Introduction à l'analyse numérique
Année : 2020-2021
Formation : L2 Mathématiques

TP2 : Recherche des zéros de fonctions

Soit f une fonction numérique dont on recherche un zéro ζ. Soit $(x_n)_{n\in\mathbb{N}}$ la suite des approximations de ζ obtenue à partir de termes initiaux en utilisant l'une des méthodes suivantes : méthode de dichotomie, de la sécante ou de Newton.

Exercice 1.

1. *Ecrivez une fonction* `dichoto(f, a, b, nbetap)` *qui renvoie la suite des* $nbetap+1$ *premières approximations du zéro de* f *localisé entre* a *et* b *calculé par une méthode de dichotomie.*
2. *Ecrivez une fonction* `secante(f, x0, x1, nbetap)` *qui renvoie la suite des* $nbetap+1$ *premières approximations du zéro de* f *calculé à partir de* x_0 *et* x_1 *par la méthode de la sécante.*
3. *Ecrivez une fonction* `newton(f, ff, x0, nbetap)` *qui renvoie la suite des* $nbetap+1$ *premières approximations du zéro de* f *calculé à l'aide de la dérivée* ff *de* f *à partir de* x_0 *par la méthode de Newton.*

Exercice 2 (Ordre de convergence)**.** *On définit les fonctions*

$$f_1(x) = 3\cos(x) - 2\ln(x+1) - 1,$$

$$f_2 = 2x-1, \quad f_3(x) = (2x-1)^3, f_4(x) = (2x-1)^5$$

1. *Après avoir calculé les dérivées à la main, testez les fonctions écrites lors de l'exercice précédent sur les fonctions* f_i, $1 \leq i \leq 4$ *ainsi définies.*
2. *Comparez sur un même graphique (n en fonction de k) le nombre n d'itérations nécessaires à chaque méthode pour obtenir le zéro des fonctions suivantes entre* $a=0$ *et* $b=1$ *avec une précision de* 10^{-k} *pour* k *entre 1 et 5.*
3. *Pour les trois dernières fonctions, représentez graphiquement les premiers termes de la suite* $\left(\frac{\ln(|\zeta-x_{n+1}|)}{\ln(|\zeta-x_n|)}\right)_{n\in\mathbb{N}}$ *où* ζ *représente la zéro de la fonction.*

Exercice 3 (Le module scipy)**.** *Certains fonctions sont déjà implémentées en Python et se trouvent dans le module optimize de la bibliothèque scipy. Il contient en particuliers les fonctions suivantes :*

- ***bisect*** *détermine la valeur approchée d'un zéro d'une fonction par la méthode de dichotomie*
- ***newton*** *qui en fonction des paramètres donnés calcule la valeur approchée du zéro d'une fonction en utilisant la méthode de la sécante ou de Newton*
- *Les fonctions* ***fsolve*** *et* ***root*** *permettent également de trouver des valeurs approchées des zéros de fonctions (avec des méthodes que nous n'avons pas vu en cours). Notez que la fonction* ***root*** *peut-être utilisé pour des fonctions de plusieurs variables. Pour plus d'informations concernant ce module, vous trouverez la documentation en ligne :* `https://docs.scipy.org/doc/scipy/reference/optimize.html`

Exemples d'utilisations du module optimize.

```
from scipy import optimize
```

```
def f(x):
    return x**2-2

a = 1
b = 2


# Dichotomie
x = optimize.bisect(f,a,b)
print(x)

# Secante
x = optimize.newton(f,a)
print(x)

# Newton
x = optimize.newton(f,a,fprime = lambda x: 2*x)
print(x)


def df(x):
    return 2*x

x = optimize.newton(f,a,df)
print(x)

# Autre fonctions

x=optimize.fsolve(f,a)
print(x)

x=optimize.root(f,a)
print(x.x)
```

Question : *Utilisez ce module pour calculer les zéros de fonctions de l'Exercice 2 et vérifiez vos résultats.*