Sup'Galilée
MACS1

Année 2018/2019

Equations non linéaires – Notes de cours

Soit f une fonction continue sur un intervalle $[a, b]$ de $\mathbb{R}$. Nous cherchons $\alpha \in [a, b]$ tel que

$$f(\alpha) = 0.$$

En général, α ne peut pas être calculé explicitement et on doit utiliser les ordinateurs pour calculer une valeur approchée de α. Les méthodes pour approcher une racine α de f sont en général itératives : elles consistent à construire une suite $(x^{(n)})_{n\geq 0}$ telle que

$$\lim_{n\to\infty} x^{(n)} = \alpha.$$

1 Convergence

Definition 1.1. *On définit l'erreur absolue à l'étape n par*

$$e^{(n)} = x^{(n)} - \alpha, \quad n \geq 0.$$

La convergence des itérations est caractérisée par :

Definition 1.2. *On dit qu'une suite $(x^{(n)})_{n\geq 0}$ construite par une méthode numérique converge vers α avec un ordre $p \geq 1$ si*

$$\exists C > 0 : |e^{(n+1)}| \leq C|e^{(n)}|^p, \quad \forall n \geq n_0 \tag{1}$$

où $n_0 \geq 0$ est un entier. Dans ce cas, on dit que la méthode est d'ordre p. Si $p = 1$, il est nécessaire que $C < 1$ dans (1) pour que $x^{(n)}$ converge vers α. On dit que la convergence est **linéaire** *si $p = 1$ ($C < 1$),* **quadratique** *si $p = 2$, et* **cubique** *si $p = 3$. La constante C est appelée facteur de convergence de la méthode.*

En général, la convergence dépend du choix de la donnée initiale $x^{(0)}$: le plus souvent on ne sait montrer la convergence que *localement*, c'est-à-dire seulement pour un $x^{(0)}$ "suffisamment proche" de la racine α. Les méthodes qui convergent vers α pour tout $x^{(0)} \in [a, b]$ sont dites *globalement convergentes* vers α.

2 Critère d'arrêt

Soit $(x^{(n)})_{n\geq 0}$ une suite qui converge vers un zéro α de f. Soit tol une tolérance fixée. En général on utilise soit un critère basé sur l'incrément, soit un critère basé sur le résidu.

2.1 Contrôle de l'incrément

Les itérations s'achèvent dès que :

$$|x^{(n+1)} - x^{(n)}| < tol \quad \text{ou} \quad \frac{|x^{(n+1)} - x^{(n)}|}{|x^{(n+1)}|} < tol, \tag{2}$$

selon que l'on regarde la difference absolue ou relative de deux itérés successifs. Une question fondamentale est de se demander si les erreurs correspondantes $|x^{(n+1)} - \alpha|$ ou $|x^{(n+1)} - \alpha|/|\alpha|$ sont petites. Ce n'est pas le cas si la convergence est très lente. Par exemple si on considère la suite obtenue par l'algorithme de point fixe (section 6), $x^{(n+1)} = \Phi(x^{(n)})$, alors, comme $\Phi(\alpha) = \alpha$, l'erreur $e^{(n+1)} = x^{(n+1)} - \alpha$ vérifie

$$e^{(n+1)} = \Phi(x^{(n)}) - \Phi(\alpha) = \Phi^{'}(\xi^{(n)})e^{(n)},$$

avec $\xi^{(n)} \in]x^{(n)}, \alpha[$. On a donc

$$x^{(n+1)} - x^{(n)} = x^{(n+1)} - \alpha + \alpha - x^{(n)} = e^{(n+1)} - e^{(n)} = (1 - \Phi^{'}(\xi^{(n)}))e^{(n)}.$$

Si $\Phi^{'}(\xi^{(n)})$ est très proche de $\Phi^{'}(\alpha)$, alors

$$e^{(n)} \simeq \frac{1}{1 - \Phi'(\alpha)}(x^{(n+1)} - x^{(n)}).$$

Ainsi, comme on peut l'observer sur la figure 1, le critère d'arrêt sur l'incrément
- — est optimal pour les méthodes telles que $\Phi^{'}(\alpha) = 0$ comme la méthode de Newton,
- — est satisfaisant si $-1 < \Phi^{'}(\alpha) < 0$,
- — n'est pas satisfaisant si $\Phi^{'}(\alpha)$ proche de 1.

