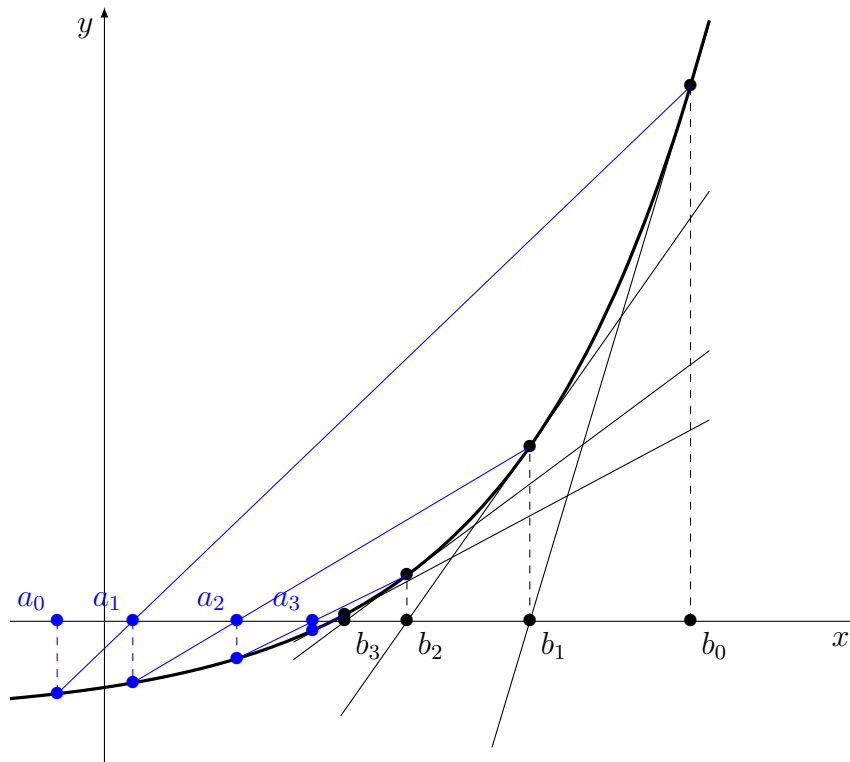


Résolution numérique des équations non linéaires



Objectifs

Il existe de nombreuses situations en ingénierie où nous avons besoin de trouver une approximation d'un zéro (ou d'une racine, dans le cas d'un polynôme) d'une fonction réelle d'une variable réelle, c'est-à-dire, étant donné un intervalle I de $\mathbb{R}$ et $f : I \rightarrow \mathbb{R}$ une fonction continue définie sur I et à valeurs réelles, trouver un réel $x \in I$ tel que

$$f(x) = 0 .$$

On appelle x racine ou zéro de cette équation. Un algorithme de recherche d'un zéro d'une fonction est une méthode numérique. Ces méthodes font partie des outils d'analyse numérique. Toutes les méthodes que nous allons présenter sont itératives et consistent en la construction d'une suite de réels $(x_k)_{k \in \mathbb{N}}$ qui, on l'espère, convergera vers x .

Table des matières

1	Méthodes d'encadrement	3
1.1	Méthode de dichotomie	3
1.2	Méthode de la corde, de Lagrange ou de la fausse position	7
2	Méthodes de point fixe	10
2.1	Méthode de la corde (de pente constante)	11
2.2	Méthode de la sécante	12
2.3	Méthode de Newton	13
3	Méthode de Newton par encadrement corde-tangente	16



Nicolas Mesnier, lycée Jean Perrin, Lyon
Compléments & mises à jour sur le site des MPSI
<https://cahier-de-prepa.fr/mpsi-perrin>

— Version du 1^{er} décembre 2023 —

1 Méthodes d'encadrement

Cette première classe de méthodes part du principe que l'intervalle dans lequel se trouve la racine est connu. Elles consistent toutes en la réduction progressive de l'intervalle dans lequel se trouve la racine. Ces méthodes sont basées sur le théorème suivant :

Théorème 1.1 (Théorème des valeurs intermédiaires)

Soit I un intervalle de $\mathbb{R}$ et $f : I \rightarrow \mathbb{R}$ une fonction continue. Soient $(a, b) \in I^2$ tels que $a < b$. Si $f(a)f(b) \leq 0$, alors il existe un réel $\ell \in [a, b]$ tel que $f(\ell) = 0$.

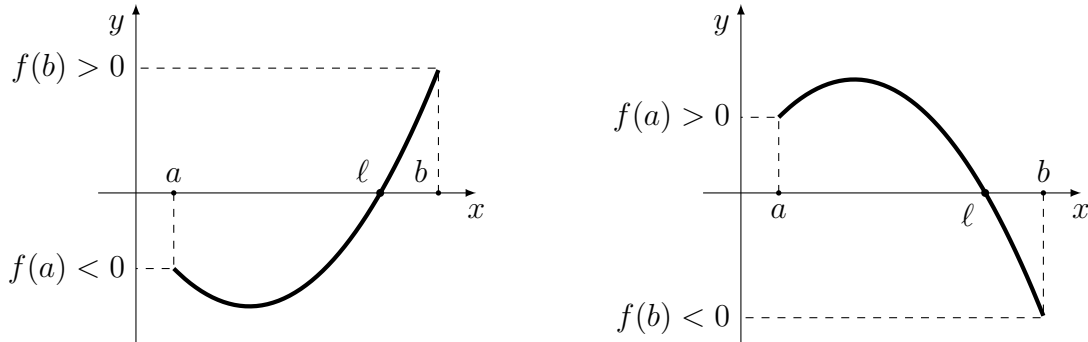


FIGURE 1 – Fonctions continues vérifiant $f(a)f(b) < 0$.

L'hypothèse de continuité est essentielle.

Définition 1.1 (Fonction continue)

Une fonction f est dite continue sur un intervalle $I \subset \mathbb{R}$ si, et seulement si, la fonction est continue en chaque point de I , c'est-à-dire :

$$\forall \alpha \in I, \quad \forall \varepsilon > 0, \quad \exists \eta > 0 \quad \forall x \in I \quad , \quad |x - \alpha| \leq \eta \implies |f(x) - f(\alpha)| \leq \varepsilon$$

1.1 Méthode de dichotomie

Le principe de la méthode de recherche de zéro par dichotomie consiste à diviser une suite d'intervalles à chaque fois en deux (étymologiquement, en grec, le mot veut dire couper en deux) et ne garder que celui dans lequel se trouve la racine.

