

Code de partage avec Cappytale : 0440-2108983

Nous verrons aujourd'hui quelques manipulations habituelles avec Python pour l'étude des suites.

1 Calculer des termes d'une suite

1.1 Avec une formule explicite

Calculer les 100 premiers termes de la suite $(u_n)_{n \in \mathbb{N}}$ définie par : $u_n = \frac{1}{1+n}$

1.2 Avec une formule récursive

Calculer les 100 premiers termes de la suite $(u_n)_{n \in \mathbb{N}}$ définie par :

$$u_0 = 1000 \text{ puis } \forall n \in \mathbb{N}, u_{n+1} = 1 + \frac{1}{1+u_n}$$

Emettre une conjecture sur sa limite

2 Un nouvel outil, la boucle *while*

La boucle *while*

Comme la boucle *for*, la boucle *while* ou « tant que », permet d'exécuter plusieurs fois une instruction, mais dans ce cas le nombre d'itération dépend d'une condition et n'est pas prédéfini.

L'algorithme de cette boucle s'écrit donc :

Tant que "condition"
→ réaliser "instruction"

Ce type de boucle peut être utilisé pour étudier l'atteinte ou le dépassement d'un seuil, ce qui correspond bien à des situations modélisées par des suites réelles. Voyons-le sur un exemple.

Avec la suite précédente, trouver le premier terme dont la valeur a un écart avec la limite inférieur à 10^{-5}

3 Représentation graphique

Comme nous l'avons vu en cours de maths, une suite peut-être vue comme une fonction, on peut donc adapter nos outils de représentation graphique de fonction avec Python au cas d'une suite.

Représenter graphiquement les 100 premiers termes de la suite $(u_n)_{n \in \mathbb{N}^*}$ définie par :

$$u_n = \left(1 + \frac{1}{n}\right)^n$$

Emettre une conjecture sur sa limite

Nota bene : nous avons représenté ici une suite explicite, pour laquelle l'analogie avec la fonction est simple. Pour une suite récursive, il y a une petite difficulté que nous verrons prochainement.