Programmation sous `python`—calcul scientifique

2025-2026

Une solution pour l'exercice n° 5.2) de la feuille n° 2

Écrire une fonction `eulerDecimalesExactes(d, k)` qui prend en argument deux entiers strictement positif d et $k \geq 2$ et qui renvoie le réel S et l'entier N où

- S est la somme d'Euler partielle $S = \sum_{n=1}^{N} \frac{1}{n^k}$
- N est le **plus petit entier** pour lequel la somme S a les d premières décimales égales à celles de la somme d'Euler (infinie)
$$E_k = \sum_{n=1}^{+\infty} \frac{1}{n^k}$$

En ne connaissant pas cette somme E_k, sauf pour quelques valeurs de k (mais pas pour $k = 3$ par exemple), il faudrait trouver une borne pour le reste

$$r_m = \sum_{n=1}^{+\infty} \frac{1}{n^k} - \sum_{n=1}^{m} \frac{1}{n^k} = \sum_{n=m+1}^{+\infty} \frac{1}{n^k}.$$

Solution.
En supposant qu'on a une borne supérieure pour r_m, l'idée est de calculer la somme partielle S_m telle que cette borne soit plus petite que 10^{-d}. Par la suite, on cherche N en ce référant à S_m, c'est-à-dire on cherche le premier N pour lequel les d premières décimales de S_N sont identiques à celles de S_m.

Le raisonnement mathématiques. Pour obtenir une borne supérieure pour r_m on majore la somme par une intégrale impropre. Dans la figure 1, en remarque que r_m représente la somme des aires des rectangles en dessous de la ligne polygonale rouge.

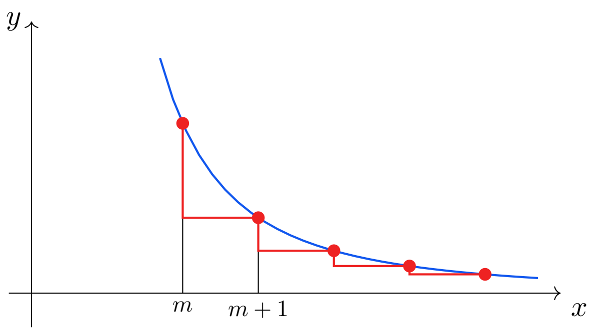

Figure 1: Le graphe de $x \mapsto \frac{1}{x^k}$ et la somme d'Euler E_k

Comme le graphe est celui de la fonction $x \mapsto 1/x^k$, on a donc

$$r_m = \sum_{n=m+1}^{+\infty} \frac{1}{n^k} \leq \int_m^{+\infty} \frac{dx}{x^k} = \left[-\frac{1}{(k-1)x^{k-1}} \right]_m^{+\infty} = \frac{1}{(k-1)m^{k-1}}.$$

Les fonctions `python`. Le rang m pour lequel le reste r_m est plus petit que $1/10^d$ sera donné par l'inégalité

$$\frac{1}{m^{k-1}} \leq \frac{1}{10^d},$$

ou encore

$$m^{k-1} \geq \frac{10^d}{k-1}.$$

Donc $m = \left\lfloor \sqrt[k-1]{\frac{10^d}{k-1}} \right\rfloor + 1$. Pour pouvoir controler les d décimales exactes de la somme partielle, on utlisera le rang

$$m = \left\lfloor \sqrt[k-1]{\frac{10^{d+2}}{k-1}} \right\rfloor .$$

Le code. Ci-dessous le code pour la fonction `eulerDecimalesExactes(d, k)` (avec les fonctions qu'elle appelle) ainsi qu'un test numérique.

```python
def eulerDecimalesExactes(d, k):
    """
    d est le nombre de decimales exactes recherche
    k est l'ordre de la somme d'Euler consideree

    Renvoie S la somme partielle avec le bon nombre de decimales exactes
    et le rang N correspondant.

    La recherche est basee sur la somme approximative obtenue a partir
    d'une borne du reste de la serie sur lequel on impose l'inegalite
    <= 10**(d +2).
    """
    bornePourLeReste = (10**(d + 2)/(k - 1))**(1/(k - 1))
    bornePourLeReste = int(bornePourLeReste)
    approximation = 0  # la somme de reference
    for n in range(1, bornePourLeReste):
        approximation += 1/n**k
    S = 0
    N = 0
    while not decimalesIdentiques(S, approximation, d):
        N = N + 1
        S += 1/N**k
    return S, N
```

```python
def eulerSomme(N, k):
    S = 0
    for n in range(1, N + 1):
        S += 1/n**k
    return S

def decimalesIdentiques(x1, x2, nb):
    """
    On verifie si x1 et x2 coincide jusqu'a la nb-eme decimale
    incluse.  Si oui, on renvoie True et si non False.
    """
    tx1 = int(10**d * x1)
    tx2 = int(10**d * x2)
    if tx1 == tx2:
        return True
    else:
        return False
```

```python
k = 3
for d in range(7):
    print(f"pour k = {k}, la somme avec {d} decimales exactes est",
    eulerDecimalesExactes(d, k))

print("pour comparaison, la somme pour N = 10000 est ",
      eulerSomme(10000, k))
```