Formation Juin 2019 :

⋆ ⋆ ⋆

Python pour l'enseignant de Physique-Chimie au lycée

⋆ ⋆ ⋆

Bonus 1

Table des matières

— *Le lien vers cet énoncé en ligne est ici sur le "padlet" (Bonus 1 formation) ; téléchargez-le et utilisez-le pour accéder aux liens cliquables de l'énoncé.*

Vous pouvez aussi télécharger l'énoncé à cette adresse :
`https://github.com/formationPythonPC-Juin/enonce-donnees/blob/master/bonus1.pdf`

— *Par rapport à l'énoncé précédent, celui-ci ne propose pas forcément de fichiers d'aides.*

Quoiqu'il en soit, les potentiels fichiers d'aides et les fichiers corrections sont donnés en lien à la fin de chaque partie.

Vous réalisez donc les exercices via le niveau 1 (pas d'aide), le niveau 2 (avec aides si le fichier est fourni) ou le niveau 3 (correction donnée, il vous faut alors commenter les lignes de code pour expliquer à quoi elles servent).

6 Exercice 6 : Importer des données expérimentales pour les traiter par Python

On a réalisé à l'aide d'un microcontrôleur le relevé de mesures correspondant à l'expérience suivante :
— un capteur de température permet d'acquérir la température d'un liquide qui refroidit : donnée `"température"`
— en parallèle, on fait calculer au microcontrôleur la résistance équivalente à un dispositif contenant une thermistance plongé dans la solution : donnée `"Réquivalente"`

Le microcontrôleur permet d'afficher ces deux valeurs au cours du temps dans un fichier que vous avez à disposition : `donnees`.

On souhaite représenter le graphe `Réquivalente` en fonction de `température` dans l'espoir de pouvoir modéliser le comportement par une droite.

1. Ouvrez le fichier `"donnees"` à cette adresse ; observez son contenu.

   Le grand nombre de lignes rend difficile le traitement par un tableur. Cliquez sur `"Raw"` puis sélectionner l'ensemble (`Ctrl+a`) et copiez-coller tout cela dans un fichier que vous pourrez appeler `donnees`.

2. Créez un fichier `exercice6.py`. Ce fichier sera placé dans le même répertoire que le fichier de données ci-dessus.

   Votre fichier `exercice6.py` doit permettre :
   — de placer les données dans 2 listes (qu'on pourra appeler `T` et `R`)
   — de représenter l'ensemble des points de coordonnées `température ; résistance`
   — de modéliser l'évolution de ces points expérimentaux par une droite

6.1 Récupérer les données dans des "listes"

On souhaite créer deux listes (ou tableaux) `R` et `T` à partir des valeurs contenues dans le fichier de données.

3. En vous servant du *vade-mecum* présent sur notre site disciplinaire à ce lien ou encore ici ("Importer des données"), construisez les deux "listes" `T` et `R` utiles.

   REMARQUE POUR LES UTILISATEURS DE PANDAS : Pour être précis, T et R ne sont pas des listes ni des tableaux mais un type particulier de données propre à la bibliothèque pandas. En tout cas, leurs constituants sont des nombres avec lesquels on peut faire toutes les opérations permises par Python.

6.2 Représentation graphique

4. Représentez alors l'ensemble des points de coordonnées `température ; résistance` sur un graphe. Donnez un titre au graphe, légendez-le ; donnez un nom aux axes. Faites afficher le graphe. Conclusion.

6.3 Modélisation de l'évolution des points expérimentaux

5. Les points précédents semblent alignés. Trouvez et faites afficher les coefficients caractéristiques de la droite modèle.
6. Construisez et affichez sur le graphe la droite modèle de ce comportement. Intégrez de même une légende pour cette droite.

► `lien vers la correction de cet exercice : exercice6-correction.py`◄

7 Exercices 7 et 8 : Mécanique – Étude énergétique

Au programme de Première Spécialité : Utiliser un langage de programmation pour effectuer le bilan énergétique d'un système en mouvement.

