

Résolution numérique des équations

1. Introduction

1.1 Présentation du problème

Dans ce chapitre nous nous intéressons à l'approximation des zéros d'une fonction réelle d'une variable réelle. Autrement dit, étant donné un intervalle I de \mathbb{R} et une application $f : I \rightarrow \mathbb{R}$, on cherche à trouver au moins une valeur approchée de $c \in I$ (s'il en existe) tel que $f(c) = 0$.

Toutes les méthodes que nous allons présenter sont itératives et consistent donc en la construction d'une suite $(x_n)_{n \in \mathbb{N}}$ telle que, on l'espère, $\lim x_n = c$. Nous verrons que la convergence de ces méthodes itératives dépend en général du choix de la donnée initiale x_0 . Ainsi, on ne sait le plus souvent qu'établir des résultats de convergence *locale*, valables lorsque x_0 appartient à un certain voisinage de c .

Remarque. Le cas des équations algébriques, c'est-à-dire de la forme $p(x) = 0$ où p est une fonction polynomiale, est particulier : les polynômes possèdent un certain nombre de propriétés qui les distinguent des fonctions quelconques. C'est la raison pour laquelle il existe des méthodes dédiées aux seuls polynômes. Ces méthodes ne seront pas abordées dans ce chapitre.

1.2 Ordre de convergence d'une méthode itérative

La vitesse de convergence est un facteur important de la qualité des algorithmes ; si la vitesse de convergence est élevée, l'algorithme converge rapidement et le temps de calcul est moindre. Ces préoccupations de rapidité de convergence ont conduit à adopter les définitions suivantes.

DÉFINITION. — Soit $(x_n)_{n \in \mathbb{N}}$ une suite qui converge vers une limite c . On dit que la convergence de $(x_n)_{n \in \mathbb{N}}$ vers c est d'ordre $r \geq 1$ lorsqu'il existe une suite $(e_n)_{n \in \mathbb{N}}$ qui converge vers 0, majore l'erreur commise : $\forall n \in \mathbb{N}, |x_n - c| \leq e_n$ et vérifie :

$$\lim \frac{e_{n+1}}{e_n^r} = \mu > 0.$$

Une méthode itérative qui fournit en général des suites dont la convergence est d'ordre r sera elle-même dite d'ordre r .

Remarque. Lorsque $r = 1$ on a nécessairement $\mu \leq 1$. En effet, si on avait $\mu > 1$ la suite $(e_n)_{n \in \mathbb{N}}$ serait strictement croissante (à partir d'un certain rang) et ne pourrait converger vers 0.

THÉORÈME. — La méthode dichotomique est une méthode de résolution numérique d'ordre 1.

Preuve. Dans le cas de la recherche d'une racine c dans l'intervalle $[a, b]$ il a été établi au chapitre précédent que $|x_n - c| \leq \frac{b-a}{2^n}$. Posons $e_n = \frac{b-a}{2^n}$. Alors $\lim \frac{e_{n+1}}{e_n} = \frac{1}{2}$. □

• Lien entre ordre de convergence et nombre de décimales exactes

Lorsqu'on pose $\delta_n = -\log_{10}(e_n)$ on a $\lim \delta_{n+1} - r\delta_n = -\log_{10}(\mu)$ donc pour n assez grand $\delta_{n+1} \approx r\delta_n - \log_{10}(\mu)$.

δ_n mesure approximativement le nombre de décimales exactes de l'approximation x_n de c : si $\delta_n = p$ alors $e_n = 10^{-p}$ et x_n et c partagent les mêmes p premières décimales.

Si $r = 1$ et $\mu < 1$, l'approximation $\delta_{n+1} \approx \delta_n - \log_{10}(\mu)$ traduit le fait qu'à partir d'un certain rang, le nombre de décimales exactes augmente linéairement avec n (ce qui explique pourquoi les méthodes d'ordre 1 sont souvent qualifiées de *linéaires*). Plus précisément, à chaque étape le nombre de décimales exactes est augmenté de $-\log_{10}(\mu)$. Par exemple, dans le cas de la méthode dichotomique on a $\mu = 1/2$ et $-\log_{10}(1/2) \approx 0,3$: toutes les trois itérations le nombre de décimales exactes est augmenté de 1.

Remarque. On peut observer que le cas $\mu = 1$ ne permet pas de tirer de telles conclusions. Ces méthodes sont dites *sous-linéaires* et sont en général très lentes.

