

Résolution approchée d'équations ordinaires (EO): $f(x) = 0$

Note 0.0.1 (Guide pour étudiants).

Vous trouverez ici le support du cours 5. Le contenu peut avoir été long pour les deux séances dédiées. Aussi il est fourni ici pour vous permettre d'accéder aux détails des commentaires fournis en séances.

- Dans sa lecture, munissez-vous de la fiche de travaux pratiques qui lui est associée et éventuellement de vos notes de cours.
- Insistez sur les remarques qui sont fournies et mises en évidence par la **couleur bleu**.
- Essayez les exercices qui sont fournis, certains vous permettront de mieux comprendre les remarques.
- Admettez le résultat de convergence sur la méthode de sécante et essayez de comprendre son interprétation donnée dans la remarque qui la suit.

Contents

1 Généralités, définitions et exemples	3
1.1 Définitions	3
1.2 Exemples	3
1.3 Remarque	4
1.4 Quelques notions sur les suites: vitesse de convergence, ordre et valeur ajoutée par une itération	4
2 Position correcte du problème (Eq)	6
2.1 Notion de conditionnement	7
2.2 Exemple	9
3 Construction de schémas numériques pour (Eq)	9
3.1 Généralités	9
3.2 Méthode de dichotomie	9
3.2.1 Principe	9
3.2.2 Algorithme	10
3.2.3 Convergence	10
3.2.4 Analyse	11
3.3 Méthode de fausse position	11

3.3.1	Principe	11
3.3.2	Algorithme	11
3.3.3	Convergence	12
3.3.4	Analyse	13
3.4	Méthode d'itération (ou du point fixe)	13
3.4.1	Principe	13
3.4.2	Interprétation géométrique:	13
3.4.3	Algorithme	13
3.4.4	Convergence	14
3.4.5	Analyse	15
3.5	Méthode de Newton	16
3.5.1	Principe	16
3.5.2	Interprétation géométrique	16
3.5.3	Algorithme	17
3.5.4	Convergence	17
3.5.5	Analyse	18
3.6	Méthode de la sécante	20
3.6.1	principe	20
3.6.2	Interprétation géométrique	20
3.6.3	Algorithme	20
3.6.4	Convergence	21
3.6.5	Analyse	22

1 Généralités, définitions et exemples

Soit $m > 0$ un entier et I un domaine fermé borné de \mathbf{R}^m . Soit $f : I \rightarrow \mathbf{R}^m$ continue. On considère le problème :

1.1 Définitions (1)

$$(Eq) \begin{cases} \text{Chercher } x^* \in I \text{ tel que :} \\ f(x^*) = 0 \end{cases}$$

Définition 1.1.1.

- Le problème (1) est appelé **équation linéaire** posé dans I lorsque f est affine. Dans le cas contraire il est dit **équation non-linéaire**.
- Tout $x^* \in I$ solution (1) est dite **racine** ou **zéro** de f dans I .
- Si f est de classe C^r sur I avec $r \geq 1$ et x^* une racine de f dans I , alors
 - x^* est dite **racine simple** si $f'(x^*) \neq 0$,
 - x^* est dite **racine de multiplicité** $p < r$, si $f^{(k)}(x^*) = 0$, $k = 0, \dots, p-1$, et $f^{(p)}(x^*) \neq 0$
 - Lorsque $m = 1$ c'est-à-dire $f : I \subset \mathbf{R} \rightarrow \mathbf{R}$, l'équation est dite **scalaire**.

1.2 Exemples

Exemple 1.2.1 (Equations nonlinéaires dans les schémas numériques pour EDO).

L'une des utilisations importantes est la résolution numérique des équations différentielles ordinaires. Nous avons vu au chapitre précédent que les méthodes implicites pour la résolution des EDOs avaient un certain avantage, sur les méthodes explicites. Pour s'en convaincre, considérons le problème

$$\dot{L}(t) = \tau_L L(t) \quad t \in]0, T[, \quad \text{avec} \quad L(0) = L_0 \quad (2)$$

Un schémas d'Euler explicite appliqué à la résolution de cette équation conduit à la suite

$$L_{n+1} = (1 + \tau_L \Delta t) L_n, \quad n = 0, \dots, N-1$$

Alors qu'un schéma d'Euler implicite conduirait à la suite:

$$L_{n+1} = L_n + \tau_L \Delta t L_{n+1}, \quad n = 0, \dots, N-1$$

La détermination de L_{n+1} se ramène ici à la résolution de l'équation $x = L_n + x \tau_L \Delta t$ qui s'écrit aussi

$$x - L_n - x \tau_L \Delta t = 0$$

On a vu que pour $L_0 = 10, \tau_L = -0.5, \Delta t = 3, T = 180$, le schéma d'Euler explicite est incapable d'approcher la solution exacte $L(t) = L_0 e^{t\tau_L}$ alors que le schéma d'Euler implicite se comporte beaucoup mieux. Ce pendant on ne pourra pas toujours résoudre explicitement le problème non-linéaire posé par les schémas à un pas implicites. C'est le cas par exemple de l'équation

$$\dot{u}(t) = e^{u(t)} \quad t \in]0, T[, \quad \text{avec} \quad u(0) = u_0. \quad (3)$$

Ici le schéma d'Euler implicite s'écrit

$$u_{n+1} = u_n + \Delta t e^{u_{n+1}}, \quad n = 0, \dots, N-1.$$

Pour tout $n = 0, \dots, N-1$, cette équation ne pourra être résoluble que de manière approchée (numériquement). Sachant que N peut être très grand en pratique, cette résolution devra être efficace: rapide (pas **coûteuse**) et précise (afin de ne pas accroître l'erreur locale de la méthode à un pas implicite).

Exemple 1.2.2 (Méthode de tir pour les problèmes aux limites du second ordre).

Dans un problème de tir à canon, on est souvent amené à déterminer l'angle vertical d'orientation du canon afin d'atteindre une cible précise.

En une dimension le problème peut se mettre sous la forme (4) où f est une certaine fonction donnée:

Chercher $x'(0)$ où

$$\begin{cases} x''(t) = f(t, x(t)), t \in]0, T[, \\ x(0) = x_0, \\ x(T) = x_T \end{cases} \quad (4)$$

On obtient ici un problème aux limites et non un problème de Cauchy pour une équation différentielle! Si l'on dispose d'un bon solveur d'Edo, on peut en faire usage. On peut en effet transformer ce problème en un problème de Cauchy (en une edo), en introduisant une inconnue nécessaire pour définir une condition initiale :

En effet, si pour $v \in \mathbb{R}$ on considère le problème :

$$\begin{cases} z''(t) = f(t, z(t)), t \in]0, T[, \\ z(0) = x_0, \\ z'(0) = v \end{cases} \quad (5)$$

On a alors un problème de Cauchy pour une Edo. Si l'on pose alors $g(t, v)$ la solution de cette Edo, en l'évaluant à $t = T$ on définit une fonction $g(T, v)$ de la seule variable v .

