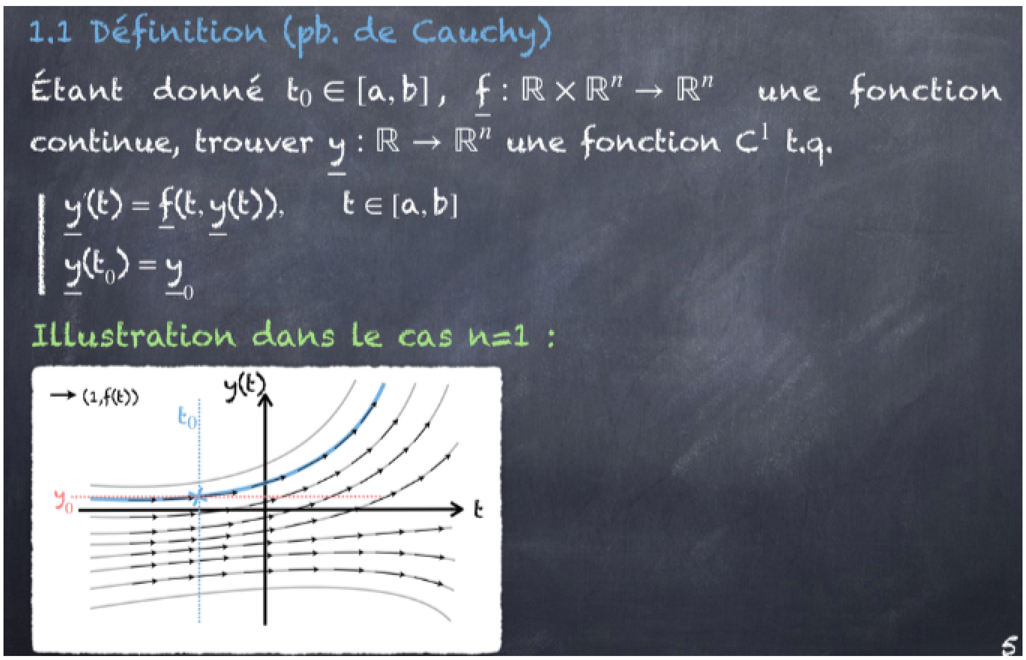
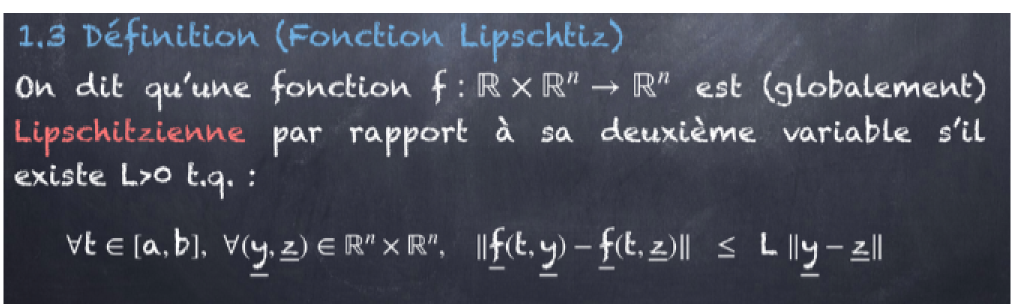
Part I

Méthode numérique pour résoudre des équations non linéaires

1 Théorème de Cauchy-Lipschitz (Rappel)





1.3 Théorème (de Cauchy-Lipschtiz)

Si f est continue et Lipschitzienne par rapport à sa 2ème variable alors le problème de Cauchy admet une unique solution.

(Rappel de la) Preuve (dans le cas n=1) :

Point de départ : y sol. ssi $y(t) = y(t_0) + \int_{t_0}^{t} f(s, y(s))\, ds$

Unicité : Soient y_1 et y_2 deux solutions du problème de Cauchy, alors on a :

$$\left| y_1(t) - y_2(t) \right| \leq \int_{t_0}^{t} \left| f(s, y_1(s)) - f(s, y_1(s)) \right| ds$$

Suite de la preuve :

Unicité : Soient y_1 et y_2 deux solutions du problème de Cauchy, alors on a :

$$\left| y_1(t) - y_2(t) \right| \leq L \int_{t_0}^{t} \left| y_1(s) - y_1(s) \right| ds$$

1.3 Lemme de Grönwall (admis)

Soient Φ et Ψ de $[t_0, t] \to \mathbb{R}$ deux fonctions continues et positives vérifiant :

$$\Phi(t) \leq K + L \int_{t_0}^{t} \Psi(s)\Phi(s)\, ds \quad \text{alors} \quad \Phi(t) \leq K e^{L \int_{t_0}^{t} \Psi(s)\Phi(s)\, ds}$$

$$\boxed{\Rightarrow \quad \left| y_1(t) - y_2(t) \right| \leq 0 \Rightarrow y_1 = y_2}$$

Existence : y sol. ssi $y(t) = y(t_0) + \int_{t_0}^{t} f(s, y(s))\, ds$

On voit alors le problème de Cauchy comme une **équation de point fixe**

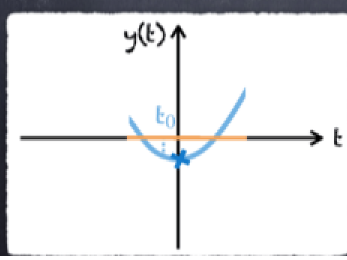
$$\mathscr{L}y = y$$

où l'opérateur $\mathscr{L} : B(a,b) = \left(C^0[a,b], \|\cdot\|_\infty\right) \to B(a,b)$.

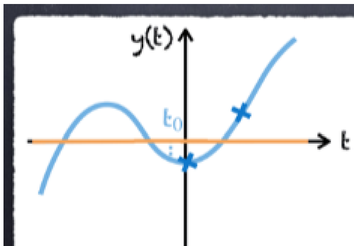
Or, on a : $\|\mathscr{L}y_1 - \mathscr{L}y_2\|_{L^\infty(a,b)} \leq L(b-a)\, \|y_1 - y_2\|_{L^\infty(a,b)}$

Si $L(b-a) < 1$, on a alors une **contraction**, et on en déduit qu'il existe une unique solution en appliquant le **Thm. de point fixe**.

Si $L(b-a) > 1$, l'idée est alors d'appliquer la même méthode sur un intervalle $[t_0 - \varepsilon, t_0 + \varepsilon]$ avec $2\varepsilon L < 1$.

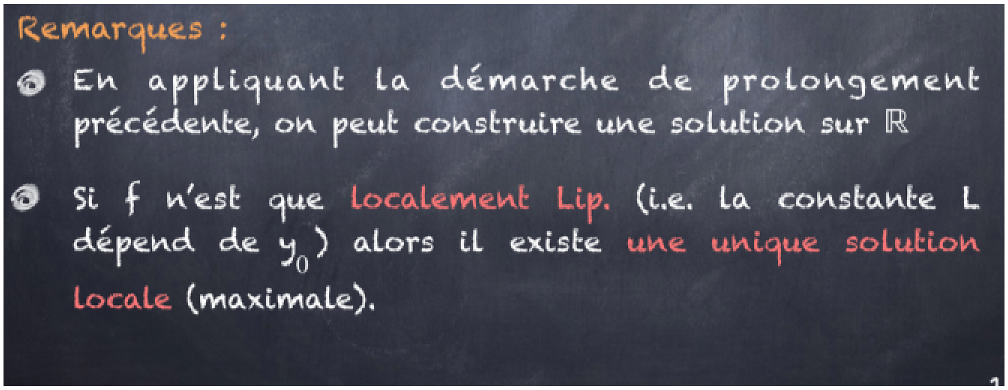


On applique le même résultat sur $[t_0, t_0 + 2\varepsilon]$ avec comme **condition de Cauchy** $y^1(t_0 + \frac{\varepsilon}{2}) = y(t_0 + \frac{\varepsilon}{2})$ pour construire y_1.



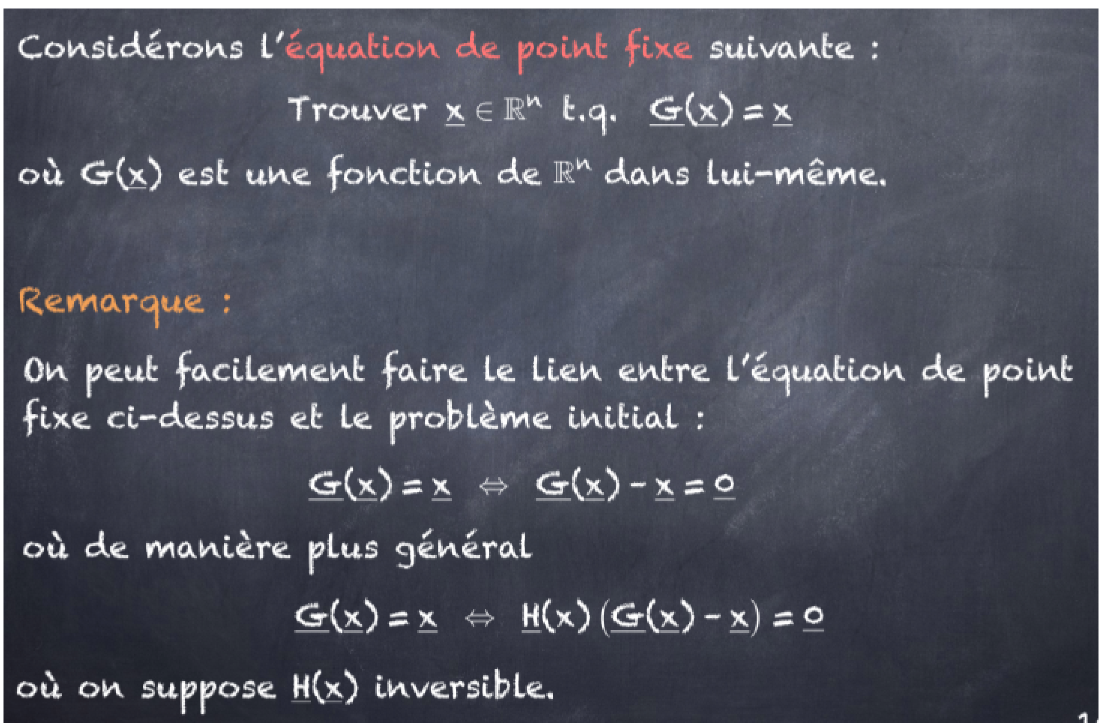
Et ainsi de suite : $y^2(t_0 + \frac{3\varepsilon}{2}) = y^1(t_0 + \frac{3\varepsilon}{2})$

Remarque : La solution étant unique sur $[a,b]$, on déduit aisément que $y^1 = y^2 = \cdots = y$.



