

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

Université 8 Mai 1945 – Guelma

Faculté des Mathématiques et de l'Informatique  
et des Sciences de la Matière  
Département de Mathématiques

510.14

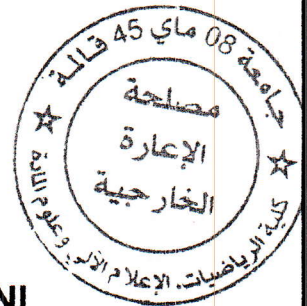


**Mémoire**

Présenté en vue de l'obtention du diplôme de

**Master Académique en Mathématiques**

Option : **Equations aux dérivées partielles**



Par :

**Melle. Sihem BENHAMIDA et Melle. Donia SOLTANI**

**Intitulé**


*Etude de quelques méthodes d'optimisation non  
linéaire sans contraintes*

**Dirigé par : Dr. Fateh ELLAGGOUNE**

Devant le jury

<b>PRÉSIDENT</b>	<b>A. MEHRI</b>	<b>M.A.A</b>	<b>Univ-Guelma</b>
<b>RAPPORTEUR</b>	<b>F. ELLAGGOUNE</b>	<b>M.C.A</b>	<b>Univ-Guelma</b>
<b>EXAMINATEUR</b>	<b>A. HITTA</b>	<b>M.C.A</b>	<b>Univ-Guelma</b>
	<b>I. BARAMA</b>	<b>M.A.A</b>	<b>Univ-Guelma</b>

**Session Juin 2011**



# Remerciements

## Remerciements

Ce mémoire symbolise –pour nous-l'aboutissement de nos efforts consentis tout au long de notre cursus universitaire en mathématique.

Nous tenons à exprimer nos remerciements et notre profonde gratitude avant tout à "ALLAH" le tout puissant qui nous a donné le courage et la force pour mener à bien ce travail.

A ce titre nous tenons à exprimer nos vifs remerciements ainsi que notre reconnaissance à tous les membres relevant du département de mathématique qui ont contribué de près et de loin à cet effort en spécifiant particulièrement:

Notre encadreur Mr Ellaggoune Fateh, dont la disponibilité et les précieux conseils nous ont permis d'orienter de façon réfléchie, voir savante, nos travaux de recherche et de documentation sur le thème choisi;

Enfin nos remerciements s'adressent à toutes les personnes qui ont participé de près et de loin dans l'élaboration de ce travail.

**S.BENHAMIDA, D.SOLTANI**



# Table des matières

Remerciements	iii
Dédicace	v
Introduction	vii
<b>1 Notions générales</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Notions d'algorithmes . . . . .	2
1.3 convergence . . . . .	4
1.3.1 Vitesse de convergence en quotient . . . . .	4
1.3.2 Vitesse de convergence en racine . . . . .	5
1.4 Notations et définitions . . . . .	6
1.4.1 Définitions et résultats classiques . . . . .	6
1.4.2 Notations . . . . .	8
1.5 Considérations numériques sur la résolution des systèmes d'équations non linéaires . . . . .	9
1.5.1 Méthode de Gauss-Seidel . . . . .	10
1.5.2 Méthode des approximations successives . . . . .	10
1.6 Définitions . . . . .	11
1.6.1 Principe général de résolution . . . . .	12
1.7 Conditions d'optimalité . . . . .	13
1.7.1 Cas différentiable . . . . .	13
1.7.2 Cas convexe . . . . .	15
1.8 Algorithmes . . . . .	17
1.8.1 Existence . . . . .	17
1.8.2 Unicité . . . . .	17
1.8.3 Méthode du gradient . . . . .	17



# Introduction

Si c'est à l'antiquité que remonte les premiers problèmes de maximum et de minimum (*Euclide, Apollonius, Archimède, Pappus*), On peut dire que c'est à la fin du dix-septième et au dix-huitième siècle qu'apparaissent les premières méthodes générales de solutions. Le développement de l'analyse mathématique en particulier le calcul différentiel (*Newton, Leibniz*), l'introduction de fonctionnelles (*Euler, Lagrange*) a permis à l'optimisation de jouer un rôle important dans le secteur des sciences exactes, où la plupart des principes physiques peuvent s'énoncer sous forme variationnelle. Dans ce contexte l'analyse des équations aux dérivées partielles de la physique et l'explication de leurs liens avec l'optimisation ont fait un grand progrès au cours des 30 dernières années (méthodes variationnelles). L'étude des modèles économiques (recherche opérationnelle) et des applications techniques (sciences de l'ingénieur, automatique...) sont devenues des domaines où l'optimisation trouve un champ d'application assez large. Et de façon générale, les techniques d'optimisations jouent un rôle de plus en plus important pour la conception des systèmes et des équipements de toute nature, et toutes les décisions techniques ou économiques. Par ailleurs, il en est de même pour les méthodes modernes de traitement du signal.

Le sujet du travail proposé ci-dessous est l'étude théorique et numérique du problème d'optimisation non linéaire sans contraintes qui se présente comme suit :

$$\text{minimiser } \{f(x) : x \in \mathbb{R}^n\} \quad (P)$$

où;  $f : \mathbb{R}^n \rightarrow \mathbb{R}$

Cette restriction - pour les problèmes sans contraintes - n'est pas fortuite car la totalité des méthodes exposées sont applicable dans le cas "avec contraintes", via une théorie appropriée. La stratégie adoptée pour la résolution du problème (P) est l'utilisation d'algorithmes à direction de descente, définis à



est due au fait que les démonstrations de convergence exige des conditions de convexité forte. La méthode de la plus profonde descente et qui remonte à Cauchy est présentée dans le but d'appliquer les conditions d'optimalité, ce qui donne une autre interprétation aux méthodes vue dans ce chapitre.