Le problème de détermination de la vitesse initiale revient alors à résoudre l'équation d'inconnue v : $G(v) = 0$, avec $G(v) = g(T, v) - x_T$

Il apparaît donc que dans certaines équations non linéaires, l'expression de la fonction dont on cherche une racine peut être bien **complexe**: ici une évaluation de la fonction G nécessite une résolution d'Edo. Il est donc nécessaire de chercher des méthodes numériques de résolution d'équations non-linéaires qui soient efficaces et si possible sans recours au calcul des dérivées.

1.3 Remarque

Il apparaît à travers ces exemples que :

Remarque 1.3.1.

- la résolution analytique des équations non-linéaires n'est pas toujours possible, même lorsqu'on sait qu'il en existe des solutions. D'où la nécessité de recourir à des algorithmes efficaces (en terme de précision et coût) permettant d'en déterminer des solutions approchées. C'est-à-dire à la **résolution numérique efficace** de ces équations.
- La plupart des équations non-linéaires sont données sous la forme $x = g(x)$. On dit que l'équation non-linéaire est posée sous forme de **recherche de point fixe**.

1.4 Quelques notions sur les suites: vitesse de convergence, ordre et valeur ajoutée par une itération

Comme nous le verrons, la recherche de solution approchée conduira à la construction de suites qui convergent vers la solution du problème. Il est donc nécessaire de quantifier cette convergence. Pour cela un rappel sur l'ordre de convergence des suites est nécessaire.

Définition 1.4.1.

Soit x_n une suite réelle qui converge vers x

- Si $K_1 = \lim_n \frac{x - x_{n+1}}{x - x_n}$ existe et si $K_1 \in]-1, 1[\setminus \{0\}$, on dit que la suite x_n converge **linéairement** vers x . On parle aussi de convergence **d'ordre 1**.
- Si $K_1 = 0$ et si $K_2 = \lim_n \frac{x - x_{n+1}}{(x - x_n)^2}$ existe et n'est pas égale à zéro, on dit que la suite x_n converge **quadratiquement** vers x . On parle aussi de convergence **d'ordre 2**.
- Si $K_1 = 0, K_2 = 0$ et si $K_3 = \lim_n \frac{x - x_{n+1}}{(x - x_n)^3}$ existe et n'est pas égale à zéro, on dit que la convergence est **cubique** ou encore **d'ordre 3**.

On généralise de manière évidente cette définition à des convergences d'ordres supérieures. Remarquons qu'il n'est pas nécessaire en pratique de calculer la limite pour conclure car cette limite peut ne pas exister. On utilise alors la définition suivante

Définition 1.4.2.

La convergence de la suite x_n vers x est d'ordre r si $\exists N$ tel que $\forall n > N$

$$0 < A \leq \frac{|x_{n+1} - x|}{|x_n - x|^r} \leq B < \infty$$

où A, B sont deux constantes indépendantes de n .

La convergence est d'ordre **au moins** r si seule l'inégalité avec B a lieu.

Remarque 1.4.1.

Signalons cependant qu'il faut passer par la définition avec la limite pour accéder à l'estimation du nombre de chiffres exactes ajoutés par itération. En effet, si la suite est convergente d'ordre r dans le sens où $K_r = \lim_n \frac{|x - x_{n+1}|}{|x - x_n|^r}$ existe et est non nul (K_r est appelé **constante asymptotique** de l'erreur), alors $d_n = -\log_{10} |x_n - x|$ mesure le nombre de chiffres décimaux de x_n et de x qui coïncident (à une constante additive près ne dépendant pas de n) et on a

$$d_{n+1} \approx r d_n - \log_{10}(K_r).$$

Ce qui montre qu'à une constante additive près, le nombre de chiffres exactes est multiplié par r à chaque itération.

En effet on a $d_{n+1} + \log_{10}(K_r)/(1-r) = r(d_n + \log_{10}(K_r)/(1-r))$.

On peut donc estimer le nombre d'itérations nécessaires pour gagner un chiffre exact. En particulier, lorsque la convergence est linéaire, comme ce sera le cas pour la plupart des méthodes que nous verrons, même celles d'ordre élevée lorsqu'elles seront confrontées à des situations particulières comme celles de recherche des racines multiples.

Proposition 1.4.1 ((importance de la constante asymptotique d'erreur dans une convergence linéaire).)

Soit x_n une suite qui converge linéairement avec une constante asymptotique d'erreur égale à K . Alors le nombre d'itérations nécessaires pour gagner un chiffre exacte est le plus petit entier supérieur à $\frac{1}{-\log_{10}(K)}$.

Démonstration.

la formule de récurrence du nombre de chiffres exactes par itération est donnée ici par $d_{n+1} = d_n - \log_{10}(K)$. Ainsi après m itérations à partir de l'itération n , on a $d_{n+m} = d_n - m \log_{10}(K)$. Par conséquent, $d_{n+m} = d_n + 1$ si et seulement si $m = \frac{1}{-\log_{10}(K)}$ (arrondi à l'entier supérieur).

Quelques fois il est suffisant d'estimer l'ordre de convergence au moyen de comparaisons:

Définition 1.4.3.

Soient x_n, y_n deux suites qui convergent respectivement vers x^*, y^* . On dit que x_n **converge plus vite que** y_n si $\lim_{n \rightarrow \infty} \frac{x_n - x^*}{y_n - y^*} = 0$.

Ainsi, si x_n est d'ordre q et converge plus rapidement que y_n qui est d'ordre p , alors $q \geq p$. Par conséquent si on ne dispose que de l'ordre p de y_n , on pourra dire pour x_n convergent plus vite que y_n qu'elle est d'ordre **au moins** p .

2 Position correcte du problème (Eq)

Avant de nous lancer dans la construction des schémas numériques pour équations non-linéaires, il est important de répondre aux questions :

1. Cette équation admet-elle une solution ?
2. Si oui cette solution est-elle unique ?
3. Cette solution dépend-t-elle continûment des données ? Autrement dit est-elle stable vis-à-vis des données du problème ?
4. La solution du problème si elle existe a-t-elle une régularité particulière ?

La réponse à ces questions est ce que l'on appelle vérification de la **position correcte** du problème. Elle est nécessaire lorsqu'on souhaite résoudre numérique (de manière approchée) tout problème donné. Si le point 3 est négligé, les conséquences peuvent être dramatiques au niveau numérique.

Puisqu'on est dans un cadre scalaire (c'est-à-dire $f : \mathbf{R} \mapsto \mathbf{R}$), l'existence de solution est une simple conséquence du théorème de Rolle (donné en fin du cours 1).

