

Objectifs : passer en revue les différentes questions posées au concours ATS depuis 2016.

NB **Compte-rendu :** vous déposerez les différents scripts (fichier *.PY) sur Éclat, en les nommant de manière à ce que je puisse identifier l'auteur et l'année du concours :

► Espace des classes > Classe ATS > Dossiers partagés > TP info-physique > **séance N°4**

1. ATS 2016

contexte

On admet que la Terre et son atmosphère constituent les deux armatures d'un condensateur sphérique. On représente l'ensemble Terre-ionosphère comme un volumineux condensateur sphérique. La Terre, de rayon R , se comporte comme un conducteur parfait de potentiel V_T , pris nul, et porte une charge négative $-Q$ uniformément répartie sur sa surface, tandis que l'ionosphère est représentée par une surface équipotentielle sphérique de rayon $R + z_0$, de potentiel V_{atm} . On suppose que l'atmosphère a la permittivité du vide.

On a calculé $\vec{E}(r) = -\frac{Q}{4\pi\epsilon_0 r^2}$

On "rappelle" que $\Delta V = V_A - V_T = U_{AT} = \int_A^T \vec{E} \cdot d\vec{r} = \int_{r=R}^{r=R+z_0} \frac{Q}{4\pi\epsilon_0 r^2} \cdot dr$

51) On souhaite calculer $\Delta V = V_{atm} - V_T$ en utilisant une procédure informatique d'intégration dite "méthode des rectangles", présentée ci-dessous :

```
def rectangles(a,b,n,f):
    #méthode d'approximation dites des 'rectangles'
    S=0
    for i in range(n-1):
        x1 = a+i*(b-a)/n
        x2 = a+(i+1)*(b-a)/n
        S = S + f((x1+x2)/2)*(x2-x1)
    return S
```

Figure 3 : intégration par la méthode des rectangles

Par identification avec les résultats précédents, donner les valeurs de a et b et l'expression de la fonction $f(x)$ utilisées dans la procédure ci-dessus.

question complémentaire. Compléter le programme pour le tester, en l'appliquant effectivement à la fonction $f(x)$: définir la fonction, entrer les paramètres a , b , n , tracer la fonction entre a et b , et afficher le résultat du calcul, ΔV , sans oublier l'unité. On arrondira le résultat à 1 V près, avec la fonction `round()`.

2. ATS 2017 (remarque : la méthode d'Euler n'est plus au programme)

contexte

Les deux électrodes d'un réacteur plasma sont des disques centrés sur l'axe z , de rayon $a/2$ et d'épaisseur négligeable. Elles sont parallèles et leur distance est notée d ($d > 0$). On applique une tension continue $U > 0$ entre les deux électrodes du réacteur. L'électrode supérieure porte alors une charge $-q$ et l'électrode inférieure une charge $+q > 0$. Le réacteur contient un gaz d'argon peu dense, encore non ionisé et de permittivité diélectrique assimilable à celle du vide, notée ϵ_0 .

L'étude faite en négligeant les effets de bords nous a conduit à $\vec{E}(M) = \frac{\sigma}{\epsilon_0} \vec{u}_z$ et $V(z) = U \left(\frac{1}{2} - \frac{z}{d} \right)$

Évaluation des effets de bord

Dans ce paragraphe, on souhaite étudier comment la prise en compte des effets de bord affecte les résultats concernant le champ et le potentiel électrique obtenus précédemment. Pour faciliter l'étude, on se limitera à un problème à deux dimensions et on admettra que dans le plan $x = 0$, le potentiel électrostatique V satisfait l'équation de Laplace suivante * :

$$\frac{\partial^2 V(y,z)}{\partial y^2} + \frac{\partial^2 V(y,z)}{\partial z^2} = 0$$

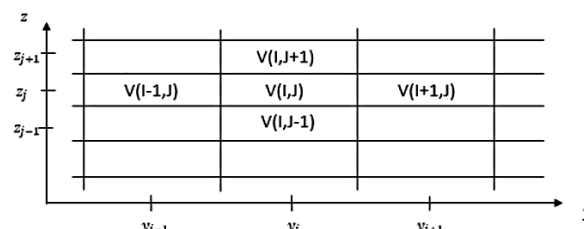
* L'équation de Laplace proposée ici s'obtient rigoureusement dans le cas d'un réacteur dont les électrodes sont assimilées à des plans rectangulaires de largeur finie (suivant y) et de longueur supposée infinie (suivant x). L'équation étudiée va donc mettre en évidence l'influence des effets de bord mais elle devrait être complétée pour décrire plus finement notre réacteur.