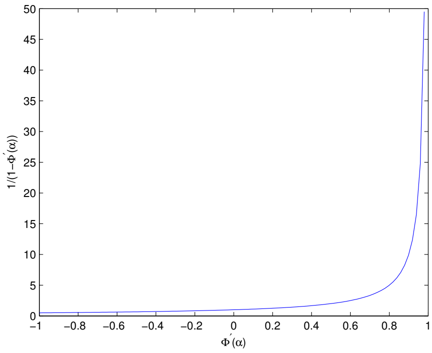

FIGURE 1 – Comportement de $\frac{1}{1-\Phi'(\alpha)}$ en fonction de $\Phi'(\alpha)$

2.2 Contrôle du résidu

Les itérations s'achèvent dès que :

$$|f(x^{(n)})| < tol \tag{3}$$

On peut montrer (voir [3]) que si α est une racine simple, alors

$$|e^{(n)}| \lesssim \frac{1}{|f'(\alpha)|}|f(x^{(n)})|.$$

Ainsi,

— si $|f'(\alpha)| \simeq 1$, alors $|e^{(n)}| \simeq tol$ et le critère d'arrêt sur le résidu est satisfaisant,
— si $|f'(\alpha)| \ll 1$, le critère d'arrêt sur le résidu n'est pas adapté car $|e^{(n)}|$ peut être grand par rapport à la tolérance tol,
— si $|f'(\alpha)| \gg 1$, le critère est trop restrictif car $|e^{(n)}| \ll tol$ dans ce cas.

2.3 Nombre d'itérations maximal

Il est nécessaire, en plus d'un test de type (2) ou (3), d'imposer un nombre d'itérations maximal (au cas où le test portant sur l'incrément ou le résidu ne serait pas vérifié), c'est-à-dire :

$$n < n_{max} \tag{4}$$

où n_{max} est le nombre d'itérations maximal que l'on se fixe.

3 Méthode de dichotomie (bissection)

Cette méthode est basée sur le théorème des valeurs intermédiaires : Soit $f : [a,b] \to \mathbb{R}$ une fonction continue, alors f prend toute les valeurs intermédiaires entre $f(a)$ et $f(b)$. En particulier, si f est telle que $f(a)f(b) < 0$, alors il existe $\alpha \in]a,b[$ tel que $f(\alpha) = 0$.
Pour trouver un zéro de f, la méthode de dichotomie consiste à calculer le point milieu m de l'intervalle et à regarder la valeur de $f(m)$. Selon le signe, on sait dans quel sous-intervalle $[a,m]$ ou $[m,b]$ se trouve le zéro. Ensuite on réitère le procédé dans le sous-intervalle correspondant. Ce qui conduit à l'algorithme suivant :

Algorithme : On pose $a^{(0)} = a,\ b^{(0)} = b,\ x^{(0)} = (a^{(0)} + b^{(0)})/2$. Alors pour $n \geq 0$:
— si $f(a^{(n)})f(x^{(n)}) > 0$ on pose $a^{(n+1)} = x^{(n)}$, $b^{(n+1)} = b^{(n)}$
— si $f(a^{(n)})f(x^{(n)}) < 0$ on pose $a^{(n+1)} = a^{(n)}$, $b^{(n+1)} = x^{(n)}$
puis on pose $x^{(n+1)} = (a^{(n+1)} + b^{(n+1)})/2$.

