

DL #2: Racine d'une fonction mathématique

Le but de ce deuxième devoir libre (DL) est d'implémenter un algorithme de calcul approché de la racine d'une fonction mathématique f continue sur un intervalle $[a, b]$.

La fonction dichotomie que vous implémenterez aura la signature suivante¹ :

```
def dichotomie(f, a, b, epsilon) :  
    # [...]
```

Pour cela, proposez une condition suffisante à l'existence d'une telle racine² : votre algorithme ne cherchera de racine que dans le cas où cette condition est remplie. Dans le cas contraire, votre fonction retournera la valeur `None`. Une fois que l'on est sûr qu'une solution existe, nous allons utiliser la dichotomie pour s'en approcher au plus près. Le principe est le suivant : s'il existe une solution à l'équation $f(x) = 0$ sur l'intervalle $[a, b]$, c'est que, avec $m = \frac{a+b}{2}$, l'une (au moins) des deux assertions suivantes est vraie :

- f possède une racine sur $[a, m]$;
- f possède une racine sur $]m, b]$.

Il suffit alors de continuer la recherche dans le sous-intervalle dont on est sûr qu'il contient une solution et, par raffinements successifs, on aura une valeur approchée d'une racine de f , l'idée étant de s'arrêter lorsque l'on trouve un intervalle dont l'une des deux bornes x_0 vérifie $|f(x_0)| < \varepsilon$, avec ε choisi suffisamment petit.

1. On pourra rendre le paramètre `epsilon` facultatif et lui donner une valeur par défaut raisonnable. Remarquez qu'il est possible de passer une fonction (`f` ici) en paramètre.

2. On pourra s'inspirer du théorème des valeurs intermédiaires.