2 Les équations non linéaires

2.1 Les équations points fixes



1.1 Définition

On dit que l'application $\underline{G}$ est contractante ssi il existe une constante K t.q. $0 < K < 1$ et

$$\forall \left(\underline{x}, \underline{y}\right) \in \mathbb{R}^n, \quad \|\underline{G}(\underline{x}) - \underline{G}(\underline{y})\| \leq K\|\underline{x} - \underline{y}\|$$

1.2 Théorème (Point fixe)

L'application $\underline{G}$ admet un unique point fixe x^* et la suite $\underline{x}^{n+1} = \underline{G}(\underline{x}^n)$ converge vers ce point fixe.

1.3 Définition

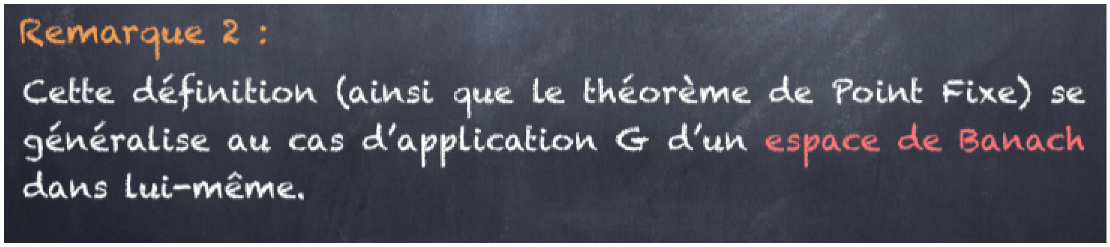
On dit que l'application $\underline{G}$ est différentiable au point $\underline{x}_0 \in \mathbb{R}^n$ s'il existe une application linéaire $d_{\underline{x}_0}\underline{G}$ vérifiant :

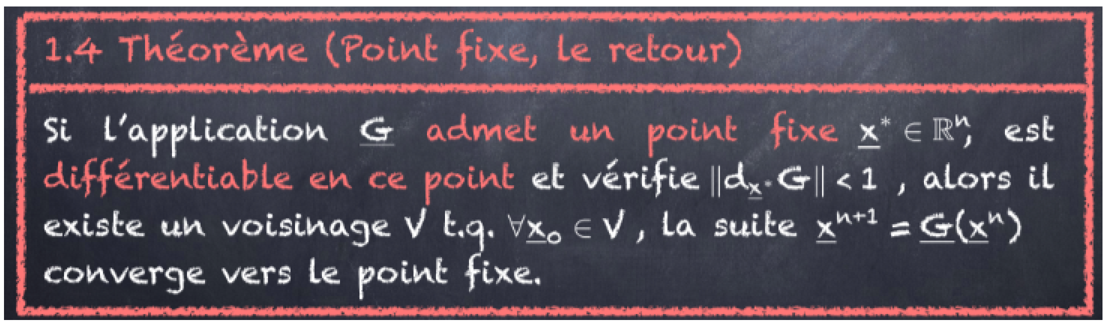
- $\forall \underline{h} \in \mathbb{R}^n, \quad \underline{G}(\underline{x}_0 + \underline{h}) = \underline{G}(\underline{x}_0) + d_{\underline{x}_0}\underline{G}(\underline{h}) + \underline{\Phi}(\underline{x}_0, \underline{h})$
- $\lim\limits_{\underline{h} \to \underline{0}} \dfrac{\|\underline{\Phi}(\underline{x}_0, \underline{h})\|}{\|\underline{h}\|} = 0$

Remarque :

L'application linéaire $d_{\underline{x}_0}\underline{G} : \mathbb{R}^n \to \mathbb{R}^n$ peut être représentée par sa matrice Jacobienne.

De plus, cette application est appelée la différentielle et elle généralise la notion de dérivée.





Preuve

Dire que G est différentiable en x^* est équivalent à dire que G est continue en x^*

$$\iff lim_{x \to x^*} d_x G = d_{x^*} G$$

$$\iff \forall \epsilon > 0, \exists \eta > 0 \; tq \; \left|\left|x - x^*\right|\right| < \eta \implies \left|\left|d_x G - d_{x^*G}\right|\right| < \epsilon$$

$$\iff d_x G \in B(d_{x^*}G, \epsilon)$$

Posons $l = 1 - ||d_x G||$ et prenons $\epsilon < l$ alors $\exists \eta$ tq $||x - x^*|| < \eta \implies ||d_x G|| < ||d_{x^*}G|| + \epsilon < 1$

car $||d_x G|| = ||d_x G + d_{x^*}G - d_{x^*}G|| < ||d_{x^*}G|| + ||d_x G - d_{x^*}G|| < ||d_{x^*}G|| + \epsilon < 1$.

$\forall x, y \in E$,

$$\begin{aligned} \left|\left|G(x) - G(y)\right|\right| &= \left|\left|G(x - y + y) - G(y)\right|\right| \\ &= \left|\left|G(y) - G(y) + d_y G(x - y) + \Phi_y(x - y)\right|\right| \end{aligned}$$

où $\Phi_y(h)$ est tel que $lim_{h \to 0} \frac{\Phi_y(h)}{||h||} = 0$

Ainsi

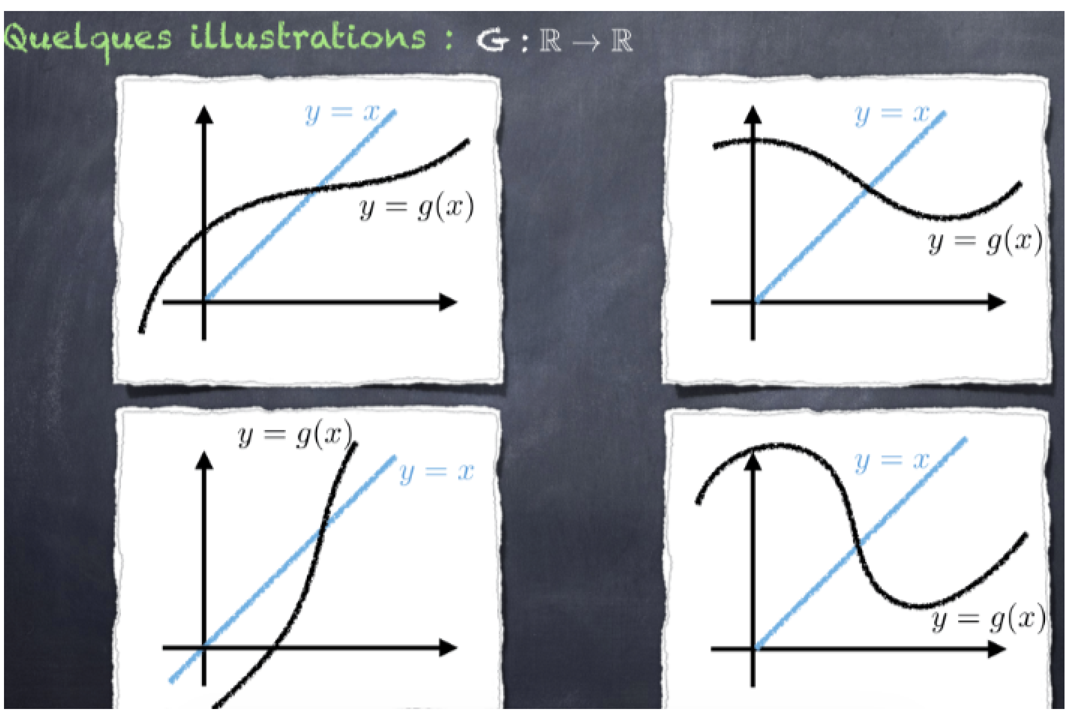
$$\big|\big|G(x)-G(y)\big|\big| \leq \big|\big|d_yG\big|\big|\big|\big|x-y\big|\big|+\frac{\Phi_y(x-y)}{||x-y||}\big|\big|x-y\big|\big|$$
$$\leq (\big|\big|d_yG\big|\big|+\frac{\Phi_y(x-y)}{||x-y||})\big|\big|x-y\big|\big|$$

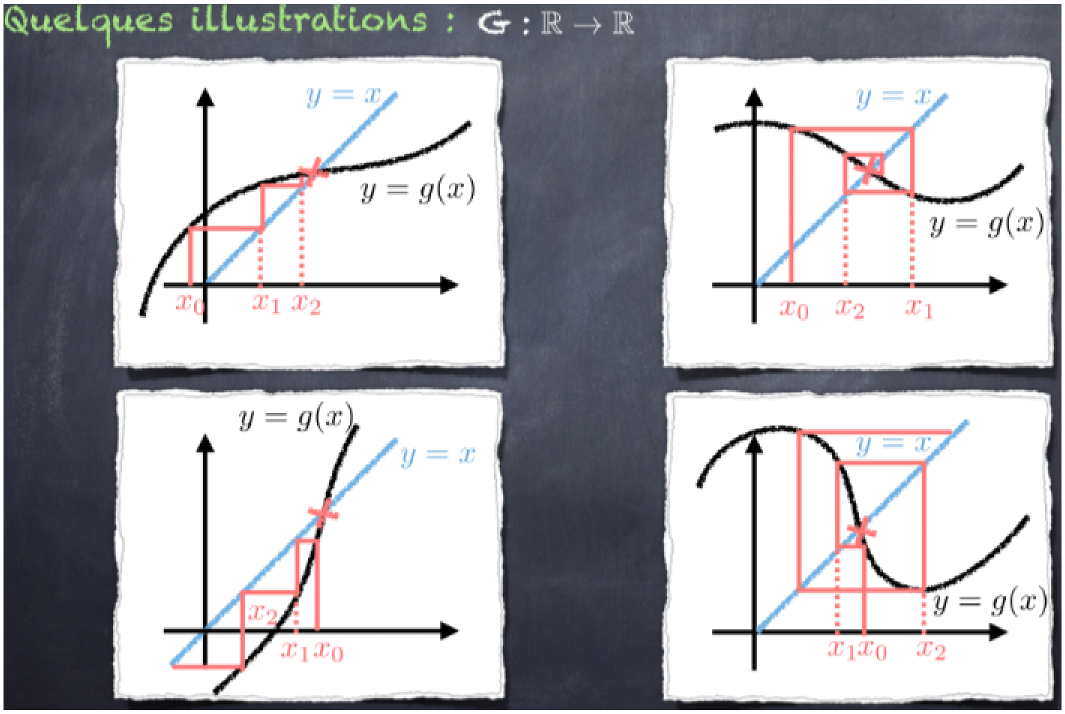
On a $\forall y \in B^f(x^*,\eta)$, $||d_yG|| < 1$
Notons $max_{y\in B^f(x^*,\eta)}||d_yG|| = M^d < 1$
posons $\widetilde{\epsilon} = \frac{1-M^d}{2} > 0$ alors $\exists \widetilde{\eta}$ tel que $||x-y|| < \widetilde{\eta} \implies \frac{\Phi_y(x-y)}{||x-y||}|| < \widetilde{\epsilon}$
Pour résumer, si $(x,y) \in (B(x^*, min(\eta,\widetilde{\eta})))^2$ alors $||G(x)-G(y)|| \leq k||x-y||$ où $k = M^d + \widetilde{\epsilon} < 1$
Ainsi, on aura