Critère d'arrêt :

Exercice 1 : Montrer qu'un test d'arrêt du type (2) est équivalent à un test d'arrêt portant sur la longueur des sous-intervalles

$$|b^{(n)} - a^{(n)}| \leq tol,$$

où tol est une tolérance fixée. On aura alors $|x^{(n)} - \alpha| \leq |b^{(n)} - a^{(n)}| \leq tol$.

Convergence :

Exercice 2 : a) Montrer que $|e^{(n)}| \leq \frac{(b-a)}{2^{n+1}}$, $n \geq 0$ et en déduire que la méthode de dichotomie converge.
b) Montrer qu'avec le choix du critère d'arrêt ci-dessus, si on veut avoir $e^{(n)} \leq \epsilon$ on doit prendre $n \geq \frac{ln((b-a)/\epsilon)}{ln(2)} - 1$.

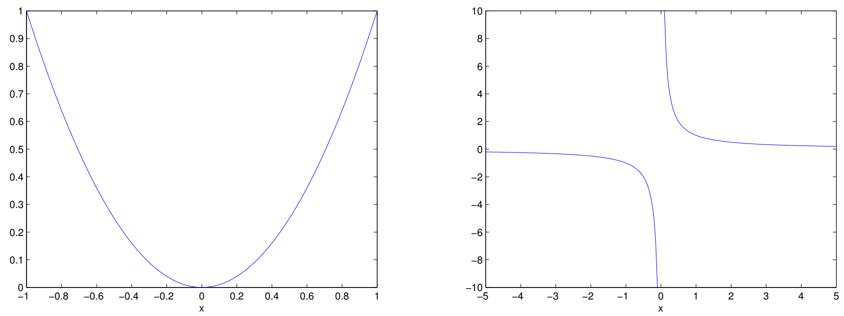

FIGURE 2 – Cas où la méthode de dichotomie a des problèmes

La figure 2 montre deux cas où la méthode de dichotomie ne marche pas : lorsque la fonction a le même signe des deux côtés du zéro, ou dans le cas d'un pôle (dans ce cas la méthode indiquerait un zéro). La lente convergence de la méthode de dichotomie suggère de n'utiliser celle-ci que pour localiser la racine et s'en approcher. En effectuant quelques itérations de dichotomie, on obtient une approximation de α qu'on peut utiliser comme point de départ pour une méthode d'ordre supérieur qui convergera plus vite vers la racine avec une précision donnée. Etudions deux méthodes d'ordre supérieur.

4 Méthode de Newton.

Supposons que f est dérivable sur $[a, b]$.

Principe de la méthode de Newton : On part d'une valeur $x^{(0)} \in [a, b]$ donnée. Soit Γ la courbe représentative de f. On construit le nouvel itéré $x^{(1)}$ de sorte que le point $(x^{(1)}, 0)$ soit le point d'intersection entre l'axe des x et la tangente à Γ passant par le point $(x^{(0)}, f(x^{(0)}))$. On itère ce processus : Connaissant $x^{(n)}$, $n \geq 0$, on construit le nouvel itéré $x^{(n+1)}$ de sorte que le point $(x^{(n+1)}, 0)$ soit le point d'intersection entre l'axe des x et la tangente à Γ passant par le point $(x^{(n)}, f(x^{(n)}))$. (voir la figure 3).

On définit ainsi la suite $x^{(n)}$ de la façon suivante : $x^{(0)} \in [a, b]$ donné,

$$x^{(n+1)} = x^{(n)} - \frac{f(x^{(n)})}{f'(x^{(n)})}, \quad n \geq 0.$$

Exercice 4 : Faire un dessin illustrant la méthode de Newton pour la fonction $y = xe^{-x}$, avec $x^{(0)} = 0.5$ puis $x^{(0)} = 2$.

