Centrale MP - 967

(Je me borne à traiter les questions de calcul numérique.)

❧ La série de terme général $a_n = {}^1/_{n^2}$ est bien convergente !
L'énoncé suggère l'approximation

$$R_n = \sum_{k=n+1}^{+\infty} \frac{1}{k^2} \approx \sum_{k=n+1}^{n+100} \frac{1}{k^2}$$

ce qui nous donne le code suivant.

```
def a(n):
    return 1/n**2

def R(n):
    s = 0
    for k in range(n+1, n+101):
        s += a(k)
    return s
```

Il reste ensuite à calculer les sommes partielles des trois séries considérées. Si on tient absolument à éviter le copier/coller, il est assez facile de définir une fonction abstraite mais ce n'est pas une nécessité : l'important ici est de produire vite un code qui fonctionne, pas un code efficace ou élégant.

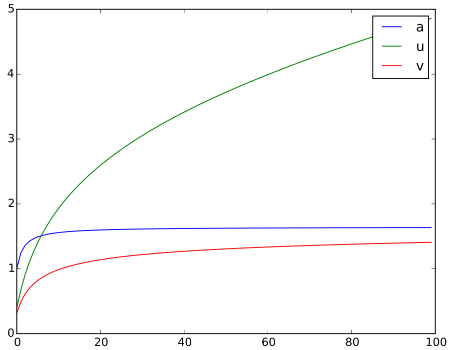
```
def somme_part(u, n):
    S = 0
    for i in range(1, n+1):
        S += u(i)
    return S
```

Ce qui suit est très laid (on fait trois fois la même chose, c'est un bon exemple de ce qu'il ne faudrait jamais faire), mais facile à coder et à vérifier.

```
import numpy as np

A = [ somme_part(a, n) for n in range(1, 101) ]

def u(n):
    return a(n+1)/R(n)
U = [ somme_part(u, n) for n in range(1, 101) ]

def v(n):
    return a(n+1)/np.sqrt(R(n))
V = [ somme_part(v, n) for n in range(1, 101) ]
```

Il reste à tracer les courbes. On rappelle qu'en fournissant une seule liste de nombres $(\ell_k)_{0 \leqslant k < n}$ à l'instruction `plt.plot`, on obtient une courbe interpolatrice des points (k, ℓ_k) pour $0 \leqslant k < n$.

```
import matplotlib.pyplot

plt.plot(A, label="a")
plt.plot(U, label="u")
plt.plot(V, label="v")
plt.legend()
```



On constate sur la figure que la série $\sum \frac{a_{n+1}}{R_n^\alpha}$ semble converger pour $\beta = 0$ et $\beta = 1/2$ alors qu'elle semble diverger pour $\alpha = 1$, ce qui est en accord avec la question **(d)** .