

Optimisation numérique  
Chapitre 2.2  
Généralités sur les algorithmes fondamentaux

Boris Zueblin  
Route des Marches 13  
CH-1636 Broc

[boris.zueblin@unifr.ch](mailto:boris.zueblin@unifr.ch)

27 septembre 2009

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Deux stratégies de base</b>	<b>1</b>
2.1	La méthode de recherche par ligne . . . . .	1
2.2	La méthode de la région de confiance . . . . .	1
2.3	Discussion sur les algorithmes fondamentaux . . . . .	2
<b>3</b>	<b>Méthode de la recherche par ligne</b>	<b>2</b>
3.1	Méthode du gradient . . . . .	3
3.2	Méthodes arbitraires . . . . .	3
3.3	Méthode de Newton . . . . .	4
3.4	Méthode Quasi-Newton . . . . .	4
<b>4</b>	<b>Modèle pour la méthode de la région de confiance</b>	<b>5</b>
<b>5</b>	<b>Condition du problème</b>	<b>5</b>
<b>6</b>	<b>Rayons de convergence</b>	<b>6</b>
6.1	Rayon de convergence . . . . .	6
6.2	R-Rayon de convergence . . . . .	7
	<b>Références</b>	<b>7</b>
	<b>Liste des figures</b>	<b>8</b>

# 1 Introduction

Nous allons traiter les algorithmes fondamentaux de l'optimisation numérique. Tous ces algorithmes fonctionnent de manière itérative. Nous allons donc construire une suite  $(x_k)_{k \in \mathbb{N}}$  de valeurs qui convergent vers la solution  $x^*$ . Nous devons donc procéder au choix de la valeur de départ  $x_0$ . Nous pouvons la choisir arbitrairement ou déjà prendre une approximation (graphique, intuitive) de la solution. Nos algorithmes construiront la valeur  $x_{k+1}$  en fonction de  $x_k$  ou de  $x_k, \dots, x_0$ . Les méthodes n'utilisant que  $x_k$ , sont dites des méthodes à un pas. Les autres, celles utilisant  $x_0, \dots, x_k$  sont dites à plusieurs pas.

Nous devons encore distinguer deux types d'algorithmes. Il y a des algorithmes qui se rapprochent à tous les pas de la solution. i.e. :

$$f(x_{k+1}) - f(x^*) \leq f(x_k) - f(x^*) \quad \forall k.$$

Mais certains ne se rapprochent de la solution qu'après un certain nombre  $m$  de pas. i.e. :

$$f(x_{k+n}) - f(x^*) \leq f(x_k) - f(x^*) \quad \forall n \leq m \quad \forall k$$

## 2 Deux stratégies de base

### 2.1 La méthode de recherche par ligne

L'idée est de partir du point  $x_k$  et de construire le point  $x_{k+1}$  en minimisant  $f$  sur une demi-droite de direction  $\vec{p}$  qui a pour point d'attache  $x_k$ . En fait,  $x_{k+1}$  est la solution de l'équation

$$f(x_{k+1}) = \min_{\beta > 0} f(x_k + \beta \vec{p}) \quad (1)$$

Il est logique que l'on arrêtera l'algorithme lorsque notre  $x_k$  sera assez proche de la solution (en la supposant connue avant) ou lorsque la différence  $\|f(x_k) - f(x_{k-1})\|$  est suffisamment petite. Bien sûr, tout cela dépend de la précision du calcul demandé.

### 2.2 La méthode de la région de confiance

L'idée est, cette fois, d'approximer  $f(x_k)$  par une fonction plus simple  $m_k$  dans une région  $R_k$  de confiance

$$R_k = x_k + \vec{p} \quad \|\vec{p}\| \leq \Delta \quad \text{avec un } \Delta \text{ fixé}$$

suffisamment petite pour que

$$m_k(x_k + \vec{p}) \cong f(x_k + \vec{p})$$

Nous avons fait le choix arbitraire de la forme de la région de confiance, nous pouvons aussi choisir une ellipse au lieu d'une boule.