Lorsque $r = 2$ l'approximation se traduit par $\delta_{n+1} \approx 2\delta_n$: à partir d'un certain rang le nombre de décimales est doublé à chaque étape. De telles méthodes sont dites *quadratiques* ; leurs vitesses de convergence sont bien supérieures à celles des méthodes linéaires.

Plus généralement, lorsque $r > 1$ le nombre de décimales exactes est à partir d'un certain rang multiplié par r à chaque étape. On a donc tout à gagner à trouver des méthodes d'ordre le plus élevé possible.

1.3 Critères d'arrêt

En cas de convergence la suite $(x_n)_{n \in \mathbb{N}}$ construite par une méthode itérative converge vers c . Pour l'utilisation pratique d'une telle méthode il faut introduire un *critère d'arrêt* pour interrompre le processus itératif lorsque l'approximation de c par x_n est jugée « satisfaisante ». Pour cela, plusieurs choix sont possibles :

- on peut imposer un nombre maximal d'itérations ;
- on peut imposer une tolérance $\varepsilon > 0$ sur l'incrément : dans ce cas les itérations s'achèvent lorsque $|x_{n+1} - x_n| \leq \varepsilon$;
- on peut imposer une tolérance $\varepsilon > 0$ sur le résidu : dans ce cas les itérations s'achèvent lorsque $|f(x_n)| \leq \varepsilon$.

Selon les cas, chacun de ces critères peut s'avérer soit trop restrictif, soit trop optimiste.

2. Principales méthodes de résolution numérique

2.1 Méthode dichotomique

Nous avons étudié dans le chapitre précédent la méthode dichotomique. Celle-ci est une méthode qualifiée de *robuste* dans le sens où la suite construite $(x_n)_{n \in \mathbb{N}}$ converge à coup sûr vers un zéro c de la fonction f . En revanche c'est une méthode plutôt lente : c'est une méthode d'ordre 1, et il faut environ trois itérations pour gagner une décimale. Cette méthode sera le plus souvent utilisée pour obtenir une première approximation *raisonnable* de c avant de poursuivre par une méthode plus rapide mais dont la convergence est uniquement locale.

2.2 Méthode de la fausse position

Appelée encore méthode *regula falsi*, il s'agit d'une méthode d'encadrement de c combinant les possibilités de la méthode de dichotomie avec la méthode de la sécante qui sera introduite plus loin.

Nous allons comme pour la méthode dichotomique considérer une fonction continue $f : [a, b] \rightarrow \mathbb{R}$ vérifiant $f(a)f(b) \leq 0$, ce qui assure l'existence d'un zéro au moins dans l'intervalle $]a, b[$.

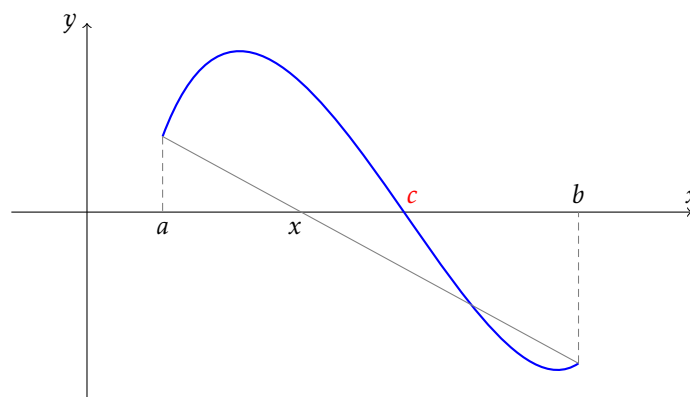


FIGURE 1 – Méthode de la fausse position. La recherche se poursuit dans l'intervalle $[x, b]$.

L'idée est d'utiliser l'information fournie par les valeurs de la fonction f aux bornes de l'intervalle $[a, b]$ pour essayer d'approcher plus finement c que par le calcul du point milieu dans la méthode dichotomique.

Concrètement, cela revient à calculer l'abscisse x de point d'intersection de la corde reliant les points de coordonnées $(a, f(a))$ et $(b, f(b))$ avec l'axe des abscisses.

L'une des deux conditions $f(a)f(x) \leq 0$ ou $f(x)f(b) \leq 0$ est nécessairement vérifiée ; la recherche se poursuit donc dans l'intervalle $[a, x]$ ou $[x, b]$ suivant les cas.