Théorème 1.2 (Recherche d'un zéro par dichotomie)

Soient $a < b$ deux réels et $f : [a, b] \rightarrow \mathbb{R}$ une fonction continue. On suppose que $f(a)f(b) < 0$. Par théorème des valeurs intermédiaires, il existe au moins un zéro sur $I = [a, b]$. On peut le localiser dans l'une des moitiés du segment en étudiant le signe de f au milieu $m = (a+b)/2$. Ainsi, si on a $f(a)f(m) \leq 0$, c'est que $x \in [a, m]$, sinon $x \in [m, b]$. Partant de ce point, on définit par récurrence deux suites (a_k) et (b_k) par $a_0 = a$, $b_0 = b$ et

$$\forall k \in \mathbb{N}, \quad m_k = \left(\frac{a_k + b_k}{2} \right), \quad (a_{k+1}, b_{k+1}) = \begin{cases} (a_k, m_k) & \text{si } f(a_k)f(m_k) \leq 0 \\ (m_k, b_k) & \text{sinon} \end{cases}$$

Alors les deux suites (a_k) et (b_k) sont adjacentes (respectivement croissante et décroissante) et convergent vers une même limite ℓ qui est un zéro de la fonction f . Si l'on choisit de prendre a_k comme valeur approchée de ℓ , on obtient la majoration de l'erreur :

$$\forall k \in \mathbb{N}, |\ell - a_k| \leq \frac{b - a}{2^k}$$

Démonstration. On vérifie par récurrence que $\forall k \in \mathbb{N}$, $a_k \leq b_k$ et que $(b_k - a_k) = \frac{b - a}{2^k} \xrightarrow{k \rightarrow +\infty} 0$. Les deux suites (a_k) et (b_k) sont donc adjacentes (respectivement croissante et décroissante). Par théorème des suites adjacentes, elles convergent vers la même limite $\ell \in \mathbb{R}$. Puisque f est continue au point ℓ , $f(a_k) \xrightarrow{k \rightarrow +\infty} f(\ell)$ et $f(b_k) \xrightarrow{k \rightarrow +\infty} f(\ell)$. On vérifie également par récurrence que, $\forall k \in \mathbb{N}$, $f(a_k)f(b_k) < 0$ et donc que l'on a :

- soit $f(a_k) \leq 0$ et $f(b_k) \geq 0$;
- soit $f(a_k) \geq 0$ et $f(b_k) \leq 0$;

ce qui montre dans les deux cas, par passage à la limite $f(\ell) \leq 0$ et $f(\ell) \geq 0$, ce qui montre que $f(\ell) = 0$. Puisque $\forall k \in \mathbb{N}$, $a_k \leq \ell \leq b_k$, $|\ell - a_k| \leq (b_k - a_k) \leq \frac{b - a}{2^k}$, ce qui montre que a_k est une valeur approchée par défaut de ℓ à $\frac{b - a}{2^k}$ près car $a_k \leq \ell \leq a_k + \frac{b - a}{2^k}$. De même, b_k est une valeur approchée par excès de ℓ à $\frac{b - a}{2^k}$ près. $\square$

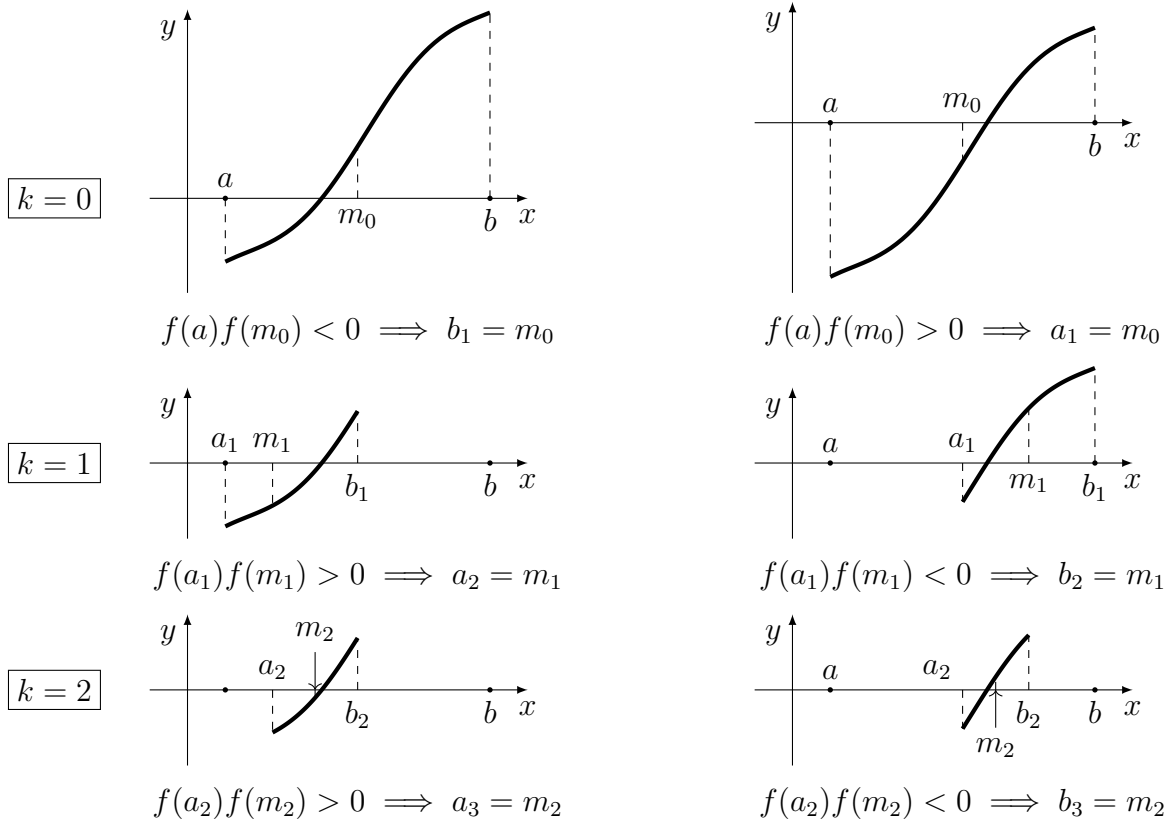


FIGURE 2 – Premières itérations de la méthode de dichotomie.

Ainsi, l'algorithme de recherche d'un zéro d'une fonction f dans un intervalle $[a, b]$ par dichotomie est basé sur la réduction progressive d'un intervalle tel que $f(a_k) f(b_k) < 0$ où a_k et b_k sont respectivement initialisés à a et b . À chaque itération, il s'agit de déterminer quelle partie du demi-intervalle sera conservée. On arrête le processus lorsque $b_k - a_k$ atteint une valeur de précision $\varepsilon > 0$ demandée.

À ce stade, il reste à montrer la terminaison et la correction. Pour la terminaison, on peut démontrer par récurrence immédiate qu'au début de la k -ième itération, on a :

$$\forall k \in \mathbb{N}, b_k - a_k = \frac{b - a}{2^k}$$

Supposant un seuil de précision $\varepsilon > 0$, on aura $b_k - a_k < 2\varepsilon$ pour k suffisamment grand, ce qui démontre la terminaison.

Pour la correction, il faut montrer que le résultat renvoyé est un réel r tel que l'équation $f(x) = 0$ possède une solution ℓ telle que $|\ell - r| \leq \varepsilon$. La clé de la démonstration est l'invariant :

$$\forall k \in \mathbb{N}, f(a_k)f(b_k) \leq 0$$

En effet :

- si $f(a_k)f(m_k) \leq 0$, on procède aux changements $a_{k+1} = a_k$ et $b_{k+1} = m_k$ et on a donc $f(a_{k+1})f(b_{k+1}) = f(a_{k+1})f(m_k) \leq 0$;
- si $f(a_k)f(m_k) > 0$, on a $b_{k+1} = b_k$ et $a_{k+1} = m_k$. Puisque $f(m_k)$ a le même signe (strict) que $f(a_k)$, le signe de $f(a_{k+1})f(b_{k+1})$ ne change pas.

Ainsi, le réel r renvoyé est le milieu d'un intervalle de longueur majorée par 2ε (condition de sortie de boucle) et qui contient un zéro ℓ de f . On a alors bien $|\ell - r| \leq \varepsilon$.