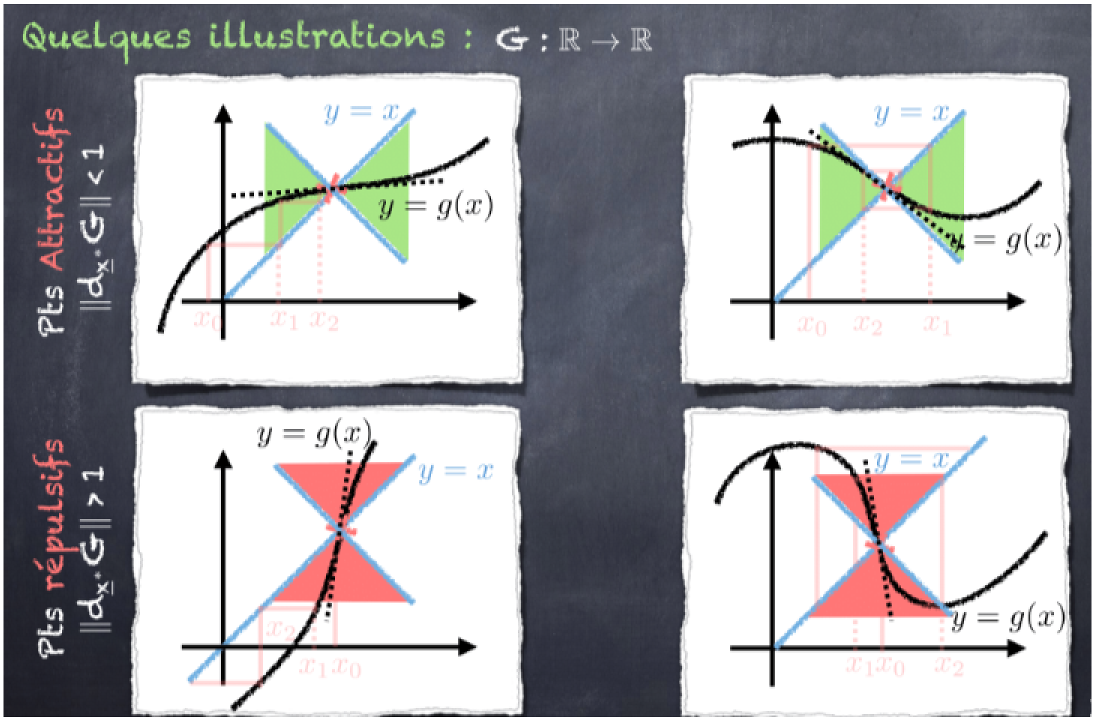
$$G : B(x^*, min(\eta,\widetilde{\eta})) \to B(x^*, min(\eta,\widetilde{\eta}))$$

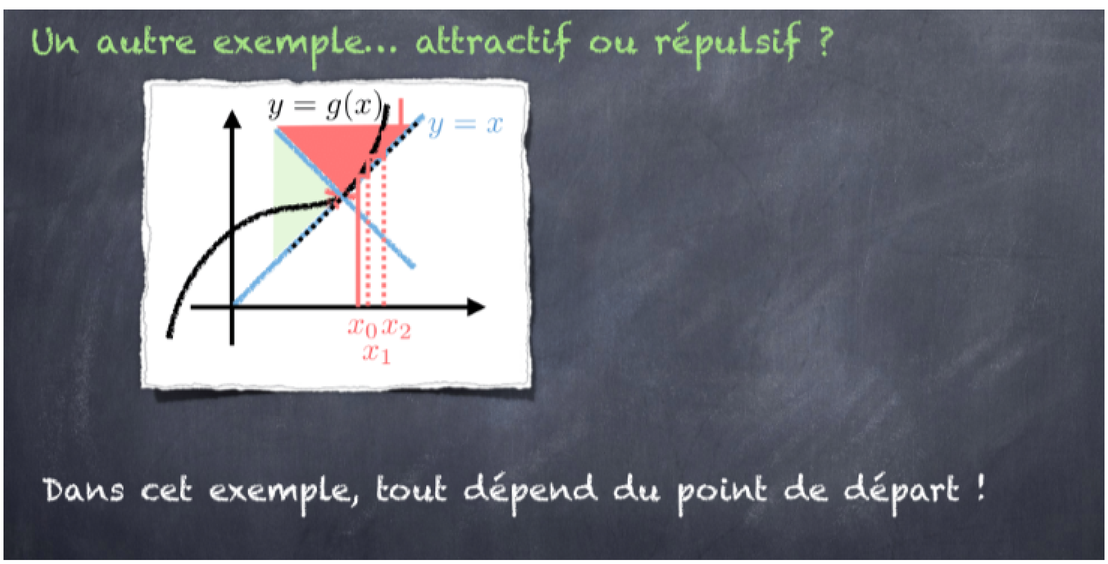
qui sera une contraction et par le théorème du point fixe, on déduit que $x_{n+1} = G(x_n) \to x^*$ si $x_0 \in B(x^*, min(\eta,\widetilde{\eta}))$
Notons que $\forall x \in B(x^*, min(\eta,\widetilde{\eta}))$, $||G(x^*)-x^*|| \leq \underbrace{(\big|\big|d_{x^*}G\big|\big|+\frac{\Phi_y(x-x^*)}{||x-x^*||})}_{<1}\underbrace{\big|\big|x-x^*\big|\big|}_{\leq min(\eta,\widetilde{\eta})}$

d'où $G(x) \in B(x^*, min(\eta,\widetilde{\eta}))$




Quelques illustrations : G : ℝ → ℝ
y = x
y = g(x)
x0 x1 x2
y = x
y = g(x)
x0 x2 x1
y = g(x)
y = x
x2
x1 x0
y = x
y = g(x)
x1 x0 x2


Quelques illustrations : G : ℝ → ℝ
Pts Attractifs
‖d_{x*}G‖ < 1
Pts répulsifs
‖d_{x*}G‖ > 1
y = x
y = g(x)
x0 x1 x2


Un autre exemple... attractif ou répulsif ?
y = g(x)
y = x
x0 x2
x1
Dans cet exemple, tout dépend du point de départ !

Algorithme de point fixe :

$\underline{\mathbf{x}}_1 \leftarrow \underline{\mathbf{G}}(\underline{\mathbf{x}}_0)$

Tant que $\|\underline{\mathbf{x}}_0 - \underline{\mathbf{x}}_1\| \geq \varepsilon$

- $\underline{\mathbf{x}}_0 \leftarrow \underline{\mathbf{x}}_1$
- $\underline{\mathbf{x}}_1 \leftarrow \underline{\mathbf{G}}(\underline{\mathbf{x}}_0)$

Équivalent mathématiques

$$\underline{\mathbf{x}}^{n+1} = \underline{\mathbf{G}}(\underline{\mathbf{x}}^{n})$$

Remarque :

Lorsque le test de convergence est satisfait, on a alors :

$$\begin{aligned}
\|\underline{\mathbf{x}}_n - \underline{\mathbf{x}}^*\| &\leq \|\underline{\mathbf{x}}_n - \underline{\mathbf{x}}_{n+1}\| + \|\underline{\mathbf{x}}_{n+1} - \underline{\mathbf{x}}^*\| \\
&\leq \epsilon + \|\underline{\mathbf{G}}(\underline{\mathbf{x}}_n) - \underline{\mathbf{G}}(\underline{\mathbf{x}}^*)\| \\
&\leq \epsilon + K\|\underline{\mathbf{x}}_n - \underline{\mathbf{x}}^*\|
\end{aligned}$$

d'où : $$\|\underline{\mathbf{x}}_n - \underline{\mathbf{x}}^*\| \leq \frac{\epsilon}{1-K} \qquad (0 < K < 1)$$

1.5 Définition (ordre d'une suite)

On dit que la suite $(\underline{\mathbf{x}}_n)_n$ est d'ordre r s'il existe une constante $C > 0$ t.q. :

$$\lim_{n \to +\infty} \frac{\|\underline{\mathbf{x}}_{n+1} - \underline{\mathbf{x}}^*\|}{\|\underline{\mathbf{x}}_n - \underline{\mathbf{x}}^*\|^r} = C$$

Remarque :

Cela revient à dire $\|\underline{\mathbf{x}}_{n+1} - \underline{\mathbf{x}}^*\| \sim C\|\underline{\mathbf{x}}_n - \underline{\mathbf{x}}^*\|^r$

Cette notion permet de « mesurer » la vitesse de convergence d'une suite (quand elle converge).