La corde en question a pour équation : $y = \frac{f(b) - f(a)}{b - a}(x - a) + f(a)$ donc l'abscisse x du point d'intersection avec l'axe des abscisses vérifie :

$$\frac{f(b) - f(a)}{b - a}(x - a) + f(a) = 0 \iff x = \frac{af(b) - bf(a)}{f(b) - f(a)}.$$

L'algorithme de recherche consiste donc à itérer deux suites $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ définies par les valeurs initiales $u_0 = a$ et $v_0 = b$ et la relation de récurrence :

$$(u_{n+1}, v_{n+1}) = \begin{cases} (u_n, x_n) & \text{si } f(u_n)f(x_n) \leq 0 \\ (x_n, v_n) & \text{sinon} \end{cases} \quad \text{avec } x_n = \frac{u_n f(v_n) - v_n f(u_n)}{f(v_n) - f(u_n)}$$

Attention, la quantité $v_n - u_n$ (la largeur de l'intervalle de recherche) est une suite décroissante, mais contrairement à la méthode dichotomique nous n'avons plus ici la garantie que cette suite tend vers 0. C'est le cas en particulier lorsque la fonction est convexe ou concave. Dans ces cas de figure l'une des deux bornes reste constante alors que l'autre converge de façon monotone vers c (illustration figure 2).

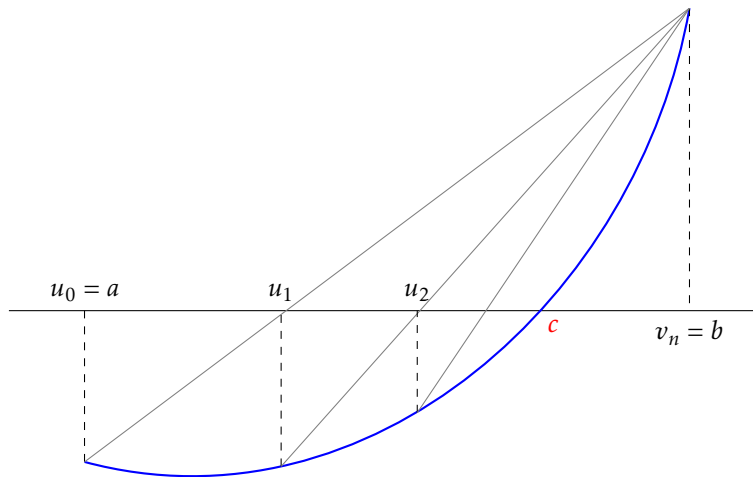


FIGURE 2 – Cas d'une fonction convexe telle que $f(a) < 0 < f(b)$: la suite $(u_n)_{n \in \mathbb{N}}$ est croissante et converge vers c ; la suite $(v_n)_{n \in \mathbb{N}}$ est constante.

Il est donc impératif que pour le critère d'arrêt soit basé sur la valeur du résidu $f(x_n)$: puisque x_n converge vers un zéro c (voir le théorème ci-dessous) et que f est continue, on a $\lim f(x_n) = 0$ et pour tout $\varepsilon > 0$ il existe un rang n à partir duquel $|f(x_n)| \leq \varepsilon$.

THÉORÈME. — Soit $f : [a, b] \rightarrow \mathbb{R}$ une fonction de classe \mathcal{C}^1 vérifiant $f(a)f(b) < 0$ et strictement convexe (respectivement concave). Alors la suite $(x_n)_{n \in \mathbb{N}}$ construite par la méthode de la fausse position converge vers l'unique zéro c de f . De plus, la convergence est linéaire (d'ordre 1).

Preuve. Supposons par exemple f strictement convexe et $f(a) < 0 < f(b)$ (les autres cas se traitent de la même façon), situation représentée figure 2.

Les fonctions convexes ont pour propriété d'avoir leurs cordes situées au dessus de leur graphe. Le point d'intersection x_n de la corde reliant les points d'abscisses u_n et v_n est donc situé au dessus du point d'abscisse x_n de la courbe, ce qui montre que $f(x_n) < 0$. Ainsi, $u_{n+1} = x_n$ et $v_{n+1} = v_n$. La suite $(v_n)_{n \in \mathbb{N}}$ est constante égale à b et la suite $(u_n)_{n \in \mathbb{N}}$ croissante et majorée par b , donc convergente. Notons c sa limite.

