

Code de partage avec Capytale : acc7-2261972

Corrigé

Echauffement - retour sur une suite réursive

Soit $(u_n)_{n \in \mathbb{N}}$ la suite définie par $u_0 = 2$ et $\forall n \in \mathbb{N}, u_{n+1} = f(u_n)$
où f est la fonction définie par $f(x) = 1 + \ln(x)$

1. écrire un programme qui calcule et représente les 100 premiers termes de la suite. En déduire que la suite $(u_n)_{n \in \mathbb{N}}$ est convergente, et donner sa limite ℓ

On calcule les termes de la suite de manière réursive et on les inclut progressivement dans une liste qu'on représente ensuite.

On crée au préalable une liste qui contient le premier terme $u_0 = 2$

```
import numpy as np
L=[2] #la liste ne contient que le premier terme initialement
u=2 # on définit le premier terme
for n in range(1,100):
    u=1+np.log(u) #on calcule le nouveau terme en fonction du précédent
    L.append(u) # on le rajoute dans la liste

plt.plot(L, '+') # par défaut, Python prend les abscisses 0, 1, ..., 99, ce
    qui nous convient ici
plt.show()
```

Il semble que la suite $(u_n)_{n \in \mathbb{N}}$ tende vers 1, ce qui est le cas

2. Déterminer le rang du premier terme de la suite tel que $|u_n - \ell| \leq 10^{-3}$
On pourra utiliser que la suite est minorée par 1

On utilise ici une boucle *while* et on calcule les termes (de manière itérative, comme plus haut) tant que la précision souhaitée n'est pas atteinte, i.e. tant que $|u_n - 1| > 10^{-3}$ que l'on peut écrire $u_n - 1 > 10^{-3}$ car la suite est minorée par 1

On incrémente également la valeur de n à chaque passage dans la boucle et on demande d'afficher la valeur de n à l'issue de la boucle (qui donne le premier terme qui ne respecte pas la condition, ce que nous cherchons).

```
u=2
n=0 # on crée une variable pour le rang
while u-1>10**(-3):
    u=1+np.log(u)
    n=n+1 # on met à jour la valeur de n

print(n)
```

Listes

Exercice 1 - liste et tri

Exécuter les commandes suivantes, puis sans la fonction `min`, trouver le minimum de la liste `L`.

```
import random
L=[random.randint(1,100) for i in range(1,101)]

min=L[0] # au début le minimum est la première
        # valeur de la liste
for i in range(1,100): # puis on parcourt le
    # reste de la liste
    if L[i]<min : # si on trouve une valeur
        # strictement inférieure au minimum
        min=L[i] # dans la variable min, on
        # remplace par cette valeur
print(min)
```

Pour trouver le minimum, il est nécessaire de parcourir toute la liste. Au début, le minimum est la première valeur de la liste puis en examinant chaque valeur de la liste, si on en trouve une plus petite on change le contenu de la « case » (i.e. variable) `min` par la nouvelle valeur minimale trouvée.

Exercice 2 - recherche de valeur

Exécuter les commandes suivantes, puis écrire un programme qui donne le rang d'apparition du premier 6

```
import random
L=[random.randint(1,6) for i in range(1,101)]
```

On parcourt la liste, **tant que** un 6 n'apparaît pas, donc la boucle *while* est l'outil indiqué. On incrémente l'indice au sein de la boucle puis on affiche l'indice final à l'issue de la boucle.

```
n=0
while L[n]!=6:
    n=n+1
print(n)
```