Exemple 1 : considérons la fonction $y = arctan(x)$. Le zéro de cette fonction est $x = 0$. En traçant la fonction on observe qu'il existe un point x^* où l'algorithme de Newton va osciller, c'est-à-dire $g(x^*) = -x^*$ où $g(x) = x - \frac{f(x)}{f'(x)}$. Pour résoudre cette équation, qui se réecrit sous la forme $(1 + x^2)\arctan(x) - 2x = 0$, on peut utiliser par exemple la méthode de dichotomie. On obtient (en Matlab/Octave) :

```
>> x=Dichotomie('g',1,2)

x = 1.3917
```

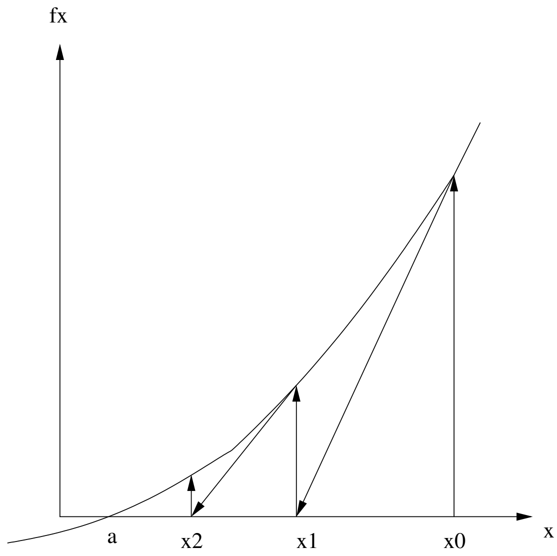

FIGURE 3 – Construction géométrique de la méthode de Newton

Il est important de noter que cette boucle où l'on oscille entre x^* et $-x^*$ a lieu théoriquement. Numériquement, on peut sortir de cette boucle (et converger ou diverger) à cause des erreurs d'arrondi. L'algorithme va converger dans le cas où x_0 est plus petit (en valeur absolue) que x^*, et diverger dans l'autre cas (voir la figure 4).

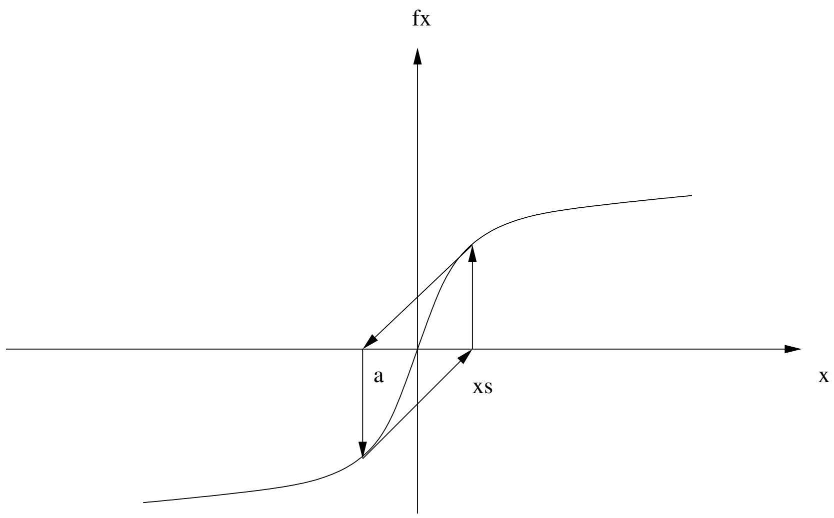

FIGURE 4 – Méthode de Newton dans le cas où $f(x) = \arctan(x)$ et $x^{(0)} = x^*$

Exemple 2 : considérons la fonction $y = x^3$. la convergence est linéaire. Ceci provient du fait que 0 est un zéro multiple, et la méthode de Newton "perd" la convergence quadratique dans ce cas :

$$x_{n+1} = x_n - \frac{x_n^3}{3x_n^2} = \frac{2}{3}x_n, \quad n \geq 0.$$

La méthode de Newton étant une méthode de point fixe, l'étude théorique de la convergence sera effectué ci-dessous. L'utilisation de la méthode de Newton nécessite le calcul de la dérivée de f en chaque point de l'itération, ce qui peut être coûteux. C'est pourquoi on remplace souvent dans les calculs la tangente par la sécante.