On a $u_{n+1} = g(u_n)$ avec $g : x \mapsto \frac{xf(b) - bf(x)}{f(b) - f(x)} = x - f(x) \frac{b - x}{f(b) - f(x)}$ et par passage à la limite $c = g(c)$, ce qui implique $f(c) = 0$. Nous avons bien prouvé la convergence de la méthode de la fausse position.

Pour montrer que cette méthode est d'ordre 1 il faut encore prouver que $\lim \frac{u_{n+1} - c}{u_n - c} = \mu > 0$. Pour ce faire on

observe que : $\frac{u_{n+1} - c}{u_n - c} = \frac{g(u_n) - g(c)}{u_n - c}$ donc $\lim \frac{u_{n+1} - c}{u_n - c} = g'(c)$.

On calcule $g'(x) = f(b) \frac{f(b) - f(x) - (b-x)f'(x)}{(f(b) - f(x))^2}$ donc $g'(c) = \frac{f(b) - (b-c)f'(c)}{f(b)}$.

Les fonctions convexes ont pour propriété d'avoir leur graphe situé au dessus de leurs tangentes, donc en particulier : $f(b) > f(c) + (b-c)f'(c) = (b-c)f'(c)$, ce qui montre que $g'(c) > 0$. La convergence est bien linéaire. \square

Remarque. Lorsque f est de classe \mathcal{C}^2 on aura le plus souvent $f''(c) \neq 0$, ce qui assure que f est strictement convexe ou concave au voisinage de c . Dans ce cas, le théorème précédent assure que l'une des deux suites $(u_n)_{n \in \mathbb{N}}$ ou $(v_n)_{n \in \mathbb{N}}$ est stationnaire (c'est-à-dire constante à partir d'un certain rang) et que la convergence est linéaire. Cette méthode ne présente donc pas de gain notable comparativement à la méthode dichotomique.

2.3 Méthode de NEWTON-RAPHSON

Les deux méthodes déjà étudiées : dichotomie et fausse position, déterminent un zéro de f en lequel se produit un changement de signe en construisant une suite décroissante d'intervalles contenant ce dernier. La méthode de NEWTON-RAPHSON s'affranchit de cette contrainte, en procédant par approximations successives pour construire une suite $(x_n)_{n \in \mathbb{N}}$ qui converge en général vers un zéro de f : l'équation $f(x) = 0$ est remplacée par l'équation affine $\varphi_n(x) = 0$, où φ_n est l'équation de la tangente à la fonction f au point d'abscisse x_n .

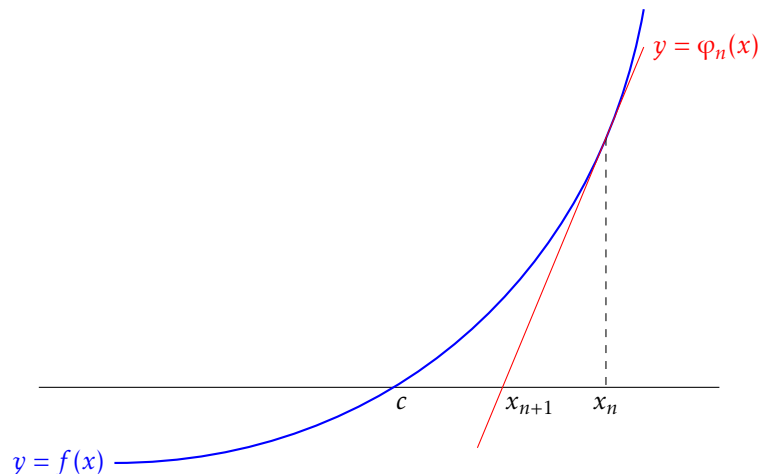


FIGURE 3 – Le principe de la méthode de NEWTON-RAPHSON.

Cette tangente a pour équation $y = f(x_n) + f'(x_n)(x - x_n)$ donc $\varphi_n(x_{n+1}) = 0 \iff x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$.

La méthode de NEWTON-RAPHSON consiste donc en l'itération d'une suite $(x_n)_{n \in \mathbb{N}}$ définie par la donnée d'une valeur x_0 et de la relation de récurrence :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

THÉORÈME. — Soit f une fonction de classe \mathcal{C}^2 définie au voisinage d'un point c pour lequel $f(c) = 0$ et $f'(c) \neq 0$. Alors il existe un voisinage \mathcal{V} de c pour lequel, quel que soit $x_0 \in \mathcal{V}$ la suite $(x_n)_{n \in \mathbb{N}}$ converge vers c . En outre, la convergence est quadratique (c'est-à-dire d'ordre 2).