Pour trouver  $x_{k+1}$ , il nous suffit cette fois de résoudre l'équation :

$$f(x_{k+1}) = \min_{\|\vec{p}\| \leq \Delta} f(x_k + \vec{p}) \quad \text{avec } f \cong m_k. \quad (2)$$

Si la différence entre  $f(x_{k+1})$  et  $m_k(x_{k+1})$  est trop grande, nous diminuons le  $\Delta$  et par conséquent, la région de confiance et l'on résoud une nouvelle fois (2). Un avantage certain de cette méthode est que toutes les directions sont prises en compte. Par contre nous devons faire attention à ne pas trop nous éloigner de  $x_k$ , car la fonction  $m_k$  n'approxime  $f$  que sur une région proche de  $x_k$ .

Il nous reste encore à déterminer une bonne approximation  $m_k(x)$  de  $f(x)$ , donc une approximation efficace qui ne coûte pas trop cher en calcul. Nous pouvons trouver  $m_k(x)$  par le développement de Taylor appliqué aux matrices autour du point  $x_k$ .

$$m_k(x_k + \vec{p}) = f(x_k) + \vec{p}^T \nabla f(x_k) + \vec{p}^T B_k \vec{p} \quad (3)$$

ou  $\nabla$  est le gradient et  $B_k$  est la Hessienne ( $= \nabla^2 f(x_k)$ ) ou une de ses approximations.

### 2.3 Discussion sur les algorithmes fondamentaux

Nous pouvons constater que ces deux algorithmes ont des approches totalement différentes. La recherche par ligne fixe une direction et optimise par rapport à la distance. La recherche par région de confiance fixe une distance et optimise la direction. Nous avons deux manières totalement différentes de penser. Il est important de bien comprendre les principes fondamentaux de chacune des méthodes afin de pouvoir les appliquer au mieux.

## 3 Méthode de la recherche par ligne

Nous allons voir différentes manières de choisir la direction de  $\vec{p}$  dans notre recherche par ligne.

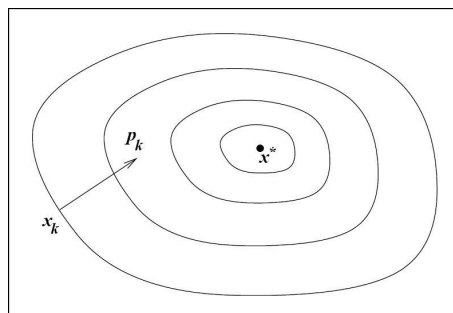


FIGURE 1 – Schéma de la méthode de la recherche par ligne

### 3.1 Méthode du gradient

Nous allons procéder de manière constructive. Regardons la fonction  $f(x)$  et son approximation de Taylor autour de  $x_k$

$$f(x_k + \alpha \vec{p}) \cong f(x_k) + \alpha \vec{p}^T \nabla f(x_k) + \frac{1}{2} \alpha^2 \vec{p}^T \nabla^2 f(x_k) \vec{p} \quad (4)$$

Par l'analyse, nous savons que la variation d'une telle fonction dans la direction de  $\vec{p}$  est donnée par  $\vec{p}^T \nabla f(x_k)$  (le facteur de  $\alpha$ ). L'idée est maintenant de maximiser la descente, donc de minimiser la pente, nous devons donc résoudre

$$\min_{\vec{p}} \vec{p}^T \nabla f(x_k) \quad \text{avec} \quad \|\vec{p}\| = 1 \quad (5)$$

Comme  $\vec{p}^T$  et  $\nabla f(x_k)$  sont des vecteurs, nous devons minimiser un produit scalaire.

$$\vec{p}^T \nabla f(x_k) = \|\vec{p}^T\| \|\nabla f(x_k)\| \cos(\angle(\vec{p}^T, \nabla f(x_k)))$$

