ÉPREUVE D'INFORMATIQUE (PYTHON) №1 (CORRIGÉ)

M. A. Ammar - IPEST : HAP 16 décembre 2020

Exercice 1 : Calculer les niveaux d'énergie dans un atome

Le n^{ime} niveau d'énergie d'un électron dans un atome d'hydrogène est donné par :

$$E_n = -\frac{m_e e^4}{8\varepsilon_0^2 h^2} \cdot \frac{1}{n^2}, \qquad (1)$$

où $m_e = 9.109410^{-31}\ kg$ est la masse de l'électron, $e = 1.602210^{19}\ C$ est la charge élémentaire, $\varepsilon_0 = 8.8542 \cdot 10^{-12} C^2 s^2\ kg^{-1} m^{-3}$ est la permittivité électrique du vide, et $h = 6.6261 \cdot 10^{34}\ Js$

Q1. On défini d'abord les constantes dans l'équation, ensuite la fonction `E(n)` qui retourne la valeur du niveau dénergie en électron-volt (eV) :

```python
# Constantes
me = 9.1094e-31
e = 1.6022e-19
eps0 = 8.8542e-12
h = 6.6261e-34
def E(n) :
    Ejoule = - (me * e**4)/(8*eps0**2 * h**2)* (1/n**2)
    return Ejoule/e
```

Q2. Le niveau d'énergie pour n = 1 :

```python
print("E(n = 1) = ", E(n = 1), " eV")
# ==> E(n = 1) =  -13.606152702370753  eV
```

Le niveau d'énergie le plus bas $E_1 = -13{,}6\ eV$ obtenu pour n = 1, correspond au niveau fondamental de l'atome d'hydrogène. C'est l'état le plus stable.

Q3. Le niveau d'énergie pour n = 100 :

```python
print("E(n = 100) = ", E(n = 100), " eV")
# ==> E(n = 100) =  -0.0013606152702370755  eV
```

Le niveau d'énergie est nul $E = 0\ eV$ lorsque n tend vers l'infini (l'électron est alors séparé du noyau).

Q4. On peut calculer et afficher les valeurs E_n pour $n = 1,,20$ en utilisant une boucle for :

```
for n in range(1, 21) :
    print("E{} = {} eV".format(n, E(n)))
```

Le résultat doit être comme suivant :

```
E1 = -13.606152702370753 eV
E2 = -3.4015381755926883 eV
............................
............................
E19 = -0.03769017369077771 eV
E20 = -0.03401538175592689 eV
```

Q5. On peut créer la matrice $\Delta E^{i,f}$ et afficher ces valeurs avec la méthode suivante :

```
from numpy import array
DEn = [[E(ni) - E(nf) for ni in range(1, 6)] for nf in range(1,6)]
print(array(DEn))
#==> DEn =
#[[  0.          10.20461453  12.09435796  12.75576816  13.06190659]
# [-10.20461453   0.           1.88974343   2.55115363   2.85729207]
# [-12.09435796  -1.88974343   0.           0.6614102    0.96754864]
# [-12.75576816  -2.55115363  -0.6614102    0.           0.30613844]
# [-13.06190659  -2.85729207  -0.96754864  -0.30613844   0.        ]]
```

Exercice 2 : Tracer la viscosité de l'eau

La viscosité de l'eau, μ, varie avec la température T (en Kelvin) selon la formule :

$$\mu(T) = A \cdot 10^{B/(T-C)} \tag{2}$$

où $A = 2.414 \cdot 10^{-5}$ Pa s, $B = 247.8\ K$ et $C = 140\ K$.

Q1. La fonction `mu(T, A, B, C)` est définie comme suivant :

```
def mu(T, A, B, C) :
    return A*10**(B/(T-C))
```

Q2. Le code qui trace la viscosité de l'eau en fonction de la température est comme suivant :

