Problème sur les limites de suites et les factorielles

Exercice 5A.1

Soit (u_n) la suite définie pour tout naturel *n* par $u_n = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + ... + \frac{1}{n!}$

On peut aussi utiliser l'écriture $u_n = \sum_{k=0}^{n} \frac{1}{k!}$.

1.
   a. Démontrer que (u_n) est strictement croissante.
   b. Démontrer par récurrence sur *n* que, pour tout naturel *n* non nul, on a : $\frac{1}{n!} \leq \frac{1}{2^{n-1}}$
   c. Montrer que, pour tout naturel *n*, on a : $u_n \leq 3 - 2 \times 0,5^n$.
   d. En déduire que la suite (u_n) est convergente.
2.
   On va conjecturer la valeur de la limite de la suite (u_n) grâce à un algorithme.
   a. Ecrire en PYTHON une fonction **factorielle(n)** qui retourne la valeur de *n*! pour tout naturel *n*. (vous utiliserez la fonction factorielle définie précédemment)
   b. Ecrire en PYTHON une fonction **u(n)** qui retourne la valeur de u_n pour tout naturel *n*.
   c. Quelles sont les valeurs de u(5), u(10), u(100) et u(1000) retournées par votre programme. Conjecturer la valeur de $\lim_{n \to +\infty} u_n$.
   d. On pose $L = \lim_{n \to +\infty} u_n$, et on admet que la valeur conjecturée à la question précédente est correcte.
   Ecrire en PYTHON une fonction **min(delta)** qui retourne le plus petit entier *n* à partir duquel $|u_n - L| < \text{delta}$, où delta est un réel strictement positif.
   e. Jean affirme qu'un tel entier *n* existe car (u_n) est strictement croissante. Jean a-t-il raison ?

CORRIGE – Notre Dame de La Merci – Montpellier – M. Quet

Exercice 5A.1

Soit (u_n) *la suite définie pour tout naturel n par* $u_n = \frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}+...+\frac{1}{n!}$

On peut aussi utiliser l'écriture $u_n = \sum_{k=0}^{n}\frac{1}{k!}$

1.

a. *Démontrer que* (u_n) *est strictement croissante.*

Soit *n* un entier naturel. On a :

$$u_{n+1}-u_n = \left(\frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}+...+\frac{1}{n!}+\frac{1}{(n+1)!}\right)-\left(\frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}+...+\frac{1}{n!}\right)=\frac{1}{(n+1)!}.$$

$u_{n+1}-u_n > 0$ pour tout entier naturel *n* donc la suite (u_n) est strictement croissante.

b. *Démontrer par récurrence sur n que, pour tout naturel n non nul, on a :* $\frac{1}{n!}\leq\frac{1}{2^{n-1}}$

Initialisation : $\frac{1}{1!}=1$ et $\frac{1}{2^{1-1}}=\frac{1}{2^0}=1$ donc : $\frac{1}{1!}\leq\frac{1}{2^{1-1}}$. L'initialisation est vérifiée.

Hérédité:

Supposons qu'il existe un entier n non nul tel que $\frac{1}{n!}\leq\frac{1}{2^{n-1}}$, cela implique-t-il $\frac{1}{(n+1)!}\leq\frac{1}{2^{(n+1)-1}}$?

Par hypothèse :

$$\frac{1}{n!}\leq\frac{1}{2^{n-1}} \Leftrightarrow \frac{1}{n!}\times\frac{1}{n+1}\leq\frac{1}{2^{n-1}}\times\frac{1}{n+1}$$

$$\Leftrightarrow \frac{1}{(n+1)!}\leq\frac{1}{2^{n-1}}\times\frac{1}{n+1}$$

Or $\quad n\geq 1 \Leftrightarrow n+1\geq 2 \Leftrightarrow \frac{1}{n+1}\leq\frac{1}{2}$

Donc : $\frac{1}{(n+1)!}\leq\frac{1}{2^{n-1}}\times\frac{1}{n+1}\leq\frac{1}{2^{n-1}}\times\frac{1}{2}$

Soit : $\frac{1}{(n+1)!}\leq\frac{1}{2^n}$ →l'hérédité est vérifiée.

Par récurrence, pour tout entier *n* non nul, $\frac{1}{n!}\leq\frac{1}{2^{n-1}}$.

c. *Montrer que, pour tout naturel n, on a :* $u_n \leq 3-2\times 0,5^n$.

