T.P. VIII - Suites de réels

Code Capytale : `7db3-1087927`

I - Ce qu'il faut savoir

II - Définition par le terme général

Solution de l'exercice 1. [D'après Ecricome - 2016 - Exercice 3]

1.

```python
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] = 1/2 * ((4/5)**(k-2) - (2/3)**(k-2))

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

2.

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

3. Comme $\mathbf{P}(X \leqslant m) + \mathbf{P}(X > m) = 1$, on cherche un réel m pour lequel $\mathbf{P}(X \leqslant m) = 0{,}5$. On obtient graphiquement $m \simeq 7$. □

Solution de l'exercice 2. Le plus petit entier naturel n pour lequel $c_n \geqslant 0{,}95$ est égal à 11. □

Solution de l'exercice 3. [D'après Ecricome - 2022 - Exercice 1]

```python
n = 0
a = 1
b = -1/3
while a > 0.334 and b < 0.333:
    n = n + 1
    a = 1/3 * (1 + 2/4**n)
    b = 1/3 * (1 - 2/4**n)

print(n)
```

□

III - Suites récurrentes

Solution de l'exercice 4.

1.

```python
import numpy as np
n = 5
u = 1
for k in range(1, n+1):
    u = np.log(1 + u**2)

print(u)
```

2. Le plus petit entier naturel n pour lequel $u_n < 0.0001$ est égal à 6. □

Solution de l'exercice 5. [D'après ESCP BSB - 2016 - Exercice 2]

1.

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

U = np.ones((21, 1))
for i in range(1, 21):
    U[i] = np.log(U[i-1] + 2)

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

2. On peut conjecturer que la suite (u_n) est croissante. □

Solution de l'exercice 6. [D'après Ecricome - 2020 - Exercice 2]

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

u = 1
n = 0
while u > 1/1000:
    u = f(u)
    n = n + 1

print(n)
```

□

IV - Suites récurrentes dépendant du rang

Solution de l'exercice 7.

```
import numpy as np
n = 20
I = 1/2 - 1/(2 * np.exp(1))
for k in range(1, n+1):
    I = k * I - 1/(2 * np.exp(1))

print(I)
```

□

Solution de l'exercice 8. [D'après Ecricome - 2016 - Exercice 2]

```
import numpy as np

n = 10
I = 1

for k in range(n+1):
    I = 1/np.exp(1) + (k + 1) * I

print(I)
```

□

V - Suites imbriquées

Solution de l'exercice 9.

```
n = 50
u = 0
v = 1
for k in range(1, n+1):
    u = (u + v)/2
    v = (u + v)/2

print("u50", u)
print("v50", v)
```

□

Solution de l'exercice 10. [D'après BCE ESCP - 2017 - Exercice 2]

1.

```
import numpy as np
n = 10
```

```
u = 1
v = 2
for k in range(2, n+1):
    a = u
    u = u**2/(u + v)
    v = v**2/(u + v)

print("u10", u)
print("v10", v)
```

2.

```
import numpy as np
import matplotlib.pyplot as plt
n = 10
u = 1
v = 2
s = np.zeros((n+1, 1))
s[1] = u
for k in range(2, n+1):
    a = u
    u = u**2/(u + v)
    v = v**2/(u + v)
    s[k] = u

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

a) La variable `s` contient la liste des termes 0, u_1, $u_2, \dots, u_{10}$.
La variable `y` contient la liste des termes 0, u_1, $u_1+u_2, \dots, u_1+\cdots+u_{10}$.

3. On peut conjecturer que la série $\sum u_n$ converge vers 2,4. □