Proposition 2.0.1 ((cas où l'équation est sous la forme $f(x) = 0$)).

Si $I = [a, b]$ et f est telle que

- $f(a)f(b) < 0$
- f continue sur I

Alors il existe $x^* \in I$ tel que $f(x^*) = 0$

Démonstration.

C'est la conséquence du théorème de Rolle.

Proposition 2.0.2 ((cas où l'équation est sous la forme $x = g(x)$)).

Si g est telle que

- $g(I) \subset I$
- g continue

Alors il existe $x^* \in I$ tel que $g(x^*) = x^*$

Démonstration.

On applique le résultat précédent à $f(x) = g(x) - x$

Remarque 2.0.1 ((accessibilité de l'intervalle I)).

- Les résultats que nous venons de présenter, font l'hypothèse que l'intervalle I est connu. Mais ce n'est pas le cas en pratique. Ainsi, I est lui même inconnu et est appelé **domaine d'attraction** de la solution dans le cas d'un problème de point fixe.
- La détermination de I et du point fixe x^* nécessite l'analyse des propriétés de contraction de la fonction g . Cette détermination repose sur une méthode constructive, qui définit en elle même un schéma numérique connu sous le nom de méthode **d'itérations (successives)**. *En effet, sous certaines hypothèses sur g , la suite définie par $x_{n+1} = g(x_n)$ pour une donnée initiale bien choisie (ie dans le domaine d'attraction du point fixe de g) converge vers le point fixe de g . Et on peut même estimer l'erreur.*
- Lorsque l'équation est sous la forme $f(x) = 0$, un autre procédé constructif permet aussi de définir l'intervalle I : on se fixe un point a et un pas Δ . Puis on se déplace selon ce pas de part et d'autre de a , jusqu'au premier changement de signe de f . On définit alors l'intervalle I . Pour Δ suffisamment petit, cette procédure approche directement la racine de l'équation et on appelle la schéma ainsi construit, **méthode de recherche incrémentale**.

On retient alors à ce stade que certaines preuves de la position correcte des équations non-linéaires, cachent en elles mêmes des schémas numériques, certes basiques, mais exploitables (pour la définition des schémas efficaces) pour la résolution de ces équations.

2.1 Notion de conditionnement

Intéressons nous à présent à la stabilité. C'est-à-dire à l'effet, sur la solution, des perturbations sur les données.

Définition 2.1.1 (Conditionnement.).

- On appelle **nombre de conditionnement** d'un algorithme, le facteur d'amplification (ou de réduction) de l'erreur d'évaluation d'un algorithme.
- Si ce facteur s'intéresse aux erreurs relatives on parle de **nombre conditionnement relatif** ou simplement **nombre conditionnement**. **Si par contre ce facteur ne s'intéresse qu'aux erreurs absolues, on parle de nombre de conditionnement absolu.** L'équation non-linéaire étant de la forme $f(x) = 0$, pour une perturbation η de f , on ne pourra, en l'absence de l'expression exacte de f , que se contenter du conditionnement absolu. Si on a l'expression de f , par exemple si f est polynomiale, on pourra considérer que la perturbation de f est due à une perturbation sur ses coefficients, et par conséquent être en mesure d'évaluer le conditionnement relatif du problème de recherche des racines de f
- **Le problème sera dit bien conditionné si le nombre de conditionnement est petit, c'est-à-dire proche de 1. Il sera dit mal conditionné si ce nombre est très grand par rapport à 1.**

Proposition 2.1.1 ((conditionnement absolu)).

Supposons f de classe C^p et soit

x^* tel que $f(x^*) = 0$.

\tilde{x} tel que $f(\tilde{x}) + \eta(\tilde{x}) = 0$, où η est une perturbation (erreur de mesure) de f que l'on suppose bornée.

Si x^* est une racine de multiplicité $m < p$ de f alors

$$|\tilde{x} - x^*| \approx (m!) \left| f^{(m)}(x^*) \right|^{-\frac{1}{m}} |\eta(\tilde{x})|^{\frac{1}{m}} \quad (6)$$

Plus particulièrement, si $f^{(m)}$ a un signe constant, on a

$$|\tilde{x} - x^*| \leq (m!) \left| f^{(m)}(x^*) \right|^{-\frac{1}{m}} |\eta(\tilde{x})|^{\frac{1}{m}} \quad (7)$$

Démonstration.

En effectuant le développement limité de Taylor Lagrange de $f(\tilde{x})$ au voisinage de x^* , on a $-f(\tilde{x}) = f(\tilde{x}) \approx \frac{1}{m!} f^{(m)}(x^*) (\tilde{x} - x^*)^m$. Et le résultat en découle.

On a la seconde estimation (7) par application du second théorème de la moyenne au développement de Taylor avec reste intégrale de $f(\tilde{x})$ au voisinage de x^* .

Ainsi, le **conditionnement absolu** est donné par $\kappa(f, x^*) = (m!) \left| f^{(m)}(x^*) \right|^{-\frac{1}{m}} |\eta(\tilde{x})|^{\frac{1}{m}-1}$

Ce qui montre que la résolution des équations non-linéaires à racines multiples sera un problème moins bien conditionné (et même mal conditionné) que celui de la recherche de racines simples.

Il faudra faire attention à ce type de problème.

2.2 Exemple

Exemple 2.2.1.

Si l'on remplace la recherche de la racine double 2 de $(x - 2)^2$ par le problème $(x - 2)^2 - 10^{-8} = 0$ on aura pour solution $x = 2 + 10^{-4}$ ou $x = 2 - 10^{-4}$. D'où un conditionnement absolu de $\kappa = 20000$. C'est-à-dire que le facteur d'amplification de l'erreur est de 20000. Ce qui signifie que lors de la recherche de la racine double de $f(x) = (x - 2)^2$, si l'on évalue la fonction f avec seulement 8 chiffres significatifs, alors, on ne pourra que garantir que 4 chiffres significatifs de la racine obtenue.

3 Construction de schémas numériques pour (Eq)

3.1 Généralités

Nous allons dans cette section construire et analyser quelques méthodes numériques de résolution du problème (Eq) dans le cas scalaire.