**Chapitre 2.** Un point délicat est soulevé dans le premier chapitre : L'ambiguïté dans le choix du pas qui est crucial pour la convergence des algorithmes : "approximation successive" et "la plus profonde pente". La solution est dans l'adoption d'une recherche linéaire; mais c'est ici qu'apparaissent les plus grandes difficultés pratiques, alors que la direction est généralement facile à calculer, une fois la théorie bien maîtrisée. Il faut déjà une solide expérience pour écrire une bonne recherche linéaire. . On distinguera deux modes de recherches linéaires : *Exacte et inexacte*. Une tendance se dessine alors vers l'adoption des recherches linéaires inexacte (plus économique), notre intérêt est porté pour l'étude plus au moins détaillé pour les plus importantes de ces méthodes à savoir Goldstein, Armijo et Wolfe. A la fin de ce chapitre, on expose un théorème essentiel de convergence globale (théorème de Zoutendijk), ce théorème est une alternative à celui de Zangwill (vue en chapitre 1).

**Chapitre 3.** L'algorithme de Newton est développé dans ce chapitre. Un théorème de convergence locale est démontré. Il servira de guide pour l'établissement de la convergence locale des méthodes quasi-Newton. On exposera aussi les avantages et les inconvénients de l'algorithme de Newton toutes en proposant quelques solutions sous formes d'algorithmes modifiés (Newton avec décomposition de Choleski et Newton tronqué), accompagné d'un théorème de convergence globale (Théorème 3.2) et des tests numériques témoignant de l'efficacité de ces méthodes. Sans parvenir à se débarrasser de la plus part des inconvénients de l'algorithme de Newton et surtout : L'effort fourni pour le calcul des dérivées de secondes ordres ainsi le nombres important d'opérations élémentaires nécessaire pour inverser la matrice hessienne. Ce qui fait que cette méthode est restreinte aux problèmes de petite taille.



# Chapitre 1

## Notions générales

### 1.1 Introduction

Qu'est ce un problème d'optimisation ?

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum d'une fonction donnée. On peut aussi trouver des problèmes d'optimisation pour lesquels les variables de la fonctions à optimiser sont contraintes d'évoluer dans une certaine partie de l'espace de recherche. On s'intéresse aux méthodes numériques pour résoudre les problèmes qui se résolvent de la manière suivante :

” Trouver  $x^* \in X$  tel que, pour tout  $x \in X$ , on ait  $f(x^*) \leq f(x)$  ”

où  $X$  est un ensemble et  $f$  est une application définie sur  $X$  à valeurs réelles (dans  $\mathbb{R}$ ). Il s'agit donc de trouver le point  $x^*$  de l'ensemble  $X$  qui minimise  $f$  sur  $X$ . On notera également ce problème comme suit :

$$\min_{x \in X} f(x). \quad (1.1)$$

On dit que  $X$  est l'ensemble admissible du problème et un point de  $X$  est dit point admissible. La fonction  $f$  est appelée fonction objective, fonction-coût, ou critère du problème.

Le problème (1.1) peut être très général. On peut toute fois déjà noter une restriction : L'ensemble d'arrivée de  $f$  est un espace vectoriel de dimension un. Nous n'aborderons pas les problèmes où l'espace d'arrivée est de dimension supérieur (on parle d'optimisation multi-critères) et le cas où  $X$  est une partie de  $\mathbb{N}^n$  et  $f$  est à valeurs dans  $\mathbb{N}$ .



1<sup>er</sup>)

$$\begin{cases} x_{k+1} \in \mathcal{A}(x_k) & \text{si } x_k \notin \Omega \\ x_{k+1} = x_k & \text{si } x_k \in \Omega \end{cases}$$

2<sup>ème</sup>) Une fonction test  $\alpha : X \rightarrow \mathbb{R}$  continue telle que :

$$x \notin \Omega \Rightarrow \alpha(y) < \alpha(x); \quad \forall y \in \mathcal{A}(x)$$

**Remarque 1.1**voici quelques exemples de l'ensemble des solutions  $\Omega$  :

- $\Omega = \{x \in X : \nabla f(x) = 0\}$
- $\Omega = \{x \in X : x \text{ est une solution optimale du problème (1.1)}\}$
- $\Omega = \{x \in X : f(x) < \nu + \varepsilon\}$ ,  $\varepsilon > 0$  et  $\nu$  est la valeur minimal de la fonction objective.

Le choix de la fonction test  $\alpha(x)$  est souvent  $f(x)$ ,  $\|\nabla f(x)\|$  ou  $\|x - x^*\|$  ( $x^* \in \Omega$ ).**Définition 1.2**Soit  $\mathcal{A}$  une fonction multivoque dans  $X$ ;  $\mathcal{A}$  est dite fermée en  $x \in X$  si :

$$\begin{cases} x_k \in X, & x_k \rightarrow x \\ y_k \in \mathcal{A}(x_k), & y_k \rightarrow y \end{cases} \Rightarrow y \in \mathcal{A}(x)$$

**Théorème 1.1 (Zangwill)**Si la fonction multivoque  $\mathcal{A}$  est localement bornée et fermée alors tout point d'accumulation de  $\{x_k\}_{k \in \mathbb{N}^*}$  est dans  $\Omega$ .**Démonstration :**Soit  $K$  tel que  $\lim_K x_k = x$  Alors  $\lim_K \alpha(x_k) = \alpha(x)$  ( $\alpha$  continue);  $\{\alpha(x_k)\}_{k \in \mathbb{N}^*}$  est une suite décroissante puisque :

$$\begin{aligned} x_k \in \Omega &\Rightarrow x_{k+1} = x_k \Rightarrow \alpha(x_{k+1}) = \alpha(x_k) \\ x_k \notin \Omega &\Rightarrow x_{k+1} \in \mathcal{A}(x_k) \Rightarrow \alpha(x_{k+1}) < \alpha(x_k). \\ \{\alpha(x_k)\}_{k \in \mathbb{N}^*} &\text{ décroissante} \Rightarrow \lim \alpha(x_k) = \alpha(x). \end{aligned}$$

 $\{x_k\}_{k \in K}$  bornée et  $\mathcal{A}$  localement bornée  $\Rightarrow \{x_{k+1}\}_{k \in K}$  bornée.Soit  $K' \subset K$  tel que  $\lim_{K'} x_{k+1} = x'$ ; alors  $\alpha(x') = \alpha(x)$ .Puisque  $\mathcal{A}$  est fermée, alors  $x' \in \mathcal{A}(x)$ .Si  $x \notin \Omega$  alors  $x' \in \mathcal{A}(x)$ , donc  $\alpha(x') < \alpha(x)$ ; qui est une contradiction avec la définition de la fonction test  $\alpha$ .