On souhaite écrire un programme informatique permettant de déterminer la fonction potentiel électrostatique dans le réacteur alimenté en ± 10 V. La solution de ce problème doit vérifier l'équation de Laplace ci-dessus et les conditions aux limites imposées au potentiel $V(y,z)$ par l'expérimentateur qui sont ici $V\left(y = \pm \frac{a}{2}; z\right) = 0$.

Dans ce programme, on va chercher à calculer le potentiel en différents points $(x = 0; y_i; z_j)$ du réacteur avec $(0 \leq z_j < 20)$ mm et $(-10 \leq y_i < +10)$ cm. Le potentiel est calculé tous les millimètres verticalement et horizontalement.

Ainsi, le potentiel est décrit dans un tableau de taille 201×21 (201 lignes \times 21 colonnes) où le réel $V_{i,j}$ contient la valeur en volt de $V(y_i, z_j)$.

Le tableau ci-contre représente une partie de la discrétisation de l'espace en cellules. $V_{i,j}$ est ici écrit $V(I,J)$.



Pour résoudre l'équation de Laplace, nous allons appliquer la méthode d'Euler. À titre d'exemple, cette méthode consiste

notamment à associer la dérivée partielle $\frac{\partial V(y,z)}{\partial y}$ à la quantité $\frac{V_{i+1,j} - V_{i,j}}{\Delta y}$. \Leftrightarrow ajout du pas Δy

8. Écrire, avec la méthode d'Euler, la quantité $\frac{\partial V(y,z)}{\partial z}$ en fonction de Δz , $V_{i,j+1}$ et $V_{i,j}$.
9. Écrire, avec la méthode d'Euler, la quantité $\frac{\partial^2 V(y,z)}{\partial z^2}$ en fonction de Δz , $V_{i,j+1}$, $V_{i,j-1}$ et $V_{i,j}$.
10. Écrire alors la définition à donner à $V_{i,j}$ avec la méthode d'Euler, en fonction de $V_{i+1,j}$, $V_{i,j+1}$, $V_{i-1,j}$ et $V_{i,j-1}$

3. ATS 2018

contexte

Une plaque à induction rayonne un champ magnétique. Nous allons supposer, pour simplifier, que ce champ magnétique est analogue à celui créé par une bobine circulaire (P) comportant N spires d'axe Oz , filiformes, jointives et de rayon a . La casserole est modélisée par une spire (S) parallèle à (P) et située à une distance z de (P).

Force magnétique

En tenant compte de l'inductance propre associée à (S) et en prenant une description plus complète du champ magnétique créé par (P), on montre que (S) est soumise à une force de Laplace \vec{F} dont la moyenne temporelle $\langle \vec{F} \rangle$ est donnée par :

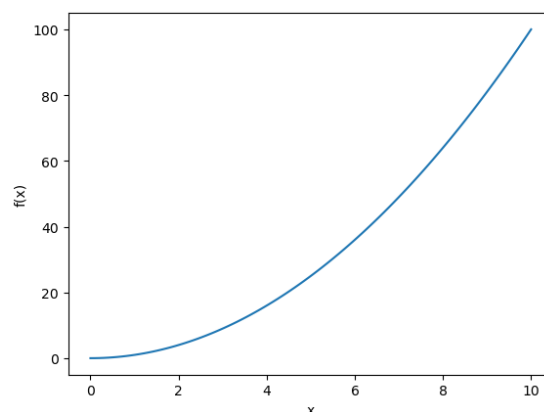
$$\langle \vec{F} \rangle \approx K_3 \frac{Z}{(1+Z^2)^4} \vec{u}_z \quad \text{avec } Z = \frac{z}{a} \quad \text{et } K_3 \approx 1 \text{ N.}$$

On souhaite apprécier la valeur maximale de la force de Laplace pouvant s'exercer sur (S). Pour cela, on va utiliser l'outil informatique. On donne ci-dessous un exemple de programmation en langage Python permettant d'obtenir le graphe de la fonction x^2 .

