au fluide est :

$$\begin{aligned} P_{\text{fluide}} &= \dot{m} \cdot \Delta h \\ &= 18,33 \text{ kg s}^{-1} \cdot 2,689 \times 10^6 \text{ J kg}^{-1} \\ &= \boxed{49,3 \text{ MW}} \end{aligned}$$

### 3.4

En régime permanent on a :

$$0 = P_{\text{abs}} - \phi_{r \rightarrow e}^{\text{nette}} - P_{\text{fluide}} + \phi_{\text{convection}}$$

Soit :

$$P_{\text{fluide}} - P_{\text{abs}} = -\phi_{r \rightarrow e}^{\text{nette}} + \phi_{\text{convection}}$$

En posant

$$A = P_{\text{fluide}} - P_{\text{abs}} \quad B = -\frac{\sigma S}{\frac{1-\epsilon}{\epsilon} + \frac{1}{F_{r \rightarrow e}}} \quad C = 2hS$$

Cela revient à résoudre

$$A = B(T_a^4 - T_r^4) + C(T_a - T_r)$$

En utilisant le script python suivant qui implémente la méthode des itérations sur l'équation linéarisé, il vient  $T_r = \dots$  (il n'y a pas de solution à l'équation proposée... (cf graphique du programme python) je pense que y'a une erreur qlq part, a voir plus tard.

```
'''
script pour résoudre une équation de la forme:
A = B(T1^4 - T2^4) + C(T1-T2)
avec A,B,C et T1 connues.
On utilise une methode de linéarisation.
'''

import matplotlib.pyplot as plt
import numpy as np

def nouveauT2(A,B,C,T1,T2):
    D = 4*B*((T1+T2)/2)**3
    return (T1*(D+C) - A)/(D+C)

A = -84.59*10**6
B = -1.04*10**(-5)
C = 3649.54
```

```

T1 = 293
precision = 100000

T2 = 3000
prochainT2 = nouveauT2(A,B,C,T1,T2)

i=0
while abs(T2 - prochainT2) > precision:

    print(T2, prochainT2)

    T2 = prochainT2
    prochainT2 = nouveauT2(A,B,C,T1,T2)
    i+=1

def f(x):
    return A - B*(T1**4-x**4) - C*(T1-x)

X = np.arange(0,1000,10)
Y = f(X)
plt.plot(X,Y)
plt.show();

print("solution trouvée:", T2)
print("verification:",A - B*(T1**4-T2**4) - C*(T1-T2))
print("En",i,"itérations")

```