Comme il n'est pas possible d'obtenir une représentation exacte d'un nombre réel, on se limitera en pratique à une approximation à ε près du zéro, que l'on obtiendra donc en n itérations, avec n tel que :

$$|\ell - a_n| \leq \frac{b - a}{2^n} \leq \varepsilon \iff n \geq \log_2 \left(\frac{b - a}{\varepsilon} \right)$$

Comme en pratique on renvoie le milieu du dernier intervalle de longueur inférieure à 2ε , la boucle sera donc exécutée exactement

$$\left\lceil \log_2 \left(\frac{b - a}{2\varepsilon} \right) \right\rceil = \left\lceil \log_2 \left(\frac{b - a}{\varepsilon} \right) - 1 \right\rceil$$

fois (où $\lceil x \rceil = - \lfloor -x \rfloor$ est l'entier supérieur ou égal à x). La complexité de cet algorithme est donc de type logarithmique en $O \left(\ln(b - a) + \ln \left(\frac{1}{\varepsilon} \right) \right)$.

Ainsi, pour avoir p chiffres significatifs exacts, c'est-à-dire que les valeurs approchées diffèrent entre elles seulement d'une unité à partir de la p -ième décimale, il faut fixer $|r - \ell| \leq \varepsilon = 10^{-(p+1)}$. Si l'intervalle de recherche de la solution $[a, b]$ est de longueur $b - a = 10^q$, alors le nombre d'itérations requises sera de

$$\lceil (1 + p + q) \log_2(10) - 1 \rceil = O(3,3 \times (1 + p + q))$$

si on suppose $\log_2(10) = 3,321\,928 \dots \approx 3,3$.

MÉMO – Méthode de dichotomie

L'objectif de la méthode de recherche par dichotomie est de trouver une solution $x \in [a, b]$ tel que $f(x) = 0$, si un tel réel existe. En partant des deux extrémités de l'intervalle $a_0 = a$ et $b_0 = b$, le principe de cette méthode est de construire deux suites $(a_k)_{k \in \mathbb{N}}$ et $(b_k)_{k \in \mathbb{N}}$ à partir des relations de récurrence :

$$\forall k \in \mathbb{N}, m_k = \left(\frac{a_k + b_k}{2} \right), \quad (a_{k+1}, b_{k+1}) = \begin{cases} (a_k, m_k) & \text{si } f(a_k)f(m_k) \leq 0 \\ (m_k, b_k) & \text{sinon} \end{cases}$$

et vérifiant $\forall k \in \mathbb{N}, a_k \leq x \leq b_k$. Pour obtenir une valeur approchée de x à ε près, il suffit de s'arrêter lorsque $b_k - a_k \leq 2\varepsilon$: m_k est alors la valeur approchée de x recherchée. Le code Python s'écrit simplement :

```
def Dichotomie(f,a,b,epsilon):
    while abs(b-a)>2*epsilon:
        m = (a+b)/2
        if f(a)*f(m) <= 0:
            b=m
        else:
            a=m
    return((a+b)/2)
```

On peut compléter le code pour s'assurer qu'il existe au moins une racine sur $[a, b]$ et que la méthode de dichotomie converge en demandant $f(a)f(b) < 0$ avec le mot-clé **assert**, qui va provoquer une erreur particulière **AssertionError** si la condition n'est pas vérifiée à l'exécution. Par contre, le fait que le test soit non valide n'implique pas la non existence d'un zéro dans l'intervalle ! En minimisant le calcul des images, il vient alors le code suivant :

```
def Dichotomie(f,a,b,epsilon):
    fa,fb = f(a),f(b)
    assert fa*fb<0
    while abs(b-a)>2*epsilon:
        m = (a+b)/2
        fm = f(m)
        if fa * fm <= 0:
            b,fb = m,fm
        else:
            a,fa = m,fm
    return((a+b)/2)
```

La méthode de recherche par dichotomie est implémentée dans le module **scipy** :

```
import scipy.optimize
x = scipy.optimize.bisect(f,a,b,epsilon)
```

Afin de mettre au point des algorithmes possédant de meilleures propriétés de convergence que la méthode de dichotomie, il est nécessaire de prendre en compte les informations données par les valeurs de f et, éventuellement, par sa dérivée f' (si f est dérivable) ou par une approximation convenable de celle-ci.

1.2 Méthode de la corde, de Lagrange ou de la fausse position

Cette méthode porte de nombreux noms : méthode de la corde, de la fausse position (ou *regula falsi*), de Lagrange, des parties proportionnelles. L'idée de cette méthode est très simple : pour une fonction f continue sur un intervalle $[a, b]$, et vérifiant $f(a) \leq 0$, $f(b) > 0$, on trace le segment $[AB]$ où $A = (a, f(a))$ et $B = (b, f(b))$. La droite (AB) s'appelle la corde et a pour équation

$$y = f(a) + \frac{f(b) - f(a)}{b - a} (x - a) .$$

C'est le plus simple des polynômes d'interpolation car c'est un polynôme du premier degré. Si la corde reste au-dessus du graphe de f alors la fonction s'annule sur l'intervalle $[a_1, b]$ où $(a_1, 0)$ est le point d'intersection de la droite (AB) avec l'axe des abscisses. L'abscisse a_1 est obtenue en prenant $y = 0$ dans l'équation, ce qui donne :

$$a_1 = a - \frac{b - a}{f(b) - f(a)} f(a) .$$

On recommence en partant maintenant de l'intervalle $[a_1, b]$ pour obtenir une valeur a_2 . Ce processus conduit à la définition d'une suite récurrente :

$$a_0 = a \quad \text{et} \quad \forall k \in \mathbb{N}, \quad a_{k+1} = a_k - \frac{b - a_k}{f(b) - f(a_k)} f(a_k) .$$

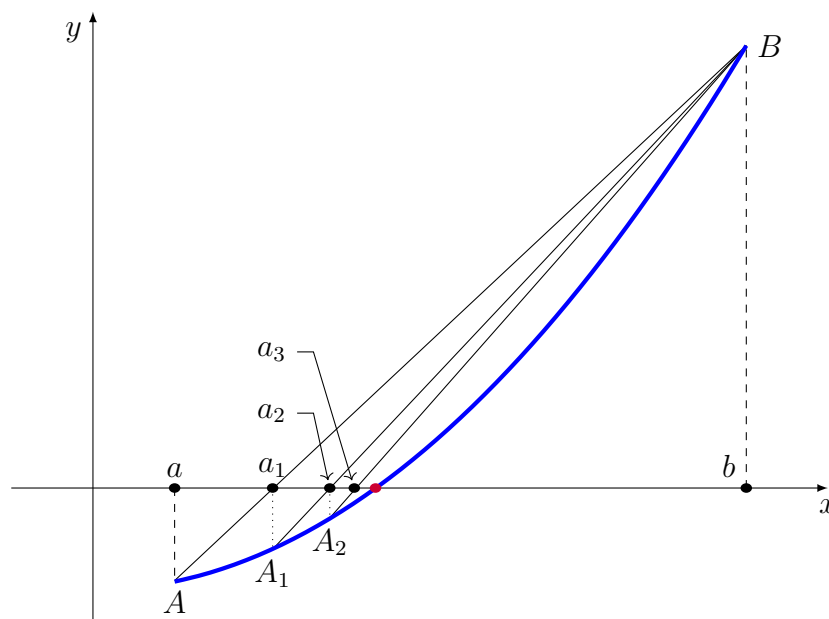


FIGURE 3 – Premières itérations de la méthode de la corde.

