RÉSOLUTION NUMÉRIQUE DE L'ÉQUATION $f(x) = 0$

méthode de dichotomie et méthode de Newton

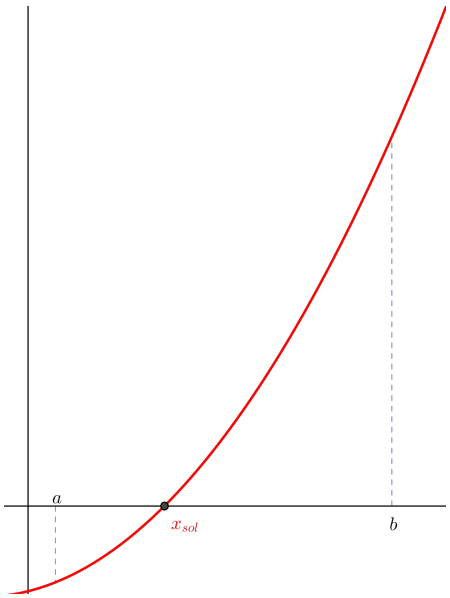
Résolution approchée d'une équation

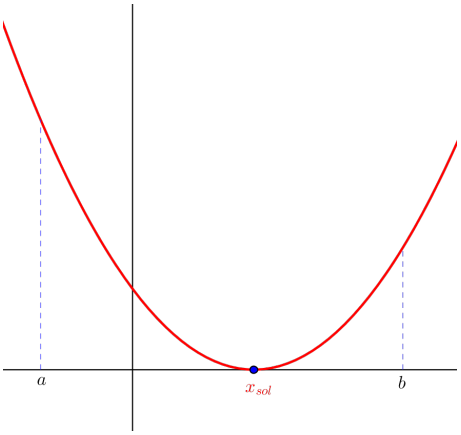
▷ Dans la mesure où on ne sait pas résoudre de manière exacte toutes les équations numériques que l'on peut être amené à rencontrer, il est légitime de mettre au point des démarches permettant d'obtenir une valeur approchée d'une solution d'équation.

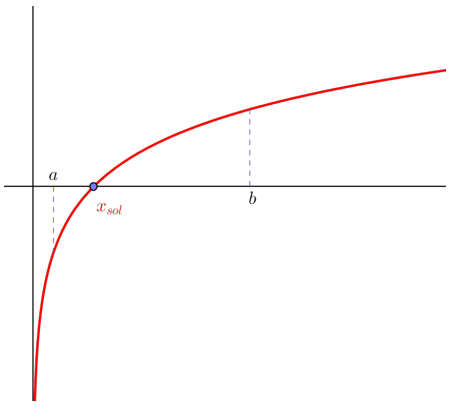
▷ La méthode de la dichotomie et la méthode de Newton sont deux techniques permettant, de manière algorithmique, de calculer une approximation d'une solution de l'équation $f(x) = 0$, où f est une fonction définie sur un intervalle et à valeurs réelles.

Cadre de travail

- Dans la suite, on notera f une fonction continue sur un intervalle $[a, b]$.
- On supposera, en outre, que f s'annule en un unique point de $[a, b]$, que l'on notera x_{sol}.







1 Méthode de dichotomie

Contexte de travail et idée de départ

On suppose que : $\begin{cases} \circ & f \text{ est continue sur } [a,b] \\ \circ & f(a) \text{ et } f(b) \text{ sont de signes contraires} \end{cases}$

Ces hypothèses suffisent à garantir que alors f s'annule (au moins une fois) sur $[a,b]$.

1.1 Démarche

Technique à répéter

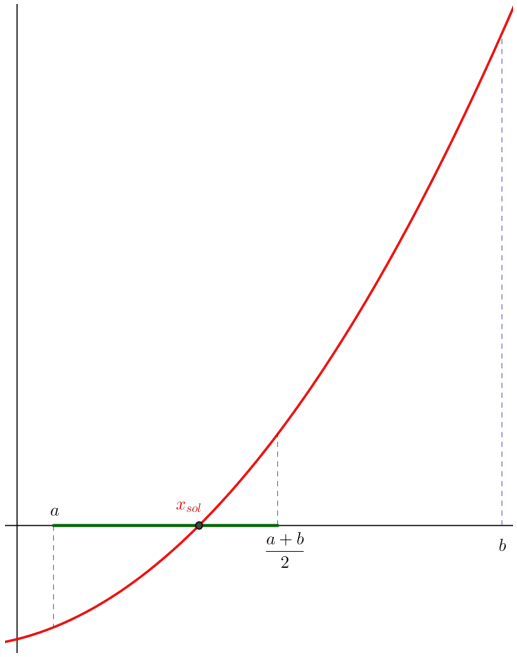
i. On divise l'intervalle $[a,b]$ en deux, et on ne garde que la section qui contient la solution.