- Ce théorème a une valeur théorique; on verra plus loin que dans des cas spécifiques, une démonstration directe de convergence est plus facile en général. Mais il montre bien que la fermeture de  $\mathcal{A}$  est cruciale.

### 1.3. CONVERGENCE

5

Puisque  $q_k \rightarrow q$ , alors

$$\forall \varepsilon > 0, \exists N \in \mathbb{N}; \forall k \geq N : |q_k - q| < \varepsilon.$$

Donc

$$\begin{aligned} q_k - q &\leq |q_k - q| < \varepsilon, \\ \Rightarrow \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} - q &\leq \varepsilon, \\ \Rightarrow \|x_{k+1} - x^*\| &\leq (\varepsilon + q)\|x_k - x^*\|. \end{aligned}$$

#### Théorème 1.2

Si  $x_k$  converge linéairement vers  $x^*$ , alors pour tout  $q > \limsup q_k$ , il existe un  $N \in \mathbb{N}$  et  $M > 0$  tels que :

$$\|x_k - x^*\| \leq Mq^k \text{ pour tout } k \geq N. \quad (1.3)$$

Démonstration :

Supposant que  $\limsup q_k = q'$  alors ;  $\exists \varepsilon > 0$  telle que  $q' + \varepsilon = q$  de (1.2) on a :

$$\|x_{k+1} - x^*\| \leq (q' + \varepsilon) \|x_k - x^*\| = q \|x_k - x^*\|, \quad k \geq N$$

Pour  $k = N + (k - N)$  on a :

$$\|x_k - x^*\| = \|x_{N+(k-N)} - x^*\| \leq q^{k-N} \|x_N - x^*\| = q^k \frac{\|x_N - x^*\|}{q^N}$$

Et le resultat s'obtient avec  $M = \frac{\|x_N - x^*\|}{q^N}$ .

#### 1.3.2 Vitesse de convergence en racine

On s'intéresse à l'étude du taux :

$$r_k = \|x_k - x^*\|^{1/k} \quad (1.4)$$

• Si  $\limsup r_k = \alpha < 1$ ; on dit que la convergence est r-linéaire et  $\alpha$  est le taux de convergence associé.

Si  $\limsup r_k = 1$ ; on dit que la convergence est r-sous-linéaire.

Si  $\lim r_k = 0$ ; on dit que la convergence est r-superlinéaire.

Remarque 1.3

## 1.4. NOTATIONS ET DÉFINITIONS

7

1. Le rayon spectral d'une matrice  $A \in \mathcal{M}_n$  est le réel positif  $\rho(A)$  défini par

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|$$

Où  $\sigma(A)$  est l'ensemble des valeurs propres de  $A$  (spectre de  $A$ )

2. La trace d'une matrice  $A \in \mathcal{M}_n$  (notée  $Tr(A)$ ) est la quantité

$$Tr(A) = \sum_{\lambda_i \in \sigma(A)} \lambda_i$$

### Théorème 1.3 (Schur)

- Si  $A$  est une matrice carrée réelle, il existe une matrice orthogonale  $Q$  telle que  $Q^{-1}AQ$  soit triangulaire.
- Si  $A$  est symétrique, il existe une matrice orthogonale  $Q$  telle que  $Q^{-1}AQ$  soit diagonale.

### Résultat 1.1

1. Soit  $A \in \mathcal{M}_n$ ; si  $A$  est semi-définie positif alors  $Tr(A) \geq 0$
2. Une matrice symétrique est dite définie positive si et seulement si ses valeurs propres sont toutes positives; elle est dite semi-définie positive si ses valeurs propres sont toutes non négatives, avec au moins l'une d'entre elles nulle.
3. Lorsque  $A$  est définie positive (resp semi définie positive) on dira que  $f(x)$  est une forme quadratique définie positive (resp semi définie positive).

### Définition 1.7

On appelle norme matricielle subordonnée à une norme définie sur  $\mathbb{R}^n$ , la norme matricielle (également notée  $\|\cdot\|$ ) définie par

$$\|A\| = \sup_{x \in \mathbb{R}^n} \frac{\|Ax\|}{\|x\|} = \sup_{x \in \mathbb{R}^n, \|x\| \leq 1} \|Ax\| = \sup_{x \in \mathbb{R}^n, \|x\|=1} \|Ax\|$$

Par définition, on a pour toute norme matricielle subordonnée, la propriété très utile  $\|Ax\| \leq \|A\| \|x\|$ , l'égalité étant toujours possible pour au moins un vecteur  $x$ , car la sphère unité est compacte en dimension finie.