7.1 Sujet

On souhaite simuler la trajectoire d'un solide matériel considéré comme ponctuel avant de réaliser un bilan énergétique.

Celui-ci est lancé depuis le point (0,0) avec un angle $\alpha = 55^\circ$ par rapport à l'horizontale avec une vitesse de norme $v_0 = 10$ m/s à l'instant $t = 0$.

La masse de l'objet est $m = 0.40$ kg, l'accélération de la pesanteur valant $g = 9,8$ N/kg.

Le modèle sur lequel on s'appuie intègre que les frottements de l'air sur l'objet peuvent être considérés comme une force $\vec{f} = -k \cdot \vec{v}$ et on prendra $k = 5,0 \cdot 10^{-2}$ SI.

On pose de plus $\tau = \dfrac{m}{k}$

Dans ces conditions, la vitesse est donnée par : $\vec{v} = \vec{v_0} \cdot \mathrm{e}^{-\left(\frac{t}{\tau}\right)} + \tau \cdot \vec{g} \times \left(1 - \mathrm{e}^{-\left(\frac{t}{\tau}\right)}\right)$

et le point se situe à $\vec{r} = \vec{v_0} \cdot \tau \times \left(1 - \mathrm{e}^{-\left(\frac{t}{\tau}\right)}\right) + \tau \cdot \vec{g} \cdot t + \tau^2 \cdot \vec{g} \times \left(\mathrm{e}^{-\left(\frac{t}{\tau}\right)} - 1\right)$

soit en projection sur les axes : $\begin{cases} v_x = v_0 \cdot \cos\alpha \cdot \mathrm{e}^{-(t/\tau)} \\ v_y = v_0 \cdot \sin\alpha \cdot \mathrm{e}^{-(t/\tau)} - \tau \cdot g \times \left(1 - \mathrm{e}^{-(t/\tau)}\right) \end{cases}$

$$\begin{cases} x_M = & v_0 \cdot \cos\alpha \cdot \tau \times \left(1 - \mathrm{e}^{-(t/\tau)}\right) \\ y_M = & \left(v_0 \cdot \sin\alpha \cdot \tau + \tau^2 \cdot g\right) \times \left(1 - \mathrm{e}^{-(t/\tau)}\right) - \tau \cdot g \cdot t \end{cases}$$

Pour cet exercice, vous pouvez créer un fichier, `exercice7.py`.

7.2 Exercice 7 : Visualisation de la trajectoire

Ici, il va nous falloir construire les listes (ou tableaux) T, X et Y donnant les coordonnées temporelles et spatiales du point.

On pourrait construire des listes, cela serait un peu long. On préfèrera utiliser la bibliothèque `numpy` qui permet de réaliser des calculs "naturels" sur des tableaux.

1. Rentrez les données de l'exercice dans des variables aux noms appropriés (les puissances de 10 se notent `e` sous Python ; de plus, la fonction `radians()` de la bibliothèque `math` convertit un angle en degrés en radians).
2. L'étude se fera entre $t_0 = 0$ et $t_f = 1,7$. On choisira un pas de temps $dt = 0,1$ s.

   Entrez ces valeurs dans des variables aux noms appropriés. Construire alors un **tableau** contenant tous les instants de l'étude.

   AIDE :

   Numpy (sous l'alias np) propose de construire très simplement des tableaux de valeurs ; 2 fonctions sont utiles : `np.arange()` et `np.linspace()` ; leurs paramètres sont les suivants :

   — `np.arange(debut, fin, intervalle_entre_les_points)`

   — `np.linspace(debut, fin, nombre_de_points)`
3. Construisez alors les tableaux X, Y, VX, VY tout au long du déplacement.

   AIDE : Numpy peut faire agir la fonction exponentielle sur tout un tableau grâce à la fonction `np.exp(...)`.
4. Tapez le code permettant de visualiser la trajectoire du mobile sous Matplotlib.