5 Méthode de la sécante.

Principe de la méthode de la sécante : On part de deux valeurs $x^{(0)}$ et $x^{(1)}$ dans $[a, b]$ données et on trace la droite passant par les points $(x^{(0)}, f(x^{(0)}))$ et $(x^{(1)}, f(x^{(1)}))$. Elle coupe l'axe des x en $x^{(2)}$. On itère ce processus : Connaissant $x^{(n-1)}$ et $x^{(n)}$, $n \geq 1$, on construit le nouvel itéré $x^{(n+1)}$ de sorte que le point $(x^{(n+1)}, 0)$ soit le point d'intersection entre l'axe des x et la sécante passant par les points $(x^{(n-1)}, f(x^{(n-1)}))$ et $(x^{(n)}, f(x^{(n)}))$ (voir la figure 5).

On définit ainsi la suite $x^{(n)}$ de la façon suivante : $x^{(0)},\ x^{(1)} \in [a,b]$ donnés,

$$x^{(n+1)} = x^{(n)} - \frac{x^{(n)} - x^{(n-1)}}{f(x^{(n)}) - f(x^{(n-1)})} f(x^{(n)}), \quad n \geq 1.$$

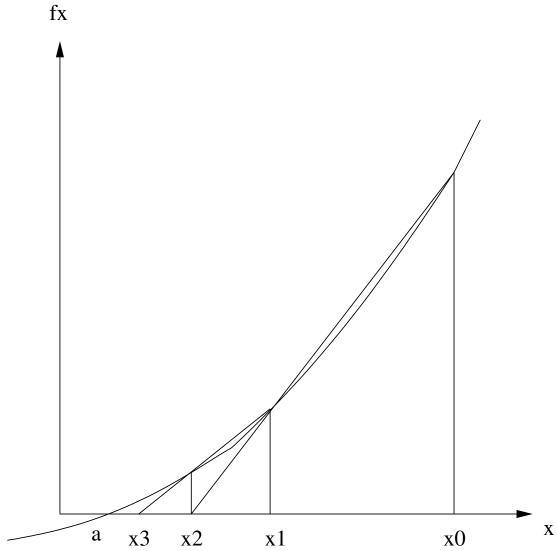

FIGURE 5 – Méthode de la sécante

Convergence :

Theorem 5.1. *Supposons que f est $\mathcal{C}^2$ dans un voisinage $\mathcal{J} =]\alpha-\delta, \alpha+\delta[,\ \delta > 0$ de la racine α, et que f' ne s'annule pas dans ce voisinage. Alors si $x^{(0)}$ et $x^{(1)}$ (choisies dans $\mathcal{J}$) sont assez proche de α, la suite $(x^{(n)})_{n\geq 0}$ définie par la méthode de la sécante converge vers α avec un ordre $p = (1+\sqrt{5})/2 \approx 1.618$. On dit que la convergence est superlinéaire.*

6 Méthode de point fixe (ou des approximations successives).

Il s'agit d'un procédé plus général pour trouver les racines d'une équation non linéaire. La méthode est fondée sur le fait que pour $f : [a,b] \to \mathbb{R}$, on peut toujours transformer le problème $f(x) = 0$ en un problème équivalent

$$\Phi(x) = x,$$

où la fonction auxiliaire $\Phi : [a,b] \to \mathbb{R}$ a été choisie de sorte que $\Phi(\alpha) = \alpha$ quand $f(\alpha) = 0$. Approcher les zéros de f revient ainsi à chercher les points fixes de Φ, ce qui se fait en utilisant l'algorithme suivant : étant donné $x^{(0)}$, on pose

$$x^{(n+1)} = \Phi(x^{(n)}), \quad n \geq 0.$$

Le choix de Φ n'est pas unique.