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return x**2

x=np.linspace(0,10,100)
plt.plot(x,f(x))
plt.xlabel('x')
plt.ylabel('f(x)')
plt.show()
```



- 48) Réécrire le programme précédent sur votre copie en l'adaptant afin d'obtenir le graphe de $\frac{Z}{(1+Z^2)^4}$ en fonction de Z pour $Z \in [0, 2]$ en utilisant 100 points (les axes devront également être renommés).

4. ATS 2019

contexte

Comment se protéger de la foudre ?

Un coup de foudre est associé à un courant de forte intensité et de courte durée. La mesure de l'intensité du courant $i(t)$ conduit typiquement au graphe ci-contre. Avec :

- un temps de montée $t_1 = 2 \text{ ms}$
- un temps de retombée à la demi-valeur $t_2 = 25 \text{ ms}$
- un courant de crête $I = 15 \text{ kA}$

Un éclair est associé à un courant électrique assimilable à celui d'un fil rectiligne, parcouru par un courant d'intensité $i(t)$ uniformément réparti. Dans le sol, on suppose que la densité de courant volumique est radiale, de la forme $\vec{j} = j(r,t)\vec{u}_r$ où \vec{u}_r est le vecteur unitaire radial de la base sphérique. Un homme se trouve à la distance moyenne d du point d'impact de la foudre et la distance entre ses pieds est notée p .

On a montré que l'expression de la différence de potentiel $U_p > 0$ entre les pieds de l'homme peut se mettre sous la forme $U_p = Ri$ avec $R = \frac{2p}{\pi\gamma(4d^2 - p^2)}$

$$R = \frac{2p}{\pi\gamma(4d^2 - p^2)}$$

On prend $\gamma = 1 \text{ S}\cdot\text{m}^{-1}$, $p = 0,5 \text{ m}$ et $d = 1 \text{ m}$. On assimile par la suite la fonction $i(t)$ à une fonction affine par morceaux.

question

- 14) Le programme Python ci-dessous permet d'obtenir le tracé de l'intensité $i(t)$ du courant. En vous aidant de ce programme, écrire sur votre copie les lignes de code *à modifier pour* obtenir le graphe de $U_p(t)$.

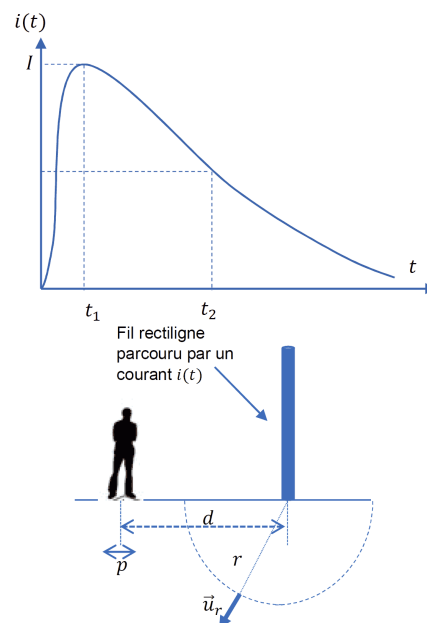
```
import numpy as np
import matplotlib.pyplot as plt

#constantes
gam=1
p=0.5
d=1
t1=2*10**-3
t2=25*10**-3
Imax=15*10**3
R=p/(2*np.pi*gam*(d**2-(p**2)/4))

#définitions
def i(t) :
    if t<=t1 :
        I=Imax*t/t1
    else :
        I=-Imax/(3*t2)*(t-t1)+Imax
    return I

def liste_valeurs_i(liste_t,i) :
    L=[] #liste vide
    for t in liste_t :
        L.append(i(t)) #rappel : append() permet d'ajouter un élément à la fin d'une liste
    return L

