TP Info 12 – Représentation des nombres en machines (II)

D.Malka – MPSI 2015-2016 – Lycée Saint-Exupéry

I1 – Nombres à virgule flottante

1. Sur une machine à processeur 64 bits, déterminer le nombre représenté par :
   0 00100000011 1101001110010101100000000000000111100000000000000000
2. ⚠ **Pas si évident :** déterminer la représentation machine du nombre d'Avogadro.

I2 – Comparaison avec des flottants

On considère le programme Python suivant :

```
x=1.0
y=x+1.0

while y-x==1.0:
    x=x*2.0
    y=x+1.0
```

1. Si l'on calculait sur des nombres décimaux exacts, que se passerait-il lors de l'exécution de ce programme ?
2. Ecrire ce programme dans un éditeur Python et l'exécuter. Que se passe-t-il ?
3. Modifier le programme de façon à déterminer au bout de combien d'itérations du corps de boucle il s'arrête, ainsi que la valeur de x à la fin de cette exécution.
4. Interpréter l'arrêt du programme.

I3 – Résolution d'une équation du second degré

1. Que fait la fonction fig.1 ?

```
from math import *


def zero_poly_deg_2(a,b,c):

    delta=b**2-4*a*c

    if delta>0:
        x1=1/(2*a)*(-b-sqrt(delta))
        x2=1/(2*a)*(-b+sqrt(delta))
    elif delta<0:
        x1=1/(2*a)*(-b-sqrt(-delta)*1j)
        x2=1/(2*a)*(-b+sqrt(-delta)*1j)
    else:
        x1=x2=-b/(2*a)

    return x1,x2
```

Figure 1 – Algorithme 1

2. Pour $a = 1,2$, $b = 1,5.10^8$ et $c = 4.10^{-4}$, on obtient le résultat suivant `(-125000000.0, 0.0)`. Commenter, expliquer.

I4 – Divergence numérique

On considère la suite $I_n = \int_0^1 x^n e^x dx$. Il est clair que $I_0 = e - 1$ et qu'une intégration par partie donne la relation de récurrence $I_n = e - nI_{n-1}$.

1. Donner un ordre de grandeur de l'erreur d'arrondi sur e.
2. Le graphe fig.2 montre le calcul des 20 premiers termes à l'aide d'une formule explicite (points sur le graphe) et le calcul des 20 premiers termes

à l'aide de la relation de récurrence. Expliquer **précisément** la divergence de résultats.

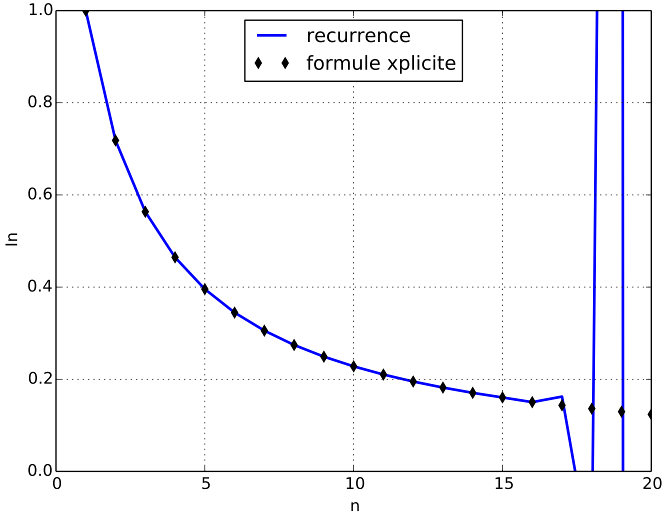

FIGURE 2 – Suite I_n

I5 – Erreur sur les multiplications

1. Montrer qu'à chaque opération arithmétique $*$ entre deux flottants A et B, l'erreur relative commise sur $A*B$ est majorée par la précision machine ε :

$$\frac{|fl(A*B) - A*B|}{|A*B|} \leq \varepsilon$$

2. Quelle est la valeur de cette erreur en base 10 pour des flottants codés sur 64 bits ?
3. Si on fait plusieurs opérations, ces erreurs s'accumulent. Quelle est l'ordre de grandeur de l'erreur relative d'un calcul qui est formé d'un million de multiplications successives (ce qui prend quelques millisecondes sur à un ordinateur usuel) ? Un raisonnement rigoureux est attendu.