NUM.1 : Resolution d'une equation, methode par dichotomie

Objectif : résoudre une équation du type $f(x) = 0$,

avec f fonction continue qui s'annule une fois sur un intervalle $[a, b]$ d'étude

Certaines équations ne sont pas (ou difficilement) solubles à la main ; on peut alors recourir à une résolution numérique. La méthode par dichotomie peut résoudre cette difficulté.

I. Description de la methode

Soit l'équation $f(x) = 0$, avec f une fonction continue qui s'annule une fois sur l'intervalle [a,b]. La méthode dichotomique permet de trouver une solution approchée de cette équation.

Prenons l'exemple de la fonction $f(x) = x^2 - 2$. On cherche à résoudre $f(x) = 0$ sur l'intervalle $[0,2]$. On sait que la solution réelle est $x = \sqrt{2} \approx 1{,}414213562\ldots$; nous allons essayer de retrouver ceci avec l'algorithme de dichotomie.

Préliminaire : tester si deux réels sont de signes opposés

Soit y et z deux réels non nuls.

→ y et z sont de même signe (tous deux positifs ou tous deux négatifs) si et seulement si $y \times z > 0$.

→ y et z sont de signes opposés (l'un positif, l'autre négatif) si et seulement si $y \times z < 0$.

La fonction $f(x)$ étant continue et ne s'annulant qu'une seule fois entre a et b, on a nécessairement $f(a)$ et $f(b)$ de signes opposés.

Ceci se traduit par $f(a) \times f(b) < 0$

Idée générale de l'algorithme de dichotomie

On considère le milieu $m = \frac{a+b}{2}$ de l'intervalle $[a, b]$, et on détermine si le zéro recherché se trouve entre a et m ou entre m et b.

Selon le résultat obtenu, on recommence la recherche selon la même méthode dans l'intervalle $[a, m]$ ou $[m, b]$, intervalle qui a été réduit de moitié.

On s'arrête lorsque l'intervalle auquel appartient la solution est jugé suffisamment petit, ce qui dépend de la précision recherchée.

Détail de l'algorithme

On définit des valeurs pour les grandeurs a, b correspondant aux bornes de l'intervalle de recherche, ainsi que pour ε qui va définir la précision recherchée pour la résolution de l'équation.

Tant que $|a - b| > \varepsilon$, on réalise la boucle suivante :

On définit le mieux de l'intervalle avec : $m = \frac{a+b}{2}$

→Si $f(a) \times f(m) < 0$, le zéro de f (la solution recherchée pour laquelle la fonction s'annule) se trouve entre a et m : la solution appartient alors à l'intervalle $[a, m]$

On définit ce nouvel intervalle de recherche par $a = a$ et $b = m$.

→ Dans le cas contraire $f(a) \times f(m) > 0$, le zéro de f se trouve entre m et b : la solution appartient alors à l'intervalle $[m, b]$.

On définit ce nouvel intervalle de recherche par $a = m$ et $b = b$.