Par le fait que  $\|\vec{p}^T\| \|\nabla f(x_k)\|$  est constant, nous devons simplement minimiser  $\cos(\angle(\vec{p}^T, \nabla f(x_k)))$ , donc l'angle entre  $\vec{p}^T$  et  $\nabla f(x_k)$  doit être de  $\pi$ , leurs sens sont donc opposés, i.e. :

$$\vec{p} = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \quad (6)$$

Le vecteur  $\vec{p}$  est donc perpendiculaire à la courbe. L'avantage de cette méthode c'est qu'elle ne requiert pas le calcul de  $\nabla^2 f(x_k)$  qui pourrait être long et fastidieux pour des problèmes complexes.

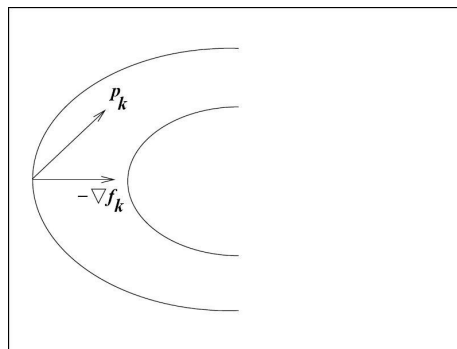


FIGURE 2 – Schéma de la méthode du gradient et d'une méthode arbitraire

### 3.2 Méthodes arbitraires

On remarque que :

$$\vec{p}^T \nabla f(x_k) = \|\vec{p}^T\| \|\nabla f(x_k)\| \cos(\angle(\vec{p}^T, \nabla f(x_k)))$$

est négatif seulement si le cos est négatif, donc si et seulement si l'angle entre  $-\nabla f(x_k)$  et  $\vec{p}$  est compris entre  $-\frac{\pi}{2}$  et  $\frac{\pi}{2}$ . Nous pouvons donc choisir arbitrairement un  $\vec{p}$  tant que cette condition est réalisée.

### 3.3 Méthode de Newton

Cette méthode se base sur le développement de Taylor du deuxième ordre. Partons encore une fois de l'approximation de

$$f(x_k + \vec{p}) = f(x_k) + \vec{p}^T \nabla f(x_k) + \frac{1}{2} \vec{p}^T \nabla^2 f(x_k) \vec{p} \cong m_k(p)$$

Décidons arbitrairement pour la stabilité que  $m'_k(p) = 0$ , nous avons donc :

$$\nabla f(x_k) + \nabla^2 f(x_k) \vec{p} = 0 \quad \Rightarrow \quad \vec{p}_k = -\nabla f(x_k) (\nabla^2 f(x_k))^{-1} \quad (7)$$

avec  $\nabla^2 f(x_k)$  définie positive. Si l'on compare notre approximation de  $f$  avec celle de l'équation (2.6), nous remarquons que nous avons simplement remplacé  $\nabla^2 f(x_k + \vec{p})$  par  $\nabla^2 f(x_k)$ . Si  $f$  est suffisamment lisse, l'erreur est proportionnelle à  $\|\vec{p}\|^3$ . Comme  $\vec{p}$  est très petit nous pouvons considérer notre approximation comme suffisamment bonne.

L'un des problèmes majeurs de cette méthode est si  $\nabla^2 f(x_k)$  n'est pas définie positive, nous ne pouvons pas garantir que notre  $\vec{p}$  engendre une diminution de  $f$ . Nous traiterons plus tard (Chapitre 6) ce genre de cas plus difficile.

En général, cette méthode converge quadratiquement (voir ci-dessous pour la définition de la convergence quadratique). Par contre, elle requiert beaucoup de calculs (surtout pour  $\nabla^2 f(x_k)$ ), les erreurs d'arrondi sont fréquentes et parfois importantes.