### Résultat 1.2

Application de la définition de norme matricielle subordonnée

$$\|A\|_1 = \max_j \sum_i |a_{ij}|, \quad \|A\|_2 = \sqrt{\rho(A^T A)} = \sqrt{\rho(AA^T)} = \|A^T\|_2,$$



## 1.5. CONSIDÉRATIONS NUMÉRIQUES SUR LA RÉOLUTION DES SYSTÈMES D'ÉQUATIONS

### Définition 1.10

$f : \mathbb{R}^n \rightarrow \mathbb{R}$  est dite deux fois différentiable en  $\bar{x}$  s'il existe une matrice  $n \times n$  symétrique  $H(\bar{x})$  appelée la matrice hessienne de  $f$  en  $\bar{x}$  et une fonction  $B(\bar{x}, x) \rightarrow 0$ , quand  $x \rightarrow \bar{x}$  telle que :

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})^\top (x - \bar{x}) + \frac{1}{2} (x - \bar{x})^\top H(\bar{x}) (x - \bar{x}) + \|x - \bar{x}\|^2 B(\bar{x}, x) \quad (1.7)$$

### Résultat 1.3

**Formule de Taylor :** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , une application de classe  $C^2$ . Alors

$$f(x+d) = f(x) + g(x)^\top d + \frac{1}{2} d^\top H(x) d + o(\|d\|^2).$$

**Formule de Taylor avec reste intégral :** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , une application de classe  $C^1$ , alors

$$f(x+d) = f(x) + \int_0^1 (g(x+td)^\top d) dt = f(x) + \int_x^{x+d} g(z)^\top dz \quad (1.8)$$

**Formule de la moyenne :** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , une application de classe  $C^1$  sur le segment  $[x_1, x_2]$ , alors il existe  $\hat{x} \in [x_1, x_2]$  tel que

$$f(x_2) = f(x_1) + g(\hat{x})^\top (x_2 - x_1).$$

Les résultats précédents restent valables, si  $f$  est une fonction vectorielle .

## 1.5 Considérations numériques sur la résolution des systèmes d'équations non linéaires

On verra plus loin qu'un problème d'optimisation puisse être transformé en une recherche de solutions d'un système d'équations non linéaires, donc la première idée naturelle est d'appliquer une méthode numérique de résolution de ces systèmes. Ce qui nous ramène à étudier deux de telles méthodes : Gauss-Seidel et approximation successives, qui sont des méthodes de premier ordre

On suppose que  $F$  est localement lipschitzien et coercif et soit  $x^*$  une solution de (1.9). Alors l'algorithme définie en 1.5.2, converge si  $t < 0$  suffisamment petit.

**Démonstration :**

Soit  $x_1$  la première itération et  $x^* \in \mathbb{R}^n$  une solution (i.e.  $F(x^*) = 0$ ), et prenant  $B = \overline{B}(x_1, \|x_1 - x^*\|)$  (la boule fermé de centre  $x_1$  et de rayon  $\|x_1 - x^*\|$ ). Puisque  $x_2 = x_1 + tF(x_1)$  et  $F(x^*) = 0$ , on a :

$$\begin{aligned} \|x_2 - x^*\|^2 &= (x_2 - x^*)^\top (x_2 - x^*) = \|x_1 - x^*\|^2 + \\ &\quad + 2t(x_1 - x^*)^\top (F(x_1) - F(x^*)) + t^2 \|F(x_1) - F(x^*)\|^2 \end{aligned}$$

Prenons  $t < 0$ ; alors d'après (1.8) et (1.9) :

$$\|x_2 - x^*\|^2 \leq (1 + 2mt + M^2t^2) \|x_1 - x^*\|^2 \quad (1.12)$$

Il suffit de prendre  $t > -2\frac{m}{M^2}$  pour avoir :

$$0 \leq K = 1 + 2mt + M^2t^2 < 1$$

On itère la formule (1.10) pour avoir :

$$\|x_k - x^*\|^2 \leq K^k \|x_1 - x^*\|^2 < \|x_1 - x^*\|^2 \quad (1.13)$$

**Remarque 1.4**

- On peut déduire de (1.12) que :

$$\|x_{k+1} - x^*\| \leq K \|x_k - x^*\|$$

Avec  $0 \leq K < 1$ , donc  $\{x_k\}$  converge linéairement vers  $x^*$ .

- La solution  $x^*$  est unique, car si  $\bar{x}^*$  est une autre solution; on a

$$\|\bar{x}^* - x^*\|^2 \leq K \|\bar{x}^* - x^*\|^2 \quad (\text{contradiction})$$

## 1.6 Définitions

On aborde le problème d'optimisation sans contrainte qui est normalement non-linéaire. Le problème est posé dans  $\mathbb{R}^n$ , donc

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1.14)$$

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ .

**Définition 1.12**

le vecteur  $d_k$  est la direction de descente en  $x_k$ , le scalaire  $t_k$  est appelé le pas de la méthode à l'itération  $k$ . On peut caractériser les directions de descente en  $x_k$  à l'aide du gradient :

**Proposition 1.1**

Si  $f$  est différentiable en un point  $x \in \mathbb{R}^n$  et  $d \in \mathbb{R}^n$  telle que :

$$g(x)^\top d < 0 \quad (1.16)$$

alors  $d$  est une direction de descente en  $x$ .

**Démonstration :**

On a pour tout  $t > 0$

$$f(x + td) = f(x) + tg(x)^\top d + t\epsilon(t)$$

Donc si on écrit

$$\frac{f(x + td) - f(x)}{t} = g(x)^\top d + \epsilon(t)$$

On voit bien que pour  $t$  suffisamment petit on aura  $f(x + td) - f(x) < 0$

## 1.7 Conditions d'optimalité

Les conditions d'optimalité d'un point  $x \in \mathbb{R}^n$  sont des équations, des inéquations ou des propriétés que vérifient les solutions de (1.14) (conditions nécessaires CN) ou qui assurent à un point d'être solution de (1.14) (conditions suffisantes CS). Ces conditions sont utiles pour :

- Vérifier l'optimalité éventuelle d'un point  $x \in \mathbb{R}^n$ , voir si c'est un minimum, un maximum ou un point stationnaire.
- Calculer les solutions de (1.14),
- Mettre en oeuvre des méthodes numériques permettant de résoudre (1.14).
- Définir des tests d'arrêt des itérations dans les algorithmes de résolution de (1.14).

### 1.7.1 Cas différentiable

On va supposer  $f$  de classe  $C^1$  ou  $C^2$ .