Proposition 1.1

Soit $f : [a, b] \rightarrow \mathbb{R}$ une fonction continue, strictement croissante et convexe telle que $f(a) \leq 0$, $f(b) > 0$. Alors la suite définie par

$$a_0 = a \quad \text{et} \quad \forall k \in \mathbb{N}, \quad a_{k+1} = a_n - \frac{b - a_k}{f(b) - f(a_k)} f(a_k)$$

est croissante et converge vers la solution ℓ de $(f(x) = 0)$.

Une fonction f deux fois dérivable est convexe sur un intervalle I si $\forall x \in I, f''(x) \geq 0$. L'hypothèse f convexe signifie exactement que pour tout $(u, v) \in ([a, b])^2$ la corde entre $(u, f(u))$ et $(v, f(v))$ est au-dessus du graphe de f .

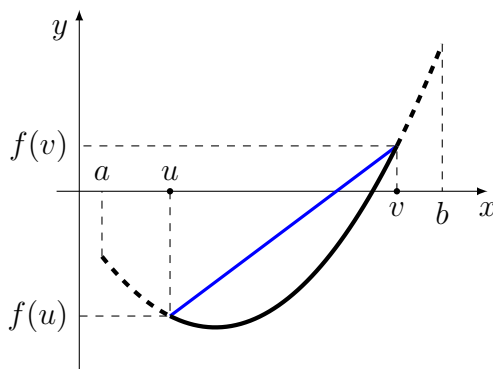


FIGURE 4 – Fonction f convexe sur $[a, b]$ et sa corde entre u et v .

Démonstration.

1. Justifions d'abord la construction de la suite récurrente.

L'équation de la droite passant par les deux points $(a, f(a))$ et $(b, f(b))$ est

$$y = (x - a) \frac{f(b) - f(a)}{b - a} + f(a)$$

Cette droite intersecte l'axe des abscisses en $(a_1, 0)$ qui vérifie donc $0 = (a_1 - a) \frac{f(b) - f(a)}{b - a} + f(a)$, donc $a_1 = a - \frac{b - a}{f(b) - f(a)} f(a)$.

2. Croissance de (a_k) .

Montrons par récurrence que $f(a_k) \leq 0$. C'est vrai au rang 0 car $f(a_0) = f(a) \leq 0$ par hypothèse. Supposons vraie l'hypothèse au rang k . Si $a_{k+1} < a_k$ (un cas qui s'avérera *a posteriori* jamais réalisé), alors comme f est strictement croissante, on a $f(a_{k+1}) < f(a_k)$, et en particulier $f(a_{k+1}) \leq 0$. Sinon $a_{k+1} \geq a_k$. Comme f est convexe : la sécante entre $(a_k, f(a_k))$ et $(b, f(b))$ est au-dessus du graphe de f . En particulier le point $(a_{k+1}, 0)$ (qui est sur cette sécante par définition a_{k+1}) est au-dessus du point $(a_{k+1}, f(a_{k+1}))$, et donc $f(a_{k+1}) \leq 0$ aussi dans ce cas, ce qui conclut la récurrence.

Comme $f(a_k) \leq 0$ et f est croissante, alors par la formule $a_{k+1} = a_k - \frac{b - a_k}{f(b) - f(a_k)} f(a_k)$, on obtient que $a_{k+1} \geq a_k$.

3. Convergence de (a_k) .

La suite (a_k) est croissante et majorée par b , donc elle converge. Notons ℓ sa limite. Par continuité $f(a_k) \rightarrow f(\ell)$. Comme pour tout k , $f(a_k) \leq 0$, on en déduit que $f(\ell) \leq 0$. En particulier, comme on suppose $f(b) > 0$, on a $\ell < b$. Comme $a_k \rightarrow \ell$, $a_{k+1} \rightarrow \ell$, $f(a_k) \rightarrow f(\ell)$, l'égalité $a_{k+1} = a_k - \frac{b - a_k}{f(b) - f(a_k)} f(a_k)$ devient à la limite (lorsque $k \rightarrow +\infty$) : $\ell = \ell - \frac{b - \ell}{f(b) - f(\ell)} f(\ell)$, ce qui implique $f(\ell) = 0$.
Conclusion : (a_k) converge vers la solution de $(f(x) = 0)$. □

La méthode de la sécante fournit l'encadrement $a_k \leq \ell \leq b$. Mais comme b est fixe cela ne donne pas d'information exploitable pour $|\ell - a_k|$. Voici une façon générale d'estimer l'erreur, à l'aide du théorème des accroissements finis.

Proposition 1.2

Soit $f : I \rightarrow \mathbb{R}$ une fonction dérivable et ℓ tel que $f(\ell) = 0$. S'il existe une constante $m > 0$ telle que pour tout $x \in I$, $|f'(x)| \geq m$ alors

$$\forall x \in I, \quad |x - \ell| \leq \frac{|f(x)|}{m}.$$

Démonstration. Par l'inégalité des accroissements finis entre x et ℓ : $|f(x) - f(\ell)| \geq m|x - \ell|$ mais $f(\ell) = 0$, d'où la majoration. □

L'algorithme correspond à la mise en œuvre de la suite récurrente (a_k) . Supposant connu un minorant $m = \inf_{t \in [a, b]} |f'(t)|$, la condition d'arrêt peut être simplement

$$|a_k - \ell| \leq \frac{|f(a_k)|}{m} \leq \varepsilon \implies |f(a_k)| \leq m\varepsilon$$

Le code suivant est une implémentation de la méthode de la fausse position avec cette condition d'arrêt :

```
def FaussePosition(f,a,b,m,epsilon):
    while abs(f(a))>m*epsilon:# supposant connu un minorant de f'
        a -= (b-a)/(f(b)-f(a))*f(a)
    return(a)
```

En pratique, dans les cas où il est difficile de calculer m , on peut utiliser une condition d'arrêt naïve du type $|f(a_k)| \leq \text{tol}$ (où tol correspond au produit $m\varepsilon$, ici inconnu) qui signifie grossièrement que $f(x_k)$ est « suffisamment proche » de 0. Le code suivant est une implémentation de la méthode de la fausse position avec la condition d'arrêt naïve $|f(a_k)| \leq \text{tol}$:

```
def FaussePosition(f,a,b,tol):
    while abs(f(a))>tol: # condition d'arrêt naïve
        a -= (b-a)/(f(b)-f(a))*f(a)
    return(a)
```

Une solution intermédiaire est de vérifier si, à l'itération k , a_k est effectivement une approximation à ε près de ℓ , c'est-à-dire si $\ell \in [a_k - \varepsilon, a_k + \varepsilon]$; ce que l'on traduit avec le théorème des valeurs intermédiaires par $f(a_k - \varepsilon) \times f(a_k + \varepsilon) < 0$. Il vient alors le code suivant :

```
def FaussePosition(f,a,b,epsilon):
    while f(a-epsilon)*f(a+epsilon)>0:
        a -= (b-a)/(f(b)-f(a))*f(a)
    return(a)
```

où l'on exclut (pour la forme) le cas $f(a_k - \varepsilon) \times f(a_k + \varepsilon) = 0$ qui signifie que $a_k - \varepsilon$ ou $a_k + \varepsilon$ est solution de $f(x) = 0$ (car le test d'égalité n'a pas beaucoup de sens sur les flottants...).