### 3.4 Méthode Quasi-Newton

Cette méthode remplace le  $\nabla^2 f(x_k)$  par un  $B_k$  qui demande moins de calculs, mais qui ne garantit qu'une convergence superlinéaire. Comme  $B_k$  dépend de  $k$ , nous devons le recalculer à chaque pas en prenant en compte les nouvelles informations calculées entre temps. Nous allons vous proposer ici une manière d'approximer notre  $\nabla^2 f(x_k)$ . Nous partirons de l'équation (2.5) du livre.

$$\begin{aligned} \nabla f(x + \vec{p}) &= \nabla f(x) + \int_0^1 \nabla^2 f(x + t\vec{p}) \vec{p} dt \\ &= \nabla f(x) + \int_0^1 \nabla^2 f(x + t\vec{p}) \vec{p} dt + \nabla^2 f(x) \vec{p} + \int_0^1 \nabla^2 f(x) \vec{p} dt \\ &= \nabla f(x) + \nabla^2 f(x) \vec{p} + \int_0^1 [\nabla^2 f(x + t\vec{p}) - \nabla^2 f(x)] \vec{p} dt \end{aligned}$$

Avec  $\vec{p} = x_{k+1} - x_k$  et  $x = x_k$ , nous avons donc :

$$\nabla f(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_{k+1} - x_k) + O(\|x_{k+1} - x_k\|)$$

Si  $\|x_{k+1} - x_k\|$  n'est pas trop grand, on a

$$\nabla^2 f(x_{k+1} - x_k) \cong \nabla f(x_{k+1}) - \nabla f(x_k) := B_{k+1} \quad (8)$$

Nous devons encore imposer quelques conditions sur notre  $B_k$  et il y a des méthodes pour exprimer explicitement  $B_{k+1}$  sans avoir recours à cette formule compliquée en fonction de  $B_k$ . Pour plus d'informations, regardez le livre (formule (2.17) et (2.18)). Nous pouvons aussi exprimer  $B_{k+1}^{-1}$  en fonction de  $B_k^{-1}$  ((2.20)).

## 4 Modèle pour la méthode de la région de confiance

Si l'on fixe  $B_k = 0$  dans l'équation 3, nous obtenons :

$$m_k(x_k + \vec{p}) = f(x_k) + \vec{p}^T \nabla f(x_k)$$

ce qui nous donne la solution :

$$\vec{p}_k = - \frac{\Delta_k \nabla f(x_k)}{\|\nabla f(x_k)\|} \quad (9)$$

Nous pouvons constater que cette solution est très proche de la solution de la recherche par ligne, où nous aurions fixé  $\alpha = \Delta_k$ . Mais pour cette méthode, il est mieux de considérer la vraie Hessienne, car même si elle n'est pas définie positive, cette algorithm fonctionne.

## 5 Condition du problème

Certaines fonctions sont beaucoup plus influencées par une variable que par une autre. Par exemple :

$$f(x, y) := 10^9 x^2 + 0.1 y^2$$

est beaucoup plus sensible à  $x$  qu'à  $y$ . C'est très important de prendre en compte cela quand on s'intéresse à la qualité d'un algorithme.

Pour éviter ce genre de problème, il existe quelques petits tricks. Prenons un exemple chimique où les  $x_i$  désignent une vitesse de réaction du composant  $c_i$ . La concentration de  $c_i$  dépendra donc des  $x_i$ . Posons :

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 10^{-10} \\ 1 \\ 1 \\ 10^5 \end{pmatrix}$$

On constate que ce problème (avec ces valeurs), est mal conditionné. Une petite perturbation de la donnée  $x_4$  donnera une grande perturbation de la solution finale. Afin de pallier à ce défaut, nous pouvons poser un système d'équations auxiliaires :

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 10^{-10} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10^5 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}$$

Nous résoudrons le problème en fonction des  $z_i$  qui est beaucoup mieux conditionné que le problème initial.

Pour résoudre nos problèmes d'optimisation numérique, il faut favoriser les algorithmes indépendants de la condition du problème, en utilisant par exemple la méthode de Newton.

## 6 Rayons de convergence

Dans ce chapitre, nous allons définir le vocabulaire et les concepts nécessaires afin de pouvoir, par la suite, s'attarder à la convergence de nos algorithmes.

