T.P. VII - Suites de réels

Code Capytale : `7db3-1087927`

I - Ce qu'il faut savoir

* **Définir** une suite par :
  * son terme général,
    la structure `for i in range(a, b):` permet de faire parcourir à `i` les valeurs de `a` à `b-1`.
  * récurrence à l'aide de la définition d'une **fonction**,
    La fonction peut être définie de manière externe avec le mot-clé `def` ou alors être définie à chaque itération.
  * récurrence pour des suites **imbriquées**,
  * récurrence linéaire avec des **matrices**.
* **Tracer** les termes successifs et **interpréter** un comportement asymptotique.
* Déterminer un **seuil**
  Les boucles conditionnelles `while` permettent d'interrompre le calcul dès qu'une condition est satisfaite.

II - Définition par le terme général

Exercice 1. (Terme général, Graphique) Soit X une variable aléatoire telle que

$$\forall\, k \geqslant 3,\ \mathbf{P}\left(X=k\right) = \frac{1}{2}\left[\left(\frac{4}{5}\right)^{k-2} - \left(\frac{2}{3}\right)^{k-2}\right].$$

1. Compléter le script suivant pour qu'il calcule et affiche les probabilités $\mathbf{P}\left(X=k\right)$ pour $k \in [\![3, 22]\!]$.

```
import numpy as np
import matplotlib.pyplot as plt
X = np.arange(0, 23)
U = np.zeros((23, 1))
for k in range(3, 23):
    U[k] = ...

plt.figure()
plt.plot(X, U, '+')
plt.show()
```

2. Compléter la suite d'instructions suivantes pour qu'elle affiche et renvoie les valeurs de la fonction de répartition de X aux points d'abscisses entières comprises entre 3 et 22.

```
F = np.cumsum(...)
plt.figure()
plt.plot(X, F)
plt.show()
```

3. Déterminer graphiquement un réel m tel que $\mathbf{P}\left([X \leqslant m]\right) = \mathbf{P}\left([X \geqslant m]\right)$.

Exercice 2. (Terme général, Seuil)
On considère la suite (c_n) définie par

$$\forall\, n \in \mathbb{N},\ c_n = 1 - \frac{2^n - 1}{3^{n-1}}.$$

Évaluer la suite d'instructions suivante puis interpréter le résultat.

```
n = 0
c = 1 - (2**n - 1) / 3**(n-1)
while c < 0.95:
    n = n + 1
    c = 1 - (2**n - 1)/3**(n-1)
print(n)
```

Exercice 3. (Terme général, Seuil)
Soit (a_n) et (b_n) les suites définies pour tout n entier naturel par

$$\begin{cases} a_n & = \frac{1}{3}\left(1 + \frac{2}{4^n}\right) \\ b_n & = \frac{1}{3}\left(1 - \frac{2}{4^n}\right) \end{cases}$$

Compléter le script ci-dessous afin qu'il calcule le plus petit entier naturel n tel que l'on ait à la fois : $a_n \leqslant 0.334$ et $b_n \geqslant 0.333$.

```
n = 0
a = 1
b = ...
while ...:
    n = ...
    a = 1/3 * (1 + 2/4**n)
    b = ...
print(...)
```

III - Suites récurrentes

Exercice 4. On définit la suite (u_n) par $u_0 = 1$ et pour tout n entier naturel, $u_{n+1} = \ln(1 + u_n^2)$.

1. Compléter la suite d'instructions suivante pour qu'elle calcule et affiche la valeur de u_5.

```
import numpy as np
n = ...
u = ...
for k in range(1, n+1):
    u = ...
print(u)
```

2. On admet que la suite (u_n) converge vers 0. Exécuter la suite d'instructions suivante et interpréter le résultat.

```
import numpy as np
n = 0
u = 1
while u >= 0.0001:
    u = np.log(1 + u**2)
    n = n + 1
print(n)
```

Le résultat affiché est 6. Quelle est la signification de ce résultat ?