Preuve. Quitte à remplacer f par $-f$ on suppose $f'(c) > 0$. La fonction f est donc strictement croissante au voisinage de c , ce qui nous autorise à considérer un intervalle $[a, b]$ sur lequel $f' > 0$ et pour lequel $f(a) < 0 < f(b)$.

Posons pour tout $x \in [a, b]$: $\varphi(x) = x - \frac{f(x)}{f'(x)}$. La fonction φ est de classe \mathcal{C}^1 sur $[a, b]$ et $f(x) = 0 \iff \varphi(x) = x$.

Alors pour tout $x \in [a, b]$,

$$\varphi(x) - c = x - c - \frac{f(x)}{f'(x)} = x - c - \frac{f(x) - f(c)}{f'(x)} = \frac{f(c) - f(x) - (c-x)f'(x)}{f'(x)}.$$

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

D'après l'inégalité de TAYLOR-LAGRANGE, $|f(c) - f(x) - (c-x)f'(x)| \leq \frac{M_2}{2}(c-x)^2$ donc :

$$|\varphi(x) - c| \leq \frac{M_2}{2m_1}(x-c)^2.$$

Posons maintenant $K = \frac{M_2}{2m_1}$ et choisissons un réel $\eta > 0$ assez petit pour que $K\eta < 1$ et $[c-\eta, c+\eta] \subset [a, b]$.

Pour tout $x \in [c-\eta, c+\eta]$, $|\varphi(x) - c| \leq K\eta^2 \leq \eta$ donc $\varphi(x) \in [c-\eta, c+\eta]$. Ceci prouve que si on choisit $x_0 \in [c-\eta, c+\eta]$ la suite $(x_n)_{n \in \mathbb{N}}$ est bien définie par la relation de récurrence $x_{n+1} = \varphi(x_n)$ et vérifie : $\forall n \in \mathbb{N}, x_n \in [c-\eta, c+\eta]$.

De plus, $|x_{n+1} - c| \leq K(x_n - c)^2$ et il est dès lors facile d'établir par récurrence que $|x_n - c| \leq \frac{1}{K}(K|x_0 - c|)^{2^n}$.

Puisque $K|x_0 - c| \leq K\eta < 1$ ceci montre que $\lim x_n = c$.

Enfin, en posant $e_n = \frac{1}{K}(K|x_0 - c|)^{2^n}$ on a $\lim \frac{e_{n+1}}{e_n^2} = K > 0$ donc la convergence est bien quadratique. \square

Remarque. Ce résultat montre que la méthode de NEWTON-RAPHSON s'avère particulièrement efficace (c'est une méthode d'ordre 2) lorsqu'on connaît déjà une première approximation de c qu'on cherche à affiner. En revanche, lorsque x_0 n'est pas suffisamment proche de c cette méthode n'offre aucune garantie de convergence. En outre, lorsque $f'(c) = 0$ la convergence ne peut être garantie, même pour des valeurs très proches de c : le terme $\frac{f(x)}{f'(x)}$, quotient de deux quantités très faibles, engendre des phénomènes de *cancellation* très importants.

• Cas d'une fonction convexe

Le théorème précédent nous assure uniquement d'une convergence *locale* ; dans le cas particulier des fonctions convexes on dispose du résultat de convergence *globale* suivant :

THÉORÈME. — Soit $f : [a, b] \rightarrow \mathbb{R}$ une fonction de classe \mathcal{C}^1 , strictement croissante et convexe, telle que $f(a) < 0 < f(b)$. Alors la suite définie par $f(x_0) > 0$ et la relation $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ existe et converge quadratiquement vers l'unique zéro c de f sur $]a, b[$.

Preuve. Une fonction convexe a la propriété d'avoir son graphe situé au dessus de ses tangentes, en conséquence de quoi, si $c < x_n < b$ alors $c < x_{n+1} < x_n$. Ceci prouve que la suite $(x_n)_{n \in \mathbb{N}}$ est bien définie par le choix d'une valeur x_0 vérifiant $c < x_0 < b$ et que cette suite est décroissante et minorée par c . Elle possède donc une limite qui est un point fixe de la fonction φ . Mais $\varphi(x) = x \iff f(x) = 0$ donc $(x_n)_{n \in \mathbb{N}}$ converge vers c .