#tracé
liste_t=np.linspace(0,3*t2,1000)
liste_i=liste_valeurs_i(liste_t,i)
plt.plot(liste_t,liste_i)
plt.xlabel('temps(s)')
plt.ylabel('i(A)')
plt.title('i(t)')
plt.show()
```



question complémentaire. Vérifier l'allure de la courbe obtenue, et comparer au graphe de $i(t)$.

5. ATS 2020

contexte

Détermination expérimentale de la conductivité thermique d'un métal.

On considère une tige en fonte, cylindrique, de rayon a et de longueur L .

- On suppose le régime stationnaire atteint. On a $L = 1,0 \text{ m}$, $a = 0,50 \text{ cm}$ donc $L \gg a$.
- On note c la capacité thermique massique de la tige assimilée à une phase condensée indilatable et incompressible, on note ρ sa masse volumique et λ sa conductivité thermique. On prendra $c = 400 \text{ J}\cdot\text{K}^{-1}\cdot\text{kg}^{-1}$ et $\rho = 5000 \text{ kg}\cdot\text{m}^{-3}$.
- On note h le coefficient de transfert conducto-convectif. Une expérience précédente permet d'obtenir la valeur du rapport $\frac{\lambda}{h}$ ici estimée à $\frac{\lambda}{h} \approx 4 \text{ SI}$.

Traitement informatique

Dans une seconde expérience, on place intégralement la tige dans un bain d'eau glacée de manière à ce que cette dernière atteigne la température $T_0 = 0 \text{ }^\circ\text{C}$. À $t=0$, on fixe la tige refroidie à une potence afin qu'elle se réchauffe au contact de l'air ambiant à la température $T_{\text{ext}} = 30 \text{ }^\circ\text{C}$. Avec ce mode opératoire, on montre que la conduction thermique au sein de la tige est négligeable et que seul le transfert conducto-convectif est à prendre en compte. Le champ des températures T de la tige est quasi-uniforme et n'est fonction que du temps lors de ce chauffage monobare. Un bilan enthalpique aboutit alors à l'équation :

$$\frac{dT(t)}{dt} + kT(t) = kT_{\text{ext}} \quad \text{où} \quad k = \frac{2h}{\rho ca}$$

La température T exprimée en $^\circ\text{C}$ suit alors l'expression : $T(t) = 30 \times (1 - e^{-kt})$

En plaçant un capteur de température au centre de la tige, on obtient N valeurs $T_{i,\text{exp}}$ de températures aux instants $t_{i,\text{exp}}$. Un traitement informatique permet alors d'obtenir deux tableaux accessibles sur un programme Python :

- `tab_temps` qui recense les instants $t_{i,\text{exp}}$ des mesures.
- `tab_T` qui contient les valeurs $T_{i,\text{exp}}$ des températures mesurées.

On cherche maintenant à déterminer la valeur de k telle que $T(t) = 30 \times (1 - e^{-kt})$ puisse effectivement correspondre aux points expérimentaux à l'aide de Python en s'inspirant de la méthode des moindres carrés.

- 21)** Écrire une fonction `Temp(t, k)` prenant en arguments d'entrée un tableau `t` de nombre décimaux et un nombre décimal `k` et renvoyant un tableau, noté `tab_Ttheo`, contenant les valeurs vérifiant $T_{i,\text{theo}} = 30 \times (1 - e^{-kt_i})$ pour tous les éléments t_i du tableau `tab_temps`.

Pour la question suivante, on pourra utiliser `sum(tableau)` qui renvoie la somme de tous les éléments d'un tableau.

- 22)** Écrire une fonction `erreur(k)` prenant pour argument un nombre décimal `k` et renvoyant la quantité
$$e = \sum_{i=1}^N (T_{i,\text{exp}} - T_{i,\text{theo}})^2$$
 donnant la somme du carré des écarts entre les températures expérimentales $T_{i,\text{exp}}$ et les températures $T_{i,\text{theo}}$ vérifiant la loi recherchée.

Dans la suite, on cherche à déterminer précisément la valeur du paramètre k qui permet de minimiser e . Dans notre cas, on estime manuellement que $(10^{-4} \leq k \leq 10^{-2}) \text{ s}^{-1}$.

- 23)** Proposer un algorithme utilisant une boucle `for` et permettant d'obtenir un tableau, noté `tab_e()`, contenant les valeurs de e calculées pour 100 valeurs de k comprises dans l'intervalle 10^{-4} s^{-1} et 10^{-2} s^{-1} . On pourra utiliser la fonction du module `numpy` `linspace(x1, x2, n)` qui renvoie un tableau de n éléments régulièrement espacés entre `x1` et `x2`.

question complémentaire

On rédigera un programme complet avec des mesures supposées,

