Code de partage avec Capytale : 2a23-1343294 **Corrigé**

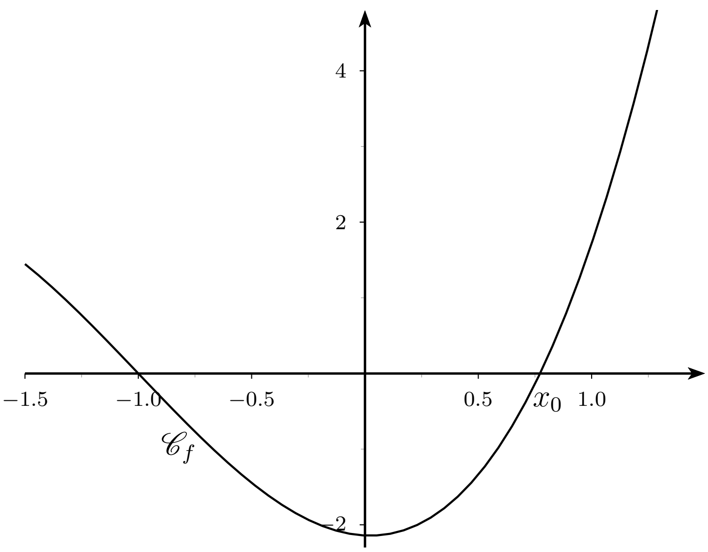
L'objectif de cette séance est de construire un algorithme de recherche d'un zéro d'une fonction par la méthode de dichotomie (on appelle **zéro d'une fonction** f une solution de l'équation $f(x) = 0$). La dichotomie, qui provient du grec « division en deux parties », sera itérée plusieurs fois, jusqu'à obtenir le niveau de précision souhaitée.

1 Principe

Expliquons le principe de dichotomie sur l'exemple ci-dessous (voir graphique).

- on voit que f est continue (et strictement croissante) sur l'intervalle $[0,1]$, et de plus que $f(0) < 0$ et que $f(1) > 0$
  de fait, grâce au théorème des valeurs intermédiaires, on en déduit que $\exists x_0 \in]0,1[, f(x_0) = 0$
- on calcule l'image du centre de l'intervalle $[0,1]$, qui est $m_1 = \dfrac{1}{2}$
  on trouve $f\left(\dfrac{1}{2}\right) < 0$, donc forcément, $x_0 > \dfrac{1}{2}$
  donc on poursuit la recherche de x_0 dans l'intervalle $\left[\dfrac{1}{2}, 1\right]$
- on coupe l'intervalle $\left[\dfrac{1}{2}, 1\right]$ en 2 avec $m_2 = \dfrac{3}{4}$
  or $f\left(\dfrac{3}{4}\right) < 0$, donc $x_0 > \dfrac{3}{4}$
  ▷ on poursuit donc la recherche de x_0 dans l'intervalle $\left[\dfrac{3}{4}, 1\right]$
- on coupe l'intervalle $\left[\dfrac{3}{4}, 1\right]$ en 2 avec $m_3 = \dfrac{7}{8}$
  or $f\left(\dfrac{7}{8}\right) > 0$, donc $x_0 < \dfrac{7}{8}$
  ▷ on poursuit donc la recherche de x_0 dans l'intervalle $\left[\dfrac{3}{4}, \dfrac{7}{8}\right]$

A ce stade, on sait que $\dfrac{3}{4} < x_0 < \dfrac{7}{8}$, soit : $0,75 < x_0 < 0,875$. L'amplitude de ce dernier intervalle est $0,875 - 0,75 = 0,125$. Si cette amplitude et plus grande que la précision choisie (c'est le cas par exemple si on souhaite avoir une valeur approchée de x_0 à 10^{-1} près, dans ce cas la précision choisie est de 10^{-1}), il faut continuer le procédé.
On le répète donc **tant que** l'amplitude de l'intervalle est supérieure à la précision choisie.



Dans cet exemple, si on appelle $[a,b]$ l'intervalle qui contiendra x_0 à chaque étape du procédé :

- au départ $[a,b] = [0;1]$
- après l'étape 1, $[a,b] = [0,5;1]$
- après l'étape 2, $[a,b] = [0,75;1]$
- après l'étape 3, $[a,b] = [0,75;0,875]$

Exercice 1 - procédé de dichotomie sur un exemple

On pose $f(x)=\ln(x)+\dfrac{x}{2}-1$ pour $x>0$

1. Justifier que f s'annule sur l'intervalle $[1,2]$ en un réel qu'on notera x_0

   $f(1)=-0,5<0$ et $f(2)=\ln 2>0$, de plus f est continue (somme de fonctions continues), donc d'après le théorème des valeurs intermédiaires, $\exists x_0\in[1;2]$ tel que $f(x_0)=0$
   En étudiant la fonction (on trouverait qu'elle est strictement croissante sur cet intervalle), on pourrait dire que x_0 est l'unique point où f s'annule sur $[1;2]$

2. Utiliser Python pour définir la fonction puis calculer $f(m)$ et faire *pas à pas* l'algorithme avec pour valeurs de départ $a=1$ et $b=2$, pour remplir le tableau suivant donnant les valeurs de a, b, m ainsi que le signe de $f(m)$:

```
import numpy as np
def f(x):
  return np.log(x)+x/2-1
```

   Il faut importer la bibliothèque numpy pour pouvoir disposer du logarithme.
   Après avoir exécuté la fonction, il suffit de calculer : `f(1.5)`, `f(1.25)`, `f(1.375)`, `f(1.3125)`.

Etape	a	b	m	$f(m)$
1	1	2	$1,5$	$+$
2	1	$1,5$	$1,25$	$-$
3	$1,25$	$1,5$	$1,375$	$+$
4	$1,25$	$1,375$	$1,3125$	$-$
5	$1,3125$	$1,375$	$1,34375$	$+$

3. Calculer l'amplitude de l'intervalle $[a,b]$ à chaque étape.

   Au début l'amplitude est de 1, puis elle est respectivement de $0,5$ à la fin de la première étape, $0,25$ à la fin de la deuxième, $0,125$ à la fin de la troisième et $0,0625$ à la fin de la quatrième.

4. Combien d'étapes faudrait-il pour avoir une valeur approchée de x_0 à $0,1$ près ? à 10^{-3} près ? à 10^{-8} près ?

   On voit qu'à la fin de l'étape 3, l'amplitude de l'intervalle vaut $\dfrac{1}{2^4}\leqslant 0,1$
   Pour obtenir une précision d'un millième, il faut $2^n\geqslant 1\,000$, ce qui sera dépassé pour $n=10$ ($2^{10}=1\,024$)
   Pour une précision de 10^{-8}, il faut $2^n\geqslant 1\,000$, soit $n\geqslant\dfrac{8\ln(10)}{\ln 2}$, donc à partir de 27

5. Ecrire un algorithme qui permet de calculer la valeur de x_0 à 10^{-3} près.

```
a=1
b=2
while (b-a)>10**(-3)
  m=(a+b)/2
  if f(m)>0 :
    b=m
  else :
    a=m
print([a,b])
```

   On sait qu'à la fin du dernier passage complet dans la boucle $f(a)<0$, $f(b)>0$ et $b-a<10^{-3}$ (ce qui arrête la boucle au passage suivant).
   On obtient ici $a\approx 1,3701$ et $b\approx 1,3711$
   On peut donc proposer $1,371$ comme valeur approchée de x_0 à 10^{-3} près.

   On peut ensuite améliorer la précision en changeant `10**(-3)` en `10**(-8)` par exemple.