Enfin, avec les notations du théorème précédent, il existe un rang N à partir duquel $x_n \in \mathcal{V}$, ce qui assure la convergence quadratique de la suite. \square

Application : méthode de HÉRON pour calculer une racine carrée

La méthode de HÉRON pour calculer une valeur approchée de $\sqrt{\alpha}$ lorsque α est un réel strictement positif consiste à appliquer la méthode de NEWTON-RAPHSON à la fonction $f : x \mapsto x^2 - \alpha$ à partir de la valeur initiale $x_0 = \alpha + 1$. On a $f(x_0) = \alpha^2 + \alpha + 1 > 0$ donc le théorème précédent assure la convergence quadratique de la méthode. Celle-ci consiste donc à itérer la suite définie par :

$$x_0 = \alpha + 1 \quad \text{et} \quad x_{n+1} = \frac{1}{2} \left(x_n + \frac{\alpha}{x_n} \right).$$

2.4 Dérivée numérique

Dans une mise en œuvre pratique de la méthode de NEWTON-RAPHSON se pose le problème du calcul de la dérivée $f'(x)$ en un point. Une solution consiste bien entendu à effectuer le calcul de la dérivée à la main (ou à l'aide d'un logiciel de calcul formel) mais ce n'est pas toujours chose facile lorsque la fonction f est compliquée.

Une autre solution, que nous allons explorer, consiste à calculer une valeur approchée de $f'(x)$.

Nous allons partir d'une idée simple : approcher $f'(x)$ par la quantité $\frac{f(x+h) - f(x)}{h}$ où h est une quantité très petite. Intuitivement, plus la valeur de h est petite, meilleure est l'approximation. Nous allons constater qu'avec un ordinateur *ceci n'est pas forcément vérifié*.

Pour fixer les idées, supposons qu'on travaille avec un ordinateur qui utilise une représentation décimale à trois chiffres significatifs et que l'on souhaite calculer à l'aide de cette formule une valeur approchée de $f'(7)$ avec $f : x \mapsto x^2$.

- Si on prend $h = 0,1$ on approche $f'(7) = 14$ par :

$$\frac{7,1^2 - 7^2}{0,1} = \frac{50,4 - 49}{0,1} = \frac{1,4}{0,1} = 14,0 \quad \text{car } 7,1^2 = 50,41 \text{ est approché par } 50,4.$$

- Si on prend $h = 0,01$ on approche $f'(7) = 14$ par :

$$\frac{7,01^2 - 7^2}{0,01} = \frac{49,1 - 49}{0,01} = \frac{0,1}{0,01} = 10,0 \quad \text{car } 7,01^2 = 49,1401 \text{ est approché par } 49,1.$$

On aboutit à une situation paradoxale où c'est la valeur la plus grande de h qui donne la meilleure approximation de la dérivée !

On comprend que pour choisir la valeur optimale de h il va falloir tenir compte de la représentation des nombres en machine.

Nous savons que les nombres flottants sont représentés en machine en base 2 avec une mantisse de 52 bits. Posons $\varepsilon = 2^{-52}$; il s'agit de la précision relative de la machine (*l'epsilon numérique*).

L'erreur absolue sur l'évaluation d'une fonction f en un point x est de l'ordre de $\varepsilon|f(x)|$; on peut donc estimer l'erreur sur le numérateur par :

$$\varepsilon|f(x+h)| + \varepsilon|f(x)| \approx 2\varepsilon|f(x)|.$$

L'erreur absolue sur le quotient qui approche la dérivée peut donc être estimée par $2\varepsilon \left| \frac{f(x)}{h} \right|$.

Il faut maintenant évaluer l'erreur que l'on commet en approchant $f'(x)$ par le quotient $\frac{f(x+h) - f(x)}{h}$. Pour cela, on utilise la formule de TAYLOR :

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + o(h^2)$$

qui, lorsque $f''(x) \neq 0$, fournit l'équivalent : $\left| \frac{f(x+h) - f(x)}{h} - f'(x) \right| \sim \frac{|h|}{2}|f''(x)|$.