2 Méthodes de point fixe

Commençons par rappeler que l'inégalité des accroissements finis permet de montrer qu'une fonction g dérivable de dérivée bornée en norme par $k < 1$ sur I est contractante¹, ce qui signifie que l'ensemble des valeurs absolues des pentes de toutes ses cordes est borné par $k < 1$. Or, une fonction g qui est k -contractante admet un unique point fixe $x \in I$ tel que $g(x) = x$. Pour tout $x_0 \in I$, la suite récurrente $x_{n+1} = g(x_n)$ converge vers ce point fixe et vérifie

$$\forall n \in \mathbb{N}, |x_{n+1} - x| = |f(x_n) - f(x)| \leq k |x_n - x|$$

d'où, par récurrence simple, la majoration :

$$\forall n \in \mathbb{N}, |x_n - x| \leq k^n |x_0 - x|$$

qui montre une convergence géométrique vers $x \in I$. Maintenant, en remarquant que l'on peut toujours écrire l'équation $f(x) = 0$ sous la forme :

$$g(x) = x$$

en posant $g(x) = x - f(x)$, il est clair que rechercher un zéro de f revient à chercher un point fixe de g . Partant de la relation de récurrence $x_{k+1} = g(x_k)$, on peut montrer que la solution de $g(x) = x$ est le point d'intersection de la première bissectrice (droite d'équation $y = x$) et de la courbe d'équation $y = g(x)$. L'ordonnée $g(x_0)$ est la nouvelle abscisse x_1 : le point $(x_1, 0)$ se déduit du point $(x_0, g(x_0))$ par une symétrie orthogonale par rapport à la droite $y = x$. Cette construction se répète pour $x_2, x_3, \dots$. Partant de ce point, il existe deux types de situations :

- les points fixes attractifs (figure 5a) ;
- les points fixes répulsifs (figure 5b).

Pour assurer la convergence, il faut satisfaire et exploiter le résultat suivant :

1. g est une fonction k -lipschitzienne avec $k \in [0, 1[$. Une fonction $f : I \rightarrow \mathbb{R}$ est dite lipschitzienne de rapport $k \geq 0$ (ou k -lipschitzienne) sur l'intervalle I si, et seulement si :

$$\forall (x, y) \in I^2, |f(x) - f(y)| \leq k |x - y| .$$

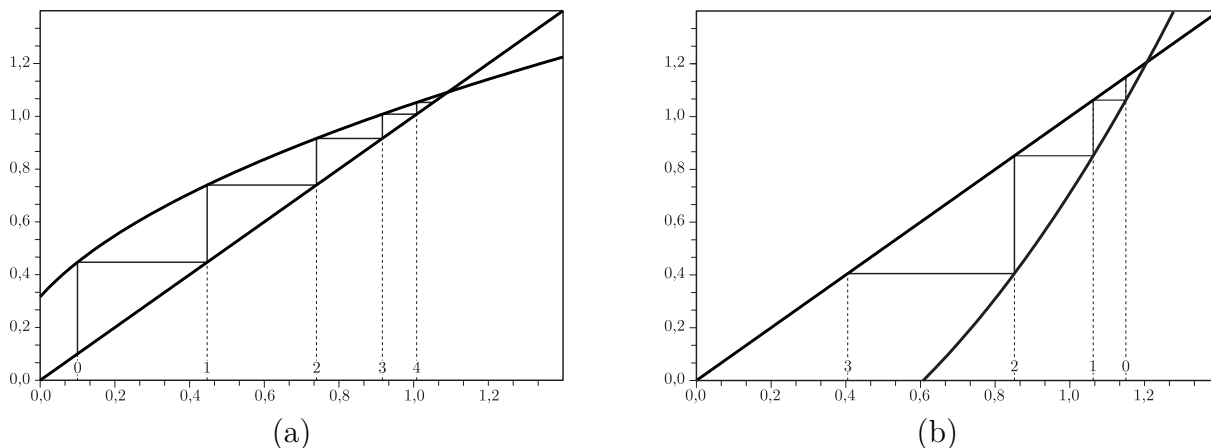


FIGURE 5 – Itérations convergentes (a) et divergentes (b) de points fixes.

Théorème 2.1

Soit $x_0 \in I$. Supposons qu'il existe un intervalle $[x_0 - \varepsilon, x_0 + \varepsilon] \subset I$ et un nombre $k \in]0, 1[$ tels que :

- $\forall x \in [x_0 - \varepsilon, x_0 + \varepsilon], |g'(x)| \leq k$;
- $|g(x_0) - x_0| \leq \varepsilon(1 - k)$.

Alors il existe un et un seul zéro $\ell \in [x_0 - \varepsilon, x_0 + \varepsilon]$ de l'équation $g(x) = x$. En outre, on peut définir par récurrence une suite (x_n) par la relation

$$\forall n \in \mathbb{N}, \quad x_{n+1} = g(x_n)$$

qui tend vers $\ell \in I$ et vérifiant à chaque itération :

$$|x_n - \ell| \leq \varepsilon k^n .$$

Comme l'erreur est multipliée par $k < 1$ à chaque itération, la convergence de la suite des (x_n) est géométrique. Comme on peut le voir sur la figure 5, un point fixe ℓ est attractif si on a $|f'(\ell)| < 1$. S'il vérifie au contraire $|f'(\ell)| > 1$, il est dit répulsif.

2.1 Méthode de la corde (de pente constante)

Beaucoup de méthodes de point fixe courantes font le choix de la forme suivante :

$$\forall x \in I, \quad g(x) = x - h(x)f(x) \quad \text{avec} \quad 0 < |h(x)| < +\infty$$

Sous cette hypothèse, on vérifie facilement que tout zéro de f est point fixe de g , et vice versa. Le choix le plus simple pour la fonction h est alors celui d'une fonction constante, ce qui conduit à la méthode de relaxation. Cette méthode consiste en la construction d'une suite $(x_k)_{k \in \mathbb{N}}$ satisfaisant la relation de récurrence :

$$\forall k \in \mathbb{N}, \quad x_{k+1} = x_k - \lambda f(x_k)$$

avec λ un réel fixé, la valeur de x_0 étant donnée.

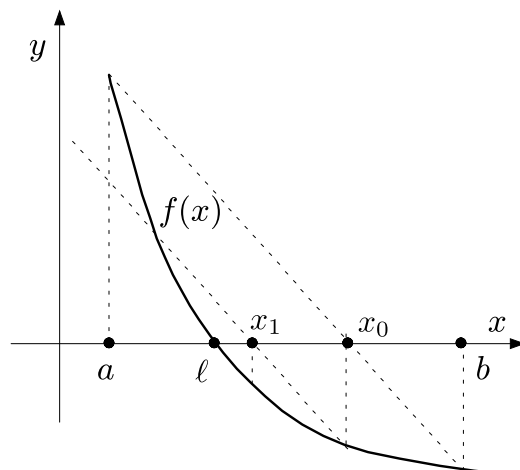


FIGURE 6 – Premières itérations de la méthode de la corde.