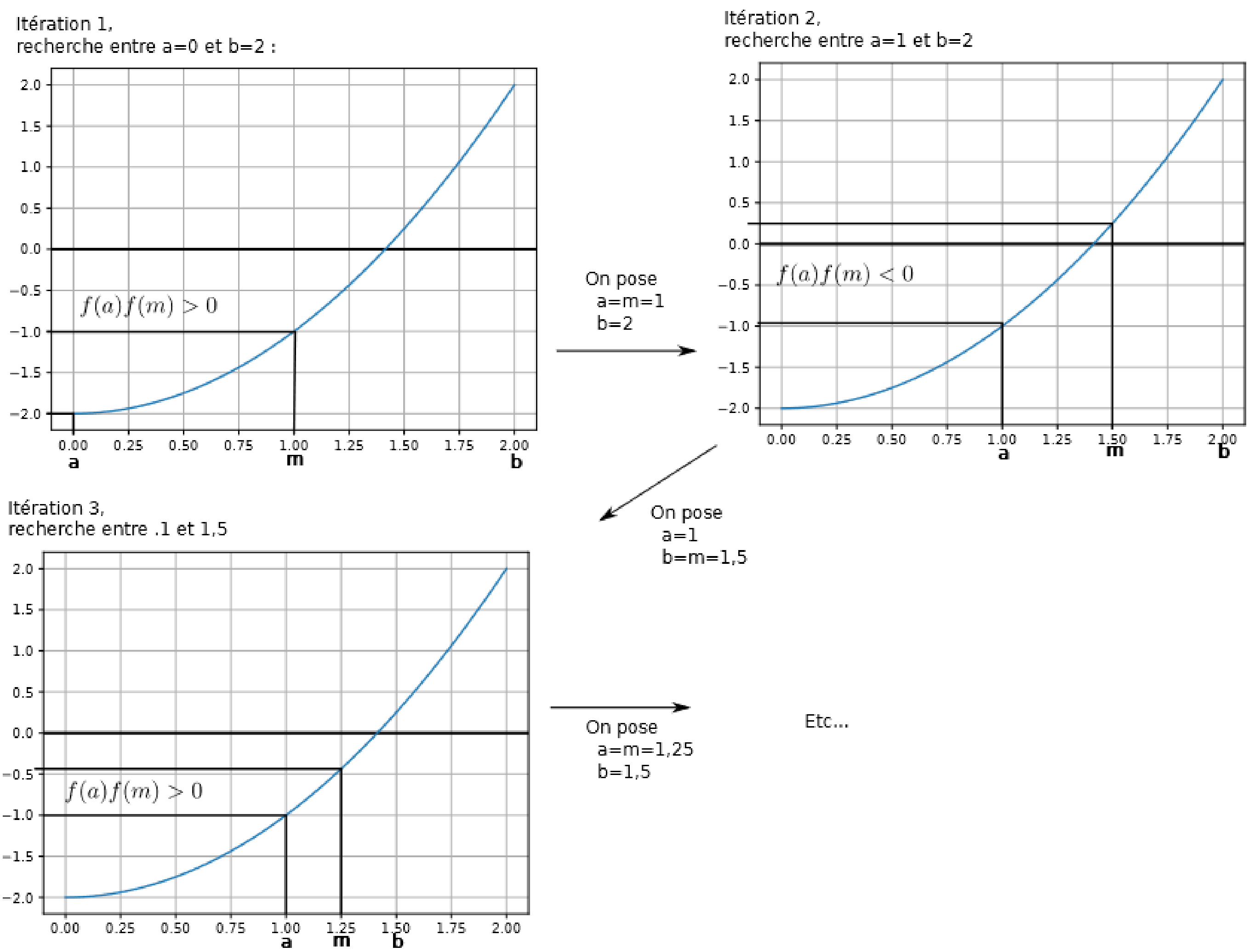
La boucle se poursuit selon ce principe tant que m n'est pas suffisamment proche de la valeur recherchée, soit tant que

$$|a - b| > \varepsilon$$

Lorsqu'on sort de la boucle, le zéro est donné par la valeur de m, à ε près.

Signification de ε : prenons par exemple $\varepsilon = 10^{-6}$. L'algorithme s'arrête donc lorsque $b - a < 10^{-6}$ Comme on sait par construction que le 0 de f (la solution recherchée pour laquelle la fonction s'annule) appartient nécessairement à l'intervalle $[a, b]$, ceci signifie que $m = \frac{a+b}{2}$ est proche de la solution recherchée à moins de 10^{-6} près.

Remarque : il existe d'autres critères de terminaison, par exemple arrêter lorsque l'algorithme trouve un x tel que $|f(x)| = 10^{-6}$

