Informatique - TP 3 **Boucle, fonctions et graphiques** 30 septembre - 4 octobre 2024

Corrigé

Code de partage avec Capytale : e3c0-1977484

Echauffement : énigme (et boucle)

Camille écrit un nombre à quatre chiffres qui commence par un 2. En déplaçant le 2 à la fin du nombre, elle remarque que le nombre obtenu vaut $\frac{4}{7}$ du nombre initial.
Que vaut ce nombre ?
Comme nous l'avons vu, on peut tâtonner et affiner la gamme de valeurs (voir que le nombre est entre 2 000 et 2 999, puis entre 2 100 et 2 199) voire résoudre le problème algébriquement.
En s'aidant de Python, on peut tester tous les nombres compris entre 2 000 et 2 999 et comparer leur valeur multipliée par $\frac{4}{7}$ avec le nombre composé des chiffres réordonnés. On ne trouver alors qu'une solution pour le nombre initial : 2 121

```
for a in range(0,9):
    for b in range (0,9):
        for c in range (0,9):
            d=2000+a*100+b*10+c;f=a*1000+b*100+c*10+2;
            if 4/7*d==f :
                print(d,f)
```

Fonctions et représentation graphique

Outils pour la représentation graphique

Outre *numpy* qui nous servira presque tout le temps, nous importerons *matplotlib.pyplot* pour la représentation graphique.

```
import matplotlib.pyplot as plt
```

Par ailleurs nous utiliserons la commande `np.linspace` qui permet de générer une liste de valeurs (d'abscisses par exemple).

Ensuite, nous représenterons un ensemble de points. Pour cela, on définira définira donc deux listes de nombres de même taille (une liste d'abscisses et une liste d'ordonnées) que l'on associera avec la commande `plt.plot`

Exercice 1

1. (a) Définir la fonction logarithme népérien.

   On utilise la syntaxe propre à la définition de fonction :

   ```
   def f(x):
       return np.log(x)
   ```

   (b) Tester la commande `np.linspace(1,10,100)` et interpréter les trois paramètres.

   `np.linspace` permet de créer une liste de valeurs espacées régulièrement (linéairement) en donnant les valeurs extrêmes et le nombre de valeurs que l'on souhaite (la commande les renvoie sous la forme d'un tableau). Par exemple `np.linspace(1,2,11)` donnera 11 valeurs réparties régulièrement entre 1 et 2, i.e. :
   `[1.0,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0]`

   (c) A l'aide du programme suivant, tracer la courbe de la fonction logarithme népérien sur l'intervalle $[1; 10]$ (où f est la fonction définie plus haut).

   ```
   x=np.linspace(1,10,100)
   y=f(x)
   plt.plot(x,y)
   plt.show()
   ```

   (d) Tracer la courbe de la fonction sur l'intervalle $[0,01; 4]$

   Pour la représentation, on a recours à la libraire *matplotlib.pyplot*, et on représente des points. Pour cela, on définit une liste d'abscisses (avec `np.linspace`) puis la liste des images.
   Par défaut, Python représente une ligne brisée (il relie les points). Si on souhaite voir les points, on peut rajouter le paramètre `'+'`

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

   def f(x):
       return np.log(x)

   x=np.linspace(0.01,4,1000)
   y=f(x)
   plt.plot(x,y) # ou plt.plot(x,y,'+') pour voir les points
   plt.show()
   ```

2. Tracer la courbe de la fonction exponentielle sur l'intervalle $[-3, 3]$

   Voici un programme simple, on utilise la fonction exp prédéfinie dans la bibliothèque `numpy` et Python. Il n'est donc pas nécessaire de définir une fonction ici.

   ```
   x = np.linspace(-3,3,100)
   y=np.exp(x)
   plt.plot(x,y)
   plt.show()
   ```

Exercice 2 - représentation graphique et variations

En représentant graphiquement la fonction $x \mapsto \dfrac{\ln x}{x}$, faire une conjecture sur ses variations.

Retrouver ce résultat par le calcul.

Comme la fonction est définie sur $]0, +\infty[$, on peut effectuer une représentation sur l'intervalle $[0,1; 100]$ en prenant par exemple `x=np.linspace(0.01,100,1000)` comme liste d'abscisses. Le problème est que l'échelle « écrase » les variations dans le cas présent et que le maximum n'est alors pas perceptible. Donc on « zoome » en représentant sur l'intervalle $[0,5; 10]$ et on observe que la fonction semble croissante puis décroissante.

En effet le calcul de la dérivée nous donne $f'(x) = \dfrac{\frac{1}{x} \times x - \ln(x) \times 1}{x^2} = \dfrac{1 - \ln(x)}{x^2}$ et le signe de $f'(x)$ est donc celui de $1 - \ln(x)$, c'est-à-dire positif sur $]0, e]$ et négatif sinon.

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

# une première tentative avec x = np.linspace(0.01,100,1000) écrase trop la
    courbe
# donc on "zoome" un peu
x = np.linspace(0.5,10,100)
y=np.log(x)/x
plt.plot(x,y)
plt.show()
```