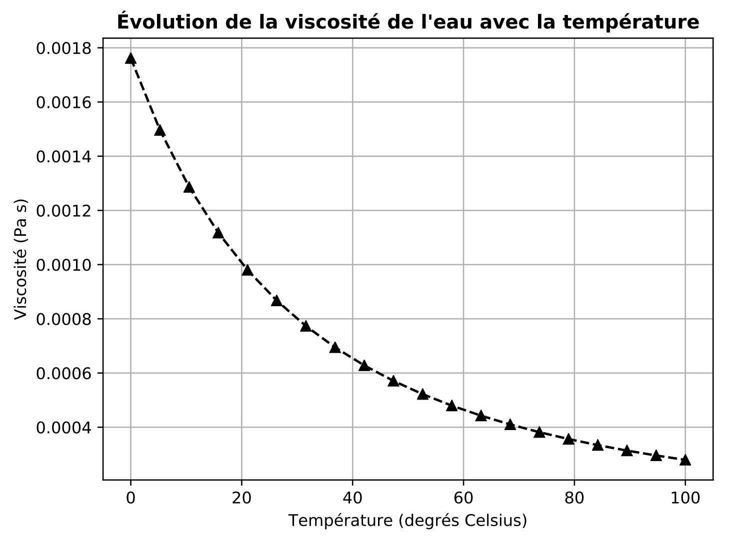
```python
import matplotlib.pyplot as plt
import numpy as np
# 0 deg C = 273 deg K
T = np.linspace(0, 100, 20)

plt.plot(T, mu(T+273, A = 2.414e-5, B = 247.8, C = 140), 'k^--')
plt.title("Évolution de la viscosité de l'eau avec la température",
          fontweight='bold')
plt.xlabel("Température (degrés Celsius)")
plt.ylabel("Viscosité (Pa s)")
plt.grid()
```

Indications.

— On vous rappelle que : 0 °C = 273 °K.

— La sortie du programme devrait ressembler à la figure ci-dessous.



Exercice 3 : Diffraction par ouverture rectangulaire

Considérons un faisceau de lumière monochromatique de longueur d'onde λ éclairant une ouverture rectangulaire située dans un plan (xOy). La largeur de l'ouverture b est dans la direction x et sa hauteur h est dans la direction y.

L'intensité normalisée de lumière en un point M situé sur un écran (E) et à une distance D de la fente peut s'écrire comme suit :

$$\frac{I(x_M, y_M)}{I_0} = \text{sinc}^2\left(B \cdot x_M\right) \text{sinc}^2\left(H \cdot y_M\right) \tag{3}$$

où $H = \frac{\pi h}{\lambda D}$, $B = \frac{\pi b}{\lambda D}$.

— La largeur de la tache centrale dans la direction x est inversement proportionnelle à la largeur de l'ouverture : $\Delta x = \frac{2\lambda D}{b}$;

— La largeur de la tache centrale dans la direction y est inversement proportionnelle à la hauteur de l'ouverture : $\Delta y = \frac{2\lambda D}{h}$.

La fonction Python `DiffRect(lamda, b, h, D)`, qui calcul Δx et Δy et affiche la figure de diffraction, peut s'écrire comme suivant :

```python
import numpy as np
import matplotlib.pyplot as plt
def DiffRect(lamda, b, h, D) :
    k = (2*np.pi)/lamda  # wavelength of light in vaccuum
    a = 30 * 1.E-2  # Side of a square-shaped screen (m)
    # The width of the central maximum along (Ox)
    delta_x = 1.E2 * (2 * lamda * D) / b
    print("Largeur de la tache centrale suivant les x  : ", delta_x)
    # The width of the central maximum along (Oy)
    delta_y = 1.E2 * (2 * lamda * D) / h
    print("Largeur de la tache centrale suivant les y  : ", delta_y)
    N = 400
    X = np.linspace(-a/2, a/2, N)
    Y = X  # coordinates of screen
    B = (k * b * X) / (2. * D)
    H = (k * h * Y) / (2. * D)  # intermediate variable
    # 2D & 3 D representation
    BB, HH = np.meshgrid(B, H)
    I = ((np.sin(BB) / BB)**2) * ((np.sin(HH) / HH)**2)
    # figure 2D
    plt.imshow(I, cmap='gray', interpolation='bilinear',
               origin='lower', vmin=0, vmax=.005)
    plt.xlabel('$X$', fontsize=12, fontweight='bold')
    plt.ylabel('$Y$', fontsize=12, fontweight='bold')
    plt.title('Diffraction de Fraunhofer par ouverture rectangulaire')
    plt.show()
```

scripts/DiffRect.py

```
>>> DiffRect(lamda= 630*1.E-9, b= 2*1.E-5, h= 4*1.E-5, D= 2)
La largeur de la tache centrale dans la direction x :  12.6
La largeur de la tache centrale dans la direction y :  6.3
```