TP INFORMATIQUE

Représentation des nombres

Exercice 1 - Erreurs de calcul

1 **Problème de représentation**

En base 2, le nombre décimal 0.1 a une écriture périodique après la virgule. Il est représenté en machine de manière arrondie. En effet on a

$$0.1 = \underline{0.00011001100110011\ldots}_2$$

ce qui se démontre à l'aide de $0.1 = \frac{1}{2^4}\mathrm{S} + \frac{1}{2^5}\mathrm{S}$ où $\mathrm{S} = \lim_{\mathrm{N}\to+\infty} \sum_{n=0}^{\mathrm{N}} \frac{1}{2^{4n}}$. (à chercher en fin de tp si vous avez le temps.)

Calculer $0.1 + 0.1 + 0.1 - 0.3$ et $0.1 + 0.2$. Commentaires ?

2 **Ordre des opérations**

1. Calculer $1 + 2^{-54} + (-1)$ et $1 + (-1) + 2^{-54}$. Commentaires ?
2. On dispose du programme ci contre. Que fait-il ? Compléter sa documentation et sa signature.
   Tester pour $n = 10^4$ et $n = 10^5$.
   Mathématiquement, quand n tend vers $+\infty$, le calcul précédent doit s'approcher de $\frac{\pi^4}{90}$.
   Est-ce le cas ?

```
def somme1(n):
    s = 0
    for i in range(1, n+1):
        s = s + 1/i**4
    return s
```

3. Écrire une fonction `somme2` de paramètre n, effectuant le même calcul que `somme1` mais à l'envers, c'est à dire en commençant par les grandes valeurs de i. Refaire les tests précédents. Commentaires ?

Exercice 2 : intégrons, dérivons

Intégration

1 on peut montrer que $\ln 2 = \int_1^2 \frac{1}{x}\,\mathrm{d}x = \lim_{n\to\infty} \left(\sum_{k=1}^{n} \frac{1}{(2\,k-1)\,2\,k} \right)$.

On rappelle que $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots$. Vérifier le calcul proposé puis codez le. À partir de quelle valeur de n obtient-on la même valeur approchée que celle obtenue en utilisant la fonction `log` de la librairie `math` ? Que remarque-t-on à l'exécution ?

2 L'intégration d'une fonction sur un intervalle par la méthode des rectangles consiste à écrire

$$\mathrm{I} = \int_a^b f(x)\,\mathrm{d}x \simeq \sum_{0}^{\mathrm{N}-1} f(a + i * p) * p \qquad \text{avec} \qquad p = (b-a)/\mathrm{N}$$

2.a Ecrire une fonction `rectangle(f : func, a : float, b : float , N : int) -> float` qui retourne la valeur approchée par cette méthode de I.

2.b Calculer (à la main...) $\mathrm{I} = \int_1^2 x\,\mathrm{d}x$, puis calculer I à l'aide de la fonction `rectangle`. Pour quelle valeur de N obtient-on le même résultat ? Avec quel temps de calcul ?

Dérivation

2 Ecrire une fonction `f(x : float) -> float` qui pour x donné en argument retourne $f(x) = x^2$. Écrire de même une fonction `Df(x : float) -> float` qui pour x donné en argument retourne la valeur de la fonction dérivée (que vous savez calculer) de f calculée en x. La tester pour $x = 2$.

3 Pour calculer la dérivée, écrire une fonction `Df2(x : float, eps : float) -> float` qui retourne la valeur approchée calculée par le taux d'accroissement

$$\frac{f(x+\varepsilon)-f(x)}{\varepsilon}$$

La tester pour $x=2$ avec $\varepsilon=2^{-n}$ en partant de $n=1$ puis en l'augmentant. Conclure.

> Vous pouvez aussi tester avec $\varepsilon=10^{-n}$ et observer les différences de comportement.

Exercice 3 : temps de calcul

On considère le programme suivant :

```
from time import perf_counter
N = 7

N0 = 2.0
t1 = perf_counter()
for i in range(N):
    N0 = N0*2**i
t2 = perf_counter()
print(t2-t1)

N0 = 2
t1 = perf_counter()
for i in range(N):
    N0 = N0*2**i
t2 = perf_counter()
print(t2-t1)

N0 = 2**180
t1 = perf_counter()
for i in range(N):
    N0 = N0**2**i
t2 = perf_counter()
print(t2-t1)
```

> Conseil : n'essayez pas d'afficher `N0`...

A votre avis quel calcul sera le plus rapide ? Coder le programme et vérifier...

Exercice 4 : une fonction mystère

Exécuter et expliquer le comportement du programme ci-dessous. Attention à bien l'écrire avec des nombres flottants, surtout pas avec des entiers ...

```
def mystere():
    x = 1.0
    y = x+1.0
    i = 0
    while (y - x == 1.0):
        i = i+1
        x = 2 * x
        y = x+1.0
        print(i)
    return y-x
```

Exercice 5 : équation du second degré

1 Écrire une fonction `nbsol` qui prend en paramètres les nombres a, b et c et qui retourne le nombre de solutions de l'équation $ax^2+bx+c=0$.

2 Utiliser cette fonction pour obtenir le nombre de solutions de $x^2+1.4x+0.49=0$ et de $x^2+0.2x+0.01=0$. Est-ce correct ?

3 Comment s'en sortir pour les deux équations précédentes ?

Exercice 6 - codage, décodage

1.a Ecrire une fonction `codage_bin( n : int) -> list` d'argument un entier naturel qui retourne son codage binaire sous forme d'une liste, le premier élément étant la plus haute puissance de deux. On pourra utiliser la méthode `instert` : on ajoute l'élément `"a"` à la position d'indice 2 de la liste `L` en écrivant `L.insert(2,"a")`.

1.b Ecrire une fonction `codage_dix( L : list) -> int` d'argument une liste contenant le codage binaire d'un nombre et qui retourne l'entier en base 10 associé.

2 Écrire une fonction `codage( r : float , n : int) -> str` d'argument un nombre `r` entier relatif écrit en base 10, un entier `n` qui retourne la chaine de caractère représentant le codage de `r` sur `n` bits en compléments à deux. On s'assurera que `r` peut effectivement être codé sur `n` bits.