Le premier point est le résultat du théorème précédent. Le deuxième point se démontre directement à partir du développement de Taylor autour du point  $\bar{x}$  suivant une direction non nulle  $d$  (voir [3], p.133).

- Moyennant un léger renforcement, les conditions du théorème précédent savèrent suffisantes pour garantir un minimum local, lorsque celui-ci est intérieur.

### (CS2) Conditions suffisantes d'optimalité du second ordre

#### Théorème 1.7

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois différentiable en  $\bar{x}$ . Si  $g(\bar{x}) = 0$  et  $H(\bar{x})$  est définie positive, alors  $\bar{x}$  est un minimum local strict.

#### 1.7.2 Cas convexe

La notion de convexité permet de passer d'une caractérisation locale d'optimalité à une caractérisation globale. Dans un cadre de différentiabilité, la théorie de la convexité est de portée moins générale. Son intérêt réside cependant dans le fait qu'elle apporte également une caractérisation dans le cas non différentiable.

##### Définition 1.14

Un ensemble  $X$  est dite convexe si pour tout couple  $(x, y) \in X^2$  et  $\forall \lambda \in [0, 1]$  on a :

$$\lambda x + (1 - \lambda)y \in X \quad (1.18)$$

##### Définition 1.15

Une fonction  $f$  définie sur un ensemble convexe  $X$ , est dite convexe si  $\forall x_1, x_2 \in X$ , et  $\forall \alpha \in [0, 1]$  on a

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2) \quad (1.19)$$

Elle est dite strictement convexe si  $\forall x_1, x_2 \in X$ ,  $x_1 \neq x_2$  et  $\forall \alpha \in ]0, 1[$  on a

$$f(\alpha x_1 + (1 - \alpha)x_2) < \alpha f(x_1) + (1 - \alpha)f(x_2) \quad (1.20)$$

- Lorsqu'une fonction est différentiable, il y a d'autres manières de caractériser la convexité.

## 1.8 Algorithmes

Considérons maintenant le problème non linéaire sans contraintes (1.14), où l'on suppose que  $f$  est continument différentiable sur  $\mathbb{R}^n$ .

### 1.8.1 Existence

Si  $f$  est minoré alors il existe une borne inférieure de  $f$  ( mais pas - nécessairement- une solution à (1.14))

**Exemple 2.1 :**  $f(x) = e^x$

$f$  est minoré par 0 mais on n'a jamais  $x \in \mathbb{R}$  tel que  $f(x) = 0$

- Il faut donc une hypothèse de plus; le lemme suivant (dû à Powell [55]) va dans ce sens.

#### Lemme 1.1

Soit  $f$  une fonction de classe  $C^2$ , son hessien  $H$  est tel que, il existe une constant  $m > 0$  verifiant :

$$d^T H(x)d \geq m \|d\|^2; \quad \forall x, d \in \mathbb{R}^n$$

Alors, pour tout vecteur  $y \in \mathbb{R}^n$ ; l'ensemble

$$\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(y)\} \quad (1.23)$$

est convexe et compact.

### 1.8.2 Unicité

#### Théorème 1.10

Soit  $f$  une fonction strictement convexe sur  $X$  convexe .le minimum de  $f$  sur  $X$  s'il existe est unique

#### Remarque importante

Théoriquement, le but d'un algorithme de minimisation est l'identification d'un minimum, dans la pratique on est plus modeste, on se contente souvent de chercher un point stationnaire (c'est-à-dire un point qui vérifie l'équation (1.11)).

### 1.8.3 Méthode du gradient

La méthode la plus classique pour minimiser une fonction de plusieurs variables est la méthode de la plus grande pente (ou méthode de gradient à pas

tel que :  $g(x_k) = (g_1(x_k), \dots, g_n(x_k))$ , donc :

$$d_k = (0, 0, \dots, \varepsilon_i, 0, \dots, 0) = \varepsilon_i e_i$$

où  $i$  et  $\varepsilon_i$  est donné par :

$$\begin{cases} g_i(x_k) = \max \{|g_j(x_k)|; 1 \leq j \leq n\} \\ \varepsilon_i = -\frac{g_i(x_k)}{|g_i(x_k)|} \end{cases}$$

• Graphiquement la solution  $d_n$  est parallèle à un certain axe de coordonnées (correspondant à la plus grande composante du gradient).  
Posant  $\theta(t) = f(x_k + td_k)$  et cherchons  $t_k$  une solution du problème

$$\min_{t>0} \theta(t)$$

l'idée immédiate est de chercher une solution de

$$\theta'(t) = \frac{d}{dt} f(x_k + td_k) = g(x_k + td_k)^\top d_k = 0, \quad t > 0 \quad (1.26)$$

Or  $d_k$  est l'un des vecteurs de base (à un signe près); ce qui donne

$$\begin{aligned} g(x_k + td_k)^\top d_k &= g(x_k + \varepsilon_i t e_i)^\top (\varepsilon_i e_i) \\ &= \varepsilon_i g_i(x_k + \varepsilon_i t e_i) = 0; \quad i = 1, 2, \dots, n \end{aligned}$$

C'est donc résoudre une seule composante du gradient, ce qui nous donne une interprétation de la méthode Gauss-Seidel.

Maintenant on choisit parmi les solutions  $t_k$  possible celle qui donne

$$f(x_k + t_k d_k) < f(x_k)$$

### Choix de la norme $L_2$

Dans ce cas  $\|x\| = \|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$

Abordant un Lemme essentiel pour la suite

#### Lemme 1.2

Supposant que  $f$  est différentiable en  $x$  tel que  $g(x) \neq 0$ . Alors le problème d'optimisation

$$\begin{cases} \min g(x)^\top d \\ \|d\| \leq 1 \end{cases}$$

admet une solution  $d_k = -\frac{g(x_k)}{\|g(x_k)\|}$



**Proposition 1.2**

On suppose le gradient  $g$  lipschitzien sur la tranche

$$\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$$

Alors la méthode définie par (i), (ii') et (iii), satisfait à

- Soit  $f(x_k)$  est non bornée inférieurement
- Soit  $g(x_k) \rightarrow 0$ .