La démarche sera la suivante:

- Nous allons d'abord regarder les **méthodes de bases**. Ici, en nous inspirant des techniques de preuve d'existence de racines, nous allons mettre en place des méthodes basiques.
 - Nous distinguerons les cas spécifiques où le problème (Eq) est sous la forme $f(x) = 0$. Ce qui conduira : à la méthode de **Dichotomie** et à la méthode de **fausse position** qui en est une amélioration.
 - Puis nous considérerons le problème (Eq) formulé comme recherche de point fixe. Ce qui conduira : à la méthode de **d'itérations (successives)** ou de **point fixe**.
- Nous construirons ensuite les **méthodes itératives**, qui seront des applications de la méthode du point fixe à des réécritures particulières de (Eq) sous la forme d'un problème de point fixe. En effet on sait qu'il n'existe pas qu'une seule écriture équivalente du problème $f(x) = 0$ sous la forme $x = g(x)$ (en guise d'illustration on a $g(x) = x - f(x)$ ou $g(x) = x + f(x)$). En utilisant alors une méthode des **puissances entières** on peut proposer une démarche systématique d'obtention de cette écriture équivalente. Ainsi suivant la régularité dont on disposera sur f , on pourra construire des méthodes itératives d'ordre 1, 2, ...:
 - Chaque méthode itérative d'ordre k construite sera à **un pas**, mais nécessitera les dérivées de f jusqu'à l'ordre $k - 1$, ce qui pour k grand sera un inconvénient.
 - On pourra alors approcher les dérivées de f par des différences divisées rétrogrades. Ce qui générera des méthodes itératives à k pas, où l'inconvénient ici sera relégué au démarrage, puisqu'il faudra compléter l'unique approximation initiale x_0 .
 - Afin de prendre en compte les cas pathologiques des racines d'ordre de multiplicité $m > 1$, des **modifications** sont apportées à ces méthodes.

Dans la pratique, pour des raisons d'efficacité, on s'arrête à l'ordre $k = 3$ et on investit plutôt, s'il le faut sur des **techniques d'accélération**.

Pour $k = 2$ on obtiendra la méthode de **Newton**, dans laquelle une approximation de la dérivée par une différence divisée rétrograde conduira à la **méthode de la sécante**.

3.2 Méthode de dichotomie

3.2.1 Principe

On a f continue sur $[a, b]$ avec $f(a)f(b) < 0$. D'après le théorème de Rolle, f admet une racine dans $[a, b]$. Cette racine est certainement dans l'une des moitiés de l'intervalle $[a, b]$. C'est-à-dire si l'on pose $c = \frac{a+b}{2}$. L'un des

intervalles $[a, c]$ ou $[c, b]$ nous placera dans la configuration de départ. On prend celui là et on rejette l'autre. On répète le processus, ce qui génère une suite d'intervalles $([a_n, b_n])$ emboîtés, dont la longueur tends vers 0 et qui vérifie pour tout n , $f(a_n)f(b_n) < 0$. La méthode de dichotomie met donc en place la procédure suivante:

- on pose $[a_0, b_0] = [a, b]$
- $[a_n, b_n]$ étant connu, on pose $c = \frac{a_n + b_n}{2}$ et on teste $f(a_n)f(c)$
 - Si cette valeur est négative, la racine de f est dans $[a_n, c]$ on pose alors $[a_{n+1}, b_{n+1}] = [a_n, c]$
 - Si cette valeur est positive, alors $f(c)f(b_n) < 0$, on pose alors $[a_{n+1}, b_{n+1}] = [c, b_n]$
 - Si elle est nulle alors c est la racine.

3.2.2 Algorithme

Listing 1: Méthode de dichotomie

```

- Initialisation
- Soit  $a, b$  tels que  $f(a)f(b) < 0$  et une tolerance  $\epsilon$ 
- Calculer  $c = \frac{a+b}{2}$ 
-  $k = 0$ 
- Itérations
- Tant que  $|f(c)| > \epsilon$  et  $k$  pas trop grand faire
  - si  $f(a)f(c) < 0$ 
    - poser  $b = c$ 
  - sinon
    - poser  $a = c$ 
  - calculer
    -  $c = \frac{a+b}{2}$ 
    -  $k = k + 1$ 
- Fin faire
  
```

Coût : chaque itération nécessite : **1 division, 1 évaluation** de signe de fonctions.

3.2.3 Convergence

Cette méthode est certaine de converger en effet on a

Proposition 3.2.1.

Soit f continue sur $[a, b]$ telle que $f(a)f(b) < 0$, et x^* l'unique racine de f dans $]a, b[$. Soit $(a_n), (b_n)$ les suites générées par la méthode de dichotomie, alors

- $x^* \in [a_n, b_n]$ pour tout n ;
- $[a_{n+1}, b_{n+1}] \subset [a_n, b_n]$ pour tout n ;
- $b_n - a_n = (b - a)/2^n$;
- $|a_n - x^*| \leq (b - a)/2^n, |b_n - x^*| \leq (b - a)/2^n$;
- $\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = x^*$.

Preuve: évidente laissée en exercice.

Remarque 3.2.1.

- La méthode produisant une suite d'intervalles, on n'a pas défini la constante asymptotique dans ce cas. Néanmoins pour des besoins de comparaison avec d'autres méthodes, on prendra $K_1 = \frac{1}{2}$ car $\frac{b_{n+1}-a_{n+1}}{b_n-a_n} = \frac{1}{2}$ pour tout n .

- On peut estimer à l'avance le nombre d'itérations nécessaires pour une tolérance tol donnée. En effet si l'on souhaite avoir une erreur $< tol$, il suffira de s'assurer que $b_n - a_n = \frac{b-a}{2^n} < tol$. Soit un nombre d'itérations

$$n = \frac{\log\left(\frac{b-a}{tol}\right)}{\log(2)}. \quad (\text{prendre sa partie entière}).$$

3.2.4 Analyse

Note 3.2.1 (Analyse et Illustration).

- Avantages:
 - La programmation de la méthode est simple.
 - La convergence est certaine.
- Inconvénients:
 - Hypothèse de départ contraignante : Il faut partir de a, b tels que $f(a)f(b) < 0$
 - Convergence lente.
- Exemples et illustration : Traiter l'exercice de la fiche de TP associée à ce cours.

3.3 Méthode de fausse position

3.3.1 Principe

Cette méthode suit le même principe que la méthode de dichotomie avec pour seule différence qu'au lieu de prendre c comme le milieu de a, b , on le prendra (définition équivalente):

- soit comme la racine du polynôme interpolateur de Lagrange associé aux nœuds a, b
- soit comme l'abscisse du point d'intersection avec l'axe des abscisses de la droite passant par $(a, f(a))$, et $(b, f(b))$. C'est-à-dire $c = b - \frac{b-a}{f(b)-f(a)}f(b)$

3.3.2 Algorithme

Listing 2: Méthode de Fausse position

```
- Initialisation
- Soit  $a, b$  tels que  $f(a)f(b) < 0$  et une tolérance  $\varepsilon$ 
- Calculer  $c = b - \frac{b-a}{f(b)-f(a)}f(b)$ 
-  $k = 0$ 
- Itérations
- Tant que  $|f(c)| > \varepsilon$  et  $k$  pas trop grand faire
  - si  $f(a)f(c) < 0$ 
    - poser  $b = c$ 
  - sinon
    - poser  $a = c$ 
  - calculer
    -  $c = b - \frac{b-a}{f(b)-f(a)}f(b)$ 
    -  $k = k + 1$ 
- Fin faire
```

Coût : chaque itération nécessite : 1 produit, 1 division, 1 évaluation de fonctions.

