

I. Calculs de sommes et de produits

Exercice 1

Définir une fonction intitulée `def facto(n)` : qui calcule $n!$ à l'aide d'une boucle `for`

Exercice 2

On considère la suite $(u_n)_{n \geq 2}$ définie par: $\forall n \geq 2, u_n = \sum_{k=2}^n \ln \left(1 - \frac{1}{k}\right)$

1. Ecrire une fonction intitulée `def u(n)` : qui calcule u_n (on suppose que n est supérieur ou égal à 2 !)
2. Calculer "à la main" u_n en fonction de n .
3. Vérifier la cohérence des résultats si $n = 5$

Exercice 3

Ecrire une fonction `def S(n)` : qui calcule

$$S_n = \sum_{k=1}^n \left[\sum_{j=k}^n \frac{k}{j+1} \right]$$

Calculer à la main S_n en fonction de n puis vérifier la cohérence des résultats si $n = 5$.

II. Utilisation de `np.prod` et `np.sum`, `np.cumprod`, `np.cumsum`

Exercice 4

En utilisant la commande `np.prod`, définir une fonction `def facto(n)` : qui calcule $n!$.

Exercice 5

On souhaite calculer la somme $S = \sum_{k=0}^{n-1} (2k+1)$. Ecrire une fonction qui calcule cette somme sans utiliser de boucle `for`. Pour cela, vous pouvez commencer par créer un tableau contenant les valeurs $[1, 3, 5, \dots, 2n+1]$.

Exercice 6

On considère la suite définie par :

$$\forall n \in \mathbb{N}, u_n = \frac{n!}{\sum_{k=0}^n k!}$$

1. Ecrire une fonction `def facto(n)` : qui renvoie la matrice ligne
$$[0! \ 1! \ 2! \ \dots \ n!]$$
2. En utilisant cette première fonction, écrire une fonction intitulée `def u(n)` : qui calcule u_n .
3. Tester avec différentes valeurs de n et conjecturer la limite de la suite (u_n) . Prouver alors ce résultat.

III. Suite récurrente linéaire d'ordre 2

On considère la suite récurrente $(v_n)_{n \in \mathbb{N}^*}$ définie par:

$$v_1 = 0, \quad v_2 = 9 \quad \text{et} \quad \forall n \in \mathbb{N}^* \quad v_{n+2} = 6v_{n+1} - 9v_n$$

Ecrire une fonction intitulée `def v(n)` : qui calcule v_n .

IV. Formule des rectangles

On note $\forall t \in [0, 1], f(t) = \frac{1}{1+t^3}$. On va s'intéresser à l'intégrale suivante $I = \int_0^1 f(t) dt$.

1. Rappeler la formule des rectangles (ou sommes de Riemann) pour une fonction définie sur $[a, b]$.
2. Dresser le tableau de variation de f sur $[0, 1]$.
3. Justifier que : $\forall n \in \mathbb{N}^*, \forall k \in [[0, n-1]], \frac{1}{n} f\left(\frac{k+1}{n}\right) \leq \int_{\frac{k}{n}}^{\frac{k+1}{n}} f(t) dt \leq \frac{1}{n} f\left(\frac{k}{n}\right)$
4. En déduire que $\forall n \in \mathbb{N}^*$,

$$\frac{1}{n} \sum_{k=1}^n f\left(\frac{k}{n}\right) \leq \int_0^1 f(t) dt \leq \frac{1}{n} \sum_{k=0}^{n-1} f\left(\frac{k}{n}\right)$$

5. Ecrire un programme donnant une valeur approchée de I à 10^{-4} près.

V. Une série alternée

Notons $(u_n)_{n \in \mathbb{N}^*}$ la suite définie par : $\forall n \in \mathbb{N}^* \quad u_n = \frac{(-1)^n \ln(n)}{n}$

et la fonction f définie par : $\forall x \in [e, +\infty[\quad f(x) = \frac{\ln(x)}{x}$.

Notons : $\forall n \in \mathbb{N}^*, S_n = \sum_{k=1}^n u_k, x_n = S_{2n}$ et $y_n = S_{2n+1}$

1. Montrer que f est décroissante sur $[e, +\infty[$.
2. Ecrire une fonction Python intitulée `def S(n)` : qui calcule la valeur de S_n .
3. (a) Montrer que les suites $(x_n)_{n \in \mathbb{N}^*}$ et $(y_n)_{n \in \mathbb{N}^*}$ sont adjacentes.
(b) En déduire que la série de terme général u_n est convergente et justifier que :

$$\forall n \in \mathbb{N}^*, S_{2n+1} \leq \sum_{k=1}^{+\infty} \frac{(-1)^k \ln(k)}{k} \leq S_{2n}$$

4. Déterminer, en utilisant Python, une valeur approchée de $\sum_{k=1}^{+\infty} \frac{(-1)^k \ln(k)}{k}$ à 10^{-3} près.