Exercice 5. (Suite récurrente, Fonction externe, Graphique) On considère la fonction g définie sur $[1, 2]$ par $g(x) = \ln(x + 2)$ et la suite (u_n) définie par $u_0 = 1$ et $u_{n+1} = g(u_n)$ pour tout $n \in \mathbb{N}$.

1. Compléter la suite d'instructions suivante pour qu'elle mémorise les valeurs de u_0 à u_{20} et qu'elle les représente graphiquement.

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

U = np.ones((21, 1))
for i in range(1, 21):
    U[i] = ...

X = np.arange(0, 21, 1)
plt.figure()
plt.plot(X, U, '+')
plt.show()
```

2. Que conjecturer sur le sens de variation et sur la limite de la suite (u_n) ?

Exercice 6. Soit (u_n) la suite définie par $u_0 = 1$ et

$$\forall\, n \in \mathbb{N},\ u_{n+1} = \frac{u_n}{1 + u_n + u_n^2}.$$

On admettra que (u_n) est décroissante et converge vers 0.
Compléter la suite d'instructions suivante pour qu'elle affiche le plus petit entier naturel n non nul tel que $u_n \leqslant 1/1000$.

```
def f(x):
    return x/(1 + x + x**2)

u = ...
n = ...
while u ...:
    u = ...
    n = ...
print(...)
```

IV - Suites récurrentes dépendant du rang

Exercice 7. On considère la suite (I_n) définie par $I_0 = \frac{1}{2} - \frac{1}{2\,\mathrm{e}}$ et

$$\forall\, k \geqslant 0,\ I_{2k+1} = kI_{2k-1} - \frac{1}{2\,\mathrm{e}}.$$

Compléter la suite d'instructions suivante pour qu'elle calcule et affiche la valeur de I_{2n+1} pour $n = 20$.

```
import numpy as np
n = ...
I = 1/2 - 1/(2 * np.exp(1))
for k in range(1, n+1):
    I = ...
print(I)
```

Exercice 8. Soit (I_n) la suite définie par $I_0 = 1$ et

$$\forall\, n \geqslant 0,\ I_{n+1} = \frac{1}{\mathrm{e}} + (n+1)I_n.$$

Écrire une suite d'instructions qui calcule et affiche la valeur de I_{20}.

V - Suites imbriquées

Exercice 9. On considère les suites (u_n) et (v_n) définies par $u_0 = 1$, $v_0 = 1$ et

$$\forall\, n \in \mathbb{N}, \begin{cases} u_{n+1} & = \frac{u_n+v_n}{2} \\ v_{n+1} & = \frac{u_{n+1}+v_n}{2} \end{cases}$$

Compléter la suite d'instructions suivante qui permet de déterminer u_n et v_n pour $n = 50$.

```
n = ...
u = ...
v = ...
for k in range(1, n+1):
    u = ...
    v = ...
print("u50", u)
print("v50", v)
```

Exercice 10. Soit $(u_n)_{n\geqslant 0}$ et $(v_n)_{n\geqslant 0}$ deux suites définies par $u1 = 1$, $v_1 = 2$ et

$$\forall\, n \in \mathbb{N},\ u_{n+1} = \frac{u_n^2}{u_n+v_n} \text{ et } v_{n+1} = \frac{v_n^2}{u_n+v_n}.$$

1. Compléter la suite d'instructions suivantes afin qu'elle calcule et affiche les valeurs de u_{10} et v_{10}.

```
import numpy as np
n = ...
u = 1
v = 2
for k in range(2, n+1):
    a = u
    u = ...
    v = ...
print("u10", u)
print("v10", v)
```

2. On considère le programme précédent avec les instructions supplémentaires :

```
import numpy as np
import matplotlib.pyplot as plt
n = ...
u = 1
v = 2
s = np.ones((n+1, 1))
for k in range(2, n+1):
    a = u
    u = ...
    v = ...
    s[k] = u

X = np.arange(0, n+1)
Y = np.cumsum(s)
plt.figure()
plt.plot(X, Y)
plt.show()
```

a) Que contiennent les variables `s` et `y` à l'issue du programme ?

b) Quel résultat le graphique obtenu permet-il de conjecturer ?