

TP 5 : TRACÉS DE COURBES

PREMIER CONTACT AVEC LES MODULES `numpy` ET `matplotlib`

1. Directement dans le Shell, importer les deux modules `numpy` et `matplotlib` :

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

2. Taper dans le Shell les instructions suivantes puis compléter la fiche de cours :

<pre>>>> L1 = list(range(21)) >>> L1 >>> L2 = np.linspace(0,20,21) >>> L2 >>> L3 = np.linspace(0,1,20) >>> L3 >>> L4 = np.linspace(0,1,10)</pre>	<pre>>>> L4 >>> L5 = np.arange(0,1,0.1) >>> L5 >>> L6 = np.arange(0,1,0.05) >>> L6</pre>
---	---

3. Taper dans le Shell les instructions suivantes puis compléter la fiche de cours.

Après chaque instruction `plt.show()`, on fermera la fenêtre de visualisation graphique avant de poursuivre.

<pre>>>> plt.plot(1,2,'o') >>> plt.show() >>> plt.plot([1,2],[3,4],'o') >>> plt.show() >>> plt.plot([1,2],[3,4]) >>> plt.show() >>> plt.plot([1,2,3],[3,4,2]) >>> plt.show()</pre>	<pre>>>> L1 = list(range(21)) >>> L2 = [x**2 for x in L1] >>> plt.plot(L1,L2,'o') >>> plt.show() >>> Abs = np.linspace(0,2*np.pi,50) >>> Ord = [np.cos(t) for t in Abs] >>> plt.plot(Abs,Ord) >>> plt.show()</pre>
---	---

REPRÉSENTER LES PREMIERS TERMES D'UNE SUITE

4. On souhaite tracer le nuage de points représentant les premiers termes de la suite définie par

$$u_0 = 0.5 \quad \text{et pour tout } n \in \mathbb{N}, \quad u_{n+1} = \sqrt{1 + u_n}.$$

- (a) Écrire une fonction `termes` qui prend en argument un entier naturel n et qui renvoie la liste des termes $u_0, \dots, u_n$.
- (b) Écrire une fonction `nuage` qui prend en argument un entier naturel n et qui affiche le nuage de points représentant les $n + 1$ premiers termes (de u_0 à u_n) de la suite $(u_n)_{n \in \mathbb{N}}$.
- (c) Afficher le graphique représentant les 10 premiers termes de la suite : qu'observe-t-on ?

.....

TRACER UN GRAPHIQUE : PREMIERS EXEMPLES FOURNIS

5. (a) Ouvrir le script `exemple.py` (à récupérer sur *ma page web*) et exécuter le programme.
 (b) Commenter la ligne `plt.axis([0,5,-4,3])` et la remplacer par l'une, puis l'autre des deux options commentées en-dessous.
 (c) Identifier le rôle de chacune des instructions du programme et compléter la fiche de cours.
6. (a) En s'inspirant du programme précédent, écrire un programme qui trace la courbe représentant la fonction exponentielle sur l'intervalle $[-2; 1]$ et la droite d'équation $y = -x$.
 (b) On constate que l'équation $e^x = -x$ admet une unique solution $\alpha \in [-2, 1]$. En utilisant la fonction `zoom` de la fenêtre graphique, donner une valeur approchée de α à 10^{-2} près.

.....

REPRÉSENTER UNE FAMILLE DE FONCTIONS

Un projectile est lancé avec une vitesse initiale $\vec{V}_0$ de norme V_0 (en $m.s^{-1}$), à une hauteur h (en m) et avec un angle α (en radians) par rapport à l'horizontale.

L'équation de la trajectoire du projectile est la suivante :

$$y = -\frac{g}{2V_0^2 \cos^2(\alpha)} x^2 + \tan(\alpha)x + h,$$

où x représente l'abscisse du projectile, y son ordonnée (sa hauteur) et g l'intensité de la pesanteur ($g = 9.8 m.s^{-2}$).

Dans toute cette partie, on prend $V_0 = 10 m.s^{-1}$ et $h = 2 m$.

On pourra donc définir des **variables globales** V_0 , h et g au début du script.

On cherche à déterminer où tombe le projectile en fonction de l'angle du lancer α .

7. Écrire une fonction `hauteur` qui prend en arguments deux réels x et α et qui renvoie la hauteur d'un projectile lancé avec un angle α lorsqu'il a pour abscisse x .
8. Écrire une fonction `trajectoire` qui prend en argument un réel α et qui génère, sans l'afficher, le graphe de la trajectoire d'un projectile lancé initialement avec un angle α par rapport à l'horizontale sur 15 mètres.
9. Écrire une fonction `plusieursTrajectoires` qui prend en arguments une liste de réels α_i et qui affiche sur le même graphique les courbes représentant la trajectoire d'un projectile lancé initialement avec un angle α_i par rapport à l'horizontale sur 15 mètres.
10. Utiliser cette fonction pour visualiser les trajectoires pour des angles de $\frac{\pi}{6}$, $\frac{\pi}{5}$, $\frac{\pi}{4}$, $\frac{\pi}{3}$ sur 15 mètres.
On modifiera à cette occasion la fonction `plusieursTrajectoires` afin de tracer en plus l'axe des abscisses pour représenter le sol et on ajustera la fenêtre graphique aux trajectoires tracées.
11. En utilisant la fonction `zoom` de la fenêtre graphique, donner des valeurs approchées à 10^{-2} près des quatre points de chute des projectiles.

$x_1 = \dots \quad x_2 = \dots \quad x_3 = \dots \quad x_4 = \dots$

12. Pour améliorer le tracé précédent, on veut générer seulement les points dont l'ordonnée est positive. Si f est une fonction et X une subdivision de l'intervalle sur lequel on souhaite faire le tracé alors on peut définir les deux nouvelles listes suivantes :

```
Xplus = [t for t in X if f(t)>=0]
Yplus = [f(t) for t in X if f(t)>=0]
```

A l'aide de ce modèle, modifier la fonction `trajectoire` afin que la trajectoire du projectile ne soit générée que jusqu'à l'impact.