Quelques gammes en Python

À l'image des gammes en musique, il est proposé ici de pianoter sur son clavier d'ordinateur pour s'entrainer à coder.

1. Do majeur : les sommes

Exemple. Ce code affiche les valeurs de $S = 6\sum_{k=1}^{100} \frac{1}{k}$ et de $T = \sum_{k=1}^{100} \frac{(-1)^{k+1}}{k}$

Attention aux indices ! `range(101)` va jusqu'à 100.

```python
S = 6 * sum([1/k for k in range(1, 101)])

T = sum([(-1)**(k+1)/k for k in range(1, 101)])

print(S, T)
```

Exercice. Classer par ordre croissant :

$$A = \sum_{k=3}^{92} \frac{1}{k} \qquad B = 3\sum_{k=1}^{70} \frac{1}{k^3} \qquad C = 5\sum_{k=1}^{50} \frac{(-1)^{k+1}}{k} \qquad D = \frac{1}{3}\sum_{k=2}^{11} \frac{k+(-1)^k}{k}$$

Exercice. Vers quelle constante bien connue semble converger pour $n \to +\infty$:

$$S_n = \sum_{k=0}^{n} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$$

2. Ré mineur : les suites définies par récurrence (ordre 1)

Exemple. Ce code donne $u_{100} \approx 2$ pour la suite définie par $u_0 = 3$ et $u_{n+1} = \dfrac{5u_n - 2}{u_n + 2}$

```python
u = 3

for _ in range(100):
    u = (5 * u - 2) / (u + 2)

print(u)
```

Exercice. Charade

Mon premier est la limite (conjecturée) de la suite définie par $u_0 = 0{,}5$; $u_{n+1} = u_n^2 - 2u_n + 2$

Mon second est la limite (conjecturée) de la suite définie par $v_0 = 18$; $v_{n+1} = \dfrac{5v_n - 49}{v_n - 9}$

Mon troisième est la valeur de w_5 pour la suite définie par $w_0 = 21$; $w_{n+1} = 3w_n + 5$

Et mon tout est la date d'une importante loi en France. Qui suis-je ?

3. Sol ♯ majeur : les suites définies explicitement

Exemple. Le code suivant affiche u_{10} et u_{50} pour $u_n = \dfrac{n+\sin(2n)}{n-\ln n}$ et v_{15} et v_{100} pour $v_n = \dfrac{n!}{\sqrt[3]{n}}$

```
from math import *

def u(n):
   return (n+sin(2*n)) / (n-log(n))

def v(n):
   return factorial(n) / n**(1/3)

print(u(10), u(50))

print(v(15), v(100))
```

Exercice. Un petit jeu d'appariemments ? C'est parti !

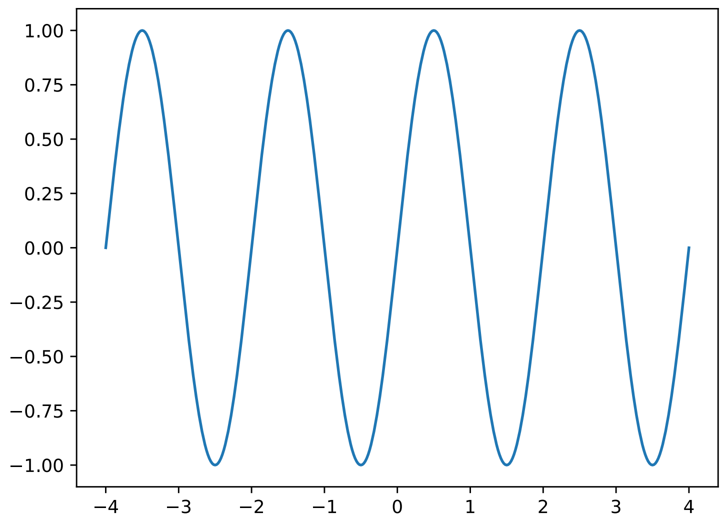
u_{10} pour $u_n = n^2\,\mathrm{e}^{-\sqrt{n}}$ •	• 3,129
v_{10} pour $v_n = \dfrac{(2n)!}{3^n\,(n!)^2}$ •	• 3,732
w_{10} pour $w_n = \left(1-\sin\left(\frac{\pi n}{3}\right)\right)(n^2-9n-8)$ •	• 4,233

4. Si ♭ mineur : les graphiques

Le code suivant trace la fonction $x \longmapsto \sin(\pi x)$ sur l'intervalle $[-4\,;4]$ avec 401 points, donc avec un pas de $\frac{4-(-4)}{400} = 0{,}02$. Attention aux majuscules/minuscules : tableau `X` et flottant `x`.