Si les hypothèses du Thm. de point fixe 1.4 sont vérifiées, on a alors :

$$\underline{x}_{n+1} = G(\underline{x}^* + \underline{x}_n - \underline{x}^*)$$

$$= \underline{x}^* + d_{\underline{x}^*}\underline{G}(\underline{x}_n - \underline{x}^*) + \underline{\Phi}(\underline{x}^*, \underline{x}_n - \underline{x}^*)$$

d'où : $\|\underline{x}_{n+1} - \underline{x}^*\| = \|(d_{\underline{x}^*}G)(\underline{x}_n - \underline{x}^*) + \underline{\Phi}(\underline{x}^*, \underline{x}_n - \underline{x}^*)\|$

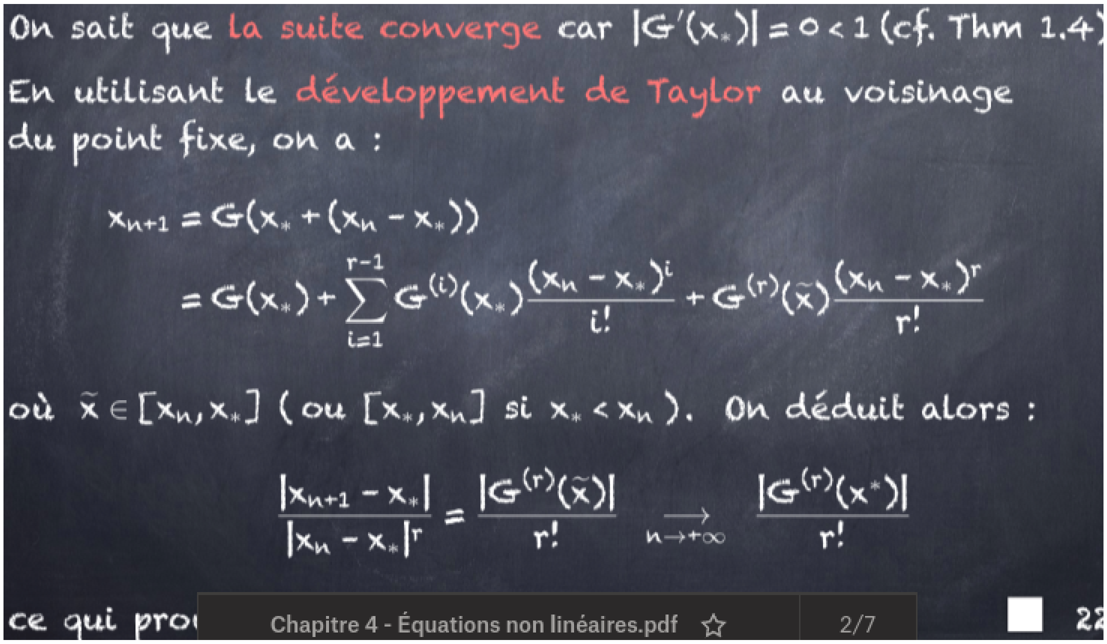
et dont on déduit que la suite est au moins d'ordre 1

1.6 Théorème

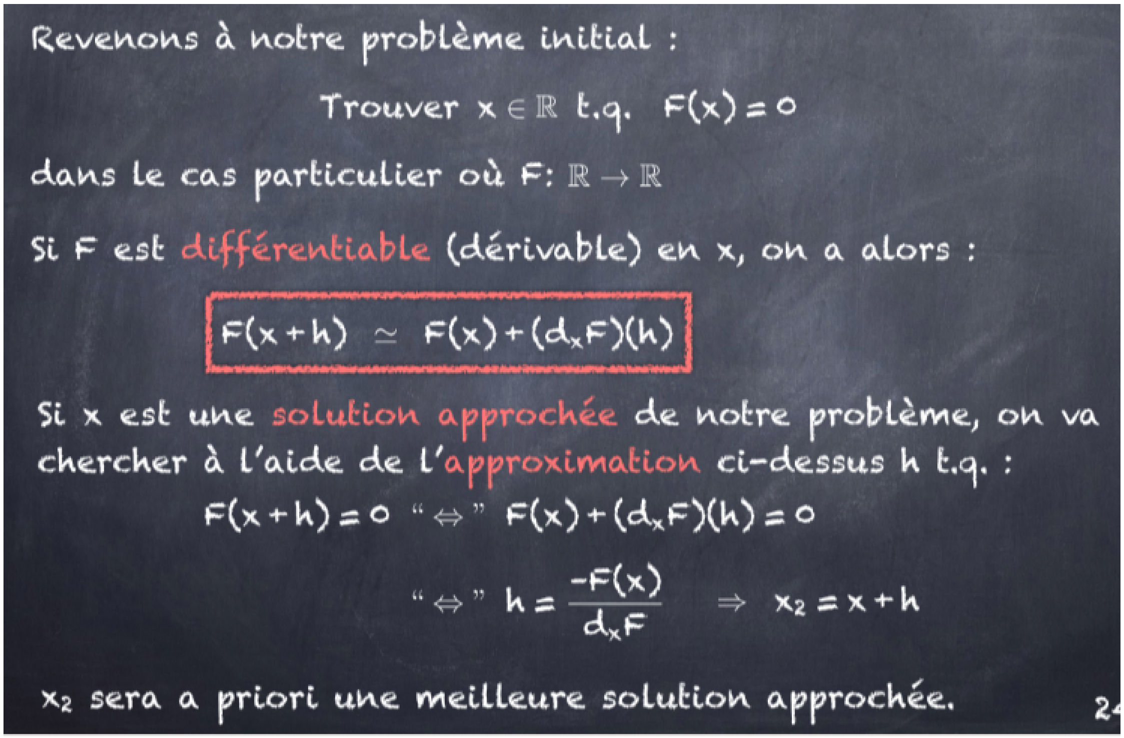
Si l'application $\underline{G}$ admet un point fixe $\underline{x}^* \in \mathbb{R}^n$, est r fois continûment différentiable et vérifie :

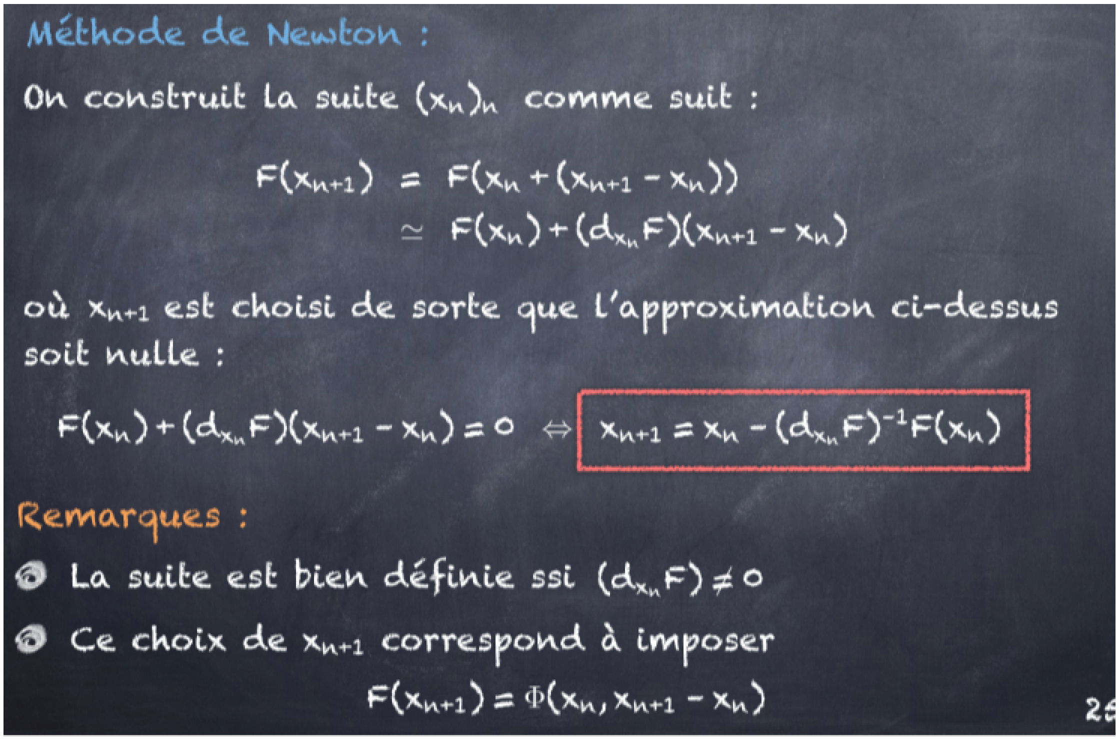
$$d_{\underline{x}^*}\underline{G} = \cdots = d_{\underline{x}^*}^{r-1}\underline{G} = \underline{0} \quad \text{et} \quad d_{\underline{x}^*}^{r}\underline{G} \neq \underline{0}$$

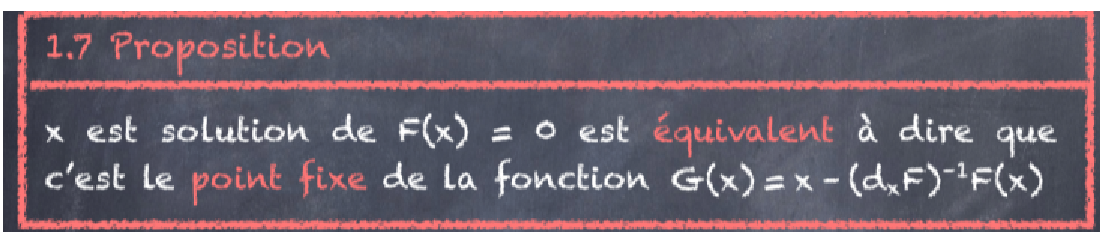
alors il existe un voisinage V t.q. $\forall \underline{x}_0 \in V$, la suite générée par $\underline{x}^{n+1} = \underline{G}(\underline{x}^n)$ converge à l'ordre r vers le point fixe.


