TP: résolution d'équation $f(x) = 0$
Informatique pour tous

On s'intéresse à une équation de la forme $f(x) = 0$ où f est une fonction définie sur un intervalle $[a, b]$ de $\mathbb{R}$. Une solution x de cette équation est appelée un zéro de f.

I Dichotomie

La **méthode par dichotomie** suppose que:

- f est continue
- les signes de $f(a)$ et $f(b)$ sont différents

D'après le théorème des valeurs intermédiaires, f s'annule entre a et b, c'est à dire que l'équation $f(x) = 0$ a une solution entre a et b.
On remplace alors soit a soit b par $\dfrac{a+b}{2}$ (le milieu de a et b) de façon à ce que les signes de $f(a)$ et $f(b)$ soient toujours différents (si $f(a)$ et $f(\dfrac{a+b}{2})$ sont de signes différents alors on remplace b par $\dfrac{a+b}{2}$, sinon on remplace a par $\dfrac{a+b}{2}$).
On continue d'appliquer cette méthode jusqu'à ce que la longueur de $[a,\ b]$ devienne suffisamment petite: on obtient alors un bon encadrement d'un zéro de f.

1. Quelle fonction f peut-on définir pour approximer $\sqrt{3}$ en utilisant cette méthode par dichotomie? Définir cette fonction en Python.

2. Écrire une fonction `dichotomie` telle que, si `f` est une fonction continue sur `[a, b]` et le signe de `f(a)` est différent de celui de `f(b)`, `dichotomie(f, a, b, epsilon)` applique la méthode de recherche par dichotomie et renvoie un encadrement d'une solution de $f(x) = 0$ par un intervalle de longueur au plus `epsilon`.
   On pourra compléter la fonction suivante:

```python
def dichotomie(f, a, b, epsilon):
    while ...:
        if ...:
            b = ...
        else:
            a = ...
    return [a, b]
```

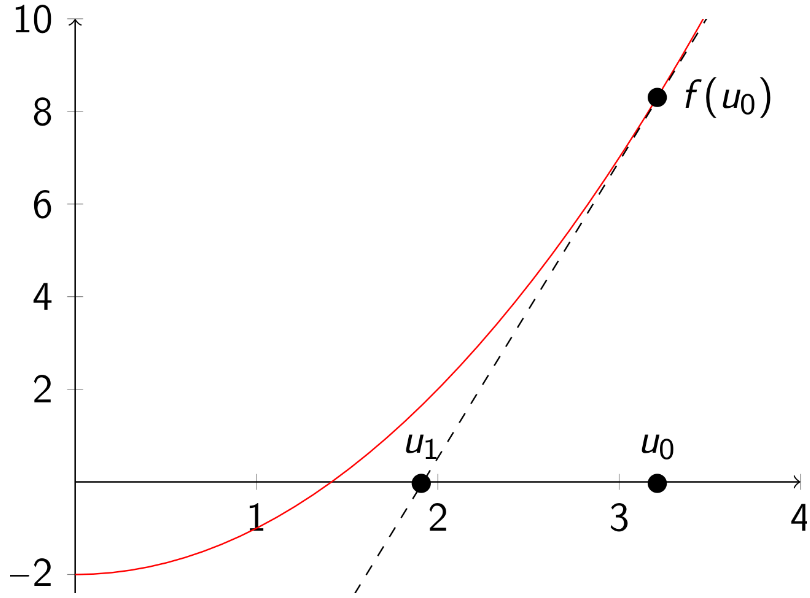
3. Utiliser `dichotomie` pour approximer $\sqrt{3}$ et π à 10^{-3} près (on pourra écrire `import numpy as np` pour utiliser `np.cos`).

4. Trouver la complexité de `dichotomie(f, a, b, epsilon)`. Pour cela il faut savoir combien de fois la boucle `while` s'exécute. Si on note a_k et b_k les valeurs de `a` et `b` après k itérations, il faut donc connaître le plus petit n tel que $b_n - a_n$ soit inférieur à `epsilon`.

II Méthode de Newton

La méthode de Newton suppose que f est dérivable et consiste à approcher un zéro de f par une suite u_n définie de la façon suivante:

- u_0 est quelconque (mais de préférence proche du zéro qu'on veut approximer)
- Pour tout $n \geq 0$, u_{n+1} est obtenu à partir de u_n comme l'intersection de l'axe des abscisses avec la tangente à f en u_n, comme sur le dessin suivant:



1. Montrer que $u_{n+1} = u_n - \dfrac{f(u_n)}{f'(u_n)}$.
   On rappelle que l'équation de la tangente à f en a est: $y = f(a) + f'(a)(x - a)$.
2. Écrire une fonction `newton(f, fp, u0, n)` renvoyant le terme u_n de la suite définie ci-dessus, avec `fp` la dérivée de `f` et $u_0 =$ `u0`.
3. Utiliser `newton` pour approximer $\sqrt{3}$, π, e, à 10^{-3} près et comparer la vitesse de convergence avec la méthode par dichotomie.