Definition 6.1. *Soit f une fonction continue sur $[a,b]$. On dit que f est lipschitzienne sur $[a,b]$, de constante de Lipschitz $L \geq 0$ si*

$$|f(x) - f(y)| \leq L|x-y|, \quad \forall x, y \in [a,b]$$

Theorem 6.2 (convergence des itérations de point fixe)**.** *On se donne $x^{(0)}$ et on considère la suite $x^{(n+1)} = \Phi(x^{(n)})$, pour $n \geq 0$. Si*

1. *$\Phi : [a,b] \to [a,b]$ continue,*
2. *ϕ lipschitzienne sur $[a,b]$, de constante de lipschitz $L < 1$*

alors Φ a un unique point fixe α dans $[a,b]$ et la suite $(x^{(n)})$ converge vers α pour tout choix de $x^{(0)} \in [a,b]$.

Démonstration. On remarque que si $\Phi(a) = a$ ou si $\Phi(b) = b$ alors on a l'existence d'un point fixe. On suppose que $\Phi(a) > a$ et $\Phi(b) < b$. La fonction h définie par $h(x) = \Phi(x) - x$ vérifie alors $h(a) > 0$ et $h(b) < 0$; comme h est continue, par le théorème des valeurs intermédiaires, il existe $\alpha \in [a,b]$ tel que $h(\alpha) = 0$, c'est-à-dire tel que $\Phi(\alpha) = \alpha$ et donc il existe un point fixe.

Pour montrer l'unicité, on raisonne par l'absurde : on suppose que l'on a deux points fixes α_1 et α_2. Alors,

$$\begin{array}{rcll} |\alpha_1 - \alpha_2| & = & |\Phi(\alpha_1) - \Phi(\alpha_2)| & \text{car } \Phi(\alpha_i) = \alpha_i \\ & \leq & L|\alpha_1 - \alpha_2| & \text{car } \Phi \text{ est lipschitzienne} \\ & < & |\alpha_1 - \alpha_2| & \text{car } L < 1 \end{array}$$

on aboutit à une contradiction, donc le point fixe est unique.

La convergence de l'algorithme provient de

$$\begin{array}{rcll} |\alpha - x^{(n+1)}| & = & |\Phi(\alpha) - \Phi(x^{(n)})| & \text{car } \Phi(\alpha) = \alpha \text{ et } x^{(n+1)} = \Phi(x^{(n)}) \\ & \leq & L|\alpha - x^{(n)}| & \text{car } \Phi \text{ est Lipschitzienne,} \end{array}$$

et par récurrence sur n on en déduit que

$$|\alpha - x^{(n)}| \leq L^n |\alpha - x^{(0)}|, \quad \forall n \in \mathbb{N}.$$

Comme $L < 1$ on a $L^n \to 0$ lorsque $n \to \infty$ et l'algorithme converge vers le point fixe. □

Lemma 6.3. *Si Φ est dérivable sur $[a, b]$ et si sa dérivée est bornée par L sur $[a, b]$, alors Φ est lipschitzienne sur $[a, b]$, de constante de lipschitz L.*

La figure 6 illustre dans quatre situations différentes, deux cas où $|\Phi'(x)| < 1$ et où l'algorithme converge, et deux cas où $|\Phi'(x)| > 1$ sur $[a, b]$ et où l'algorithme diverge.