3.3.3 Convergence

Proposition 3.3.1.

Soit $f \in C^2([a, b])$ telle que $f(a)f(b) < 0$. Si f'' n'a aucune racine dans l'intervalle $[a, b]$, alors une des suites (a_n) et (b_n) demeure constante quand on applique la méthode de fausse position à f sur $[a, b]$.

Démonstration.

f'' étant continue et ne s'annulant pas dans $[a, b]$, elle y a un signe constant. Comme $f(a)f(b) < 0$, on a 4 configurations possibles

- cas 1) : $f'' > 0$ sur (a, b) ; $f(a) < 0 < f(b)$
- cas 2) : $f'' < 0$ sur (a, b) ; $f(a) > 0 > f(b)$
- cas 3) : $f'' > 0$ sur (a, b) ; $f(a) > 0 > f(b)$
- cas 4) : $f'' < 0$ sur (a, b) ; $f(a) < 0 < f(b)$

Traisons le cas 1), les autres se traitent de manière analogue. $f'' > 0$ est une indication que la fonction est strictement convexe sur $[a, b]$. Donc la courbe de f est en dessous de la droite joignant $(a, f(a))$ et $(b, f(b))$ pour tout $x \in [a, b]$. D'où pour tout $x \in [a, b]$, on a

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

Par conséquent le point c de départ qui est celui qui annule le membre de droite vérifie : $f(c) < 0$. Or $f(a) < 0$, ainsi après la première itération on retient l'intervalle $[a_1, b_1] = [c, b]$. Et b n'aura pas bougé. On se retrouve encore dans une configuration analogue à celle de départ à savoir le cas 1). Et b restera encore inchangé. On conclut alors par récurrence que b ne changera jamais.

On remarquera que dans les cas 1 et 2, b_n sera constante. Et dans les cas 3 et 4 c 'est a_n qui restera constante.

On peut alors donner un résultat de convergence:

Proposition 3.3.2 (Convergence de la méthode de fausse position).

Soit $f \in C^2([a, b])$ telle que $f(a)f(b) < 0$ et f'' n'a aucune racine dans l'intervalle $[a, b]$. Soit $(a_n), (b_n)$ générées par la méthode de fausse position.

- Si b_n est constante, alors (a_n) converge linéairement vers la racine x^* de f et on a $K_1 = \lim_{n \rightarrow \infty} \frac{x^* - a_{n+1}}{x^* - a_n} = 1 + f'(x^*) \frac{x^* - b}{f(b)}$.
- Si (a_n) est constante, alors (b_n) converge linéairement vers la racine x^* de f et on a $K_1 = \lim_{n \rightarrow \infty} \frac{x^* - b_{n+1}}{x^* - b_n} = 1 + f'(x^*) \frac{x^* - a}{f(a)}$.

Démonstration.

Puisque les suites $(a_n), (b_n)$ s'écrivent dans ce cas sous la forme $y_{n+1} = G(y_n)$. Cette démonstration est un cas particulier d'une démonstration qui sera donnée plus loin.

3.3.4 Analyse

Note 3.3.1 (Analyse et Illustration).

- Avantages:
 - Convergence généralement meilleure que celle de la méthode de dichotomie, pour racine simple.
- Inconvénients:
 - Hypothèse de départ contraignante : Il faut partir de a, b tels que $f(a)f(b) < 0$.
 - Complexité plus grande que la méthode de dichotomie.
 - Convergence très lente vers la racine multiple.
- **Exemples et illustration** : Traiter l'exercice de la fiche de TP associée à ce cours.

3.4 Méthode d'itération (ou du point fixe)

3.4.1 Principe

C'est le schéma numérique naturel lorsque l'équation est écrite sous la forme

$$\text{Chercher } x^* \text{ tel que } x^* = g(x^*)$$

Le schéma numérique est simplement

$$\begin{cases} x_{n+1} = g(x_n), & n = 0, \dots, \\ x_0 & \text{donné} \end{cases} \quad (8)$$

Cette méthode est appelée **méthode d'itérations (successives)** et la fonction g est appelée **fonction d'itération**.

3.4.2 Interprétation géométrique:

L'opération $x_{n+1} = g(x_n)$ se traduit géométriquement ainsi: dans un repère cartésien, on part de $(x_n, 0)$, on se déplace verticalement jusqu'à la courbe $y = g(x)$. Puis on se déplace horizontalement jusqu'à la droite $y = x$, puis verticalement jusqu'à la droite $y = 0$, ce qui donne le point $(x_{n+1}, 0)$.

3.4.3 Algorithme

Listing 3: Méthode du point fixe

```
- Initialisation
- Soit  $x_0$  une approximation initiale et  $\varepsilon > 0$  une étolrance
- Poser  $k = 0$ 
- Poser  $r_k = \varepsilon + 1$ 
- Itérations
- Tant que  $r_k > \varepsilon$  et k pas trop grand faire
  -  $x_{k+1} = g(x_k)$ 
  -  $r_{k+1} = |x_{k+1} - x_k|$ 
  -  $k = k + 1$ 
- Fin faire
```

Coût : chaque itération nécessite : **1 évaluation** de fonctions. Mais attention cette évaluation peut être chère.

3.4.4 Convergence

Proposition 3.4.1.

Soit I un intervalle fermé et borné, et une fonction g continue sur I telle que $g(I) \subset I$. La fonction d'itération g admet au moins un point fixe x^* dans I .

- Si $\max_{x \in I} |g'(x)| = K < 1$, alors ce point fixe est unique.
- Si $\max_{x \in I} |g'(x)| = K < 1$, et si en plus $x_0 \in I$ et x_n est définie par $x_{n+1} = g(x_n)$, alors
 - $|x^* - x_{n+1}| < K|x^* - x_n|$
 - x_n converge vers x^*
 - $K_1 = \lim_{n \rightarrow \infty} \frac{x^* - x_{n+1}}{x^* - x_n} = g'(x^*)$. Et $|K_1| \leq K$.

Démonstration.

- **Existence:** Posons $I = [a, b]$, l'existence est une conséquence du théorème de Rolle appliquée à $f(x) = g(x) - x$.
- **Unicité:** Si g possédait deux points fixes z^*, y^* , on aurait, $1 = \frac{g(z^*) - g(y^*)}{z^* - y^*} = g'(\eta)$ où $\eta \in I$. Et il existerait $\eta \in I$ tel que $1 = g'(\eta)$. Ce qui contredirait l'hypothèse $K < 1$.
- **Convergence:**
On a $\frac{x^* - x_{n+1}}{x^* - x_n} = \frac{g(x^*) - g(x_n)}{x^* - x_n} = g'(\zeta)$ pour un ζ entre x^* et x_n . Comme $|g'(\zeta)| < K$, on déduit que $|x^* - x_{n+1}| < K|x^* - x_n|$.
Et par induction il vient $|x^* - x_n| < K^n|x^* - x_0|$. D'où x_n converge vers x^* (car $0 \leq K < 1$);
- Le dernier résultat découle de la continuité de g' et par application du théorème de la moyenne.

