TP n° 10 : Dichotomie et méthode de Newton

Le but de ce TP est de résoudre des équations de la forme $f(x) = 0$ où f est une fonction continue sur un intervalle $[a\,;b]$ donné. Plus précisément, on va supposer qu'au moins une solution à cette équation existe et on va chercher une approximation de cette solution grâce à deux méthodes :

- la dichotomie ;
- la méthode de Newton.

Dans la suite, on va considérer la fonction g définie sur $[0\,;\pi]$ par $g(x) = \sqrt{\sin x} + x - 1$. Comme cette fonction est continue et vérifie $g(0) = -1 < 0$ et $g(\pi) = \pi - 1 > 0$, on sait qu'il existe au moins un réel $\alpha \in [0\,;\pi]$ tel que $g(\alpha) = 0$.

1. Écrire une fonction `g(x)` prenant en argument un flottant `x` et qui renvoie la valeur de $g(x)$ que l'on a définie ci-dessus.

   Indication : on importera le module `math` via la commande

   ```
   from math import *
   ```

   afin de pouvoir utiliser les fonctions usuelles comme `sqrt`, `sin` ou le nombre `pi`.
2. Se servir de cette fonction pour retrouver les valeurs de $g(0)$ et $g(\pi)$.

1 Dichotomie

Comme l'indique son nom, l'algorithme de dichotomie consiste à *couper en deux* l'intervalle de recherche de la solution à chaque étape. Pour cela on calcule la valeur de la fonction au milieu de l'intervalle et on regarde si le changement de signe à lieu à gauche ou à droite de celui-ci, on réitère alors le procédé sur le demi-intervalle restant et ainsi de suite jusqu'à la précision souhaitée.

3. Recopier la fonction suivante et compléter les lignes 3, 5, 7 et 8 afin qu'elle renvoie une solution approchée de $f(x) = 0$ à la précision `eps`.

   ```
   1 def Dichotomie(f, a, b, eps):
   2     while abs(b-a)>eps:
   3         milieu =
   4         if f(a)*f(milieu)<0:
   5             b =
   6         else:
   7             a =
   8     return
   ```

4. Utiliser cette fonction pour déterminer une valeur approchée de la solution de $g(x) = 0$ à 10^{-3} près.
5. Modifier la fonction précédente pour qu'elle renvoie également le nombre d'itérations de la boucle effectuées.
6. Donner le nombre d'itérations effectuées pour obtenir une valeur approchée de la solution de $g(x) = 0$ à respectivement 10^{-3}, 10^{-9} et 10^{-15} près.
7. Déterminer une valeur approchée à 10^{-15} près de l'équation $\cos x = x^3$.

2 Méthode de Newton

La méthode de Newton consiste à approximer successivement la fonction f par certaines de ses tangentes. Plus précisément, on part d'une valeur x_0. On trace la tangente à la courbe représentative de f au point d'abscisse x_0 et on note x_1 l'intersection de cette tangente avec l'axe des abscisses. On réitère le procédé à partir du point d'abscisse x_1 afin d'obtenir x_2 et ainsi de suite jusqu'à la précision souhaitée.

Remarque. Faire un dessin ou regarder l'animation dans le cours en ligne pour illustrer cette construction.

8. Rappeler l'équation de la tangente à courbe représentative de f au point d'abscisse x_n.
9. En déduire l'expression de x_{n+1} en fonction de x_n. (On rappelle que x_{n+1} est l'abscisse du point d'intersection de la tangente au point d'abscisse x_n avec l'axe des abscisses.)
10. Écrire une fonction `Newton(f,fp,x0,eps)` qui :
    – prend en arguments une fonction `f`, sa fonction dérivée `fp` et deux flottants `x0` et `eps` ;
    – calcule les termes successifs de la suite (x_n) définie à la question précédente à partir de la valeur x_0 et qui s'arrête lorsque $|x_{n+1} - x_n|$ est plus petit que `eps` ;
    – renvoie la dernière valeur calculée de la suite (x_n).
11. Écrire une fonction `gp(x)` prenant en argument un flottant `x` et qui renvoie la valeur de $g'(x)$, valeur en x de la dérivée de la fonction g définie au début de ce TP.
12. Utiliser la fonction `Newton` pour déterminer une valeur approchée de la solution de $g(x) = 0$ à 10^{-3} près. (Vérifier que le résultat coïncide avec celui obtenu par dichotomie.)
13. Modifier la fonction `Newton` pour qu'elle renvoie également le nombre d'itérations de la boucle effectuées.
14. Donner le nombre d'itérations effectuées pour obtenir une valeur approchée de la solution de $g(x) = 0$ à respectivement 10^{-3}, 10^{-9} et 10^{-15} près. Comparer avec les résultats obtenus à la question 6.

Remarque. En pratique, la convergence de la suite (x_n) n'est pas assurée : elle peut sortir du domaine de définition de la fonction, prendre une valeur qui annule la dérivée de f (pourquoi est-ce un problème ?) ou encore alterner entre deux valeurs (considérer la fonction $x \mapsto \sqrt{|x|}$). Pour pallier en partie ces problèmes, en général on rajoute dans la boucle `while` une condition d'arrêt sur un nombre maximal d'itérations possibles.

Un inconvénient de la méthode de Newton est qu'elle nécessite la connaissance de la fonction dérivée f'. Si l'on ne connaît pas la valeur de f' en x, on peut l'approximer par le taux d'accroissement de f entre x et $x + h$, *i.e.*

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

où h est choisi assez petit (10^{-3} par exemple).

15. Écrire une fonction `Derivee(f,x)` qui prend en arguments une fonction `f` et un flottant `x` et qui renvoie la valeur approchée de $f'(x)$.
16. En s'inspirant de la fonction `Newton` écrite à la question 10, écrire une fonction `Newton2(f,x0,eps)` qui prend en arguments une fonction `f` ainsi que deux flottants `x0` et `eps` et qui renvoie une valeur approchée à `eps` près de la solution de $f(x) = 0$.
17. Tester la fonction sur les exemples précédents.