D'un point de vue géométrique, le point x_{k+1} correspond à l'abscisse du point d'intersection entre la droite de pente $1/\lambda$ passant par $(x_k, f(x_k))$ et l'axe des abscisses. Ainsi, connaissant un intervalle d'encadrement $[a, b]$, on définit la méthode de la corde (de pente constante) en prenant λ comme l'inverse de la pente de la corde, soit :

$$\forall k \in \mathbb{N}, \quad x_{k+1} = x_k - \frac{b-a}{f(b)-f(a)} f(x_k)$$

avec $x_0 \in [a, b]$ donné. En utilisant le théorème des valeurs intermédiaires, on peut vérifier à chaque itération k si $f(x_k - \varepsilon) \times f(x_k + \varepsilon) < 0$ auquel cas x_k est alors une approximation à ε près de ℓ . Il vient alors le code suivant :

```
def Corde(f,a,b,x0,epsilon):
    h=(b-a)/(f(b)-f(a))
    x=x0
    while f(x-epsilon)*f(x+epsilon)>=0:
        x-=h*f(x)
    return(x)
```

2.2 Méthode de la sécante

La méthode la sécante peut être considérée comme une variante de la méthode de la corde pour laquelle la pente de la corde est mise à jour à chaque itération. À partir de la donnée de deux variables x_{-1} et x_0 , telles que $x_{-1} \neq x_0$, la méthode de la sécante consiste en la relation de récurrence :

$$\forall k \in \mathbb{N}, \quad x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k) .$$

À chaque itération, le point $(x_k, 0)$ correspond à l'intersection de la droite passant par les points $(x_{k-1}, f(x_{k-1}))$ et $(x_k, f(x_k))$ de la courbe représentation de f avec l'axe des abscisses.

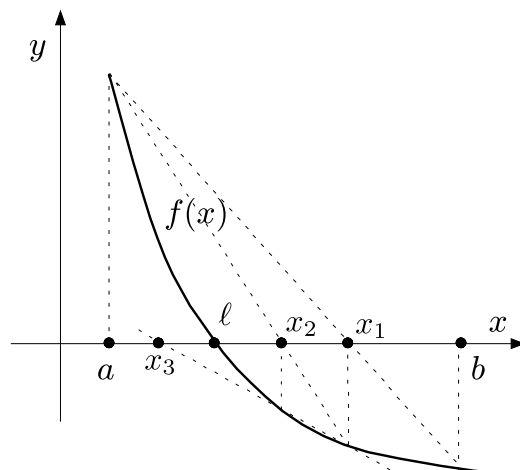


FIGURE 7 – Premières itérations de la méthode de la sécante.

En utilisant le théorème des valeurs intermédiaires, on peut vérifier à chaque itération k si $f(x_k - \varepsilon) \times f(x_k + \varepsilon) < 0$ auquel cas x_k est alors une approximation à ε près de ℓ . Il vient alors le code suivant :

```
def Secante(f,a,b,epsilon):
    while f(b-epsilon)*f(b+epsilon)>=0:
        a, b = b, b-(b-a)/(f(b)-f(a))*f(b)
    return(x)
```

2.3 Méthode de Newton

Considérons une fonction $f : [a, b] \rightarrow \mathbb{R}$ de classe $\mathcal{C}^1$ telle que f' ne s'annule pas sur $I = [a, b]$. Sa tangente en $x_0 \in [a, b]$ a pour équation :

$$y = f(x_0) + (x - x_0) f'(x_0)$$

On appelle $(x_1, 0)$ l'intersection de la tangente au graphe de f en $(x_0, f(x_0))$ avec l'axe des abscisses. L'abscisse x_1 est obtenue en prenant $y = 0$ dans l'équation, ce qui donne :

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Si $x_1 \in [a, b]$ alors on recommence l'opération avec la tangente au point d'abscisse x_1 . Ce processus conduit à la définition d'une suite récurrente :

$$x_0 \in [a, b] \quad \text{et} \quad \forall k \in \mathbb{N}, \quad x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

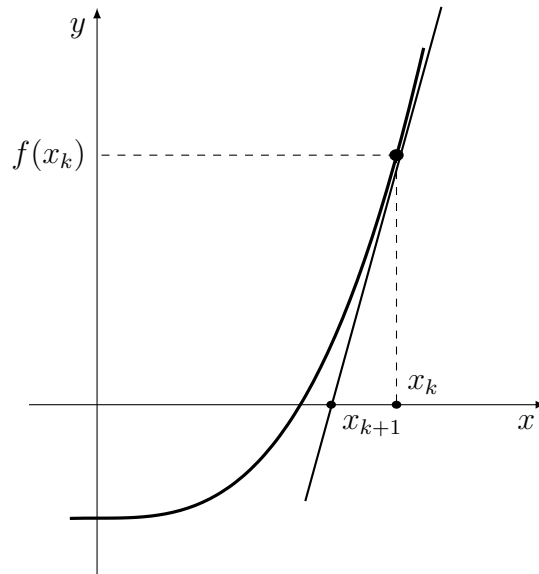


FIGURE 8 – Tangente au point $(x_k, f(x_k))$ permettant de définir x_{k+1} .

La méthode de Newton est une méthode de point fixe où au lieu de prendre la corde ou la sécante, on prend la tangente, telle que $h(x) = 1/f'(x)$. Ainsi, partant d'une valeur approchée suffisamment proche du zéro recherché, il s'agit de glisser le long de la tangente associée, jusqu'à rencontrer l'axes des abscisses, puis de recommencer... Elle est d'une redoutable efficacité.

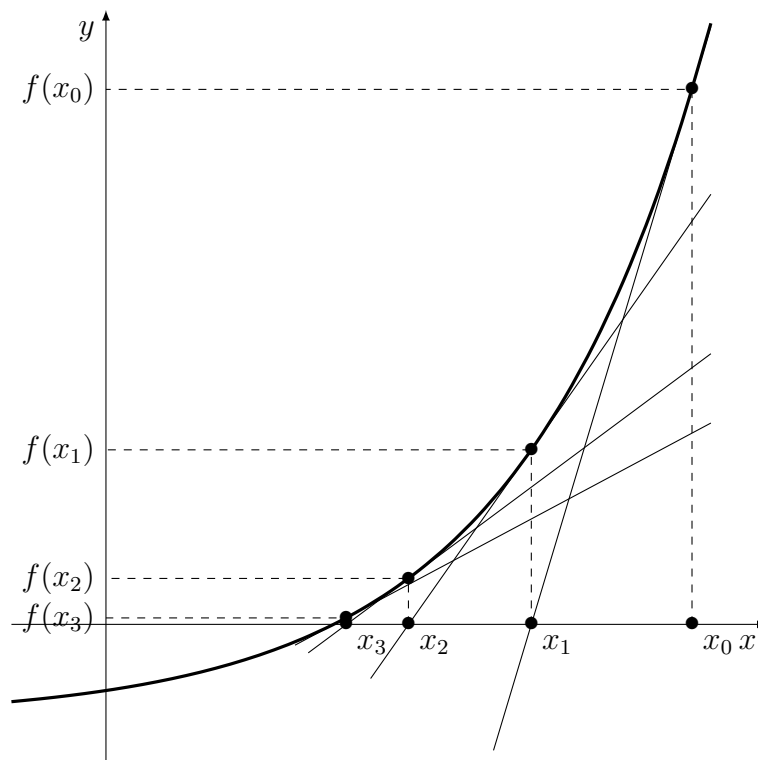


FIGURE 9 – Premières itérations de la méthode de Newton.