D'après le résultat précédent, pour tout naturel *k* non nul, on a :

$$\frac{1}{k!}\leq\frac{1}{2^{k-1}}.$$

Donc
$$\frac{1}{1!}+\frac{1}{2!}+...+\frac{1}{n!}\leq\frac{1}{2^{1-1}}+\frac{1}{2^{2-1}}+...+\frac{1}{2^{n-1}}$$

$$\Leftrightarrow \frac{1}{1!}+\frac{1}{2!}+...+\frac{1}{n!}\leq\frac{1}{2^0}+\frac{1}{2^1}+...+\frac{1}{2^{n-1}}$$

On reconnait la somme des termes d'une suite géométrique de premier terme égal à 1 et de raison $q=\frac{1}{2}$, avec :

$$1+q+q^2+...+q^{n-1}=\frac{1-q^n}{1-q}.$$

Ainsi : $\dfrac{1}{1!}+\dfrac{1}{2!}+...+\dfrac{1}{n!}\leq\dfrac{1-\left(\dfrac{1}{2}\right)^n}{1-\dfrac{1}{2}}$

$\Leftrightarrow\ \dfrac{1}{1!}+\dfrac{1}{2!}+...+\dfrac{1}{n!}\leq 2\left[1-\left(\dfrac{1}{2}\right)^n\right]$

D'où : $(u_n)\leq 1+2\left[1-\left(\dfrac{1}{2}\right)^n\right]$

$u_n\leq 3-2\times\left(\dfrac{1}{2}\right)^n$

d. *En déduire que la suite (u_n) est convergente.*

Comme $(u_n)\leq 3-2\times\left(\dfrac{1}{2}\right)^n$, on en déduit immédiatement que : $u_n\leq 3$.

La suite (u_n) est strictement croissante et majorée, donc elle est **convergente**.

2. *On va conjecturer la valeur de la limite de la suite (u_n) grâce à un algorithme.*

a. *Ecrire en PYTHON une fonction **factorielle(n)** qui retourne la valeur de n! pour tout naturel n. (vous utiliserez la fonction factorielle définie précédemment)*

Voici une fonction **factorielle(n)** qui retourne la valeur de *n*! pour tout naturel *n*.

```
def factorielle(n):
    fact = 1
    for i in range (1,n+1):
        fact *= i
    return fact

valeur = int(input("Veuillez saisir un entier:"))
resultat = factorielle(valeur)
print(valeur , '! = ' , resultat)
```

Pour $n = 6$, on obtient : 6 ! = 720.

b. *Ecrire en PYTHON une fonction **u(n)** qui retourne la valeur de u_n pour tout naturel n.*

Voici une fonction **u(n)** qui retourne la valeur de u_n pour tout naturel *n*.

```
def factorielle(n):
    fact = 1
    for i in range (1,n+1):
        fact *= i
    return fact

def u(n):
    U = 0
    for k in range (n+1):
        U += 1/factorielle(k)
    return U

n=int(input("Veuillez saisir un rang:"))
resultat=u(n)

print('u(' , n ,') = ' , resultat)
```

On aurait aussi pu commencer le programme par **from math import factorial** et remplacer **factorielle(k)** par **factorial(k)** dans le programme, mais cela n'aurait pas respecté l'énoncé!

c. Le programme nous donne :

u(5) = 2.7166666666666663 et u(10) = 2.7182818011463845
u(100) = 2.7182818284590455 et u(1000) = 2.7182818284590455

On conjecture que $\lim\limits_{n\to+\infty} u_n = e$.

d. *On pose* $L = \lim\limits_{n\to+\infty} u_n$*, et on admet que la valeur conjecturée à la question précédente est correcte.*
Ecrire en PYTHON une fonction ***min(delta)*** *qui retourne le plus petit entier n à partir duquel* $|u_n - L| < \text{delta}$*, où delta est un réel strictement positif.*

```
from math import *

def factorielle(n):
   fact = 1
   for i in range (1,n+1):
      fact *= i
   return fact

def u(n):
   U = 0
   for k in range (n+1):
      U += 1/factorielle(k)
   return U

def min(delta):
   n = 0
   e = exp(1)
   while abs(u(n) - e) > delta :
      n += 1
   return n

ecart = eval(input("Veuillez saisir un écart:"))
resultat = min(ecart)
print('Pour un écart égal à' , ecart , ', le rang cherché est :' , resultat)
```

Pour que PYTHON comprenne ce que représente **exp(1)**, il suffit de commencer le programme par **from math import exp**
On obtient :

Pour un écart égal à 0.001 , le rang cherché est : 6
Pour un écart égal à 1e-06 , le rang cherché est : 9
Pour un écart égal à 1e-09 , le rang cherché est : 12

e. Jean affirme qu'un tel entier *n* existe car (u_n) est strictement croissante.

Jean a tort. Un tel entier *n* existe et $\lim\limits_{n\to+\infty} u_n = e$.

Le fait que (u_n) soit strictement croissante nous assure que le programme proposé renvoie bien le plus petit entier convenable.