```
tab_temps=np.linspace(0,3600,60) # toutes les minutes pendant 1 heure
tab_T=30*(1-np.exp(-0.001*tab_temps))+.01
```

pour une utilisation possible : affichage du tableau `tab_Ttheo` avec `k=0.001`, tracé du graphe de l'évolution théorique de la température (`tab_Ttheo` en fonction de `tab_temps`), affichage de l'erreur, affichage du tableau `tab_e`.

6. ATS 2021

première partie

contexte

Nous nous intéressons ici au temps de réponse d'un capteur de température présent dans un smartphone.

Nous allons assimiler ce capteur à une phase solide idéale, et nous admettons la loi de Newton décrivant la puissance thermique P algébriquement reçue par le capteur.

On souhaite tester le modèle précédent. On place alors le téléphone étudié dans un réfrigérateur à la température $T_0 = 5\text{ °C}$ suffisamment longtemps pour que le capteur atteigne cette température T_0 . À $t=0$, on expose le téléphone à une température $T_{\text{ext}} = 20\text{ °C}$.

On montre alors que $T(t) = T_{\text{ext}} + (T_0 - T_{\text{ext}})e^{-\frac{t}{\tau}}$, τ étant le temps caractéristique du régime transitoire, la température se stabilisant au bout de quelques τ .

question

Les valeurs des températures mesurées depuis l'instant $t=0$ sont reportées dans un fichier nommé *mesures.csv*.

Remarque : il s'agit d'un fichier texte où les données d'une même ligne sont séparées par des tabulations.

Sous Python, on peut récupérer ces données dans un tableau, appelé M , avec les deux lignes de code ci-dessous :

```
import numpy as np
M = np.loadtxt('mesures.csv')
```

```
M = [[ 0.  5. ]
      [ 3.  9.1]
      [ 6. 11.8]
      [ 9. 13.7]
      [12. 15.4]
      [15. 16.7]
      [18. 17.4]
      [21. 18.2]
      [24. 18.7]
      [27. 19.1]
      [30. 19.3]
      [33. 19.5]
      [36. 19.5]
      [39. 19.6]
      [42. 19.7]
      [45. 19.8]
      [48. 19.9]
      [51. 19.9]
      [54. 19.9]]
```

Ci-contre est affiché le tableau M .

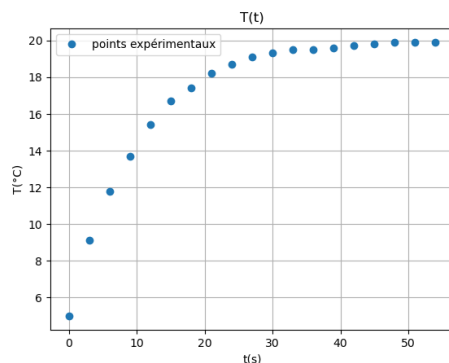
La première colonne contient les instants d'acquisition et la deuxième colonne contient les températures associées à ces instants d'acquisition.

19) Écrire un programme permettant :

- d'obtenir deux tableaux à une dimension appelés t et T contenant respectivement les valeurs des temps d'acquisition et les valeurs des températures mesurées ;
- d'afficher le graphe expérimental $T(t)$.

Précision : le graphe comportera uniquement les points, et devra comporter un titre, une légende et un étiquetage des axes, suivant l'indication suivante.

On obtient le graphe expérimental ci-dessous :



question complémentaire