Reste alors la question de la condition d'arrêt. En effet, si l'on cherche une valeur approchée de $x \in [a, b]$ à $\varepsilon > 0$ près, quand doit-on arrêter l'algorithme ? Le théorème suivant nous donne la réponse, et précise un jeu de conditions sous lesquelles la méthode converge.

Théorème 2.2

Soit $f : [a, b] \rightarrow \mathbb{R}$ une fonction de classe $\mathcal{C}^2$ telle que :

- i) $f(a) < 0 < f(b)$, ii) $\forall x \in [a, b], f'(x) > 0$, iii) $\forall x \in [a, b], f''(x) > 0$.

Alors la suite définie par $x_0 = b$ et $\forall k \in \mathbb{N}, x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ converge vers un réel $\ell \in [a, b]$ tel que $f(\ell) = 0$, de sorte que

$$0 \leq x_{k+1} - \ell \leq \frac{M_2}{2m_1} (x_k - \ell)^2,$$

où m_1 est un minorant de f' et où M_2 est un majorant de f'' sur $[a, b]$.

Ce théorème montre que la convergence de la méthode de Newton est très rapide. En effet, si $\frac{M_2}{2m_1} \leq 1$ alors $\forall p \in \mathbb{N}, |x_k - \ell| \leq 10^{-p} \implies |x_{k+1} - \ell| \leq 10^{-2p}$; ce qui signifie que lorsque le nombre d'itérations passe de k à $k+1$, on double quasiment le nombre de décimales exactes. On parle alors de **convergence quadratique**.

En pratique, dans les cas où il est difficile de calculer m_1 et M_2 , on peut utiliser une condition d'arrêt naïve du type :

- $|f(x_k)| \leq \varepsilon'$ pour les courbes à fort gradient (on s'arrête lorsque $f(x_k)$ est « suffisamment proche » de 0), ce qui se traduit par le code suivant :

```
def Newton(f,df,x0,tol):
    x=x0
    while abs(f(x))>tol:
        x-=f(x)/df(x)
    return(x)
```

- $|x_{k+1} - x_k| \leq \varepsilon'$ pour les courbes à faible gradient (on s'arrête lorsque l'algorithme semble « stagner »), qui conduit au code suivant :

```
def Newton(f,df,x0,tol):
    x=x0-f(x0)/df(x0)
    while abs(x-x0)>tol:
        x0=x
        x-=f(x)/df(x)
    return(x)
```

Évidemment, l'inconvénient de ces deux conditions d'arrêt est que l'on perd le contrôle de la marge d'erreur (la valeur finale n'est pas une valeur approchée de ℓ à ε près), mais elles présentent l'avantage de simplifier le code et donnent des résultats assez satisfaisants si l'on

est pas trop exigeant sur la valeur approchée recherchée. Une solution plus raisonnable est de vérifier si, à l'itération k , a_k est effectivement une approximation à ε près de ℓ , c'est-à-dire si $\ell \in [a_k - \varepsilon, a_k + \varepsilon]$; ce que l'on traduit avec le théorème des valeurs intermédiaires par $f(a_k - \varepsilon) \times f(a_k + \varepsilon) < 0$. Il vient alors le code suivant :

```
def Newton(f,df,x0,epsilon):
    x=x0
    while f(x-epsilon)*f(x+epsilon)>0:
        x-=f(x)/df(x)
    return(x)
```

où l'on exclut (pour la forme) le cas $f(x - \varepsilon) \times f(x + \varepsilon) = 0$ qui signifie que $x - \varepsilon$ ou $x + \varepsilon$ est solution de $f(x) = 0$ (car le test d'égalité n'a pas beaucoup de sens sur les flottants...).

La méthode de Newton avec la condition d'arrêt naïve $|x_{k+1} - x_k| \leq \varepsilon'$ est implémentée dans le module `scipy` :

```
import scipy.optimize
x = scipy.optimize.newton(f,x0,df,tol)
```

où `tol` est le flottant associé à tolérance ε' .

3 Méthode de Newton par encadrement corde-tangente

L'utilisation conjointe des méthodes de Newton et de la corde pour encadrer une solution ℓ de $f(x) = 0$ est particulièrement simple lorsque f est convexe (ou concave).

Soit f une fonction convexe sur un intervalle ouvert $I = [a, b]$ telle que $f(a)f(b) < 0$. On peut supposer f croissante telle que $f(a) < 0$ et $f(b) > 0$. Le principe de calcul de cet encadrement corde-tangente est le suivant : on part de l'encadrement initial $a = a_0 < \ell < b = b_0$. On calcule ensuite b_1 en suivant la tangente à partir du point $(b_0, f(b_0))$ comme précédemment, et on calcule a_1 comme étant l'abscisse du point d'intersection de l'axe des abscisses et de la corde joignant les points $(a_0, f(a_0))$ et $(b_0, f(b_0))$ de la courbe représentative de f d'équation

$$y = f(a_0) + \frac{f(b_0) - f(a_0)}{b_0 - a_0} (x - a_0) .$$

L'abscisse a_1 est obtenue en prenant $y = 0$ dans l'équation, ce qui donne :

$$a_1 = a_0 - \frac{b_0 - a_0}{f(b_0) - f(a_0)} f(a_0) .$$

On recommence en partant maintenant de l'intervalle $[a_1, b_1]$. En itérant ce processus, on construit deux suites adjacentes $(a_k)_{k \in \mathbb{N}}$ et $(b_k)_{k \in \mathbb{N}}$, respectivement croissante et décroissante, vérifiant à chaque itération $\forall k \in \mathbb{N}$, $a_k \leq x \leq b_k$ et qui convergent vers l'unique solution sur $[a, b]$ de l'équation $f(x) = 0$.

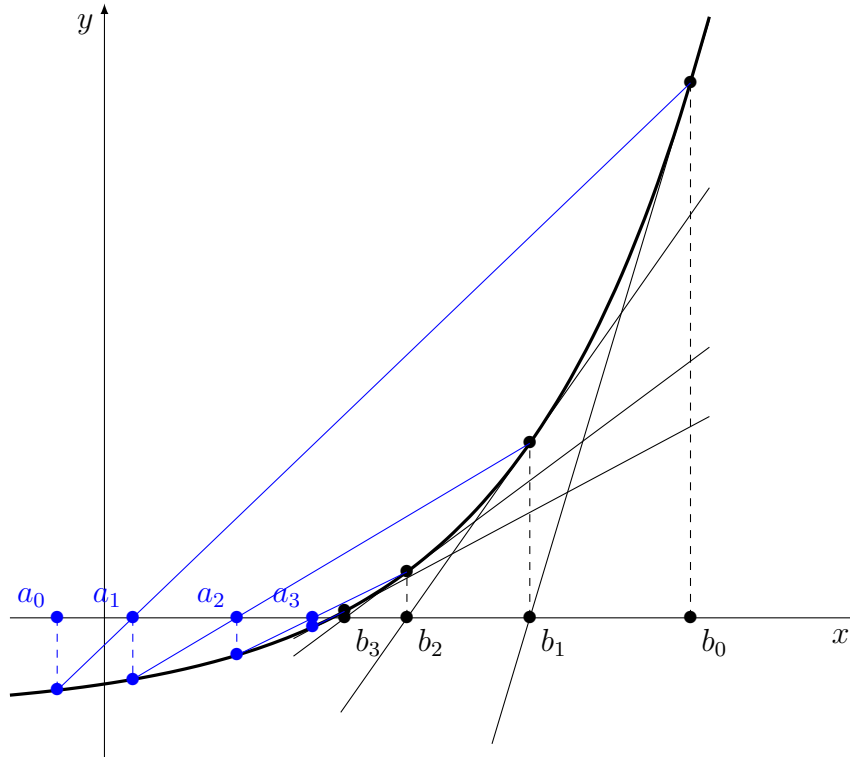


FIGURE 10 – Premières itérations de la méthode de Newton par encadrement corde-tangente.

