Corrigé

Code de partage avec Capytale : 0058-1395988

Limites

Exercice 1

On considère la fonction f, de domaine de définition $\mathscr{D}_f$, définie par :

$$\forall x \in \mathscr{D}_f, \quad f(x) = \frac{1}{\ln(x)}$$

1. Déterminer $\mathscr{D}_f$

   Le ln nous limite à $\mathbb{R}_+^*$ et de plus il ne peut s'annuler car il est au dénominateur, ce qui exclut 1, donc $\mathscr{D}_f = \mathbb{R}_+^* \setminus \{1\}$

2. On considère g, définie seulement sur $]0,1[$, par : $g(x) = f(x)$
   Calculer la limite de g en 0 et la limite de g en 1

   $\lim\limits_{x\to 0} \ln x = -\infty$ et $\lim\limits_{x\to 1^-} \ln x = 0^-$ donc par opération (inverse) $\lim\limits_{x\to 0} g(x) = 0$ et $\lim\limits_{x\to 1} g(x) = -\infty$

3. Peut-on parler d'une limite de f en 1 ? Que vaut $\lim\limits_{x\to 1^+} f$? $\lim\limits_{x\to +\infty} f$?

   Non car $\lim\limits_{x\to 1^+} \ln x = 0^-$ donc par opération (inverse) $\lim\limits_{x\to 1^+} f(x) = +\infty \neq \lim\limits_{x\to 1^-} f(x) = -\infty$
   $\lim\limits_{x\to +\infty} \ln x = +\infty$ donc par opération (inverse) $\lim\limits_{x\to +\infty} f(x) = 0$

4. Grâce à Python, vérifier graphiquement les résultats trouvés (bien choisir les arguments dans la commande `linspace`).

   On définit d'abord la fonction, puis avec la commande linspace, on définit les abscisses souhaitées : valeurs extremes d'abord, puis le nombre de valeurs. Pour la question posée ici, la commande `x=np.linspace(0.01,0.99,100)` convient bien (100 points équirépartis sur l'intervalle $[0,01; 0,99]$.

```
def g(x) :
    return 1/np.log(x)
x=np.linspace(0.01,0.99,100)
y=f(x)
import matplotlib.pyplot as plt
plt.plot(x,f)
plt.show
\hline
```

   Et enfin la commande `plt.plot(x,f)` permet de représenter g pour ces abscisses.
   La représentation graphique est conforme à nos résultats.

Exercice 2

On considère la fonction f, définie sur $]0,+\infty[$ par :

$$\forall x > 0, \quad f(x) = \frac{x}{1 - e^{-x}}$$

1. Calculer la limite de f en $+\infty$

   $\lim\limits_{x\to 1^+} \ln x = 0^-$ donc par opération (inverse) $\lim\limits_{x\to 1^+} f(x) = +\infty$

2. Que peut-on dire de la limite de f en 0 ?

   Rien, on se retrouve avec une forme indéterminée : « $\frac{0}{0}$ » qui ne nous permet pas de conclure.

3. Grâce à Python, vérifier graphiquement le résultat trouvé au 1, et conjecturer la valeur de la limite de f en 0

On peut dans un premier tracer la courbe pour des grandes valeurs de x (`x=np.linspace(1,1000,1000)`) et on a l'impression d'avoir à faire à une droite qui tend bien vers $+\infty$ quand x tend vers $+\infty$
Mais cela ne permet pas d'y voir clair en 0 et on effectue donc un deuxième tracé pour zoomer.

```
import numpy as np
def f(x):
    return x/(1-np.exp(-x))
x=np.linspace(0.001,0.1,100)
y=f(x)
import matplotlib.pyplot as plt
plt.plot(x,y)
plt.show()
```

4. Montrer qu'il existe $x \in]0,2[$, tel que $f(x) = 2$

On extrapole un peu le théorème des valeurs intermédaires (en l'utilisant avec une limite) et on s'appuie sur l'étude graphique : $\lim_{x\to 0} f(x) = 0$

$f(2) = \dfrac{2}{1-e^{-2}} > 2$ car $0 < e^{-2} < 1$ donc $0 < 1-e^{-2} < 1$ et donc $\dfrac{1}{1-e^{-2}} > 1$ et $\dfrac{2}{1-e^{-2}} > 2$
f étant continue, on en déduit (TVI) qu'il existe $x \in]0,2]$, $f(x) = 2$ (et même $]0,2[$ car $f(2) \neq 2$)

5. En utilisant un algorithme dichotomique, trouver une valeur approchée de cette solution à 10^{-5} près. Combien de calculs par l'ordinateur a nécessité votre algorithme ?

On adapte l'algorithme classique à la situation ($f(x) = 2$ ici) et on peut ajouter un compteur (`n` dans le programme ci-dessous) pour compter le nombre de passages dans la boucle, et on trouve $n = 18$.

```
a = 0
b = 1
while b-a>10**(-5):
  c = (a + b) / 2
  if f(c)< 2 :
    a=c
  else :
    b=c
print(c, "est une valeur approchée à 0,00001 près")
```

Continuité

Exercice 4

On considère la fonction f, définie sur $\mathbb{R}$ par :

$$\forall x \in \mathbb{R}, \quad f(x) = \begin{cases} x+1 & \text{si} \quad x \leqslant 2 \\ x^2-1 & \text{si} \quad x > 2 \end{cases}$$

1. f est-elle continue en 2 ?

   Oui car $\lim\limits_{x\to 2^-} f(x) = \lim\limits_{x\to 2^+} f(x) = f(2) = 3$ donc f est continue en 2

```
def f(x):
    if x <= 2
     return x+1
    else :
     x**2-1
```

2. Définir la fonction f avec Python.

   Il faut traduire la condition, ce que l'on fait avec `if`.

3. Grâce à Python, vérifier graphiquement le résultat trouvé.

   Comme on s'intéresse à la continuité en 2, on peut représenter la fonction autour de ce point, par exemple en prenant l'intervalle $[0, 4]$. Il n'y a pas de valeur interdite, donc on peut utiliser `np.linspace(0,4,100)`. Mais on obtient un message d'erreur, Python n'arrive pas à calculer `y=f(x)` (où x contient 100 abscisses) à cause de la condition dans la définition de f. On contourne le problème en calculant les images une par une.

```
import numpy as np
import matplotlib.pyplot as plt
x=np.linspace(0,4,100)
y=f(x)
plt.plot(x,y)
plt.show()
# ça bugge, d'où l'alternative :
x=np.linspace(0,4,100)
y=[f(x[i]) for i in range(0,100)]
plt.plot(x,y)
plt.show()
```