On sait que la suite converge car $|G'(x_*)| = 0 < 1$ (cf. Thm 1.4
En utilisant le développement de Taylor au voisinage du point fixe, on a :
$$x_{n+1} = G(x_* + (x_n - x_*))$$
$$= G(x_*) + \sum_{i=1}^{r-1} G^{(i)}(x_*) \frac{(x_n - x_*)^i}{i!} + G^{(r)}(\tilde{x}) \frac{(x_n - x_*)^r}{r!}$$
où $\tilde{x} \in [x_n, x_*]$ (ou $[x_*, x_n]$ si $x_* < x_n$). On déduit alors :
$$\frac{|x_{n+1} - x_*|}{|x_n - x_*|^r} = \frac{|G^{(r)}(\tilde{x})|}{r!} \xrightarrow[n \to +\infty]{} \frac{|G^{(r)}(x^*)|}{r!}$$
ce qui pro
Chapitre 4 - Équations non linéaires.pdf
2/7

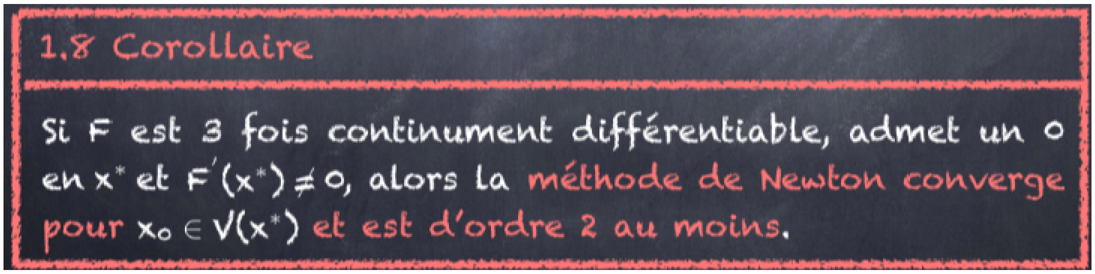
2.2 La méthode de Newton



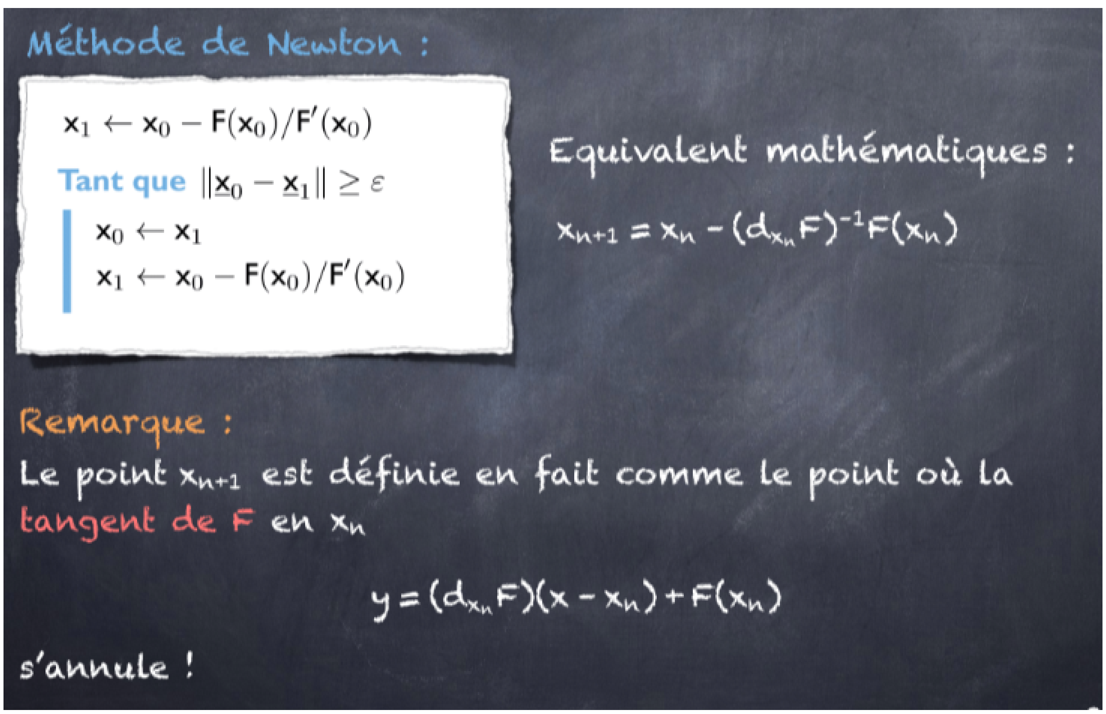


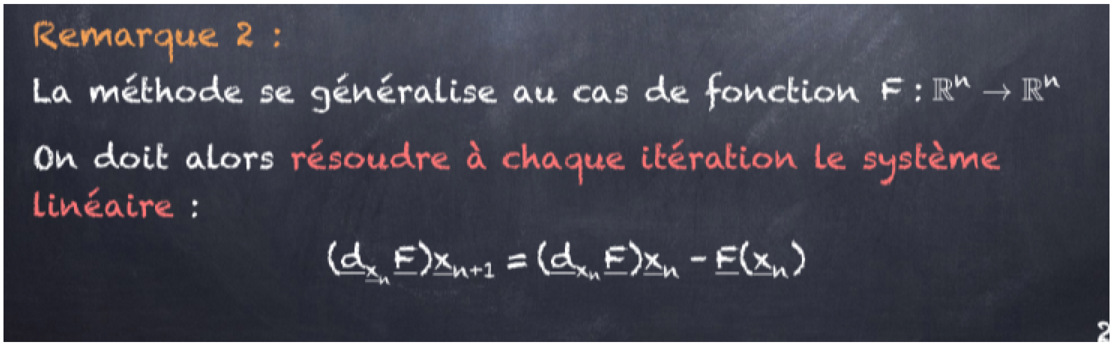


Preuve
$F(x^*) = 0$.
$g(x^*) = x^*$où $g(x) = x - \frac{f(x)}{f'(x)}$ revient à $x^* - \frac{f(x^*)}{f'(x^*)} = x^* \iff f(x^*) = 0$



On a $g(x) = x - \frac{f(x)}{f'(x)}$ donc $g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{(f'(x))^2} = 1 - 1 - \frac{f''(x)}{(f'(x))^2}f(x) \iff g'(x^*) = \frac{f''(x^*)}{(f'(x^*))^2}f(x^*) = 0$ donc la méthode est convergente à l'ordre 2 au moins




Remarque 2 :
La méthode se généralise au cas de fonction $F : \mathbb{R}^n \to \mathbb{R}^n$
On doit alors résoudre à chaque itération le système linéaire :
$(\underline{d}_{\underline{x}_n}\underline{F})\underline{x}_{n+1} = (\underline{d}_{\underline{x}_n}\underline{F})\underline{x}_n - \underline{F}(\underline{x}_n)$

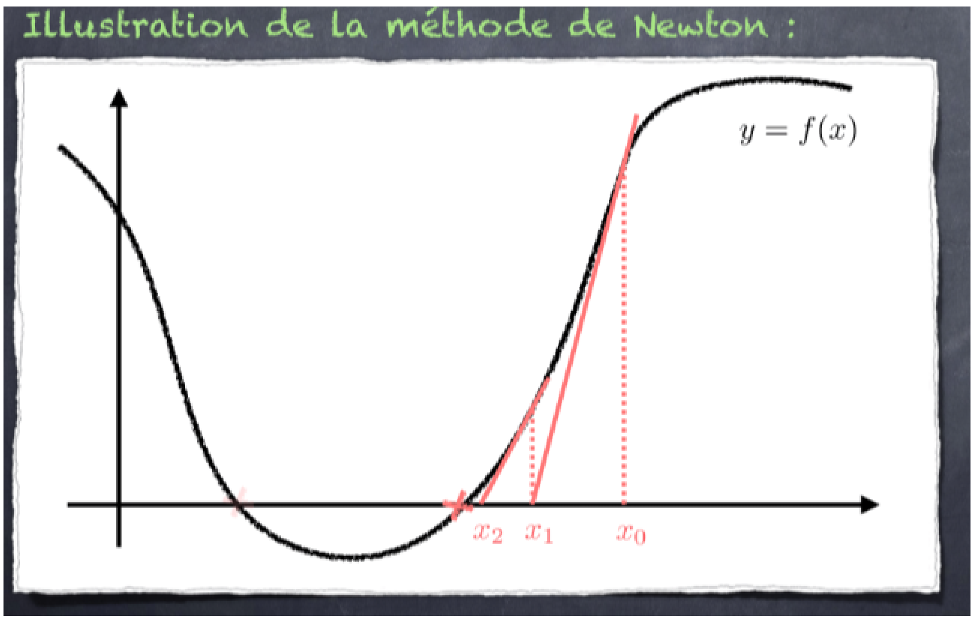
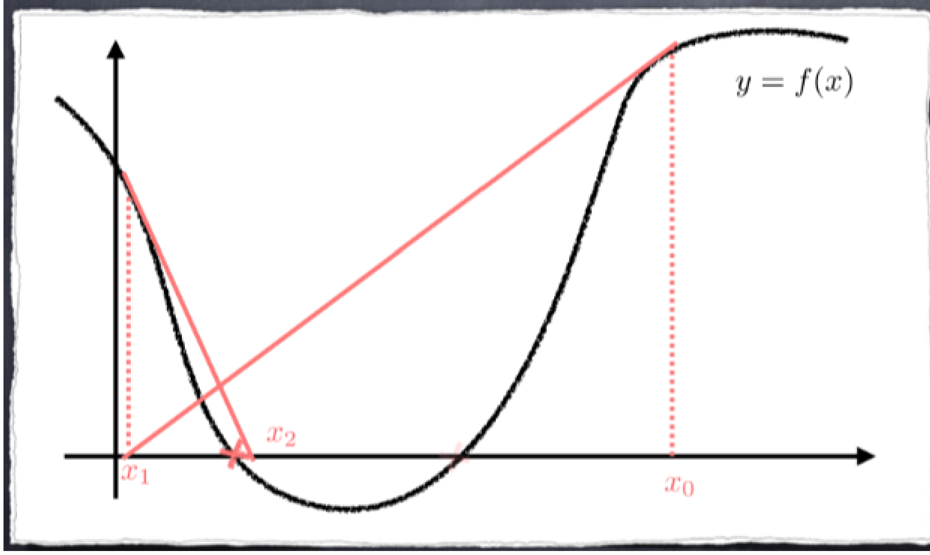


Illustration de la méthode de Newton :
$y = f(x)$
x_2
x_1
x_0

Illustration de la méthode de Newton :



Méthode de Newton :