Remarque 3.4.1.

La proposition ci-dessus, indique une convergence linéaire. Mais en pratique, la convergence est plutôt au moins linéaire.

Proposition 3.4.2.

Si la suite x_n définie par x_{n+1} converge vers x^* et s'il existe $p \geq 1$ entier tel que $g \in C^p$, $g^{(p)}(x^*) \neq 0$, $g^{(i)}(x^*) = 0$ pour $1 \leq i \leq p-1$, alors

$$K_p = \lim_{n \rightarrow \infty} \frac{x^* - x_{n+1}}{(x^* - x_n)^p} = \frac{g^{(p)}(x^*)}{(-1)^{(p-1)}p!}$$

Démonstration.

$$\text{On a } \lim_{n \rightarrow \infty} \frac{x^* - x_{n+1}}{(x^* - x_n)^p} = \lim_{n \rightarrow \infty} \frac{g(x^*) - g(x_n)}{(x^* - x_n)^p} = \lim_{x \rightarrow x^*} \frac{g(x^*) - g(x)}{(x^* - x)^p}$$

D'où le résultat en utilisant le développement de Taylor de $g(x)$ au voisinage de x^* :

$$g(x) = g(x^*) + \frac{g^{(p)}(x^*)}{p!}(x - x^*)^p + \mathcal{O}((x - x^*)^{p+1}), \text{ (car } g^{(i)}(x^*) = 0 \text{ pour } 1 \leq i \leq p - 1).$$

3.4.5 Analyse**Note 3.4.1 (Analyse et Illustration).**

- Avantages:
 - Simple à mettre en oeuvre .
- Inconvénients:
 - La convergence n'est pas assurée.
- **Exemples et illustration :**
Traiter l'exercice de la fiche de TP associée à ce cours.

Exemple 3.4.1 (Exercice d'application).

Montrer le résultat de convergence de la méthode de **fausse position**. On rappelle que cette méthode, du fait que l'une des extrémités de l'intervalle est fixe peut s'écrire comme méthode de point fixe. Si c'est b qui est fixe, on aura $a_{n+1} = a_n - f(a_n) \frac{a_n - b}{f(a_n) - f(b)}$. C'est-à-dire $a_{n+1} = g(a_n)$ avec $g(x) = x - f(x) \frac{x - b}{f(x) - f(b)}$.

Exemple 3.4.2 (Exercices).

- **Exercice 1:**
 1. Montrer que si le point fixe x^* de g est tel que $|g'(x^*)| > 1$, alors la méthode d'itération du point fixe diverge.
 2. Peut-on aussi conclure si $|g'(x^*)| = 1$?
- **Exercice 2:**
Soient g de classe C^2 , x^* son point fixe tel que $|g'(x^*)| = 1$ (on supposera $g'(x^*) = 1$).
Soit x_n la suite $x_{n+1} = g(x_n)$ et $e_n = x^* - x_n$, l'erreur à l'itération n .
 1. Montrer qu'on a

$$\frac{e_{n+1}}{e_n} = 1 - \frac{1}{2}g''(\eta)e_n$$
 , où η est compris entre x^* et x_n .
 2. Etudier alors la convergence ou la divergence de la suite (x_n) en fonction du signe de $g''(x^*)$ et de ceux des e_n .

Remarque 3.4.2.

Le résultat montre que la méthode du point fixe peut converger ou diverger suivants les cas:

- Si $|g'(x^*)| < 1$, la méthode convergera si l'on choisi le point de départ proche de x^* avec g est C^1 .
- Si $|g'(x^*)| > 1$, la méthode divergera si g est au moins C^1 .

Le cas $|g'(x^*)| = 1$ est indéterminé et l'exercice 2 permet d'y apporter une réponse: par exemple, dans le cas où $g'(x^*) = 1$

- Si la fonction g est **strictement convexe** au voisinage de x^* , et si l'on démarre proche et à gauche du point fixe, la méthode convergera. Mais si l'on démarre à droite du point fixe, la méthode divergera.
- Si la fonction g est **strictement concave** au voisinage de x^* , et si l'on démarre proche et à droite du point fixe, la méthode convergera. Mais si l'on démarre à gauche du point fixe, la méthode divergera.

3.5 Méthode de Newton

3.5.1 Principe

La méthode de Newton, repose sur l'approximation de la fonction par son polynôme de Taylor.

Plus généralement, on pourra construire un processus itératif en partant de x_0 et en définissant à l'itération n le terme x_{n+1} , comme la racine d'une *bonne* approximation (polynomiale) de $f(x)$. Si le polynôme d'interpolation est un choix possible, la méthode de **Newton** utilise le développement de Taylor pour construire un tel polynôme. Elle se contente du développement à d'ordre 1 au voisinage de x_n , on dit qu'on effectue une **approximation affine** de f au voisinage de x_n .

Supposons disposer de x_n , qui est l'approximation courante de la racine. Supposons f dérivable. Le développement limité d'ordre 1 de f au voisinage de x_n est $f(x) = f(x_n) + f'(x_n)(x - x_n) + \iota(x - x_n)$. D'où l'approximation de degré 1 de f au voisinage de x_n donnée par $P_1(x) = f(x_n) + f'(x_n)(x - x_n)$. On détermine x_{n+1} comme la racine de ce polynôme, soit $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$.

Le schéma obtenu est le suivant

$$\begin{cases} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)}, & n = 0, \dots, \\ x_0 &\text{donné} \end{cases} \quad (9)$$

On retrouve la méthode du point fixe, avec pour fonction d'itération $g(x) = x - \frac{f(x)}{f'(x)}$.

3.5.2 Interprétation géométrique

La droite $y = f(x_n) + f'(x_n)(x - x_n)$ est l'équation de la tangente à la courbe $y = f(x)$ au point $(x_n, f(x_n))$. Par conséquent x_{n+1} est l'abscisse du point d'intersection de l'axe des abscisses (doite $y = 0$) avec la tangente à la courbe $y = f(x)$ au point $(x_n, f(x_n))$.

Ainsi, l'opération $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ se traduit géométriquement par: A partir de $(x_n, 0)$, on se déplace verticalement jusqu'à la courbe $y = f(x)$, puis le long de la tangente à courbe jusqu'à la droite $y = 0$ (l'axe des abscisses) pour définir le point $(x_{n+1}, 0)$.

