



Mathématiques et nombres

Quelques programmes et algorithmes reliés aux mathématiques et aux nombres.


-  [Théorie des nombres](#)
-  [Nombre_remarquable](#)
- ...

Calculs en précision arbitraire

Les calculs suivants renvoient des nombres avec des décimales bien particulières :

- $1/9^2 = 0.0123456790123456790123456790123456790123456790123457...$
- $1/99^2 = 0.0001020304050607080910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667686970717273747576777879808182838485868788899091929394959697990001020304050607080910111...$

Quelle est l'explication de ces particularités. Comment manipuler de tels nombres et les construire ?

L'idée est de s'intéresser au développement en  [série de Taylor](#) de $1/x^2$ autour de $a=1$, ou de manière équivalente à la série de Maclaurin de $1/(1-x)^2$

Taylor, pour $f(x) = 1/x^2$:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n = 1 - 2(x-1) + 3(x-1)^2 - 4(x-1)^3 + 5(x-1)^4 - \dots$$

Maclaurin :

$$1/(1-x)^2 \approx 1 + 2x + 3x^2 + 4x^3 + 5x^4 + 6x^5 + 7x^6 + 8x^7 + 9x^8 + \dots$$

Ce développement va introduire des décimales particulières si $x = 0.1$, 0.01 ou $0.001, \dots$

Programme

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
'Nombres magiques' 100/81 (et 10000/9801) dont les valeurs montrent des
```

décimales
consécutives

Librairie de multiprécision : <http://mpmath.org/doc/current/basics.html>

■■ ■■ ■■

```
from mpmath import mp # multiprecision library
n = 41
mp.dps = n
#mp.pretty = True
print('standard float value : ', 1/0.81)

print('Multiprécision :')
sumt = mp.mpf(0)
for i in range(40, 0, -1):
    t = mp.mpf(i) * (mp.mpf(1)/mp.mpf(10))**mp.mpf(i-1)

    sumt += t
    print(mp.nstr(t, n, strip_zeros=False, min_fixed=-n)[:n],
          mp.nstr(sumt, n, strip_zeros=False, min_fixed=-n)[:n])

print(100/mpf(9**2))

n = 51
mp.dps = n
#mp.pretty = True
print('standard float value : ', 1/0.9801)

print('Multiprécision :')
sumt = mp.mpf(0)
for i in range(25, 0, -1):
    t = mp.mpf(i) * (mp.mpf(1)/mp.mpf(100))**mp.mpf(i-1)

    sumt += t
    print(mp.nstr(t, n, strip_zeros=False, min_fixed=-n)[:n],
          mp.nstr(sumt, n, strip_zeros=False, min_fixed=-n)[:n])

print(10000/mpf(99**2))
```

Résultats

standard float value : 1.2345679012345678

Multiprécision :

```
0.0000000000000000000000000000000000000000000000040
0.0000000000000000000000000000000000000000000000040
0.00000000000000000000000000000000000000000000000390
```

[illegible]

[illegible]

Références

- <https://mrob.com/> → $1/99^{**2}$
 - <https://mrob.com/pub/math/numbers-14.html>
 - <https://mrob.com/pub/seq/digits.html>
- https://fr.wikipedia.org/wiki/S%C3%A9rie_de_Taylor
- <https://socratic.org/questions/what-is-the-taylor-series-expansion-of-f-x-1-x-2-at-a-1>
- Multiprécision en Python : <http://mpmath.org/>
 - [Documentation](#)

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

https://dvillers.umons.ac.be/wiki/teaching:progappchim:math_nombres

Last update: **2020/01/14 13:58**