**Démonstration**

Observant que  $\theta'(t)$  est décroissante sur  $]0, t_k^*]$  puisque

$$\begin{cases} \theta'(0) < 0 \\ \theta'(t) \text{ continue (} g \text{ continue)} \\ t_k^* \text{ est le premier zero de } \theta'(t) = 0, t > 0 \end{cases} \Rightarrow \theta'(t) < 0, \forall t \in ]0, t_k^*]$$

Donc  $t_k^*$  est le plus petit minimum de  $\theta(t)$ ,  $t > 0$

Si on pose  $x^* = x_k + t_k^* d_k$ , et on a :  $x^*, x_k \in \mathcal{L}$  (par construction)

**Argument 1**

On va montrer que dans le voisinage de  $t = 0$ ,  $f(x_k + td_k)$  décroît avec un taux non négligeable. Prenant  $z = x_k + td_k$ ,  $t > 0$  assez petit de façon à ce que  $z \in \mathcal{L}$ ; la formule de la moyenne pour  $z'$  entre  $x_k$  et  $z$  donne

$$f(z) = f(x_k) + g(z')^\top (z - x_k) = f(x_k) + (g(z') - g(x_k))^\top (z - x_k) + g(x_k)^\top (z - x_k) \quad (1.29)$$

On a

$$g(x_k)^\top (z - x_k) = -\|g(x_k)\| \|z - x_k\|$$

Et puisque  $z' \in \mathcal{L}$  du fait que  $z' = x_k + \tau d_k$ ,  $\tau \in [0, t]$

$$\Rightarrow f(z') \leq f(z) \leq f(x_k)$$

$g$  est lipschitzien sur  $\mathcal{L}$ , et grâce l'inégalité de Cauchy-Schwarz on a

$$\begin{aligned} (g(z') - g(x_k))^\top (z - x_k) &\leq \|g(z') - g(x_k)\| \|z - x_k\| \leq M \|z' - x_k\| \|z - x_k\| \\ &\leq M \|z - x_k\|^2 \end{aligned}$$

$$\Rightarrow f(z) \leq f(x_k) - \|g(x_k)\| \|z - x_k\| + M \|z - x_k\|^2$$

De l'argument 1 et (1.30)

$$f(z) \leq f(x_k) - \frac{\|g(x_k)\|^2}{4M}$$

Donc  $\|g(x_k)\|^2 \leq 4M[f(x_k) - f(z)] \leq 4M[f(x_k) - f(x_{k+1})]$ , ce qui implique

$$\sum_{i=1}^{i=k} \|g(x_i)\| \leq 4M[f(x_1) - f(x_{k+1})]$$

Si  $f$  est bornée inférieurement on a  $[f(x_1) - f(x_{k+1})] \leq C, \forall k \in \mathbb{N}^*$  ce qui donne

$$\sum_{i=1}^{i=\infty} \|g(x_i)\| \leq C \Rightarrow g(x_k) \rightarrow 0$$

### 1.8.5 Etude locale de la convergence

Maintenant, on va analyser la convergence locale de cette méthode, c'est-à-dire la vitesse de convergence des suites générées par l'algorithme au voisinage du point optimum.

#### Théorème 1.11

Si  $f$  est quadratique de la forme  $f(x) = \frac{1}{2}x^\top Qx - b^\top x$ , avec  $Q \in \mathcal{M}_n$  symétrique définie positive,  $b \in \mathbb{R}^n$ , la méthode "steepest descent" (i),(ii),(iii), converge vers la solution du problème avec une vitesse linéaire de rapport

$$r = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}$$

Où :  $\lambda_n = \max(\sigma(Q)); \lambda_1 = \min(\sigma(Q))$ .

#### Démonstration :

La condition suffisante d'optimalité (CS2) permet d'identifier la solution de (1.14), il suffit de résoudre l'équation (1.26), ce qui donne

$$t_k = \frac{g(x_k)^\top g(x_k)}{g(x_k)^\top Q g(x_k)} \quad (1.31)$$

Avec un calcul simple, on trouve

$$f(x_{k+1}) = \left[ 1 - \frac{(g(x_k)^\top g(x_k))^2}{(g(x_k)^\top Q g(x_k))(g(x_k)^\top Q^{-1} g(x_k))} \right] f(x_k) \quad (1.32)$$

## Chapitre 2

### Recherche linéaire

Dans ce chapitre, on considère le problème du calcul de la direction comme résolu, nous nous intéressons uniquement au calcul du pas optimal  $t^*$  qui est solution du problème

$$\min_{t>0} \theta(t) \quad (2.1)$$

- $\theta(t)$  la fonction test qui représente  $f(x + td)$  définie pour  $t \geq 0$
- $x$  le point de départ de la recherche linéaire
- $d$  la direction de la recherche linéaire

On suppose toujours que  $\theta'(0) < 0$

#### 2.1 Schéma général de résolution

Un algorithme de recherche linéaire résolvant le problème (2.1) génère une suite de valeurs  $\{t^1, t^2, \dots, t^k, \dots\}_{k \in \mathbb{N}^*}$  est un test à trois sorties qui étant donné un  $t^1 > 0$  (une valeur initial) répond si

- $t^k$  est satisfaisant donc c'est une solution
  - $t^k$  est trop grand
  - $t^k$  est trop petit
- Ce test s'opère on vue de la valeur de  $\theta(t^k)$  est éventuellement de  $\theta'(t^k)$ .
  - On appellera  $t^*$  un pas optimal,  $t_d$  un pas trop petit et  $t_a$  un pas trop grand.



### 2.1.1 Propriétés

- 1) Les ensembles suivants constituent une partition de  $\mathbb{R}^+$

$$\{t \in \mathbb{R}^+ : \theta'(t) = 0\}, \{t \in \mathbb{R}^+ : \theta'(t) > 0\}, \{t \in \mathbb{R}^+ : \theta'(t) < 0\}$$

pour que tout  $t^{k+1}$  soit classé sans ambiguïté.