3.5.3 Algorithme

Listing 4: Méthode de Newton

```
- Initialisation
- Soit  $x_0$  une approximation initiale et  $\varepsilon > 0$  une étolrance
- Poser  $k = 0$ 
- Itérations
- Tant que  $|f(x_k)| > \varepsilon$  et k pas trop grand faire
  -  $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ 
  -  $k = k + 1$ 
- Fin faire
```

Coût : chaque itération nécessite : **1 division** et **2 évaluations** de fonctions dont une de la dérivée de la fonction.

3.5.4 Convergence

Proposition 3.5.1.

Soit (x_n) une suite générée par la méthode de Newton appliquée à la fonction f . Supposons que x_n converge vers x^* .

- Si
 - Si f est C^1 alors x^* est racine de f .
 - Si m est un entier tel que f est C^{m+2} , $f^{(m)}(x^*) \neq 0$, $f^{(i)}(x^*) = 0$ pour $i \leq m - 1$,

alors,

$$K_1 = \lim_{n \rightarrow \infty} \frac{x^* - x_{n+1}}{x^* - x_n} = 1 - \frac{1}{m}.$$

- Si p est un entier tel que f est C^{p+1} , $f'(x^*) \neq 0$, $f^{(p)}(x^*) \neq 0$, $f^{(i)}(x^*) = 0$ pour $2 \leq i \leq p - 1$, alors

$$K_p = \lim_{n \rightarrow \infty} \frac{x^* - x_{n+1}}{(x^* - x_n)^p} = \frac{(p-1)f^{(p)}(x^*)}{(-1)^{(p-1)}p!f'(x^*)}.$$

Démonstration.

(On peut le faire comme exercice, car c'est une application de la méthode du point fixe.)

- Le point 1 résulte du passage à la limite dans $f(x_n) = f'(x_n)(x_n - x_{n+1})$.
- Pour la suite on applique le résultat sur la méthode du point fixe, car la méthode de Newton s'écrit $x_{n+1} = g(x_n)$ avec $g(x) = x - \frac{f(x)}{f'(x)}$.
- Pour le point 2, on remarque que $g'(x) = -\frac{f(x)f''(x)}{(f'(x))^2}$ et qu'on peut écrire $f(x) = (x - x^*)^m h(x)$, avec $h(x^*) \neq 0$. Il vient donc $K_1 = \lim_{x \rightarrow x^*} g'(x) = 1 - \frac{1}{m}$.
- Le point 3 s'obtient simplement par dérivations successives (c'est un peu fastidieux, limitez vous à $p = 3$).

Remarque 3.5.1.

- Le résultat ci-dessus montre que la méthode de Newton appliquée à la recherche d'une racine de multiplicité $m > 1$ convergera à l'ordre 1 avec pour constante asymptotique $K_1 = 1 - \frac{1}{m}$
- Mais si la racine est simple ($f'(x^*) \neq 0$), l'ordre de convergence est au moins d'ordre 2 et peut être supérieure à 2. Plus précisément, elle sera d'ordre p , où p est le plus petit entier $p \geq 2$ tel que $f^{(p)}(x^*) \neq 0$. Pour s'en convaincre :
 - la méthode de Newton sera d'ordre 2 pour la recherche de la racine simple $x^* = 0$ de $f(x) = e^x - 1$,
 - elle sera d'ordre 3 pour la recherche de la racine simple $x^* = 0$ de $f(x) = e^x - 1 - \frac{x}{2} - \frac{x^2}{2}$,
 - elle sera d'ordre 4 pour la recherche de la racine simple $x^* = 0$ de $f(x) = e^x - 1 - \frac{x}{2} - \frac{x^2}{2} - \frac{x^3}{6}$.

3.5.5 Analyse

Note 3.5.1 (Analyse et Illustration).

- Avantages:
 - Toujours plus rapide que la méthode de fausse position.
 - Hypothèses de départ moins contraignantes que celles de dichotomie, de fausse position.
- Inconvénients:
 - Evaluation obligatoire de la dérivée.
 - Convergence plus lente vers les racines multiples que celle de la méthode de dichotomie.
- Exemples et illustration :

Traiter l'exercice de la fiche de TP associée à ce cours.

Remarque 3.5.2 (Montée en ordre en cas de racine d'ordre de multiplicité $m > 1$).

- Le résultat de convergence montre une convergence d'ordre 1 dans le cas de recherche de racine d'ordre de multiplicité $m > 1$.
- On peut corriger cela de plusieurs manières, suivant que l'on dispose ou pas de l'ordre de multiplicité de la racine.
 - Si l'on ne connaît pas l'ordre de multiplicité de la racine, et si f est au moins de classe C^2 , on peut obtenir une convergence d'ordre 2 au moins en appliquant la méthode de Newton à la nouvelle équation $F(x) = f(x)/f'(x)$. Car la racine de multiplicité $m > 1$ de f sera simple pour F .
 - Si l'on dispose de l'ordre de multiplicité m , ou d'une approximation r , alors on pourra considérer la méthode de **Newton pondérée**:

$$\begin{cases} x_{n+1} &= x_n - r \frac{f(x_n)}{f'(x_n)}, & n = 0, \dots, \\ x_0 &\text{donné} \end{cases} \quad (10)$$

Exemple 3.5.1 (Exercices : Sur la montée en ordre).

• **Exercice 1:**

Soit $f \in C^{m+1}$ avec m ordre de multiplicité de racine recherchée de f . Posons

$$\tilde{f}(x) = \begin{cases} \frac{f(x)}{f'(x)}, & \text{si } f'(x) \neq 0 \\ 0, & \text{si } f'(x) = 0 \\ \text{indéfinie,} & \text{si } f(x) \neq 0, \quad f'(x) = 0 \end{cases}$$

1. Montrer que \tilde{f} possède exactement les mêmes racines que f , mais que ses racines sont toutes simples.
2. Montrer que la méthode de Newton appliquée à \tilde{f} est donnée par

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{(f'(x_n))^2 - f(x_n)f''(x_n)}$$

• **Exercice 2:**

Soit r un réel et (x_n) la suite générée par la méthode de Newton pondérée

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

. Supposons que (x_n) converge vers x^* racine de f de multiplicité m et que la limite suivante

$$K_1 = \lim_{n \rightarrow \infty} \frac{x^* - x_{n+1}}{x^* - x_n}$$

. Montrer que si $f \in C^{m+2}$ alors $K_1 = 1 - \frac{r}{m}$.

Remarque 3.5.3.