↪ `aide à la résolution : exercice7-aide.py`

7.3 Exercice 8 : Étude énergétique

1. Avec utilisation de Numpy : Après avoir importé la bibliothèque Numpy, implémentez 3 listes (ou tableaux) `Ep`, `Ec` et `E` qui permettront de connaître les énergies potentielle, cinétique et mécanique du mobile au cours de son déplacement.

   RAPPEL : a^2 s'écrit `a**2` en Python.

2. Tracez alors les courbes représentant ces 3 grandeurs au cours du temps. Faites de même afficher en console les valeurs des 3 types d'énergies pour chaque instant t de la liste T.

3. On constate que les valeurs d'énergies possèdent beaucoup trop de décimales par rapport aux nombres de chiffres significatifs de l'énoncé.

   Par malheur, Python ne dispose pas de fonction toute prête pour gérer le nombre de chiffres significatifs ; il faut la fabriquer...

   Par contre, Python dispose d'une fonction qui arrondit un nombre à n chiffres après la virgule, c'est la fonction `round(nbe_à_arrondir, n)`.

   À titre d'exemple, construire la liste `Xarrondie` contenant les éléments de X arrondis à 1 chiffre après la virgule.

   Faites afficher `X` et `Xarrondie`.

↬ aide à la résolution : exercice8-aide.py

► lien vers la correction de cette partie (exercices 7 et 8) : exercices7-8-correction.py◄

REMARQUE :

En fait Numpy propose de base une fonction `round()` qui arrondit tous les éléments du tableau à un certain nombre de chiffres après la virgule.

Ainsi, pour répondre à la question 3, il aurait suffi de taper :

```
Xarrondie = np.round(X, 1)
```

8 Exercice 9 : Onde progressive, périodicité spatiale et temporelle

Au programme de Première (Spécialité) : Simuler à l'aide d'un langage de programmation, la propagation d'une onde périodique.

8.1 Sujet

On souhaite représenter la propagation d'un signal périodique qui s'écrit sous la forme : $I(x) = I_0 \times \cos(\omega \cdot t - k \cdot x + \phi)$ et visualiser la progression sur $[0\,;\,5]$

- L'étude se fait dans la région de l'espace des x : [0,5]
- On prendra $I_0 = 3$
- On prendra $k = \pi$
- On prendra $\phi = \dfrac{\pi}{2}$
- On prendra $\omega = \pi$

8.2 Point cours

POINT COURS :

C'est sans doute la partie la plus difficile (en terme de code) de ce que nous devrons traiter (En 2^{nde} et $1^{\text{ère}}$). Le principe est le suivant :

1. On construit une courbe vide en phase d'initialisation
2. On va remplir cette courbe par des listes x et I
3. La liste I va varier en fonction du temps, il faudra donc qu'elle soit actualisée (mise à jour) pour chaque instant avec la fonction `set_data()`
4. En incluant cela dans une boucle, on peut gérer ainsi une série de graphes différents pour des instants différents
5. `matplotlib.pyplot` dispose de la fonction `pause()` qui permet dans une boucle, d'afficher plusieurs images séparées d'un certain laps de temps, passé en paramètre de `pause()`.

REMARQUE :

Matplotlib dispose d'une fonction prête à l'emploi pour réaliser des animations, c'est la fonction `matplotlib.animation.FuncAnimation`.

Elle ne me semble pas très intéressante pour nos élèves ; en effet, on perd le sens physique de la propagation en l'utilisant. Pour cette raison nous ne l'étudierons pas.

Il faut noter tout de même que notre méthode ne permet pas de réaliser des animations en 3D, ce que permet de faire `FuncAnimation`.

8.3 Application

Allez chercher le code présent ici, copiez-le dans un nouveau fichier que vous pourrez appeler `exercice9.py`.

Ce code est déjà fonctionnel, vous pouvez le tester.

Votre travail : écrire les commentaires partout où ils manquent.

► `lien vers la correction de cet exercice : exercice9-correction.py` ◄