On génère la suite $(x_n)_n$ ainsi : $x_{n+1} = x_n - (d_{x_n}F)^{-1}F(x_n)$

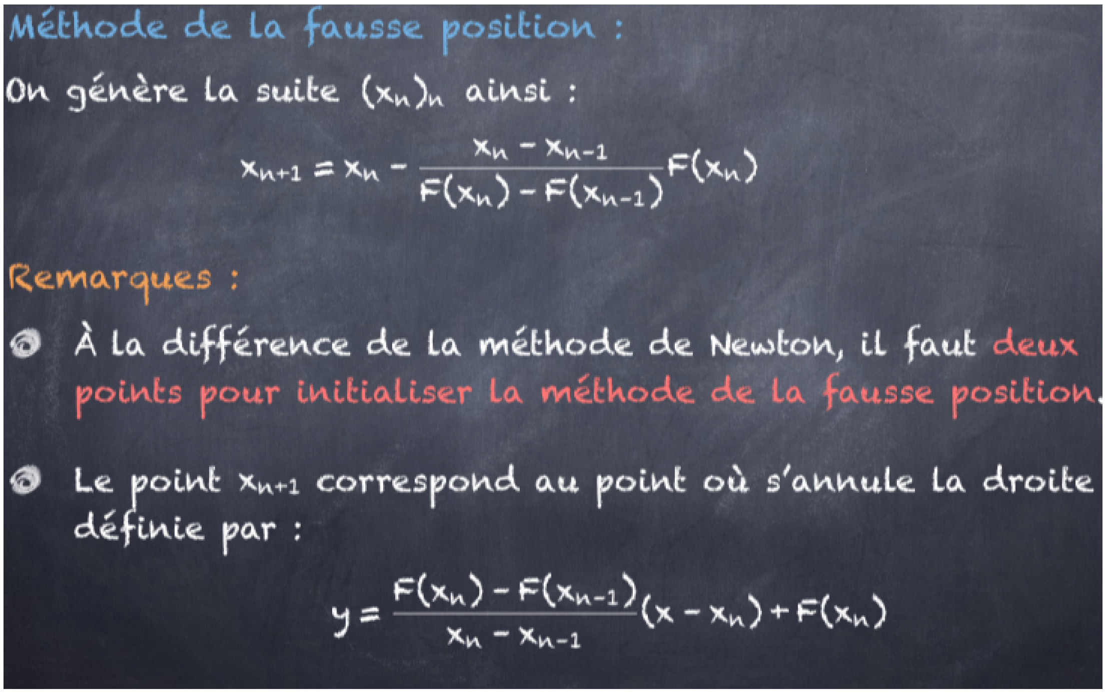
Dans certaines situations, le calcul de $(d_{x_n}F)^{-1}$ est couteux voir impossible.

Une idée est alors d'utiliser l'approximation de la dérivée :

$$(d_{x_n}F) \simeq \frac{F(x_n) - F(x_{n-1})}{x_n - x_{n-1}}$$

On déduit alors la méthode de la fausse position

2.3 Méthode de la fausse position



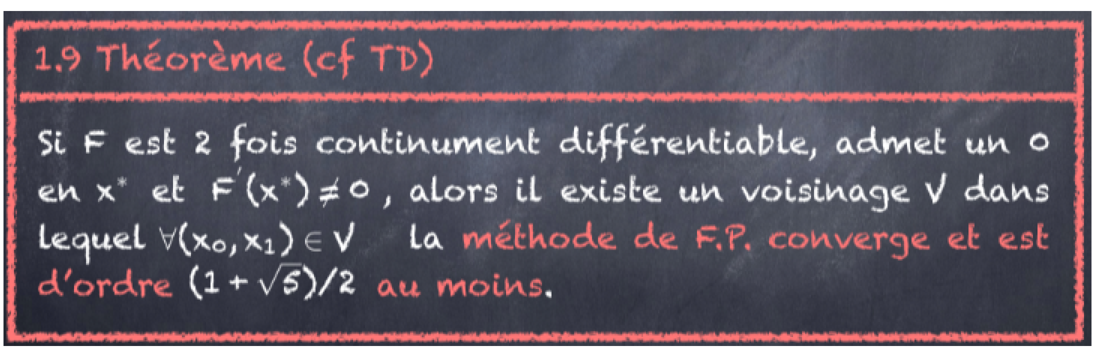
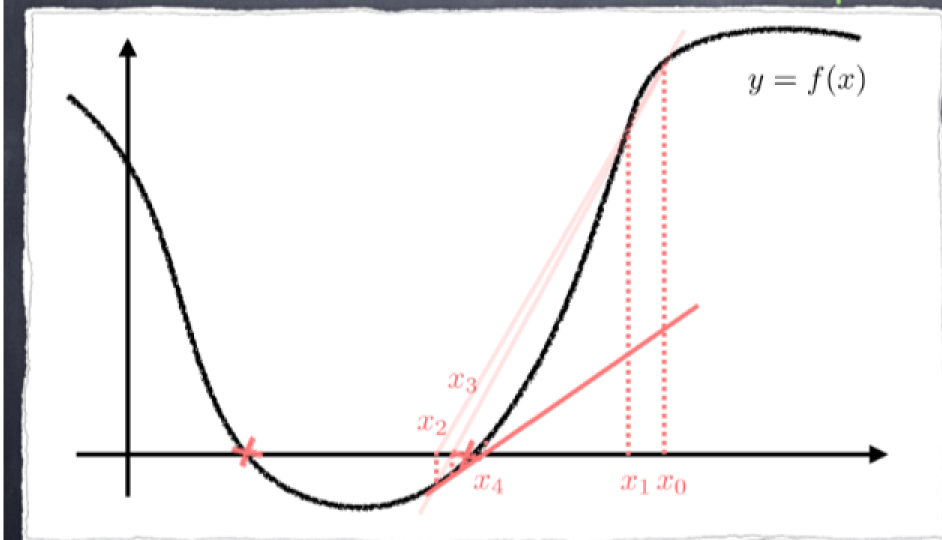


Illustration de la méthode de la fausse position :



Méthode de la fausse position :

Tant que $\|\underline{x}_0 - \underline{x}_1\| \geq \varepsilon$

$x_2 \leftarrow x_1 - (x_1 - x_0) \times F(x_1)/(F(x_1) - F(x_0))$
$x_0 \leftarrow x_1$
$x_1 \leftarrow x_2$

Equivalent mathématiques : $x_{n+1} = x_n - \dfrac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n)$

Remarque :

Pour la mise en oeuvre de la méthode, en particulier si le calcul de F(x) est couteux, il est intéressant de remarquer qu'à chaque étape une seule évaluation de F est nécessaire...

35

Remarques de conclusion

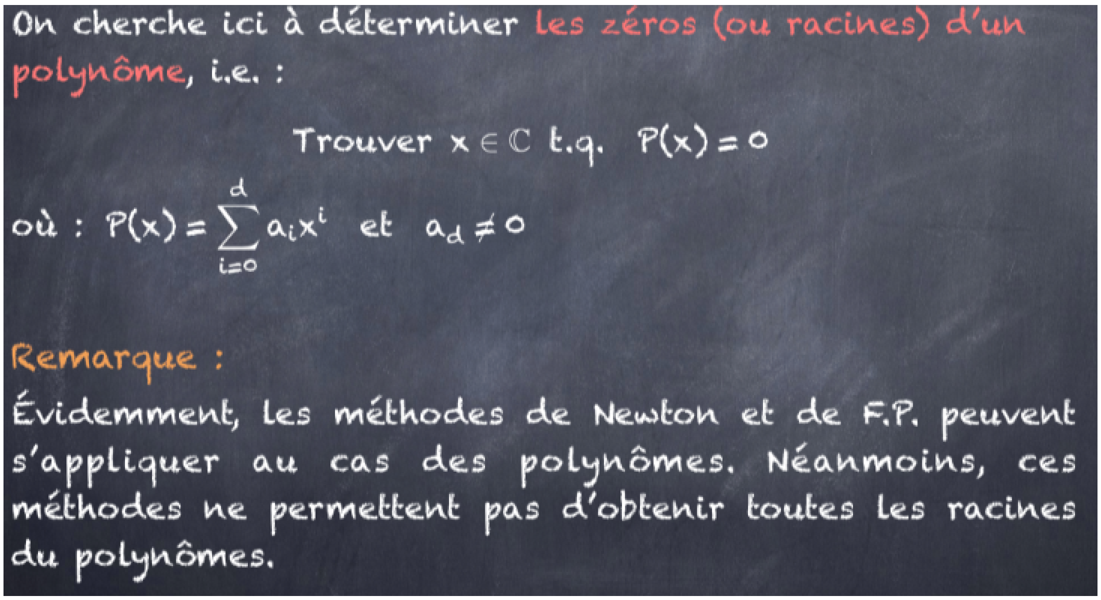
- On est assuré de la convergence des méthodes de Newton et de F.P. seulement lorsque l'initialisation est dans un voisinage proche de la solution... Il faut donc avoir a priori une bonne solution approchée !

- Lorsque la racine est double, i.e. $F(x_*) = d_{x_*}F = 0$ les méthodes de Newton et F.P. convergent mais à un ordre plus petit (i.e. moins vite, cf TD !)

- On peut adapter la méthode de Newton pour résoudre des problèmes de minimisation (cf TD !).

3 Le cas des polynômes

3.1 Algorithme de Horner



On cherche ici à déterminer les zéros (ou racines) d'un polynôme, i.e. :

$$\text{Trouver } x \in \mathbb{C} \text{ t.q. } P(x) = 0$$

où : $P(x) = \sum_{i=0}^{d} a_i x^i$ et $a_d \neq 0$

2.1 Théorème Fondamentale (admis)

Si $d > 0$, alors le polynôme admet au moins une racine complexe. Par conséquent, tout polynôme de degré d admet d racines complexes (possiblement multiples).

Depuis la théorie de Galois, on sait que les équations polynomiales sont résolubles (on peut déterminer les racines) de manière générale si et seulement si le degré du polynôme est inférieur strictement à 5...

Cela motive d'autant plus l'utilisation de méthodes numériques pour déterminer les racines de P.