`plt.show()` montre ce graphique à l'écran. Il est ensuite possible de l'enregistrer en cliquant sur l'icône . Différents formats d'images sont possibles : `.svg` pour une image vectorielle (courbe sans pixels), `.png` et `.jpg` pour des images matricielles (avec pixels).

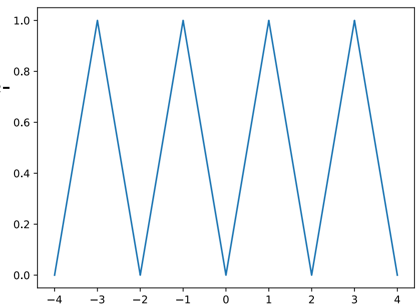
```
from math import *
import numpy as np
import matplotlib.pyplot as plt

X = np.linspace(-4, 4, 401)
Y = [sin(pi*x) for x in X]

plt.plot(X, Y)
plt.show()
```



Exercice. Tracer, sur 400 points, pour $x \in [-4\,;4]$, les courbes représentatives de :

1. $f(x) = \ln(1 + x^2)$
2. $g(x) = \dfrac{1}{\sqrt{2\pi}}\,\mathrm{e}^{\frac{-x^2}{2}}$

Exercice. Trouver comment tracer une fonction ayant le graphe ci-contre (c'est possible en une seule ligne, sans développement en série entière ou série de Fourier)



Indice : Mélanger valeur absolue et calcul modulaire réel

Solutions

1. Do majeur : les sommes

gammes-sommes.py

```python
A = sum([1/k for k in range(3, 93)])
B = 3 * sum([1/k**3 for k in range(1, 71)])
C = 5 * sum([(-1)**(k+1)/k for k in range(1, 51)])
D = 1/3 * sum([(k+(-1)**k)/k for k in range(2, 12)])

print(A, B, C, D)
```

Exécution du code

```
3.6044291790549177 3.6058689289727854 3.4162358028795907 3.4211519961519956
```

On a donc $C < D < A < B$

gammes-sommes.py

```python
S = sum([1/16**k * (4/(8*k+1) - 2/(8*k+4) - 1/(8*k+5) - 1/(8*k+6))
    for k in range(0, 13)])

print(S)
```

Exécution du code

```
3.141592653589793
```

S_n semble converger vers π. C'est effectivement le cas. Il s'agit de la formule BBP pour π. Elle a été découverte par Simon Plouffe en 1995.

2. Ré mineur : les suites définies par récurrence (ordre 1)

gammes-rec1.py

```python
# jour
u = 0.5

for _ in range(10):
    u = u**2 - 2*u + 2

print("u10 =", u)

# mois
v = 18

for _ in range(1000):
    v = (5*v - 49) / (v - 9)

print("v1000 =", v)

# année
w = 21

for _ in range(4):
    w = 3*w + 5

print("w4 =", w)
```

Exécution du code

```
u10 = 1.0
v1000 = 6.9979996362975045
w4 = 1901
```

Je suis le **1er juillet 1901**. La loi votée ce jour-là est celle de la liberté d'association, encore en vigueur de nos jours. Environ 2 millions d'« associations loi de 1901 », sans but lucratif, sont référencées en 2026.

Côté mathématiques :

Pour $(u_n)_{n\in\mathbb{N}}$: on pose $f(x) = x^2 - 2x + 2$. La suite étant définie par $u_{n+1} = f(u_n)$, les points fixes vérifient $L = f(L)$ L'équation $L = L^2 - 2L + 2$ admet $L = 1$ et $L = 2$ comme solutions. On calcule la dérivée $f'(x) = 2x - 2$. En 1, $|f'(1)| = 0 < 1$, donc le point est attractif. En 2, $|f'(2)| = 2 > 1$, donc le point est répulsif. Pour tout $u_0 \in]0\,;2[$, la suite converge vers 1.

Pour $(v_n)_{n\in\mathbb{N}}$, la limite se trouve par étude des suites homographiques. L'équation du second degré $L = \frac{aL+b}{cL+d}$ a une racine double égale à 7 pour $(a,b,c,d) = (5,-49,1,-9)$, et la suite converge vers cette racine pour tout $u_0 \neq \frac{-d}{c}$

3. Sol# majeur : les suites définies explicitement

gammes-suites.py

```
from math import *

def u(n):
    return n**2 * e**(-sqrt(n))

def v(n):
    return factorial(2*n) / (3**n * factorial(n)**2)

def w(n):
    return (1 - sin(pi*n/3)) * (n**2 - 9*n - 8)

print(u(10), v(10), w(10))
```

Exécution du code

```
4.2329219623205 3.128859083134346 3.7320508075688776
```

4. Si ♭ mineur : les graphiques

```
Y = [log(1+x**2) for x in X]

Y = [e**(-x**2/2) / sqrt(2*pi) for x in X]

Y = [abs((x-1)%2-1) for x in X]
```