Pour appliquer la méthode de Newton par encadrement corde-tangente, il est nécessaire de prendre quelques précautions (figure 11) :

- avec f convexe et croissante, on doit considérer la tangente au point $(b, f(b))$. La tangente au point $(a, f(a))$, $f(a) < 0$, rencontre l'axe des abscisses en un point dont l'abscisse n'appartient pas en général à $[a, b]$;
- avec f convexe décroissante, on doit considérer la tangente au point $(a, f(a))$;
- avec f concave croissante, on doit considérer la tangente au point $(a, f(a))$;
- avec f concave décroissante, on doit considérer la tangente au point $(b, f(b))$.

On peut remarquer que le choix du point $(b, f(b))$ pour la tangente est commun aux fonctions convexes croissantes ou concaves décroissantes pour lesquelles, $\forall x \in [a, b]$, on a $f'(x) \times f''(x) > 0$. De même, le choix du point $(a, f(a))$ pour la tangente est commun aux fonctions convexes décroissantes ou concaves croissantes pour lesquelles on a cette fois, $\forall x \in [a, b]$, on a $f'(x) \times f''(x) < 0$. Ainsi, le choix du point où utiliser la tangente peut se résumer à :

$$b_0 = \begin{cases} b & \text{si } f'(b) \times f''(b) > 0 \\ a & \text{sinon} \end{cases}$$

l'autre point servant évidemment pour la corde.

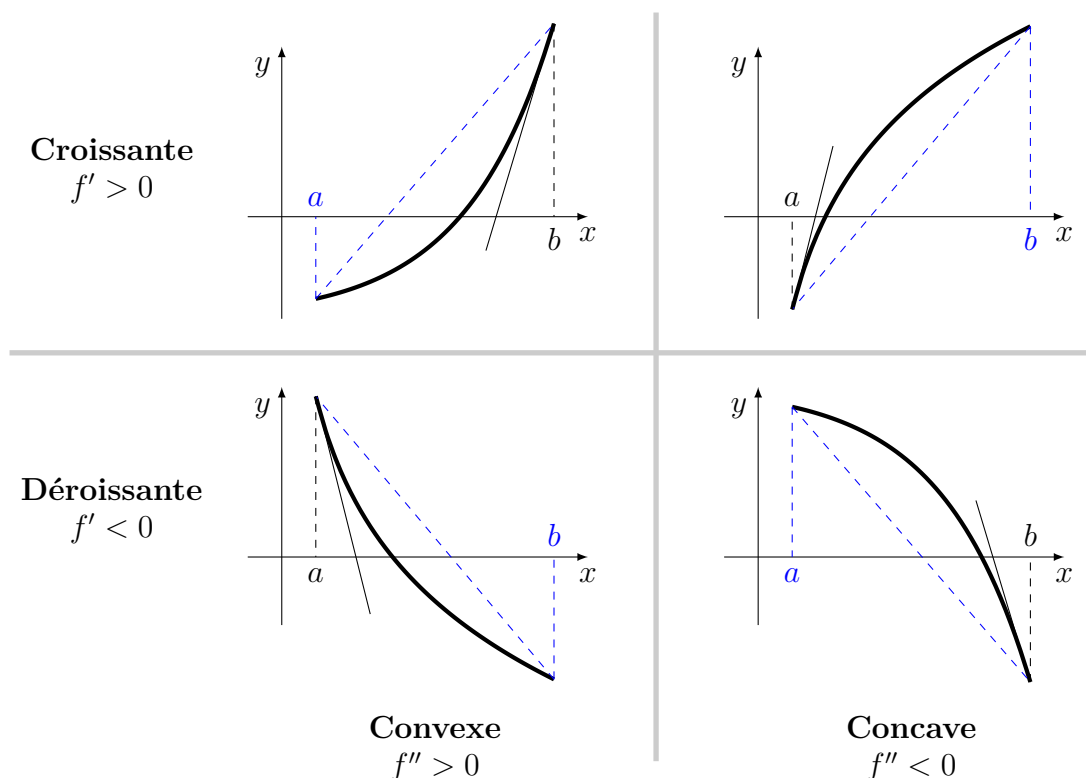


FIGURE 11 – Conditions d’applications de la méthode de Newton par encadrement corde-tangente aux fonctions convexes ou concaves, croissantes ou décroissantes.

Il n’est pas toujours facile d’avoir une expression de la dérivée seconde d’une fonction f pour vérifier si elle est convexe sur un intervalle $[a, b]$. Pour l’évaluer, on peut vérifier si le milieu de la corde

$$f(a) + \frac{f(b) - f(a)}{b - a} \left(\frac{a + b}{2} - a \right) = \frac{f(a) + f(b)}{2}$$

est effectivement au-dessus de l’image du milieu $f\left(\frac{a + b}{2}\right)$; ce que l’on peut traduire par :

$$f(a) + f(b) - 2f\left(\frac{a + b}{2}\right) > 0 \implies f'' > 0 .$$

Ainsi, le choix du point où utiliser la tangente peut être évalué par :

$$b_0 = \begin{cases} b & \text{si } f'(b) \times \left[f(a) + f(b) - 2f\left(\frac{a + b}{2}\right) \right] > 0 \\ a & \text{sinon} \end{cases}$$

l’autre point servant pour la corde.

MÉMO – Méthode de Newton par encadrement corde-tangente

L'objectif de la méthode de Newton par encadrement corde-tangente est de trouver une solution $x \in [a, b]$ tel que $f(x) = 0$, si un tel réel existe. Le principe de cette méthode est de construire deux suites $(a_k)_{k \in \mathbb{N}}$ et $(b_k)_{k \in \mathbb{N}}$ vérifiant $\forall k \in \mathbb{N}, a_k \leq x \leq b_k$. En partant des deux extrémités de l'intervalle $a_0 = a$ et $b_0 = b$ (sous l'hypothèse que l'on ait $f' f'' > 0$ sur l'intervalle $[a, b]$, sinon on inverse les bornes), on construit ces suites avec les relations de récurrence :

$$\forall k \in \mathbb{N}, \quad \begin{cases} a_{k+1} = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)} \\ b_{k+1} = b_k - \frac{f(b_k)}{f'(b_k)} \end{cases}$$

où f' est la dérivée de la fonction f dont on cherche un zéro. Pour obtenir une valeur approchée de x à ε près, il suffit de s'arrêter lorsque $b_k - a_k < \varepsilon$: a_k et b_k sont alors une valeur approchée de x recherchée.

Le code Python s'écrit simplement :

```
def Newton(f,df,a,b,epsilon):
    if (df(b)*(f(a)+f(b)-2*f((a+b)/2))<0):
        a,b=b,a
    while abs(b-a)>epsilon:
        a=(a*f(b)-b*f(a))/(f(b)-f(a))
        b-=f(b)/df(b)
    return(a)
```

* *

*