L'erreur totale de l'approximation de $f'(x)$ par $\frac{f(x+h) - f(x)}{h}$ peut donc être évaluée par :

$$E(h) = 2\varepsilon \frac{|f(x)|}{|h|} + \frac{|h|}{2}|f''(x)|.$$

Une étude de cette fonction sur $[0, +\infty[$ montre que l'erreur est minimale pour $h_0 = 2\sqrt{\varepsilon \frac{|f(x)|}{|f''(x)|}}$.

h	0	h_0	$+\infty$
$E'(h)$	-	0	+
$E(h)$	$+\infty$	$E(h_0)$	$+\infty$

Dans des domaines dans lesquels le quotient $\frac{|f(x)|}{|f''(x)|}$ reste borné on choisit en général $h \approx \sqrt{\varepsilon}$ soit $\boxed{h = 10^{-8}}$.

Remarque. L'erreur mathématique peut être améliorée en observant que :

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f^{(3)}(x) + o(h^2) \quad \text{et} \quad f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f^{(3)}(x) + o(h^2)$$

donc il est possible d'approcher $f'(x)$ par $\frac{f(x+h) - f(x-h)}{2h}$ et dans ce cas l'erreur commise vaut :

$$\left| \frac{f(x+h) - f(x-h)}{2h} - f'(x) \right| \sim \frac{h^2}{3} |f^{(3)}(x)|.$$

Dans ce cas, l'erreur totale d'approximation vaut :

$$E(h) = 2\varepsilon \frac{|f(x)|}{|h|} + \frac{h^2}{3} |f^{(3)}(x)|.$$

et celle-ci est minimale pour $h_0 = \sqrt[3]{\frac{3\varepsilon|f(x)|}{|f^{(3)}(x)|}}$.

Lorsqu'on choisit d'approcher $f'(x)$ par $\frac{f(x+h) - f(x-h)}{2h}$ on choisit en général $h \approx \sqrt[3]{\varepsilon}$ soit $h = 10^{-5}$.

2.5 Méthode de la sécante

Une autre solution si on ne souhaite pas à avoir à se préoccuper du calcul de la dérivée dans la méthode de NEWTON-RAPHSON consiste à remplacer cette dernière par la pente de la corde reliant les points d'abscisses x_{n-1} et x_{n-2} . Autrement dit, la relation :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

est remplacée par la relation :

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$$

qui peut aussi s'écrire $x_{n+1} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$.

Le point de coordonnées $(x_{n+1}, 0)$ apparaît maintenant comme l'intersection avec l'axe des abscisses de la corde reliant les points de coordonnées $(x_{n-1}, f(x_{n-1}))$ et $(x_n, f(x_n))$.

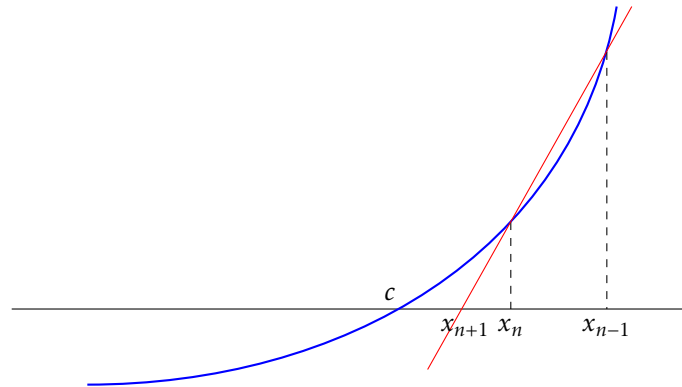


FIGURE 4 – Le principe de la méthode de la sécante.

Notons que cette méthode nécessite le choix de deux valeurs initiales x_0 et x_1 .

Nous admettrons le résultat suivant :

THÉORÈME. — Soit f une fonction de classe \mathcal{C}^2 définie au voisinage d'un point c pour lequel $f(c) = 0$ et $f'(c) \neq 0$. Alors il existe un voisinage \mathcal{V} de c pour lequel, quel que soit $(x_0, x_1) \in \mathcal{V}^2$ vérifiant $x_0 \neq x_1$ la suite $(x_n)_{n \in \mathbb{N}}$ converge vers c . En outre, la convergence est d'ordre ϕ avec $\phi = \frac{1 + \sqrt{5}}{2} \approx 1,618$.

Autrement dit, cette méthode est un peu moins bonne que la méthode de NEWTON-RAPHSON qui elle est quadratique, mais reste bien meilleure que les méthodes linéaires évoquées au début de ce chapitre (en particulier la méthode dichotomique).