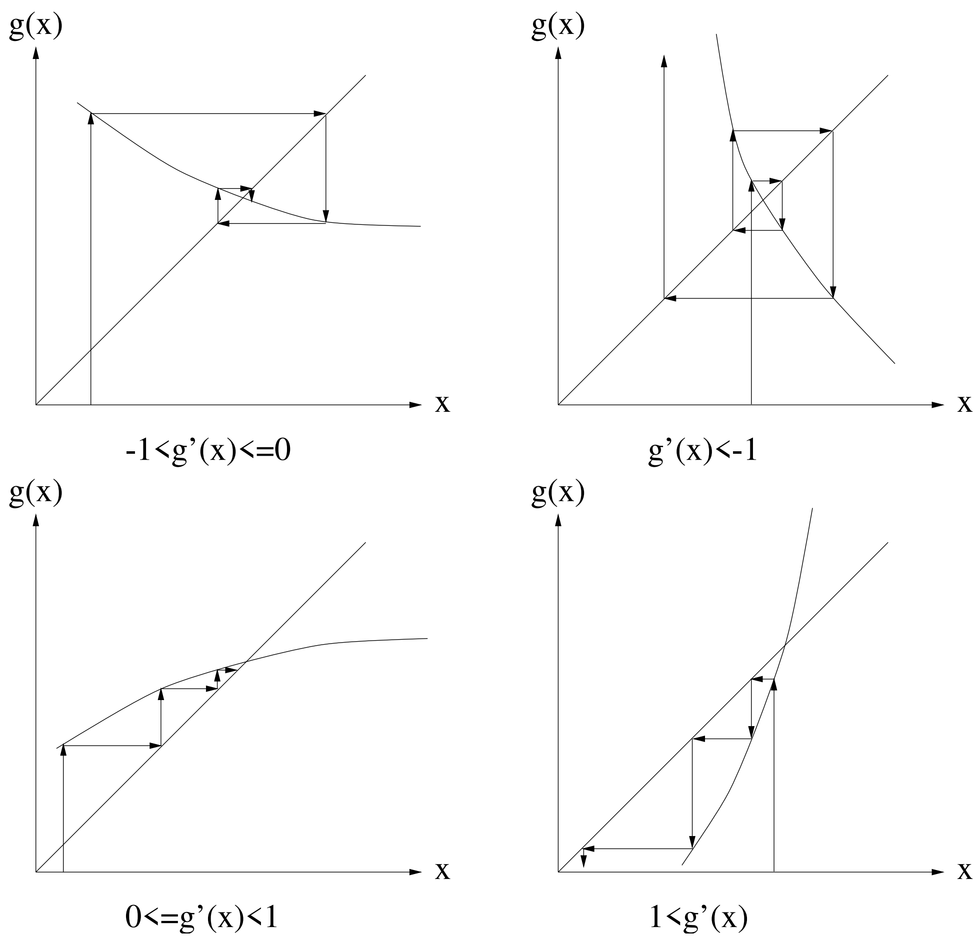

FIGURE 6 – Iterations de point fixe dans quatre situations différentes

La convergence est en général linéaire, du fait que l'erreur $e^{(n)} = |\alpha - x^{(n)}|$ vérifie

$$e^{(n+1)} \leq L e^{(n)}.$$

Propriété 6.4. *Si $\Phi \in \mathcal{C}^{p+1}(\mathcal{J})$ pour un certain voisinage $\mathcal{J}$ de α et un entier $p \geq 0$, et si $\Phi^{(i)}(\alpha) = 0$ pour $0 \leq i \leq p$ et $\Phi^{(p+1)}(\alpha) \neq 0$, alors la méthode de point fixe est d'ordre $p+1$ et*

$$\lim_{n \to \infty} \frac{x^{(n+1)} - \alpha}{(x^{(n)} - \alpha)^{p+1}} = \frac{\Phi^{(p+1)}(\alpha)}{(p+1)!}, \quad p \geq 0$$

Exercice 5 : Interpréter la méthode de Newton comme un méthode de point fixe et en déduire le théorème 6.5 ci-dessous.

Theorem 6.5. *Supposons que f est $\mathcal{C}^2$ dans un voisinage $\mathcal{J} =]\alpha - \delta, \alpha + \delta[$, $\delta > 0$ de la racine α, et que f' ne s'annule pas dans ce voisinage. Alors il existe $\mu < \delta$ tel que si $|x^{(0)} - \alpha| < \mu$, la suite $(x^{(n)})_{n \geq 0}$ définie par la méthode de Newton converge quadratiquement vers α.*

