

Recherche des zéros d'une fonction par dichotomie

Rappelons le programme officiel :

2. Équations algébriques	
Résolution d'une équation algébrique ou d'une équation transcendante : méthode dichotomique.	Déterminer, en s'appuyant sur une représentation graphique, un intervalle adapté à la recherche numérique d'une racine par une méthode dichotomique. Mettre en œuvre une méthode dichotomique afin de résoudre une équation avec une précision donnée. Utiliser la fonction bisect de la bibliothèque scipy.optimize (sa spécification étant fournie).

I. Méthode :

Principe : Soit f la fonction de la variable x dont on cherche le zéro x_0 .

- On donne la représentation graphique de f de sorte que x_0 appartienne à un intervalle $[a, b]$ tel que:
 - $f(a)$ et $f(b)$ sont de signes opposés ;
 - f est continue et strictement monotone sur $[a, b]$.
- A partir du couple de valeurs (a, b) , évaluer la fonction en $a + b$.
- Si $f(a+b)$ a le même signe que $f(a)$, alors f ne s'annule pas entre a et $a+b$, mais entre $a+b$ et b : On recommence en remplaçant a par $a + b$.
- Sinon : remplacer b par $a + b$.
- On recommence ainsi de suite en réduisant la taille de l'intervalle sur lequel f s'annule, et qui contient donc x_0 , jusqu'à ce que la différence entre les deux bornes soit inférieure à la précision voulue.

Proposition de script pour la fonction dichotomie (f, a, b, eps) qui prend comme arguments la fonction f , les bornes a et b de l'intervalle initial et l'erreur maximale eps (c'est-à-dire $|\text{largeur de l'intervalle final}| \leq \text{eps}$) - pensez à définir au préalable la fonction f dont on cherche le zéro.

```
def dichotomie(f, a, b, eps):
    if f(a)*f(b) > 0:
        return None
    while abs(b-a) > eps:
        c = (a+b) / 2
        if f(a)*f(c) > 0:
            a=c
        else:
            b=c
    return a, b
```

On demande simplement ensuite à ce que le résultat s'affiche (avec print).

II. Utilisation de la fonction bisect de la bibliothèque scipy.optimize

Il y a encore plus simple pour résoudre une équation algébrique par dichotomie, avec la fonction bisect de la bibliothèque scipy.optimize :

On importe au préalable ladite bibliothèque.

On définit la fonction f dont on cherche le zéro.

On demande la solution avec bisect en précisant l'intervalle [a ; b] :

```
import scipy.optimize as sc

def f(x) :
    return ...

solution = sc.bisect( f, a, b)
print(solution)
```

Approfondissement facultatif :

Il existe d'autres méthodes (méthode de Newton, utilisation de fsolve dans le module scipy.optimize...)

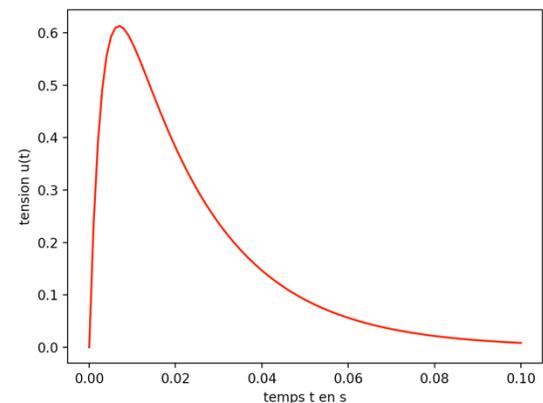
Si cela vous amuse de les regarder...

III. Exercice 4 du TD Elec Chap 5

Lors de l'exercice 4 du TD Elec Chapitre 5 , nous avons obtenu que la tension u suivait la loi :

$$u(t) = A(e^{r_2 t} - e^{r_1 t})$$

Avec $r_1 = -327 \text{ s}^{-1}$ et $r_2 = -48 \text{ s}^{-1}$.



On cherche à quelle date t_m la tension est maximale. Evidemment, cela revient à résoudre l'équation :

$$r_2 e^{r_2 t} - r_1 e^{r_1 t} = 0$$

- 1) Tracer la courbe $u(t)$.
- 2) Chercher « à la main » la date t_m .
- 3) Déterminer par dichotomie la date à laquelle l'équation ci-dessus s'annule.
- 4) Utiliser bisect pour retrouver la date t_m .