

0.1 Dosage des ions Calcium et Magnesium dans de l'eau

**

Des mesures par dosage de la concentration en ions Calcium et Magnesium sont réalisées. A partir d'une série de mesures, vous allez calculer des grandeurs caractéristiques de cette série.

Afin de générer artificiellement cette série de mesures, vous téléchargerez depuis moodle le fichier `giveConcentration.py`. Vous écrirez en début de votre programme les lignes suivantes qui appellent la fonction `genereC` du module `giveConcentration` pour créer une liste appelée `C` de `N` concentrations mesurées.

```
1 import giveConcentration as gc
2
3 N = 150
4 C = gc.genereC(N)
5 print(C)
```

Dans cet exercice, vous ne devez pas utiliser de boucles.

1. Convertir la liste `C` en un tableau.
2. Calculer la moyenne de la concentration : \bar{c} .
3. Trouver le minimum et le maximum de la concentration
4. Calculer l'écart-type de la série de mesure sans faire de boucles. On donne la formule permettant le calcul de l'écart-type

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (c_i - \bar{c})^2}$$

où N est le nombre de valeurs de la série générée par `genereC`. Les c_i correspondent aux éléments de la matrice des concentrations `C` générée par le code ci-dessus.

0.2 Vitesse des molécules dans un gaz parfait

**

La théorie cinétique des gaz permet de connaître pour une température T donnée (en Kelvin) la quantité d'atomes (ou de molécules pour un gaz moléculaire) dont la vitesse v est comprise entre v et $v + dv$ où dv représente un élément de taille très petite de la vitesse comparé à la valeur de v (on parle de variation infinitésimale). En langage plus familier, cela permet de connaître la fraction d'atomes ayant une certaine vitesse. Nous appellerons cette quantité $\rho(v, T)$. Cette loi de variation s'appelle la loi de Maxwell-Boltzmann.

La fonction `ddp` du module `distribVT` permet de calculer la valeur de cette loi pour une vitesse v et une température T données. Pour pouvoir l'utiliser, il faut que le fichier `distribVT.py` soit présent dans votre répertoire de travail. Dans ce module, le gaz considéré est l'Argon. L'exemple ci-dessous montre comment utiliser ce module pour calculer la valeur de la loi de Maxwell-Boltzmann pour une température $T = 273$ K et une vitesse de 500 m s^{-1} , c'est à dire $\rho(500, 273)$.

```

1 import distribVT as vt
2
3 T=273
4 print(vt.ddp(500,T))

```

1. Faire le nécessaire pour qu'un utilisateur puisse saisir la température de travail en faisant en sorte que l'utilisateur ne puisse pas saisir une température inférieure à la température de liquéfaction de l'argon qui est de -185.85°C à la pression atmosphérique. Cette valeur sera placée dans une variable **T**. Vous choisirez dans un premier temps une valeur 25°C .
2. Faire le nécessaire pour qu'un utilisateur puisse saisir la vitesse maximale du domaine des vitesses, v_{\max} . Cette valeur sera placée dans une variable **vMax**. Vous choisirez dans un premier temps une valeur de 1500 m s^{-1} .
3. Créer un tableau vitesse **v** comprenant $n = 2000$ valeurs entre 0 m s^{-1} et v_{\max} . Calculer dans une variable **pas**, l'écart de vitesse entre 2 vitesses successives dans le tableau **v**.
4. Créer un tableau **val** contenant les valeurs de la loi de Maxwell-Boltzmann pour chacune des vitesses du tableau **v**.
5. Donner la vitesse la plus probable, c'est à dire, celle qui correspond au maximum du tableau **val**. Il ne s'agit pas d'en faire un calcul théorique (une formule existe) mais de trouver sa valeur dans le tableau vitesse **v**.
6. Calculer la vitesse moyenne \bar{v} . Pour cela, vous utiliserez la méthode des trapèzes et la formule suivante :

$$\bar{v} = \int_0^{+\infty} v \rho(v, T) dv$$

Appliquée à la méthodes des trapèzes, on a dans le cas qui nous intéresse :

$$\bar{v} = pas \left(\frac{(v_0 \rho(v_0, T) + v_{n-1} \rho(v_{n-1}, T))}{2} + \sum_{i=1}^{n-2} v_i \rho(v_i, T) \right)$$

Vous remarquerez que la vitesse la vitesse moyenne n'est pas la plus probable !.