7 Calcul de l'erreur en pratique

Le calcul numérique de l'erreur peut servir d'une part à vérifier les estimations théoriques, et d'autre part à comparer – en vitesse de convergence– différentes méthodes pour approcher une racine α de f. Ainsi, en plus du calcul d'une approximation de α, on souhaite calculer $|e^{(n)}| = |x^{(n)} - \alpha|,\ n \geq 0$. Le problème est que l'on ne connait pas α en général, et même si α est connu, sa valeur risque d'être approchée sur un ordinateur (par exemple si $\alpha = \pi$). Aussi on calculera en pratique un vecteur $\boldsymbol{e}$ de $(n+1)^{\text{ième}}$ composante $e_{n+1} = |x^{(n)} - \alpha_a|,\ 0 \leq n \leq n_{it}$, où $n_{it} \leq n_{max}$ est le nombre d'itérations, et

— α_a vaut α si α est connu,
— sinon α_a sera une valeur approchée de α aussi précise que possible.

Représentation de l'erreur en pratique (en Matlab/Octave) : pour visualiser l'erreur on utilisera deux types de graphes :

• d'une part le tracé du vecteur $\boldsymbol{e}$ (en échelle logarithmique) en fonction du nombre d'itérations, c'est-à-dire le tracé de $log_{10}(\boldsymbol{e})$ en fonction de n, $1 \leq n \leq n_{it}$. Pour cela on peut utiliser la fonction *semilogy* de Matlab :

```
>> semilogy(e)
```

En particulier, on utilisera cette représentation de l'erreur pour observer et comparer les vitesses de convergence des différentes méthodes. Par exemple si $\boldsymbol{e_D}$ est le vecteur erreur correspondant à l'algorithme de dichotomie, et $\boldsymbol{e_N}$ celui correspondant à l'algorithme de Newton, on utilisera la représentation suivante :

```
>> semilogy(e_D)
>> hold on
>> semilogy(e_N)
>> hold off
```

• d'autre part, le tracé de e_{n+1} en fonction de e_n ,$1 \leq n \leq n_{it-1}$, de façon à trouver l'ordre de la méthode. Rappelons qu'une méthode est d'ordre $p \geq 1$ si

$$\exists C > 0 : |e^{(n+1)}| \leq C|e^{(n)}|^p, \quad \forall n \geq n_0$$

où $n_0 \geq 0$ est un entier. En pratique on aura alors

$$|e^{(n+1)}| \approx C|e^{(n)}|^p, \quad \forall n \geq n_0$$

soit encore

$$ln(|e^{(n+1)}|) \approx ln(C) + p * ln(|e^{(n)}|), \quad \forall n \geq n_0$$

Ainsi, les points $(ln(|e^{(n)}|), ln(|e^{(n+1)}|))$ sont sur une droite de pente p, pour $n \geq n_0$. Donc pour trouver l'ordre de la méthode, il faut tracer cette droite et regarder sa pente. Pour cela, on introduit les vecteurs $\boldsymbol{y}$ et $\boldsymbol{x}$ définis par :

$$y_n = ln(e_{n+1}) \text{ et } x_n = ln(e_n), \quad 1 \leq n \leq n_{it-1},$$

et on trace $\boldsymbol{y}$ en fonction de $\boldsymbol{x}$. Sauf éventuellement sur les premières itérations ($n \leq n_0$) la courbe que l'on obtient est une droite de pente p. On peut tracer sur la même figure (en rouge) la droite $y = px$:

```
>> y=log(e(2:length(e)));
>> x=log(e(1:length(e)-1));
>> plot(x,y,'b',x,p*x,'r')
```

Références

[1] J. P. Demailly. *Analyse Numérique et Equations Différentielles*, PUG, 1994.

[2] M.J. Gander and W. Gander. *Scientific Computing. An introduction using MAPLE and MATLAB*, to appear.

[3] A. Quarteroni, R. Sacco and F. Saleri. *Méthodes numériques pour le calcul scientifique. Programmes en MATLAB*, Springer, 2000.

[4] J. Rappaz and M. Picasso. *Introduction à l'analyse numérique*, PPUR, 1997.