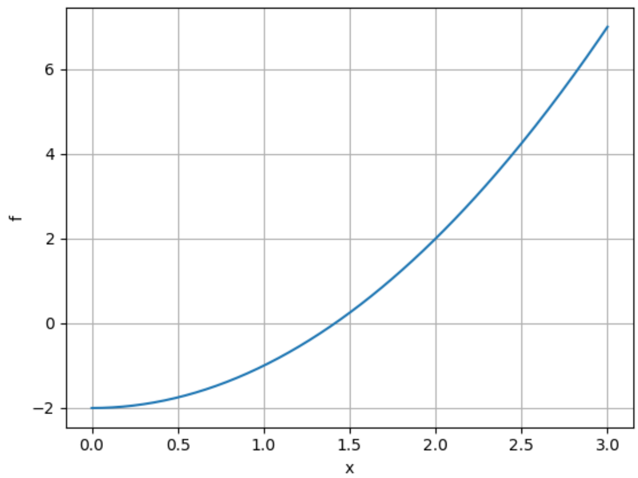


II. Programmation sous Python :

Exemple :

Nous cherchons à résoudre : $f(x) = x^2 - 2 = 0$

```python
import numpy as np
import matplotlib . pyplot as plt
from math import *
def f(x):

    return x**2-2

epsilon = 1e-6
a=0
b=3

#affichage de la courbe, cela permet d'optimiser l'intervalle a, b

x=np.linspace(0, 3, 200) #créé un tableau de valeurs de x, compris entre 0 et 3
plt.figure(1) #création de la figure 1
plt.plot(x, f(x))  #tracé

plt.xlabel('x')  #légende de l'axe des abscisses
plt.ylabel('f') #légende de l'axe des ordonnées
plt.grid()          #ajout d'une grille
plt.show()   #affiche le graphe

nb_iter = 0

while abs(b-a) > epsilon:
    m = (a+b)/2
    if f(a)*f(m) <0:  #si f(a) et f(b) n'ont pas le même signe
        b=m        #alors c'est que le 0 de f se trouve entre  a et m
    else:
        a=m      #sinon, c'est que le 0 de f est entre m et b
    nb_iter= nb_iter +1

print(m)
print(nb_iter)
```



Autre méthode :

La fonction **bisect** de la bibliothèque scipy.optimize permet de trouver directement le zéro d'une fonction. Elle utilise une méthode de type dichotomie.

On l'utilise ainsi :

```python
import scipy.optimize as sp
x = sp.bisect(f ,a , b )
print ( x )
```

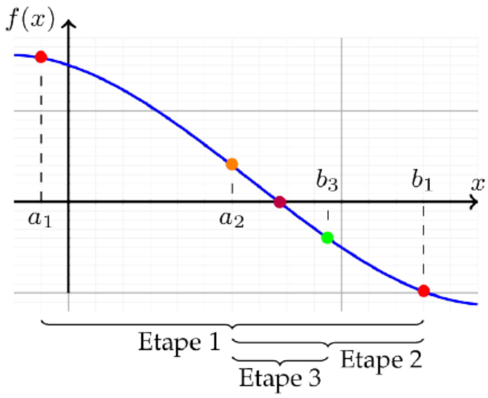
a et b sont les limites de l'intervalle de recherche, et f la fonction pour laquelle on résout $f\ (x)\ =\ 0$.

Résumé de la méthode et autre exemple de programme :

On cherche une solution x de l'équation $f(x) = 0$ (avec f continue) dans un intervalle $[a, b]$ (avec $f(a)f(b) < 0$).

1. On calcule le milieu de l'intervalle $[a, b]$:
$$m = \frac{a+b}{2}$$
2. Il y a deux cas possibles.
   — Si $f(a)$ et $f(m)$ sont de signes opposés, alors x se trouve à gauche de m, c'est-à-dire $x \in [a, m]$. On réaffecte alors $b \leftarrow m$.
   — Sinon, x se trouve à droite de m, c'est-à-dire $x \in [m, b]$. On réaffecte alors $a \leftarrow m$.
3. On recommence à l'étape 1 avec le nouvel intervalle $[a, b]$.
4. On arrête la recherche lorsque $b - a < \varepsilon$ où ε est la précision souhaitée.

```python
def dicho(f:function,a:float,b:float,eps:float)->float:
    "renvoie la solution de f(x)=0 sur [a,b] à eps près"
    while b-a>eps:
        m=(a+b)/2
        if f(a)*f(m)<=0:
            b=m
        else:
            a=m
    return m
```



On peut également utiliser la fonction `bisect(f,a,b)` du module scipy.optimize.