AMÉLIORATIONS FACULTATIVES (demander à l'enseignant si vous avez le temps de la traiter en séance ou s'il vaut mieux passer à l'exercice suivant) : Si vous avez réussi les questions précédentes, C'est que le sujet a été écrit pour vous faciliter la tâche. On vous a fixé les données suivantes : la température et la vitesse maximale v_{\max} . Mais si vous choisissez une température plus élevée qui vous garantit qu'un nombre important d'atomes ne vont pas dépasser les 1500 m s^{-1} ? Il faut donc vérifier avant les calculs de la vitesse la plus probable et de \bar{v} que l'ensemble des vitesses choisies est suffisant. S'il n'est pas suffisant il faudra alors demander à l'utilisateur de choisir une valeur v_{\max} supérieure à celle choisie précédemment. Mais comment savoir si v_{\max} est suffisamment grande ? Pour cela, il faut s'assurer que l'intégrale $I = \int_0^{v_{\max}} \rho(v, T) dv$ est proche de 1. Par exemple, on peut considérer que si $1 - I < 10^{-3}$, la plage de vitesses est suffisante. Sinon, cela signifie que la valeur saisie pour v_{\max} est trop faible et qu'il faut demander une valeur plus importante et ainsi de suite jusqu'à ce que I vérifie $1 - I < 10^{-3}$. Les questions suivantes portent sur l'implémentation de ce processus.

7. Calculer I par la méthode des trapèzes :

$$I = pas \left(\frac{(\rho(v_0, T) + \rho(v_{n-1}, T))}{2} \right) + \sum_{i=1}^{n-2} \rho(v_i, T)$$

8. Faire en sorte que l'utilisateur saisisse à nouveau une valeur de v_{\max} tant que la relation $1 - I < 10^{-3}$ n'est pas respectée avant de calculer la vitesse la plus probable et \bar{v} .

Vous pouvez maintenant faire les calculs pour toutes les températures que vous voulez :)

0.3 écrêtage et lissage de courbes

Des mesures donnent l'évolution d'une concentration en fonction du temps.

Afin de générer artificiellement cette série de mesures, vous téléchargerez depuis moodle le fichier **decConcentration.py**. Vous écrirez en début de votre programme les lignes suivantes qui appellent la fonction **genereC** du module **decConcentration** pour créer une **liste** appelée **C**.

```
1 import genereConcentration as gc
2 from pylab import *
3
4 C = gc.decC()
5 C=array(C)
6 plot(C)
```

1. Convertir la liste **C** en un tableau.
2. Certaines mesures de concentration sont négatives suite à des erreurs de mesure. Sans faire de boucles, remplacer par la valeur 0 les valeurs négatives dans le tableau **C**.
3. Les données présentent beaucoup de variations d'une mesure à l'autre. Vous allez lisser ces valeurs en utilisant la méthode de la moyenne mobile. Cette méthode consiste à calculer la i ème valeur du tableau des valeurs lissées, que nous appellerons L , comme étant la moyenne centrées autour de C_i des k valeurs qui le précèdent, des k valeurs qui le suivent et de lui-même. On a ainsi la formule,

$$L_i = \frac{1}{2k + 1} \sum_{j=i-k}^{i+k} C_j$$

Calculer les valeurs du tableau L . Vous laisserez les k premières valeurs de L_i égales à celle de C_i et de même pour les k dernières.

Vous pouvez en exécutant **plot(L)** voir l'effet de votre lissage. Vous pouvez vous amuser à le faire pour différentes valeurs de k .

4. (facultatif) Trouver un moyen d'améliorer le traitement des k premières et des k dernières valeurs pour que le lissage concernent également cette partie des valeurs.