Informatique en CPGE (2015-2016) Corrigé TP 9 : méthodes de dichotomie et de Newton

Exercice 1

```
from math import cos

def f(x):
    return cos(x)-x

def zeroDic(f,a,b,eps):
    cpt=0  # compteur pour le nombre d'itérations
    while b-a > eps:
        cpt+=1
        c = (a+b)/2
        if f(a) * f(c) > 0:
            a = c
        else:
            b = c
    return (a+b)/2,cpt

a,b=0,1  # on initialise a et b
eps=10**(-2)
print('solution de cos(x)-x=0 à',eps,"près",zeroDic(f,a,b,eps))
eps=10**(-4)
print('solution de cos(x)-x=0 à',eps,"près",zeroDic(f,a,b,eps))
eps=10**(-8)
print('solution de cos(x)-x=0 à',eps,"près",zeroDic(f,a,b,eps))
```

Si k est le nombre d'itérations, alors $\dfrac{b-a}{2^k} \leq \epsilon < \dfrac{b-a}{2^{k-1}}$, et on obtient :

$$\ln\left(\frac{b-a}{\epsilon}\right) \leq \ln(2^k) < \ln\left(2 \times \frac{b-a}{\epsilon}\right)$$

puis :

$$\ln\left(\frac{b-a}{\epsilon}\right) \leq k\ln 2 < \ln 2 + \ln\left(\frac{b-a}{\epsilon}\right)$$

Si $b-a=1$ et $\epsilon = 10^{-p}$, alors

$$\ln(10^p) \leq k\ln 2 < \ln 2 + \ln(10^p)$$

soit,

$$p\ln 10/\ln 2 \leq k < 1 + p\ln 10/\ln 2$$

Pour afficher la solution à 10^{-p} près pour p entier variant de 1 à 10, on ajoute une boucle qui appelle pour chaque valeur de p la fonction **zeroDic** et affiche la solution avec le nombre d'itérations ainsi que la partie entière (fonction **int**) de $1+p\ln 10/\ln 2$. Ne pas oublier d'importer la fonction **log** du module **math**.

```
from math import log

for p in range(1,11):
    print("p=",p,"sol",zeroDic(f,x,1,10**(-p)),int(p*log(10)/log(2))+1)
```

Exercice 2

On utilise le programme précédent avec quelques modifications. On importe les fonctions **cos**, **sin**, et **log** du module **math** et on définit la fonction dérivée **df(x)**.

```
from math import cos, sin, log

def df(x):
    return -sin(x)-1
```

Ensuite on remplace la fonction **zeroDic** par la fonction **newton**.

```
def newton(f,x,df,eps,N=100):
    cpt=0
    while abs(f(x))>eps and cpt<N: # test pas plus de N itérations
        x=x-f(x)/df(x)
        cpt+=1
    return x, cpt
```

On teste :

```
x=0
print("pour x = ",x,"\t",newton(f,x,df,10**(-3)))
x=-3
print("pour x=",x,"\t",newton(f,x,df,10**(-3)))
x=-5
print("pour x=",x,"\t",newton(f,x,df,10**(-3)))
x=3
print("pour x=",x,"\t",newton(f,x,df,10**(-3)))
x=10
print("pour x=",x,"\t",newton(f,x,df,10**(-3)))
```

On modifie la boucle pour afficher la solution à 10^{-n} près pour n entier variant de 1 à 25 et on affiche la valeur de $\text{int}(\ln n/\ln 2)+1$.

```
x=0
for n in range(1,26):
    print("n=",n,"\t",newton(f,x,df,10**(-n)),1+int(log(n)/log(2)))
```

Exercice 3

On définit les fonctions **f**, **df** et **newton**.

```
def f(z):
    return z**3-1

def df(z):
    return 3*z**2

def newton(f,z,df,eps,N=100):
    cpt=0
    while abs(f(z)) > eps and cpt<=N:
        if abs(z)<10**(-6):
            return z
        z=z-f(z)/df(z)
        cpt+=1
    return z
```

On affecte les valeurs des trois solutions à trois variables nommées **sol1**, **sol2** et **sol3**.

```
sol1=1+0j
sol2=complex(-0.5,0.5*3**(0.5))
sol3=complex(-0.5,-0.5*3**(0.5))
```

On ouvre quatre fichiers en écriture :

```
fic0=open('sol0.dat','w') # pas de convergence
fic1=open('sol1.dat','w') # converge vers sol1
fic2=open('sol2.dat','w') # converge vers sol2
fic3=open('sol3.dat','w') # converge vers sol3
```

On initialise x avec la valeur $-1,5$ et pour chaque valeur de x, on initialise y avec la valeur $-1,5$; le pas est de 0,004 pour x et y.

On affecte à z le complexe $x + \mathrm{i}y$ et on appelle la fonction **newton(f,z,df,10**(-2))** qui renvoie la solution. Ensuite il faut écrire dans le fichier correspondant à cette solution les valeurs de x et y.

```
x=-1.5
while x<=1.5:
    y=-1.5
    while y<=1.5:
        z=complex(x,y)
        sol=newton(f,z,df,10**(-2))
        if abs(sol-sol1)<10**(-2):
            fic1.write(str(x)+'\t'+str(y)+'\n')
        elif abs(sol-sol2)<10**(-2):
            fic2.write(str(x)+'\t'+str(y)+'\n')
        elif abs(sol-sol3)<10**(-2):
            fic3.write(str(x)+'\t'+str(y)+'\n')
        else:
            fic0.write(str(x)+'\t'+str(x)+'\n')
        y+=0.004
    x+=0.004
```

On ferme les quatre fichiers.

```
fic0.close()
fic1.close()
fic2.close()
fic3.close()
```

On trace avec Gnuplot la figure et on obtient une "fractale de Newton". Le choix du pas (0,004) est un compromis entre la vitesse et la qualité.