- 2)  $\exists N \in \mathbb{N}$  telle que  $t_d^N \neq 0$  ( $\theta'(t_d^N) > 0$ ), pour éviter que  $t_g^k \rightarrow +\infty$ .  
 3) Tout intervalle  $[t_g^k, t_d^k]$  contient un  $t^*$  tel que  $\theta'(t^*) = 0$  pour éviter le cas :  $|t_g^k - t_d^k| \rightarrow 0$

### 2.1.2 Remarques sur le calcul de $t^{k+1}$

1. La recherche linéaire est caractérisé par le choix de  $t^{k+1}$ , pour cela considérons les deux cas suivants :
- Cas  $t_d^{k+1} = 0$  (extrapolation)  
On peut prendre  $t^{k+1} = \alpha t_g^{k+1}$  avec  $\alpha$  fixe par exemple  $\alpha = 10$
  - Cas  $t_d^{k+1} \neq 0$  (interpolation)  
Donc ;  $t^{k+1} \in ]t_g^{k+1}, t_d^{k+1}[$ , on peut procéder à une simple interpolation par dichotomie, c'est à dire

$$t^{k+1} = \frac{t_g^{k+1} + t_d^{k+1}}{2}$$

Ce qui nous donne un intervalle de confiance (où d'incertitude) divisé par deux pour chaque itération.

2. Dans le but d'accélérer la recherche linéaire (c'est essentiel puisqu'elle est exécutée à chaque itération de minimisation) il est bon d'orienter le choix de  $t^{k+1}$ . Pour cela l'idée général est de construire un polynôme d'interpolation à partir de points déjà testé puis minimiser ce dernier. L'usage est de ce limité à deux points  $t^k$  et  $t^{k-1}$ ; soit

$$\theta(t^k), \theta(t^{k-1}), \theta'(t^k), \theta'(t^{k-1})$$

Les quatres informations qui permettent de calculer les quatres coefficients d'un polynôme  $P_3(x)$  de troisième degré le quel (sauf exception) admet un minimum local. Les calculs donnent les coefficient du polynôme

$$P_3(x) = ax^3 + bx^2 + cx + d$$

## 2.2 Recherche linéaire exacte

Historiquement (Cauchy 1847); on s'est d'abord efforcé de chercher un  $t^*$  tel que

$$\theta(t^*) < \theta(0) \text{ et } \theta'(t^*) = 0 \quad (2.3)$$

Sous ces conditions, on peut dresser l'algorithme suivant

### Algorithme 2.2

- **Etape 0 : (Initialisation)**

$k = 1, t^1 > 0, t_g^1 = 0, t_d^1 = 0, \varepsilon > 0$  (suffisamment petit)

- **Etape 1 :**

si  $|\theta'(t^k)| \leq \varepsilon$ ; stop  $t^* = t^k$

si  $\theta(t^k) \geq \theta(0)$  où  $\theta'(t^k) > \varepsilon$  alors;

$t_d^{k+1} = t^k, t_g^{k+1} = t_g^k$ , on va à l'étape 2

si  $\theta(t^k) < \theta(0)$  et  $\theta'(t^k) < -\varepsilon$  alors;

$t_g^{k+1} = t^k, t_d^{k+1} = t_d^k$ , on va à l'étape 2

- **Etape 2 :**

si  $t_d^{k+1} = 0$  déterminer  $t^{k+1} \in ]t_g^{k+1}, +\infty[$

si  $t_d^{k+1} \neq 0$  déterminer  $t^{k+1} \in ]t_g^{k+1}, t_d^{k+1}[$

remplacer  $k$  par  $k + 1$  et aller à l'étape 1

- On peut noter que ce test est plus raffiné que l'algorithme 2.1 (ce dernier n'assurait même pas la décroissance de  $\theta$ ).

### Théorème 2.2

Supposant que  $\theta$  est de classe  $C^1, \theta'(0) < 0$  et  $\exists t^* \in \mathbb{R}^+$  tel que;  $\theta(t) \geq \theta(t^*), \forall t \in \mathbb{R}^+$  alors l'algorithme 2.2 est finie.

#### Démonstration :

Puisque  $\exists t^* \in \mathbb{R}^+$  tel que  $\theta(t) \geq \theta(t^*), \forall t \in \mathbb{R}^+$  alors  $\exists N \in \mathbb{N}^*; t_d^N \neq 0$  donc  $\{t_g^k\}$  et  $\{t_d^k\}$  sont des suites adjacentes avec

$$\lim t_d^k = \lim t_g^k = t^*$$

$\theta$  est continue et  $\theta(t_d^k) < \theta(0)$  donc;

$$\theta(t_g^k) \rightarrow \theta(t^*) \leq \theta(0)$$

la même chose pour  $\theta'$  donne

$$\theta'(t^*) \leq -\varepsilon \leq -\frac{\varepsilon}{2} < 0$$

dite conditions de Goldestein.

**Algorithme 2.3**

• **Etape 0 : (Initialisation)**

$k = 1, t^1 > 0, t_g^1 = 0, t_d^1 = 0, 0 < m_1 < m_2 < 1$

• **Etape 1 :**

si  $m_2\theta'(0)t^k + \theta(0) \leq \theta(t^k) \leq m_1\theta'(0)t^k + \theta(0)$ ; stop.  $t^* = t^k$

si  $\theta(t^k) > m_1\theta'(0)t^k + \theta(0)$  alors;

$t_d^{k+1} = t^k; t_g^{k+1} = t_g^k$ , on va à l'étape 2

si  $\theta(t^k) < m_2\theta'(0)t^k + \theta(0)$  alors;

$t_g^{k+1} = t^k; t_d^{k+1} = t_d^k$ , on va à l'étape 2

• **Etape 2 :**

si  $t_d^{k+1} = 0$  déterminer  $t^{k+1} \in ]t_g^{k+1}, +\infty[$

si  $t_d^{k+1} \neq 0$  déterminer  $t^{k+1} \in ]t_g^{k+1}, t_d^{k+1}[$

remplacer  $k$  par  $k + 1$  et aller à l'étape 1

**Théorème 2.3**

Supposant que  $\theta$  est de classe  $C^1, \theta'(0) < 0$  et  $\exists t^* \in \mathbb{R}^+$  tel que;  $\theta(t) \geq \theta(t^*), \forall t \in \mathbb{R}^+$  alors la méthode de Goldestein est finie.