Écrire un programme utilisable, commençant par la définition et l'affichage du tableau M à l'aide de `np.array()` par exemple, simulant l'importation des données d'un fichier CSV. Vous pourrez ensuite afficher les tableaux t et T demandés, et compléter le programme par le tracé du graphe demandé.

deuxième partie**contexte**

Les smartphones sont munis d'un capteur de pression. Le jour de l'expérience, un opérateur mesure une pression de $P_0 = 1000,00 \text{ hPa}$ au niveau du sol et de $P(H) = 999,80 \text{ hPa}$ en posant le téléphone sur sa tête.

On en déduit alors la taille H de cet opérateur : $H \approx \delta \left(\frac{P_0 - P(H)}{P_0} \right) \approx 1,7 \text{ m}$

question

Dans la suite, on suppose que l'incertitude-type $u(H)$ sur la valeur de H n'est due qu'à l'incertitude-type $u(P) = 0,02 \text{ hPa}$ sur la lecture des pressions. Afin d'estimer $u(H)$, on va utiliser la méthode numérique de Monte-Carlo. On donne ci-dessous des renseignements sur quelques fonctions *Python*, nécessitant l'importation préalable du module `numpy` (as np) :

`P=np.random.normal(loc=Pmes, scale=u_P)` : P est un nombre tiré au hasard parmi des valeurs réparties selon une distribution de valeur moyenne P_{mes} et d'écart type u_P .

`u=np.std(tab)` : u est l'incertitude-type des valeurs du tableau `tab`.

- 31) Écrire un programme utilisant une boucle `for` et permettant d'obtenir un tableau, appelé `tab_H`, de $N = 10000$ valeurs possibles de H du fait des incertitudes sur les lectures de pression. Compléter ce programme afin d'obtenir $u(H)$.

7. ATS 2022 (remarque : la méthode d'Euler n'est plus au programme)**contexte**

Chute d'un grêlon avec frottements quadratiques.

On considère un grêlon de masse m , qui chute dans le champ de pesanteur \vec{g} . On note z un axe descendant vers le sol. $z = 0$ marque la position initiale du grêlon lorsqu'il est lâché dans le nuage. La vitesse initiale est nulle. On note \vec{e}_z un vecteur unitaire orienté vers le bas. On prend en compte des frottements entre le grêlon et l'air. On note $\vec{v} = v(t)\vec{e}_z$ la vitesse du grêlon. La force de frottement de l'air sur le grêlon peut s'écrire : $\vec{f} = -\alpha v^2 \vec{e}_z$

Pour les vitesses atteintes par les grêlons, des études en soufflerie sur des sphères montrent que le coefficient α est donné par $\alpha = \frac{1}{2} \rho_{air} \pi R^2 C$, avec ρ_{air} la masse volumique de l'air, R le rayon du grêlon et $C \approx 0,5$.

On montre alors que la vitesse obéit à l'équation différentielle $\dot{v} + \frac{\alpha}{m} v^2 = g$ qui conduit à une vitesse limite $v_{lim} = \sqrt{\frac{mg}{\alpha}}$.

questions

On étudie ensuite le mouvement du grêlon à l'aide d'une résolution numérique. Les hypothèses et notations sont toujours les mêmes. On utilise pour cela la méthode d'Euler, dans un algorithme écrit en *Python* retranscrit ci-dessous :

