TD 7 - *Recherche de racines d'une fonction. Méthode de Newton Méthode de Babylone*

Informatique
MPSI - Lycée Thiers

2014/2015

Exercice 1

Racines d'un polynôme de degré 2

1. Ecrire une fonction `racine_trinome(a,b,c)` prenant en paramètres 3 nombres à virgule flottante `a`, `b`, `c` et qui à l'aide du discriminant retourne la ou les racines du polynôme de degré 2 : $P(X) = aX^2 + bX + c$.
2. Appliquer la recherche de racines au polynôme :

$$X^2 + X + \frac{1 - 2^{-54}}{4}$$

en appelant `racine_trinome(1,1,(1.0-2**-54)/4.0)`.

3. Définir la variable `e = -54`, et lancer l'instruction de comparaison avec 0 :

```
>>> e = - 54
>>> 1.0 + 2**e -1 == 0
```

Recommencer après l'affectation : `e=-52`. Discuter de la validité du résultat trouvé à la question précédente.

Exercice 1 : Réponse

1)

```
def racine_trinome(a,b,c):
    assert a != 0
    Delta = b**2 - 4*a*c
    if Delta != 0 :
        print("Deux racines")
        return (-b-Delta**0.5) / (2*a), (-b+Delta**0.5) / (2*a)
    else:
        print("Une seule racine")
        return -b/(2*a)
```

2)

```
>>> e = -54
>>> a, b, c = 1, 1, (1.0-2**e)/4.0
>>> print(racine_trinome(a,b,c))
Une seule racine
-0.5
```

```
>>> e = -54
>>> print(1.0 + 2**e - 1 == 0)
True
>>> e = -52
>>> print(1.0 + 2**e - 1 == 0)
False
```

Problème dans la comparaison d'un nombre à virgule flottante très petit avec 0.

Exercice 2

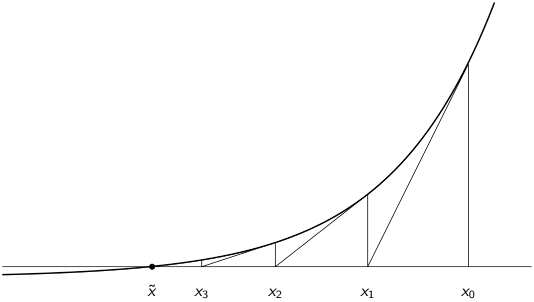
La méthode de Newton permet de déterminer une valeur approchée d'une racine :

Théorème

Si $f : I \longrightarrow \mathbb{R}$ est de classe C^1 (dérivable à dérivée continue) et a une racine r isolée sur I, si x_0 est suffisamment proche de r et si $f'(r) \neq 0$, alors la suite : (x_n) définie par $\forall\, n \in \mathbb{N},\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$ converge vers la racine r.

1. En quel point la tangente à la courbe représentative en x_n intersecte-t-elle la droite des abscisses ? En déduire une interprétation graphique de la suite (x_n).
2. Ecrire une fonction `newton(f,df,x,n)` qui retourne la valeur approchée x_n de la racine de f. Retrouver le résultat de l'Exercice 2, en partant du point $x = 1$. Que se passe-t-il si l'on prend $x = 0$ ou 2 ?

Exercice 2



La méthode de descente de Newton pour la recherche du zéro d'une application $f : \mathbb{R} \longrightarrow \mathbb{R}$.

Exercice 2

```
def newton(f,g,x,n):
    for i in range(n):
        x=x-f(x)/g(x)
return x
```

```
>>> f=lambda x:x**3-3*x**2+1
>>> g=lambda x:3*x**2-6*x
>>> newton(f,g,1,5)
0.6527036446661393
```

Exercice 3

Exercice 1. Pour la résolution de $x^2 = a$, c'est à dire pour l'extraction de racines carrée la méthode de Newton s'appelle la méthode de Babylone ou la méthode de Héron d'Alexandrie. L'appliquer pour le calcul approché de $\sqrt{2}$ et comparer le résultat obtenu à la valeur réelle.

Exercice 3

```
def heron(a,n):
    x=a
    for i in range(n):
        x=0.5*(x+a/x)
    return x
```

```
>>> heron(2,5)
1.414213562373095
>>> 2**0.5
1.4142135623730951
```

Après 5 itérations tous les 16 chiffres significatifs sont exacts ! !
La convergence est particulièrement rapide, et donc la méthode particulièrement efficace pour l'extraction de racines carrées.