↪ *Pour cela, on examine le signe de* $f\left(\dfrac{a+b}{2}\right)$ *:*

→ si $f(a)$ et $f\left(\dfrac{a+b}{2}\right)$ sont de signes contraires, on se place alors sur $\left[a, \dfrac{a+b}{2}\right]$;

→ sinon, il faudra travailler sur $\left[\dfrac{a+b}{2}, b\right]$.

ii. Puis on recommence.



Condition d'arrêt

Plusieurs conditions peuvent justifier l'arrêt des itérations décrites précédemment :

- ◇ lorsque la taille de notre intervalle de travail est « suffisamment petite »
  ↪ *l'écart entre le milieu de cet intervalle et x_{sol} est encore plus petit*
- ◇ lorsque l'image du milieu de notre intervalle par f est « suffisamment petit »
  ↪ *cette condition nécessite que f prend des « petites » valeurs exclusivement au voisinage de x_{sol}*
- ◇ lorsqu'on estime que le nombre d'itérations est suffisant.

Le dernier milieu calculé sera alors considéré comme une approximation de x_{sol}.

1.2 Programmation en Python

```
def recherche_solution_dichotomie(fonction,borne_inf,borne_sup,tolerance,nb_iterations_max) :
    milieu=(borne_sup+borne_inf)/2
    nombre_iterations=0
    while (fonction(milieu)>tolerance) and (nombre_iterations<=nb_iterations_max) :
        if f(borne_inf)*f(milieu)<0 :
            borne_sup=milieu
        else :
            borne_inf=milieu
        milieu=(borne_sup+borne_inf)/2
        nombre_iterations=nombre_iterations+1
    return milieu
```

Remarque

- ▷ Dans Python, la bibliothèque `scipy.optimize` contient la méthode de dichotomie.
- ▷ Pour l'utiliser, il suffit de charger la bibliothèque, puis d'appliquer la fonction `bisec` en précisant (au moins) la fonction `f` considérée, ainsi que les bornes `borne_inf` et `borne_sup` de l'intervalle de travail.

```
from scipy.optimize import *
...
bisect(f,borne_inf,borne_sup)
```

1.3 Vitesse de convergence

Comme, à chaque étape, la taille de l'intervalle de travail est divisée par 2, l'intervalle de recherche à la n-ième étape est de longueur $\dfrac{b-a}{2^n}$: le milieu de cet intervalle est alors distant de x_{sol} d'au plus $\dfrac{b-a}{2^n}$.

2 Méthode de Newton

Contexte de travail et idée de départ

On suppose que : $\begin{cases} \circ & f \text{ est dérivable sur } [a,b] \\ \circ & f' \text{ ne s'annule pas sur } [a,b] \end{cases}$

On va considérer que la représentation graphique $\mathscr{C}_f$ de f est « proche » de sa tangente en un point : l'intersection de cette tangente avec (Ox) doit nous « rapprocher » de x_{sol}

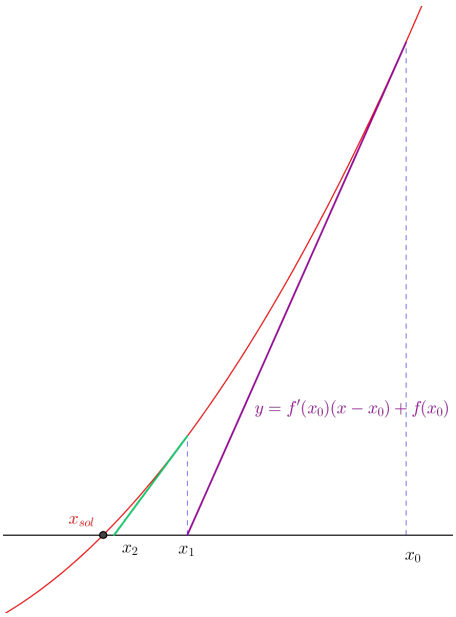
2.1 Démarche

Technique à répéter

i. On considère un réel x_0 dans I

ii. → On considère alors la tangente à $\mathscr{C}_f$ au point d'abscisse x_0, qui a pour équation cartésienne $y = f'(x_0)(x-x_0)+f(x_0)$.
→ Le point d'intersection de cette droite et de l'axe des abscisses permet d'approcher x_{sol} :
on note ainsi x_1 le réel vérifiant $0 = f'(x_0)(x_1-x_0)+f(x_0)$,
c'est-à-dire que l'on pose $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$.

iii. Puis on recommence : le réel x_n étant construit, on pose $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$.



