

12. Recherche du zéro d'une fonction

Nous utiliserons dans ce TP le langage de programmation *Python* au moyen de l'environnement de développement *IDLE*.

Exemple 1. Fonction *lambda* en Python.

```
>>> f=lambda x:x**3-x-1
>>> f(3)
23
```

Exercice 1. Créer puis tester une fonction *evaluation* des variables *f* et *a* qui retourne l'image de *a* par la fonction *f*.

(par exemple *evaluation(lambda x : x ** 3 - x - 1, 3)* vaut 23)

Exercice 2. On rappelle l'algorithme d'encadrement du zéro d'une fonction continue strictement monotone par dichotomie :

Fonction: dichotomie(*f*, *a*, *b*, *e*)
Action: Encadrement d'amplitude *e* du zéro d'une fonction *f* sur l'intervalle [*a*; *b*] par dichotomie
Début
 $u \leftarrow a$
 $v \leftarrow b$
TantQue $v - u > e$ **faire**
 $w \leftarrow \frac{u+v}{2}$
 Si $f(u)f(w) > 0$ **alors**
 $u \leftarrow w$
 sinon
 $v \leftarrow w$
 FinSi
FinTantQue
Renvoyer u, v
Fin

Créer puis tester une fonction *dichotomie* des variables *f*, *a*, *b* et *e* qui retourne un encadrement d'amplitude *e* du zéro de la fonction *f* sur l'intervalle [*a*; *b*].

(par exemple *dichotomie(lambda x : x ** 3 - x - 1, 1, 2, 10 ** (-5))* vaut (1.3247146606445312, 1.3247222900390625))

Exercice 3. Créer puis tester une fonction *newton* des variables *f*, *Df*, *a* et *n* qui retourne le terme d'indice

n de la suite $(u_k)_{k \in \mathbb{N}}$ définie par
$$\begin{cases} u_0 = a \\ u_{k+1} = u_k - \frac{f(u_k)}{Df(u_k)} \end{cases} .$$

(par exemple *newton(lambda x : x ** 3 - x - 1, lambda x : 3 * x ** 2 - 1, 1, 7)* vaut 1.324717957244746)

Exercice 4. Créer puis tester une fonction *secante* des variables *f*, *a*, *b* et *n* qui retourne le terme d'indice

n de la suite $(u_k)_{k \in \mathbb{N}}$ définie par
$$\begin{cases} u_0 = a \\ u_1 = b \\ u_{k+2} = u_{k+1} - f(u_{k+1}) \frac{u_{k+1} - u_k}{f(u_{k+1}) - f(u_k)} \end{cases} .$$

(par exemple *secante(lambda x : x ** 3 - x - 1, 1, 2, 7)* vaut 1.3247179572446703)

Réponses

- 1)

```
def evaluation(f,a):  
    '''calcul de l'image de a par la fonction f'''  
    return(f(a))
```
- 2)

```
def dichotomie(f,a,b,e):  
    '''encadrement d'amplitude e du zéro de f sur [a;b] par dichotomie'''  
    u=a  
    v=b  
    while v-u>e :  
        w=(u+v)/2  
        if f(u)*f(w)>0:  
            u=w  
        else:  
            v=w  
    return(u,v)
```
- 3)

```
def newton(f,Df,a,n):  
    '''valeur approchée du zéro de f par la méthode de newton en partant de a avec n itérations'''  
    u=a  
    for k in range(0,n):  
        u=u-f(u)/Df(u)  
    return(u)
```
- 4)

```
def secante(f,a,b,n):  
    '''valeur approchée du zéro de f sur [a,b] par la méthode de la sécante avec n itérations'''  
    u=a  
    v=b  
    for k in range(0,n):  
        w=v-f(v)*(v-u)/(f(v)-f(u))  
        u=v  
        v=w  
    return(v)
```