```
g = 9.8
alpha = 1.5e-3
m = 0.24
dt = 0.02 # pas d'intégration en secondes
fin = 15 # durée de la simulation en secondes

t(1) = 0 # temps initial
z(1) = 0 # position initiale
v(1) = 0 # vitesse initiale

nb_iterations = int(fin/dt)
# pour information : int = troncature (arrondi par suppression des décimales)
for i in range(1, nb_iterations):
    t(i+1) = t(i) + dt
    z(i+1) = [case 1 à compléter]
    v(i+1) = [case 2 à compléter]
end
```

Les valeurs de m et α sont ici données en unités S.I. et sont valables pour un grêlon de diamètre 8 cm.

33 - Compléter la case 1 de l'algorithme.

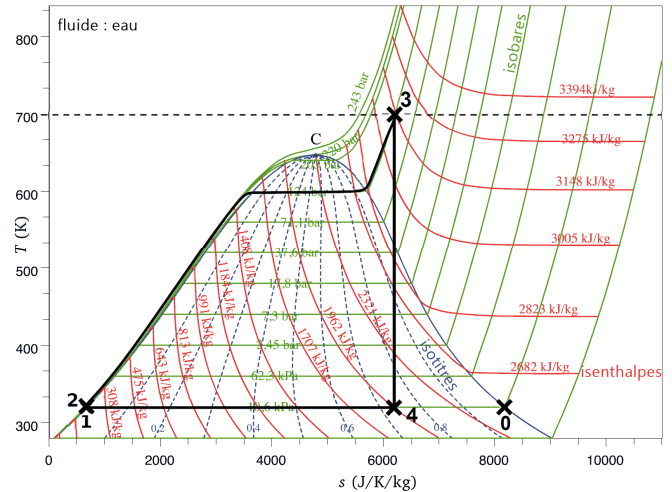
34 - Compléter la case 2 de l'algorithme.

8. ATS 2023

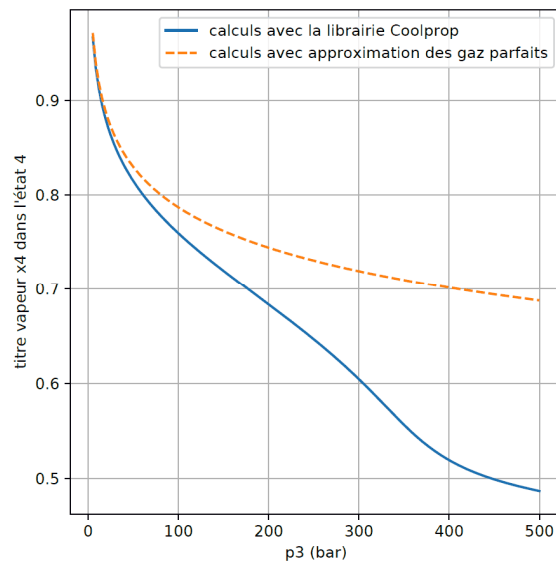
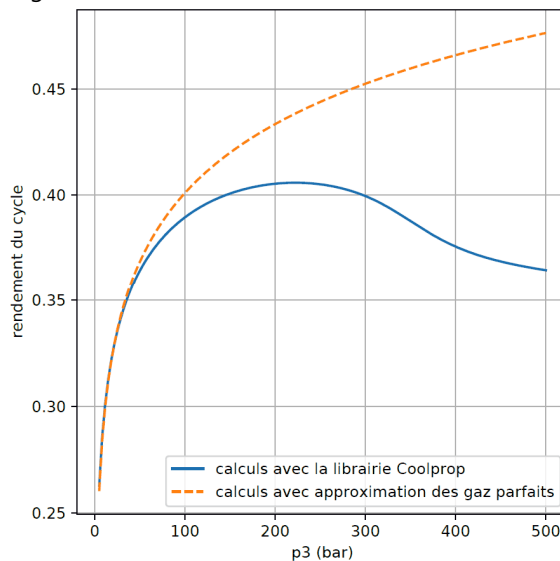
contexte

Nous étudions ici le cycle thermodynamique suivi par de l'eau, dans un exemple simplifié et typique de ce qui est mis en œuvre dans une centrale thermique de puissance moyenne (5 à 20 MW).

On montre que le cycle a l'allure suivante →



On fournit également ce document :



Document 12 : rendement du cycle (à gauche) et titre en vapeur dans l'état 4 (à droite) en fonction de la valeur choisie pour la pression p_3 . Ici $T_3 = 700$ K est fixé et l'état 1 est toujours le même. Courbes en traits pointillés : d'après les expressions approchées établies dans la sous-partie III.2. Courbes en traits pleins : calculs réalisés avec la librairie Coolprop (qui sert aussi au tracé du diagramme T - s de l'énoncé).

Recherche du rendement maximal

Cette sous-partie mobilise des capacités numériques et peut être traitée sans avoir abordé les autres.

On dispose de tableaux qui contiennent les données suivantes :

- p_3 contient des valeurs de pression allant de 5 bar à 500 bar avec $N=150$ valeurs.
- η_a contient les valeurs calculées pour le rendement, c'est-à-dire que $\eta_a[i]$ est la valeur du rendement lorsque la pression p_3 vaut $p_3[i]$. Ces valeurs correspondent à la courbe en trait plein du graphique de gauche du document 12.
- De même, x_4 contient les valeurs calculées pour le titre en vapeur dans l'état 4 (courbe en trait plein du graphique de droite du document 12).

Il apparaît graphiquement que le tableau η_a admet une valeur maximale (courbe en trait plein du graphique de gauche du document 12), dont on cherche à déterminer la valeur à l'aide d'un algorithme écrit dans le langage Python :

```

maxi = 0
imax = 0

for i in range(1,150):
    if eta[i] >= maxi :
        [bloc 1 à compléter]

print(imax) # affichage dans la console
print(maxi) # affichage dans la console
[bloc 2 à compléter]

```

- 42 - Proposer les instructions manquantes dans le bloc 1 ci-dessus, afin que le programme affiche la valeur maximale du tableau `eta`, ainsi que l'indice `i` qui correspond.
- 43 - Proposer les instructions manquantes dans le bloc 2 ci-dessus, afin que le programme affiche également les valeurs de la pression `p3` et du titre `x4` correspondant au maximum de `eta`.

question complémentaire

Tester le programme en insérant au début la série de données à traiter (simulation) :

