TP # 19

Illustrations sur les séries

Rappel : représentation graphique d'une suite

Pour afficher des représentation graphiques en Python, il faut effectuer l'importation :

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

Pour représenter les premières valeurs d'une suite $(u_n)_{n\in\mathbb{N}^*}$, on peut alors :

1. Créer le vecteur des abscisses : `X = [` $1,\dots,n$ `]`
2. Construire le vecteur des ordonnées : `Y = [` $u_1,\dots,u_n$ `]`
3. Relier les points avec `plt.plot(X,Y)` puis afficher la représentation avec `plt.show()`.

Séries de Riemann

Exercice 1

Pour un réel $a>0$, on considère la suite : $\forall n \geqslant 1,\ S_n = \sum_{k=1}^{n} \frac{1}{k^a}$.

1. Compléter la fonction `riemann` qui prend en entrée un réel a et un entier n et renvoie le vecteur `Y = [` $S_1, S_2, \dots, S_n$ `]`.

```
import numpy as np

def riemann(a,n) :

    Y = ..............................

    Y[0] = .........

    for k in range(............) :

        Y[k] = ........................

    return(Y)
```

2. (a) Représenter les 20 premiers termes de la suite $(S_n)_{n\in\mathbb{N}^*}$ pour $a=4$.

```
X = ..............................

Y = ..............................

plt.plot(X,Y)
plt.show()
```

(b) Superposer à ce graphe, en pointillés rouges, la droite d'équation $y = \frac{\pi^4}{90}$.

Noter qu'il suffit pour cela de relier les 2 points de coordonnées $(1, \frac{\pi^4}{90})$ et $(20, \frac{\pi^4}{90})$.

Juste avant le `plt.show()`, on rajoute ainsi :

```
c = np.pi**4/90

plt.plot( [..................] , [..................] , 'r--' )
```

En déduire, graphiquement, pour $a=4$, $\lim\limits_{n\to+\infty} S_n =$ .....

3. Afficher les 1000 premiers termes de la suite $(S_n)_{n\in\mathbb{N}^*}$ pour $a=1/2$.

```
X = ..............................

Y = ..............................

plt.plot(X,Y)
plt.show()
```

En déduire, graphiquement, pour $a=1/2$, $\lim\limits_{n\to+\infty} S_n =$ .....

4. (a) Afficher les 1000 premiers termes $(S_n)_{n\in\mathbb{N}^*}$ pour $a=1$ (série harmonique). Superposer, en pointillés rouges, les 1000 premiers termes de la suite $(\ln(n))_{n\in\mathbb{N}^*}$

```
X = ..............................

Y = ..............................

plt.plot(X,Y)

plt.plot( .......... , ........... , 'r--')

plt.show()
```

(b) A l'aide de Python, calculer une valeur approchée de $\gamma = \lim\limits_{n\to+\infty} (S_n - \ln(n))$.

On pourra par exemple calculer la valeur $S_{1000} - \ln(1000)$.

On trouve $\gamma \simeq$ .....

Adapter alors la portion de code précédente pour représenter, en rouge, les 1000 premiers termes de la suite $(\gamma + \ln(n))_{n\in\mathbb{N}^*}$.

```
gamma = ...................

plt.plot( .......... , ........................... , 'r--')
```

Constater graphiquement que $S_n \simeq \gamma + \ln(n)$.

Plus précisément, on a $S_n \underset{n\to+\infty}{=} \gamma + \ln(n) + o(1)$.

Séries exponentielles

Exercice 2

Pour un réel x, on considère la suite $\forall n \geqslant 0,\ S_n = \sum_{k=0}^{n} \frac{x^k}{k!}$.

1. Compléter la fonction `expo` qui prend en entrée un réel x et un entier n et renvoie le vecteur `Y = [` $S_0, S_1, \ldots, S_n$ `]` . (On n'utilisera pas la fonction factorielle...)

```
def expo(x,n) :

    Y = ..........................

    a = 1

    Y[0] = a

    for k in range(..................) :

        a = a * ..................

        Y[k] = Y[k-1] + a

    return(Y)
```

2. Afficher les 51 premiers termes de la suite $(S_n)_{n\geqslant 0}$ pour $x = 5$.
Superposer, en pointillés rouges, la droite d'équation $y = e^5$.
Constater la convergence $\lim\limits_{n\to+\infty} S_n = e^5$.

```
X = ..............................

Y = ..............................

plt.plot(X,Y)

plt.plot( .................. , ............................ , 'r--')

plt.show()
```

3. Reprendre en variant les valeurs de x : 10, -20, etc...