2.6 Utilisation du module `scipy.optimize`

Dans le chapitre précédent, nous avons appris que dans le module `scipy.optimize` se trouve une fonction nommée `bisect` qui applique la méthode dichotomique pour trouver un zéro d'une fonction. Dans ce même module se trouve une fonction nommée `newton` qui, comme son nom l'indique, applique la méthode de NEWTON-RAPHSON. Examinons-en la page d'aide (légèrement simplifiée) :

```
newton(func, x0, fprime=None, tol=1.48e-08, maxiter=50)
    Find a zero using the Newton-Raphson or secant method.

    Find a zero of the function func given a nearby starting point x0.
    The Newton-Raphson method is used if the derivative fprime of func
    is provided, otherwise the secant method is used.

Parameters
-----
func : function
    The function whose zero is wanted.
x0 : float
    An initial estimate of the zero that should be somewhere near the
    actual zero.
fprime : function, optional
    The derivative of the function when available and convenient. If it
    is None (default), then the secant method is used.
tol : float, optional
    The allowable error of the zero value.
maxiter : int, optional
    Maximum number of iterations.

Returns
-----
zero : float
    Estimated location where function is zero.
```

On y apprend notamment que si la dérivée f' n'est pas fournie par l'utilisateur c'est la méthode de la sécante qui est utilisée.

Pour finir, illustrons ce chapitre en considérant la fonction $f : x \mapsto 2x^3 - 4x - 1$, dont le graphe figure ci dessous.

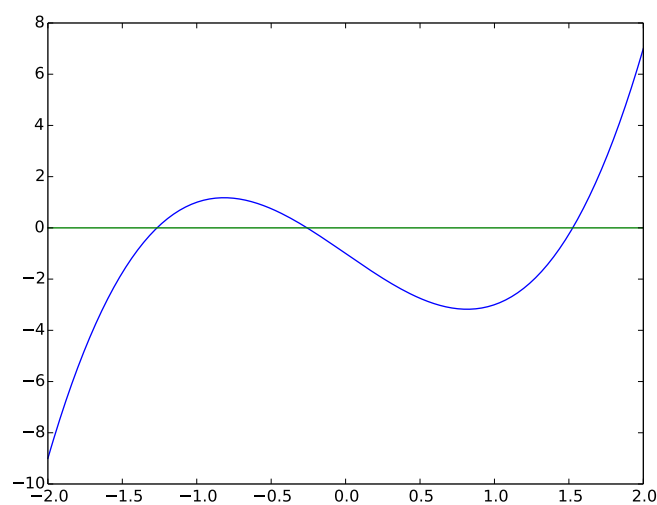


FIGURE 5 – Le graphe de la fonction $f : x \mapsto 2x^3 - 4x - 1$ sur $[-2, 2]$.

On constate que cette fonction possède trois zéros dans l'intervalle $[-2, 2]$; le script ci-dessous montre que suivant la valeur de x_0 choisie, la méthode de NEWTON-RAPHSON converge vers l'un de ces trois zéros.


```
>>> def f(x): return 2 * x**3 - 4 * x - 1
>>> def df(x): return 6 * x**2 - 4

>>> newton(f, -1, df)
-1.2670350983613659

>>> newton(f, 0, df)
-0.25865202250415276

>>> newton(f, 1, df)
1.5256871208655185
```

Si maintenant on omet de préciser la valeur de la dérivée, c'est la méthode de la sécante qui est appliquée, avec pour conséquence une légère variation des résultats :

```
>>> newton(f, 0)
-0.25865202250415226
```

Dans tous les cas, le manque de fiabilité de la méthode impose une vérification des résultats *a posteriori* : pour certaines valeurs de x_0 il se peut que la méthode fournisse un résultat qui est éloigné d'un zéro de la fonction (en particulier si on choisit pour critère d'arrêt une tolérance sur l'incrément $|x_{n+1} - x_n|$). Il se peut aussi qu'il n'y ait pas de convergence (ce qui explique la présence du paramètre `maxiter`) :

```
>>> def f(x): return x**3 - 2 * x + 2
>>> def df(x): return 3 * x**2 - 2

>>> newton(f, 0, df)
RuntimeError: Failed to converge after 50 iterations, value is 0.0
```

Dans l'exemple ci-dessus, la suite $(x_n)_{n \in \mathbb{N}}$ oscille indéfiniment entre les deux valeurs 0 et 1.