```

# données approximatives relevées sur les courbes (12 points) suivies de 138 zéros pour faire 150 termes, afin de tester le programme
p3= [0, 50, 100, 150, 200, 225, 250, 300, 350, 400, 450, 500] + [0]*138
eta=[0, 0.36, 0.39, 0.40, 0.41, 0.415, 0.41, 0.40, 0.385, 0.375, 0.37, 0.365]+[0]*138
x4 =[1, 0.81, 0.76, 0.72, 0.68, 0.67, 0.65, 0.605, 0.55, 0.52, 0.50, 0.49] + [0]*138

```

Vous en profiterez pour améliorer l'affichage des résultats en ajoutant du texte dans l'instruction `print()`.

9. ATS 2024

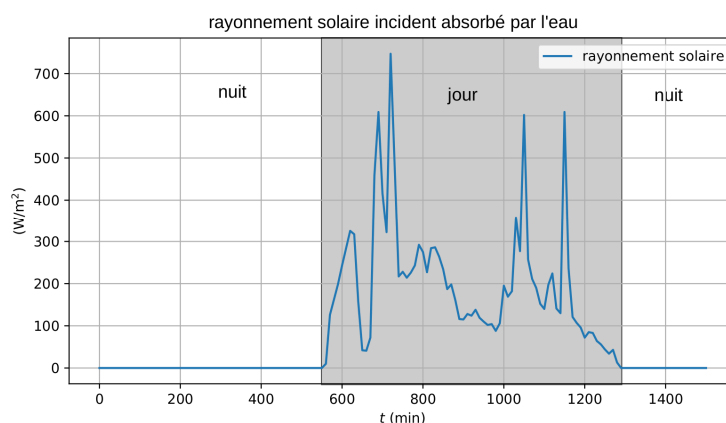
Le sujet aborde différentes notions sur le thème de la piscine. Tout d'abord le chauffage par le soleil.

1.3.1 Rayonnement solaire

Le rayonnement solaire est un apport important d'énergie. Un capteur permet d'enregistrer la puissance thermique surfacique de ce rayonnement (document 5). Notons-la y . Afin d'obtenir le transfert thermique reçu, il faut intégrer cette puissance et la multiplier par la surface S du bassin :

$$Q_{\text{solaire}} = \int_0^{t_{\text{fin}}} y(t) dt \times S \quad (2)$$

Il est nécessaire de calculer cette intégrale numériquement à partir des données.



Document 5 : puissance thermique surfacique liée au rayonnement solaire direct.

On dispose de tableaux, déjà définis dans l'algorithme, qui contiennent les données suivantes :

- `t` contient les instants auxquels les enregistrements ont lieu, en minutes.
- `y` contient les valeurs de la puissance thermique surfacique du rayonnement solaire reçu, c'est-à-dire que y_i est la valeur du rayonnement à l'instant t_i , en $W \cdot m^{-2}$.

- 16 - Proposer l'instruction manquante dans l'algorithme ci-dessous, afin qu'il retourne bien la grandeur Q_{solaire} en joules. On utilisera la méthode des rectangles pour le calcul de l'intégrale.

```

n = len(t)
somme = 0
for i in range(n-1):
    # À COMPLÉTER

print(somme*300) # affiche l'énergie reçue, en joules

```

question complémentaire

Tester le programme en ajoutant en début du script :

```
import numpy as np
t = np.linspace(600,1200,1000) # 1000 valeurs entre 600 et 1200 min → assez bon résultat
y = np.linspace(0,600,1000) # rayonnement croissant linéairement de 0 à 600 W/m²
Q = 3.24E+09 # valeur exacte de l'intégrale correspondante (aire d'un triangle rectangle)
print('Q attendue =',Q, 'J')
```

Le sujet aborde maintenant ce qui concerne la mesure du niveau d'eau.

II.3 Traitement des données

En pratique, le niveau d'eau peut fluctuer autour de sa valeur moyenne à cause de la présence de vagues. Pour remédier à ceci, plusieurs valeurs du niveau sont relevées dans un intervalle de temps de l'ordre de la minute.

On dispose ainsi de $n = 100$ valeurs de hauteur stockées dans le tableau `h`. On les traite à l'aide d'un algorithme (le tableau `h` est déjà défini), écrit en langage Python :

```
n = len(h)
somme = 0
for i in range(n):
    # instruction 1

    # instruction 2
print(moyenne) # affiche la moyenne
```

33 - Compléter les instructions 1 et 2 ci-dessus afin que l'algorithme retourne la valeur moyenne du tableau `h`.

34 - Proposer, sur le même modèle que ci-dessus, un algorithme qui permet de calculer la variance :

$$V = \frac{1}{n} \sum_i (h_i - \text{moyenne})^2,$$

où l'indice i parcourt toute l'étendue des indices du tableau `h`. On supposera que `moyenne` est déjà calculée.

question complémentaire

Tester le programme en ajoutant un tableau type en début de script :

```
import numpy as np
# exemple de tableau :
h=np.linspace(9,11,100) #liste de 100 valeurs entre 9 et 11 : moyenne ≈ 10
```