Voyons tout d'abord comment évaluer un polynôme $P(x)$ en un point s donné :

$$\begin{aligned} P(s) &= a_0 + a_1 s + \cdots + a_d s^d \\ &= a_0 + s\left(a_1 + a_2 s + \cdots + a_d s^{d-1}\right) \\ &= a_0 + s\left(a_1 + s\left(a_2 + \cdots + a_d s^{d-2}\right)\right) \\ &= a_0 + s\left(a_1 + s\left(a_2 + s\left(a_3 + \cdots + s\left(a_{d-1} + s a_d\right) \cdots\right)\right)\right) \end{aligned}$$

L'idée est de factoriser autant que possible s afin de réduire le cout de calcul en évitant de calculer tous les s^i

Voyons tout d'abord comment évaluer un polynôme $P(x)$ en un point s donné :

$$P(s) = a_0 + s\left(a_1 + s\left(a_2 + s\left(a_3 + \cdots + s\left(a_{d-1} + s a_d\right) \cdots\right)\right)\right)$$

2.1 Proposition

En notant $q_{i-1} = a_i + s q_i \ \forall i \in \{1, d\}$ et $q_d = 0$, on a

$$P(x) = (x - s)Q(x) + r_0$$

où $Q(x) = q_0 + q_1 x + \cdots + q_{d-1} x^{d-1}$ et $r_0 = a_0 + s q_0 = P(s)$

Algorithme d'évaluation de Horner :

$q_d \leftarrow 0$

Pour i=d **à** 1

  $q_{i-1} \leftarrow a_i + sq_i$

$r_0 \leftarrow a_0 + sq_0$

Equivalent mathématiques :

$$P(s) = a_0 + s\,(a_1 + s\,(a_2 + s\,\underbrace{(a_3 + \cdots + \underbrace{s\,\underbrace{(a_{d-1} + s a_d)}_{i\,=\,d-1}}_{i\,=\,d-2} \cdots)}_{i\,=\,3}))$$

On appelle division Euclidienne d'un polynôme $P(x)$ par $D(x)$ l'opération consistant à déterminer $Q(x)$ et $R(x)$ t.q. :

$$P(x) = Q(x)D(x) + R(x)$$

où $\deg(R(x)) < \deg(D(x))$.

Exemple :

Calculer la division Euclidienne de :

$$P(x) = 3x^3 - 8x^2 + 9x - 9 \quad \text{par} \quad Q(x) = x - 2$$

La division euclidienne de $P(x)$ par $D(x) = (x-s)$ donne :

$$P(x) = (x-s)Q(x) + R$$

où :

- $P(x) = a_0 + a_1 x + \cdots + a_d x^d$
- $Q(x) = q_0 + q_1 x + \cdots + q_{d-1} x^{d-1}$

  où $\boxed{q_{i-1} = a_i + s q_i \quad \forall i \in \{1, d\} \quad \text{et} \quad q_d = 0}$
- $\boxed{R = r_0 = a_0 + s q_0}$

Pour déterminer une racine de $P(x)$, on cherche s t.q. $R(s) = 0$.

Méthode de Newton-Horner :

La division euclidienne de $P(x)$ par $D(x) = (x-s)$ donne :

$$P(x) = (x-s)Q(x) + R$$

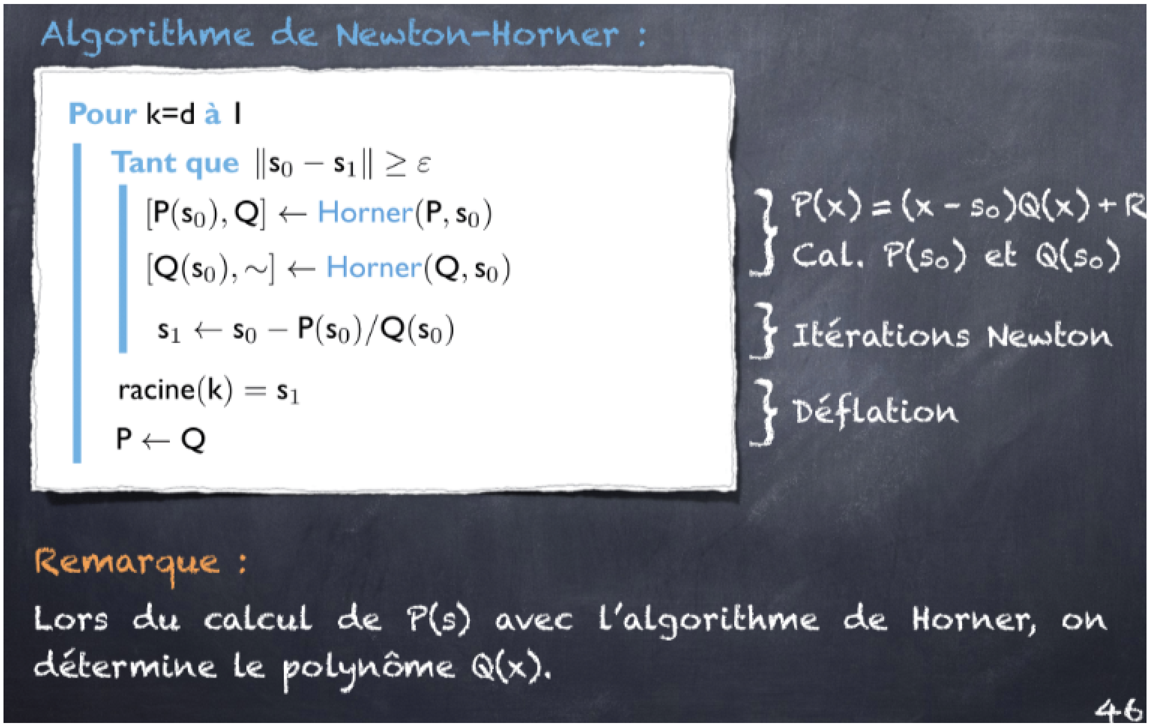
où $R = r_0 = a_0 + s q_0$ et on cherche s t.q. $R(s) = 0$.

Pour résoudre l'équation $R(s) = 0$, on va appliquer la **méthode de Newton** :

$$s_{n+1} = s_n - \frac{R(s_n)}{R'(s_n)} = s_n - \frac{P(s_n)}{Q(s_n)} \qquad \left(= s_n - \frac{P(s_n)}{P'(s_n)} \right)$$

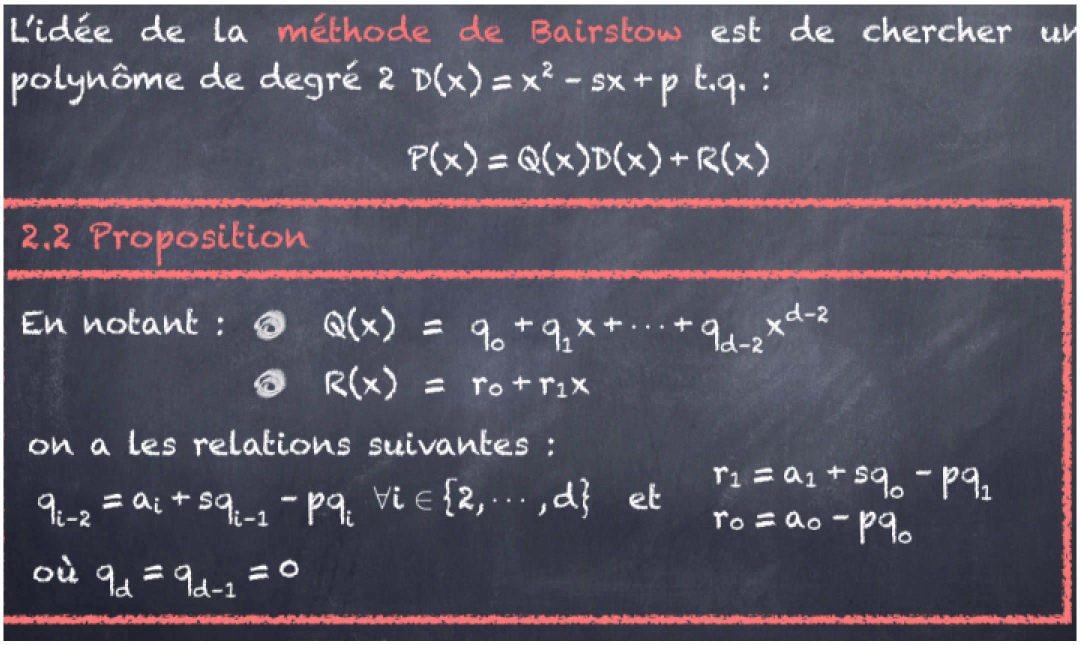
On exploitera l'**algorithme de Horner** pour évaluer les polynômes P et Q en s_n.

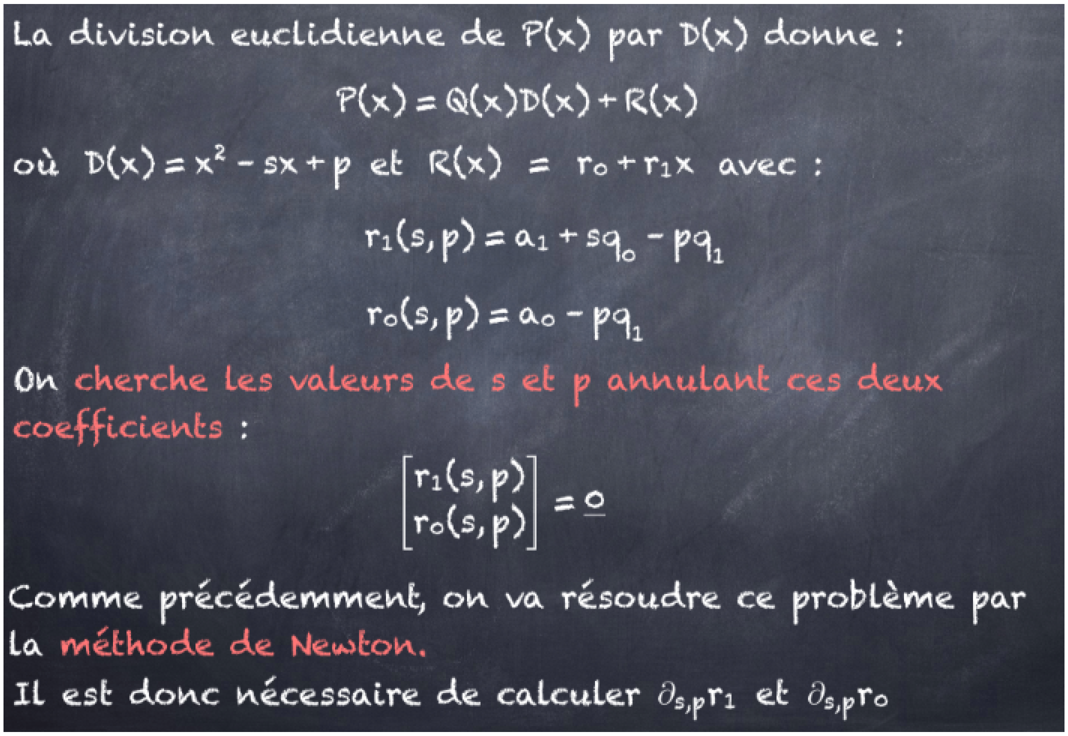
Ensuite, pour déterminer les autres racines, on procédera de même sur le polynôme $Q(x)$ (déflation).


