Python : TP1

1er octobre 2019

Considérons une fonction $f(t)$ de type $f(t) = cos(\omega_1 t) + cos(\omega_2 t)$.

Le fichier "data.txt" contient une suite de nombres (sous forme de charactères) obtenu en échantillonant la fonction $f(t)$ pour t allant de 0 à 100 par pas de 1. Chaque valeur de $f(t)$ a été arrondi à 3 décimales près.

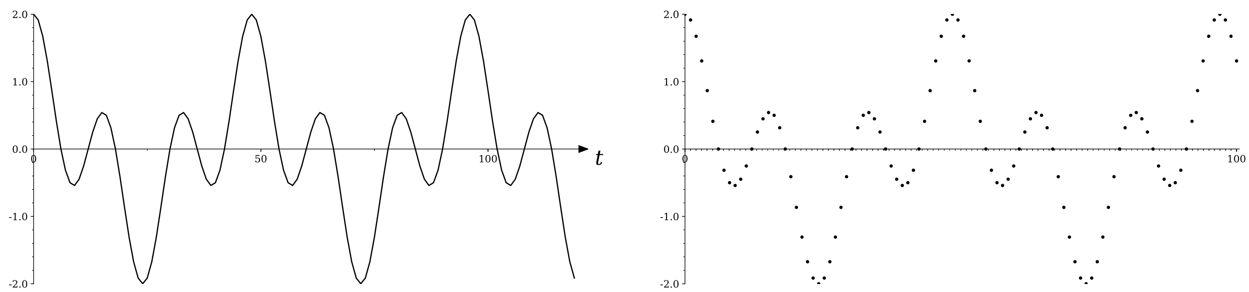

FIGURE 1 – La fonction $f(t)$ et l'échantillonage correspondant aux valeurs contenus dans le fichier "data.txt".

Exercice 1 Réalisez une fonction permettant d'ouvrir le fichier "data.txt" et de stocker les valeurs, convertis en nombres flottants, de chaque ligne dans une liste **L**. Cette méthode aura comme argument d'entrée le chemin du fichier "data.txt" et retournera la liste **L**.

Exercice 2 Réalisez une méthode permettant d'afficher les **n** premiers éléments de **L** à l'aide d'une boucle **for**. Cette méthode prendra en entrées la liste **L** et la valeur **n**. Affichez le résultat de la manière suivante :
"Les **n** premiers éléments de L sont : el_1 el_2 ... el_n"

Exercice 3 En reprenant les consignes de l'exercice précédent, écrivez une méthode permettant de stocker les **n** premiers éléments de **L** dans un fichier en respectant le même format que le fichier "data.txt". Le chemin du fichier contenant ces **n** valeurs sera passé en argument d'entrée.

Exercice 4 Réalisez une méthode permettant de retourner le maximum de **L** à l'aide d'une boucle.

Exercice 5 Implémentez une méthode permettant de rechercher les maximums locaux de **L**, de les afficher ainsi que de les écrire dans un fichier, dont le chemin sera spécifié en argument d'entrée, en respectant le format suivant :
(valeur1, indice1)
(valeur2, indice2)
(valeur3, indice3)
etc...

Exercice 6 Ecrivez une méthode retournant la norme Euclidienne de la fonction échantillonnée :

$$norm = \sqrt{\sum_i L[i]^2}$$

Le calcul de la racine se fait en important le module **math** et en utilisant sa fonction **sqrt**.

Exercice 7 Ecrivez une méthode retournant la moyenne et la variance de la fonction échantillonnée :

$$m = \frac{\sum_i L[i]}{N}$$

$$var = \frac{\sum_i (L[i] - m)^2}{N}$$

où N est la longeur de **L**.

Exercice 8 La fonction échantillonnée est-elle périodique ? Réalisez une méthode vérifiant cette hypothèse à l'aide d'une boucle sur des périodes entières et retournant la valeur de la période dans le cas échéant où -1 sinon.

Exercice 9 La fonction représentée par la f est de la forme

$$f(t) = cos(\omega_1 t) + cos(\omega_2 t)$$

avec $\omega_i = 2\pi/T_i$ et $T_i \in \{10, .., 100\}$. Implémenter une méthode permettant de trouver ω_1 et ω_2 et d'afficher leurs valeurs de la manière suivante : "La fonction est somme de deux fonctions cosinus de pulsation ω_1 et ω_2".

Aide : La partie principale de votre code où sera réalisé l'appel aux fonctions correspondantes aux différents excercices pourra ressembler au code présenté ci-dessous :

```
# Main
L = ex1_lire_data("./data.txt")

ex2_afficher_n_data(L, 10)

ex3_ecrire_n_data(L, 10, "./data_n.txt")

maxL = ex4_maximum(L)

ex5_maxi_locaux(L, "./maxi_locaux.txt")

norm_eucli = ex6_norm_eucli(L)
print("\nLa norme Euclidienne est {:.4f}".format(norm_eucli))

(moy, var) = ex7_moyenne_variance(L)
print("\nLa moyenne est {:.4f}".format(moy))
print("La variance est {:.4f}".format(var))

period = ex8_periodique(L)
if period > 0:
    print("\nLa fonction est {}-periodique".format(period))
else:
    print("\nLa fonction n'est pas periodique")

(pulsT1, pulsT2) = ex9_trouver_pulsT(L)
print("\nLa fonction est somme de deux fonctions cosinus
de pulsation {} et {}".format(pulsT1, pulsT2))
```