Condition d'arrêt

Plusieurs conditions peuvent justifier l'arrêt des itérations décrites précédemment :
- ◇ lorsque l'écart entre deux termes consécutifs est « suffisamment petit »
- ◇ lorsque l'image du milieu de notre intervalle par f est « suffisamment petit »
  ↪ *cette condition nécessite que f prend des « petites » valeurs exclusivement au voisinage de x_{sol}*
- ◇ lorsqu'on estime que le nombre d'itérations est suffisant.

Le dernier terme x_n calculé sera alors considéré comme une approximation de x_{sol}.

2.2 Programmation en Python

```
def recherche_solution_Newton(fonction,derivee_fonction,position_depart,tolerance,nb_iterations_max) :
    approximation=position_depart
    nombre_iterations=0
    while (fonction(approximation)>tolerance) and (nombre_iterations<=nb_iterations_max) :
        approximation=approximation-fonction(approximation)/fonction_derivee(approximation)
        nombre_iterations=nombre_iterations+1
    return approximation
```

Remarque

▷ Dans Python, la bibliothèque `scipy.optimize` contient la méthode de Newton.

▷ Pour l'utiliser, il suffit de charger la bibliothèque, puis d'appliquer la fonction `newton` en précisant (au moins) la fonction `f` considérée, sa dérivée `der_f` et la valeur initiale `position_depart`.

```
from scipy.optimize import *
...
newton(f,position_depart,der_f)
```

2.3 Vitesse de convergence

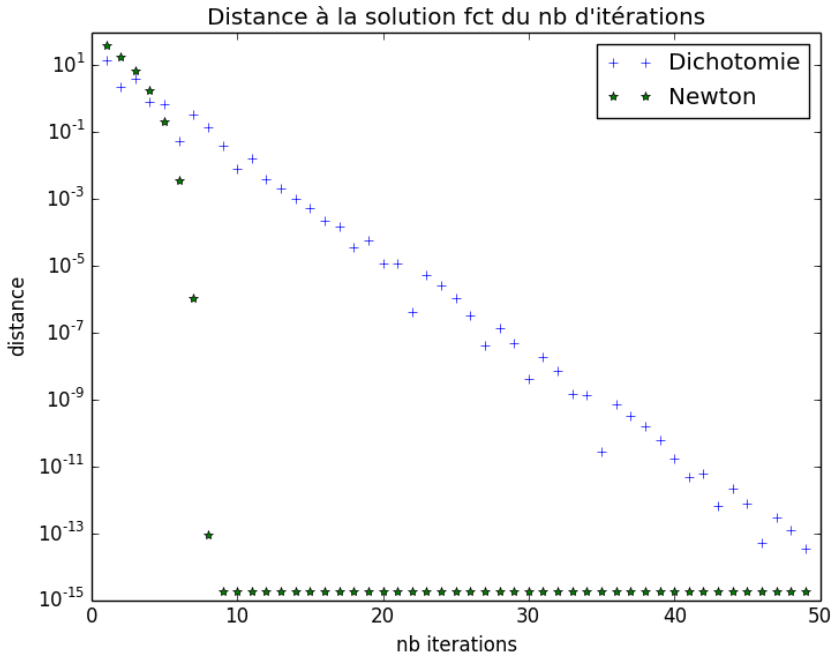
Sous certaines hypothèses concernant f (que nous supposerons vérifiées), on montre qu'il existe une constante positive C telle que, pour tout $n \in \mathbb{N}$, $|x_{n+1} - x_{sol}| \leq C\,|x_n - x_{sol}|^2$.
On en déduit que, pour tout $n \in \mathbb{N}$, $|x_n - x_{sol}| \leq \frac{1}{C} \times (C\,|x_0 - x_{sol}|)^{2^n}$.
Il « suffit » alors que $|x_0 - x_{sol}| \leq \frac{1}{C}$ pour être en mesure de garantir la convergence de (x_n) vers x_{sol}... et de majorer la distance de x_n à x_{sol}.

3 Comparaison des deux méthodes

Pour comparer l'« efficacité » des deux méthodes de résolution *numérique* précédemment étudiées, plusieurs critères doivent être considérés :

▷ *La rapidité de convergence*

De ce point de vue, dans les hypothèses où les deux méthodes convergent, la méthode de Newton met en place une suite de nombres qui converge plus « rapidement » (on parle ici de convergence *quadratique*) vers x_{sol} que celle construite par la méthode de dichotomie (qui converge de façon *linéaire*) :



▷ *Le caractère « contraignant » des hypothèses*

La méthode de dichotomie nécessite, pour converger, des hypothèses plus simples que la méthode de Newton : il suffit de disposer d'une fonction continue qui change de signe pour que cette construction converge vers un point d'annulation de cette fonction. Dans certains cas (respectant pourtant les hypothèses initialement formulées), la méthode de Newton risque de ne pas converger...