Algorithme de Newton-Horner :
Pour k=d à 1
Tant que $\|s_0 - s_1\| \geq \varepsilon$
$[P(s_0), Q] \leftarrow$ Horner(P, s_0)
$[Q(s_0), \sim] \leftarrow$ Horner(Q, s_0)
$s_1 \leftarrow s_0 - P(s_0)/Q(s_0)$
racine$(k) = s_1$
$P \leftarrow Q$
$P(x) = (x - s_0)Q(x) + R$
Cal. $P(s_0)$ et $Q(s_0)$
Itérations Newton
Déflation
Remarque :
Lors du calcul de P(s) avec l'algorithme de Horner, on détermine le polynôme Q(x).
46


Remarque 2 :
Pour déterminer les racines complexes, si le polynôme est à coefficients réels, il faut initialiser s complexe.

3.2 Algorithme de Bairstow





On a

$$
\begin{aligned}
& p(x) = (x-s)q(x) + r_0 \\
\iff & \sum_{i=0}^{d} a_i x^i = (x-s)\sum_{i=0}^{d-1} q_i x^i + r_0 \\
\iff & \sum_{i=0}^{d} a_i x^i = \sum_{i=0}^{d-1} q_i x^{i+1} - s\sum_{i=0}^{d-1} q_i x^i + r_0 \\
\iff & \sum_{i=0}^{d} a_i x^i = \sum_{i=1}^{d} q_{i-1} x^i - s\sum_{i=0}^{d-1} q_i x^i + r_0
\end{aligned}
$$

Pour $i = d$,$a_d = q_{d-1}$
pour $i < d$ et $i \geq 1$, $a_i = q_{i-1} - sq_i \iff q_{i-1} = a_i + sq_i$
pour $i = 0$, $a_0 = -sq_0 + r_0 \iff r_0 = a_0 + sq_0$
donc $\forall i \geq 1$, $q_{i-1} = a_i + sq_i$ avec $q_d = 0$
Rq : la suite $(q_i)_i$ sert à

- évaluer $p(s)$

- connaître $p'(x)$ car $p(x) = q(x)(x-s) + r_0 \implies p'(x) = q'(x)(x-s) + q(x) \implies p'(s) = q(s)$

Remarque

$$\begin{cases} r_1(s,p) & = 0 \\ r_0(s,p) & = 0 \end{cases}$$
$$\iff \begin{cases} a_1 + sq_0 - pq_1(s,p) & = 0 \\ a_0 - pq_1(s,p) & = 0 \end{cases}$$

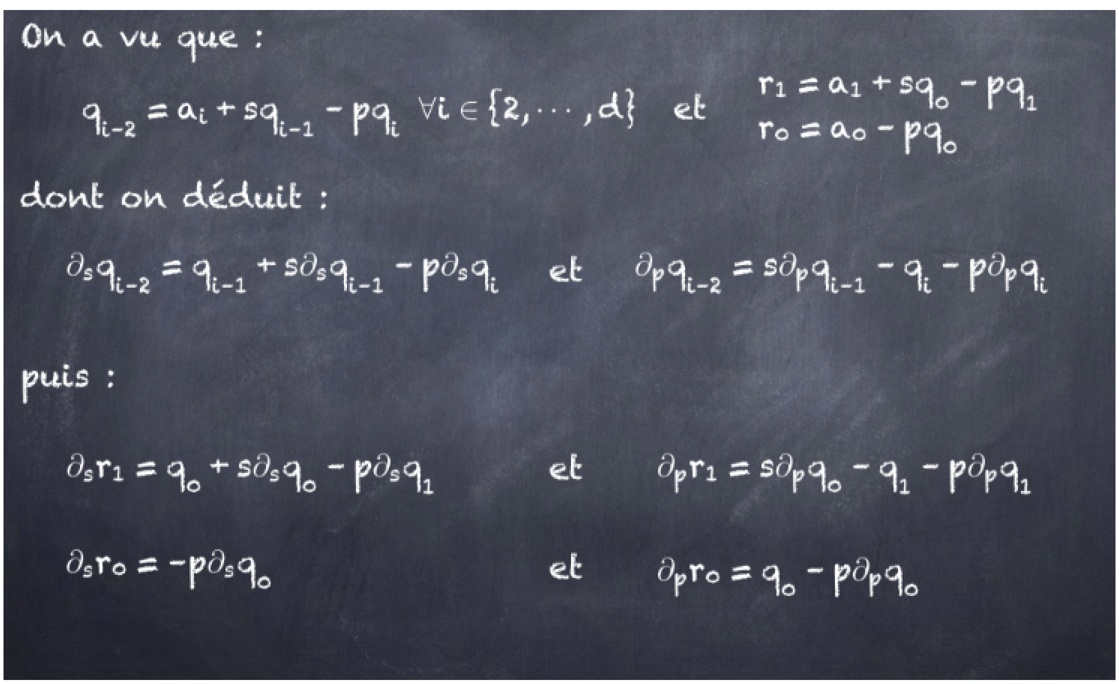
Rappel : $F : \mathbb{R}^2 \to \mathbb{R}^2$,

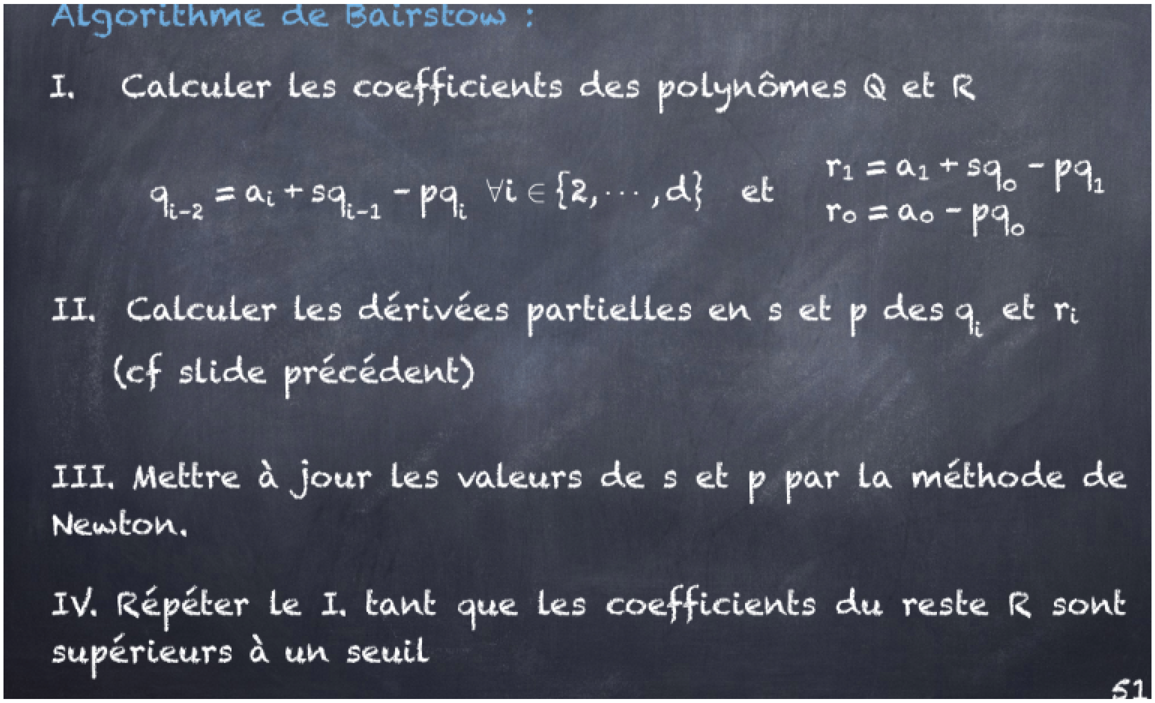
$$x_{n+1} = x_n - \left[J_{x_n} F\right]^{-1} F(x_n)$$
$$\iff x_{n+1} = x_n + \begin{bmatrix} \frac{\partial F_1}{\partial x_1}(x^n) & \frac{\partial F_1}{\partial x_2}(x^n) \\ \frac{\partial F_2}{\partial x_1}(x^n) & \frac{\partial F_2}{\partial x_2}(x^n) \end{bmatrix}^{-1} \begin{bmatrix} F_1(x_n) \\ F_2(x_n) \end{bmatrix}$$

Pour l'algorithme de Bairstow, on a

$$F(s,p) = \begin{bmatrix} a_1 + sq_0 - pq_1(s,p) \\ a_0 - pq_1(s,p) \end{bmatrix}$$

Typiquement, on doit calculer $\frac{\partial F_1}{\partial s} = q_0(s,p) + s\frac{\partial q_0}{\partial s} - p\frac{\partial q_1}{\partial s} \ldots$




Algorithme de Bairstow :
I. Calculer les coefficients des polynômes Q et R
$q_{i-2} = a_i + sq_{i-1} - pq_i \;\; \forall i \in \{2, \cdots, d\}$ et $r_1 = a_1 + sq_0 - pq_1$
$r_0 = a_0 - pq_0$
II. Calculer les dérivées partielles en s et p des q_i et r_i
(cf slide précédent)
III. Mettre à jour les valeurs de s et p par la méthode de Newton.
IV. Répéter le I. tant que les coefficients du reste R sont supérieurs à un seuil
51

Quelques remarques :

- Dans la méthode de Bairstow, une fois les coefficients s et p du diviseur D(x) obtenus, on détermine 2 racines de P(x). On peut ensuite continuer l'algorithme en l'appliquant à Q(x) qui est de degré p-2

- La méthode de Bairstow est comme la méthode de Newton, pour une racine simple, au moins d'ordre 2. Néanmoins, elle converge en général plus vite.

- En cas de racine double, la méthode de Bairstow converge à l'ordre 2 alors que celle de Newton seulement à l'ordre 1.