### 6.1 Rayon de convergence

Types de convergence d'une suite

**Convergence 1.** Q-Linéaire

$$\Leftrightarrow \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq r \quad \in [0, 1] \text{ avec un } k \text{ suffisamment grand}$$

i.e, la suite se rapproche de la valeur limite à chaque pas à partir d'un certain  $k$ . (le Q vient du mot quotient).

**Convergence 2.** Q-Superlinéaire

$$\Leftrightarrow \lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$$

**Convergence 3.** Q-quadratique

$$\Leftrightarrow \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \leq M \quad \in \mathbb{R} \quad \text{avec un } k \text{ suffisamment grand}$$

Nous pouvons facilement prouver que Q-Quadratique  $\Rightarrow$  Q-Superlinéaire  $\Rightarrow$  Q-Linéaire.



**Convergence 4.** de Q-ordre  $p$

Si  $p$  est le plus grand réel t.q.

$$\Leftrightarrow \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} \leq M \in \mathbb{R} \quad \text{avec un } k \text{ suffisamment grand}$$

## Exemples de suites convergentes

**Exemple 1.** Q-Linéaire

La suite  $(1 + 0.5^k)_{k \in \mathbb{N}}$  converge de manière Q-Linéaire vers 1. En effet

$$\frac{\|1 + 0.5^{k+1} - 1\|}{\|1 + 0.5^k - 1\|} \leq 0.5$$

**Exemple 2.** Q-Superlinéaire

La suite  $(1 + k^{-k})_{k \in \mathbb{N}}$  converge de manière Q-superlinéaire vers 1. En effet

$$\lim_{k \rightarrow \infty} \frac{\|1 + (k+1)^{-(k+1)} - 1\|}{\|1 + k^{-k} - 1\|} \leq \lim_{k \rightarrow \infty} \frac{\|1 + k^{-(k+1)} - 1\|}{\|1 + k^{-k} - 1\|} = \lim_{k \rightarrow \infty} \frac{1}{k} = 0$$

**Exemple 3.** de Q-ordre  $p$

La suite  $(1 + 0.5^{pk})_{k \in \mathbb{N}}$  est une suite convergente de Q-ordre  $p$  vers 1. En effet

$$\frac{\|1 + 0.5^{p(k+1)} - 1\|}{\|1 + 0.5^{pk} - 1\|^p} \leq 0.5^{(p-p^2)k+p}$$

Par la suite, nous omettrons le Q. Dans les exemples traités auparavant, nous pouvons déjà vous indiquer que la méthode de Newton converge Quadratiquement, la quasi-Newton, Superlinéairement et la méthode du gradient, seulement linéairement (de plus, si le problème n'est pas trop bien conditionné, le  $r$  sera très proche de 1, la convergence sera donc extrêmement lente).

## 6.2 R-Rayon de convergence

Dans certaines suites, il se peut que  $\|x_{k+1} - x^*\| > \|x_k - x^*\|$  pour une sous-suite de  $\mathbb{N}$ , comme par exemple la suite :

$$x_k = \begin{cases} 1 + 0.5^k & \text{si } k \text{ est pair} \\ 1. & \text{sinon} \end{cases}$$

Une telle suite est appelée R-Linéaire, R-Superlinéaire, R-Quadratique, de R-ordre  $p$ , si elle est bornée respectivement par une suite Q-Linéaire, Q-Superlinéaire, Q-Quadratique, de Q-ordre  $p$ . Notre exemple est lui borné par la suite  $(1 + 0.5^k)_{k \in \mathbb{N}}$ , c'est donc une suite R-Linéaire.

## Références

- [1] Jorge Nocedal/ J. Wright *Numerical Optimization*. Springer-Verlag, 1999.

## Table des figures

1	Schéma de la méthode de la recherche par ligne . . . . .	2
2	Schéma de la méthode du gradient et d'une méthode arbitraire . . . . .	3