- L'exercice 2 montre que l'on peut annuler la constante asymptotique linéaire dans la méthode de Newton pondérée appliquée à la recherche de la racine multiple, et garantir par la même occasion **une convergence bien plus que linéaire**. Mais pour cela il faut disposer de **l'ordre de multiplicité exacte** de la racine.
- Cet exercice indique aussi que si l'on ne dispose que d'une **approximation de l'ordre de multiplicité**, on aura une convergence linéaire mais avec une plus petite constante asymptotique d'erreur et par conséquent, un bien **plus grand gain de chiffres significatifs par itération**, que dans une méthode de Newton non pondérée appliquée au même problème.

3.6 Méthode de la sécante

3.6.1 principe

Cette méthode suit le même principe que la méthode de Newton avec pour la seule différence qu'au lieu d'approcher f par son développement limité d'ordre 1, au voisinage de x_0 , on utilise cette fois le polynôme interpolateur de Lagrange de degré 1 associé aux points x_1, x_0 . Soit $P_f(x) = f(x_1) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$.

Et on construit x_2 comme racine de ce polynôme. Soit

$$x_2 = x_1 - f(x_1) \frac{x_1 - x_0}{f(x_1) - f(x_0)}$$

Puisque la méthode de Newton donnerait

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

On remarque que cette méthode se déduit de la méthode de Newton en remplaçant la dérivée de f en x_1 par son approximation

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

Le schéma obtenu est le suivant

$$\begin{cases} x_{n+2} &= x_{n+1} - f(x_{n+1}) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)}, \quad n = 0, \dots, \\ x_0, x_1 &\text{donnés} \end{cases} \quad (11)$$

3.6.2 Interprétation géométrique

L'opération $x_{n+2} = x_{n+1} - f(x_{n+1}) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)}$ se traduit géométriquement ainsi: A partir de $(x_{n+1}, 0)$, on se déplace verticalement jusqu'à la courbe $y = f(x)$, puis le long de la droite reliant ce point (à savoir $(x_{n+1}, f(x_{n+1}))$) au point $(x_n, f(x_n))$ jusqu'à la droite $y = 0$ (l'axe des abscisses) pour définir le point $(x_{n+2}, 0)$.

3.6.3 Algorithme

Listing 5: Méthode de Sécante

```
- Initialisation
- Soit  $x_0, x_1$  deux approximations initiales et  $\varepsilon > 0$  une étolrance
- Poser  $k = 1$ 
- Itérations
- Tant que  $|f(x_k)| > \varepsilon$  et  $k$  pas trop grand faire
  -  $x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$ 
  -  $k = k + 1$ 
- Fin faire
```

Coût : chaque itération nécessite : **1 produit, 1 division, 1 évaluation** de fonctions.

3.6.4 Convergence

Proposition 3.6.1.

Soit (x_n) la suite produite par application de la méthode de sécante à une fonction f . Supposons que (x_n) converge vers x^* .

- Si f est C^1 , alors x^* est racine de f .
- Si on a
 - m entier, $m \geq 1$;
 - $f \in C^{m+1}$, $f^{(m)}(x^*) \neq 0$, $f^{(i)}(x^*) = 0$ pour $i \leq m - 1$;
 - $\lim_{n \rightarrow \infty} \frac{x^* - x_{n+1}}{x^* - x_n}$ existe;

alors pour $m > 1$ cette limite est racine de $K^m + K^{m-1} - 1 = 0$ et elle appartient à l'intervalle $[\frac{1}{2^{m-1}}, \frac{1}{2^m}]$. Pour $m = 1$, cette limite est 0.

- Si p est un entier, $p \geq 2$, si $f \in C^{p+1}$, $f'(x^*) \neq 0$, $f^{(p)}(x^*) \neq 0$, $f^{(i)}(x^*) = 0$ pour $2 \leq i \leq p - 1$, alors
 - K_1 existe et est égale à 0;
 - $\lim_{n \rightarrow \infty} \frac{x^* - x_{n+2}}{(x^* - x_{n+1})(x^* - x_n)^{p-1}} = (-1)^{p-1} \frac{f^{(p)}(x^*)}{p! f'(x^*)}$;
 - Appelons ω la racine positive de $\omega^2 - \omega - (p - 1) = 0$. Si $\lim_{n \rightarrow \infty} \frac{|x^* - x_{n+1}|}{|x^* - x_n|^\omega}$ existe (réel ou infini), alors

$$\lim_{n \rightarrow \infty} \frac{|x^* - x_{n+1}|}{|x^* - x_n|^\omega} = \left| \frac{f^{(p)}(x^*)}{p! f'(x^*)} \right|^{\frac{1}{\omega}}$$

Démonstration.

(admise pour l'instant) et du fait de sa longueur pourra être transformée en exercices.

Remarque 3.6.1.

- Dans la plupart des applications, on n'a pas accès aux dérivées d'ordre > 1 de f . Le résultat précédent nous renseigne alors que la méthode de sécante **converge au moins à l'ordre** $\frac{1+\sqrt{5}}{2} \approx 1,618033989$ (obtenu pour $p = 2$) vers une racine simple de f .
- Mais comparée à la méthode de Newton, où la convergence vers la racine simple est d'ordre p où p est le plus petit entier $p \geq 2$ tel que $f^{(p)}(x^*) \neq 0$., la méthode de sécante dans la même configuration convergera à l'ordre $\omega = \frac{1 + \sqrt{1 + 4(p - 1)}}{2}$.

3.6.5 Analyse

Note 3.6.1 (Analyse et Illustration).

- Avantages:
 - Toujours plus rapide que la méthode de fausse position.
 - Hypothèses de départ moins contraignantes que celles de dichotomie, de fausse position.
 - Pas d'évaluation de la dérivée de fonction et donc convergence plus rapidement (en terme de temps de calcul, pas en nombre d'itérations) que la méthode de Newton.
- Inconvénients:
 - Instabilité numérique en convergence avancée, du fait que $x_{n-1} \approx x_n$, $f(x_{n-1}) \approx f(x_n)$. Situation qui ne se présente pas dans la méthode de fausse position, où $f(x_{n-1})$ et $f(x_n)$ sont de signes opposés.
 - Convergence plus lente vers les racines multiples que celle de la méthode de dichotomie.
- Exemples et illustration :

Traiter l'exercice de la fiche de TP associée à ce cours.

Remarque 3.6.2 (Montée en ordre en cas de racine d'ordre de multiplicité $m > 1$).

- On fait les mêmes remarques que dans la méthode de Newton.
- On peut appliquer la méthode de sécante à la fonction $\frac{f}{f'}$ (éventuellement prolongée par continuité). on obtient le schéma

$$x_{n+2} = x_{n+1} - f(x_{n+1})f'(x_n) \frac{x_{n+1} - x_n}{f(x_{n+1})f'(x_n) - f(x_n)f'(x_{n+1})}$$

- Si l'on dispose de l'ordre de multiplicité $m > 1$ ou une estimation r , on peut faire usage de la méthode de sécante pondérée :

$$x_{n+2} = x_{n+1} - r f(x_n) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)}$$