**Démonstration :**

Puisque  $\exists t^* \in \mathbb{R}^+$  tel que  $\theta(t) \geq \theta(t^*), \forall t \in \mathbb{R}^+$  alors  $\exists N \in \mathbb{N}; t_d^N \neq 0$  et de la remarque 2.1,  $\{t_g^k\}$  et  $\{t_d^k\}$  sont des suites adjacentes avec  $\lim t_d^k = \lim t_g^k = t^*, \theta$  est continue et  $\theta(t_d^k) < \theta(0)$  donc;

de b) on a;

$$\theta(t_d^k) > m_1 t_d^k \theta'(0) + \theta(0) \Rightarrow \theta(t^*) \geq m_1 t^* \theta'(0) + \theta(0)$$

$$\theta(t_g^k) < m_2 t_g^k \theta'(0) + \theta(0) \Rightarrow \theta(t^*) \leq m_2 t^* \theta'(0) + \theta(0)$$

Ce qui donne  $m_2 \leq m_1$  et puisque  $\theta'(0) < 0 \Rightarrow$  une contradiction avec les hypothèses

- Dans ce test on demande à  $t^*$  soit de façon à ce que

$$\theta_1(t^*) \leq \theta(t^*) \leq \theta_2(t^*); t > 0$$

où

$$\theta_1(t) = m_2 \theta'(0)t + \theta(0)$$

$$\theta_2(t) = m_1 \theta'(0)t + \theta(0)$$



**Algorithme 2.5**• **Etape 0 : (Initialisation)**

$$k = 1, t^1 > 0, 0 < m_1 < m_2 < 1$$

• **Etape 1 :**

si  $\theta(t^k) \leq m_1 \theta'(0)t^k + \theta(0)$  et  $\theta'(t^k) \geq m_2 \theta'(0)$ ; stop  $t^* = t^k$

si  $\theta(t^k) > m_1 \theta'(0)t^k + \theta(0)$  alors;

$$t_d^{k+1} = t^k, t_g^{k+1} = t^k \text{ on va à l'étape 2}$$

si  $\theta(t^k) \leq m_1 \theta'(0)t^k + \theta(0)$  et  $\theta'(t^k) < m_2 \theta'(0)$  alors;

$$t_g^{k+1} = t^k, t_d^{k+1} = t^k \text{ on va à l'étape 2}$$

• **Etape 2 :**

si  $t_d^{k+1} = 0$  déterminer  $t^{k+1} \in ]t_g^{k+1}, +\infty[$

si  $t_d^{k+1} \neq 0$  déterminer  $t^{k+1} \in ]t_g^{k+1}, t_d^{k+1}[$

remplacer  $k$  par  $k + 1$  et aller à l'étape 1

**Remarque 2.4**

Ceci a une bonne interprétation : On assure d'une part la décroissance de  $\theta$  (donc  $x_{k+1}$  ne sera pas trop loin de  $x_k$ ), puisque

$$\frac{\theta(t^*) - \theta(0)}{t^*} \leq m_1 \theta'(0) < 0 \Rightarrow \theta(t^*) < \theta(0)$$

Et d'autre part du fait que  $\theta'(t^*) \geq m_2 \theta'(0)$ ; la dérivée a suffisamment augmenté (donc  $x_{k+1}$  ne sera pas trop proche de  $x_k$ ).

**Théorème 2.6**

Supposant que :  $\theta$  est de classe  $C^1$ , bornée inférieurement et  $\theta'(0) < 0$ . Alors la recherche linéaire de Wolfe est finie.

**Démonstration :**

$\exists N \in \mathbb{N}$  tel que  $t_d^k \neq 0$  sinon  $t^k \rightarrow +\infty$ , avec

$$\theta(t^k) \leq \theta(0) + m_1 t^k \theta'(0)$$

Donc  $\theta(t^k)$  tendrait vers  $-\infty$  d'où une contradiction avec les hypothèses

Les suites  $\{t_g^k\}$  et  $\{t_d^k\}$  sont des suites adjacentes avec

$$\lim t_d^k = \lim t_g^k = t^*$$

$$\theta(t_d^k) > m_1 t_d^k \theta'(0) + \theta(0) \Rightarrow \theta(t^*) \geq m_1 t^* \theta'(0) + \theta(0)$$

$$\theta(t_g^k) \leq m_1 t_g^k \theta'(0) + \theta(0) \Rightarrow \theta(t^*) \leq m_1 t^* \theta'(0) + \theta(0)$$

Ce qui donne

$$\theta(t^*) = m_1 t^* \theta'(0) + \theta(0)$$

Pour  $t = \theta$  on a

$$\theta(0) = c \text{ et } \theta'(0) = b$$

Ce qui permet d'écrire

$$\theta(t) = \frac{1}{2}at^2 + \theta'(0)t + \theta(0)$$

Auquel cas

$$t^* = \frac{-\theta'(0)}{a}, \theta(t^*) = \frac{-\theta'^2(0)}{a} + \theta(0) \text{ et } \frac{[\theta(t^*) - \theta(0)]}{t^*} = \frac{1}{2}\theta'(0)$$

C'est-à-dire si  $\theta$  est quadratique, le pas optimale  $t^* = 1$  sera refusé par le test de descente si  $m_1 > \frac{1}{2}$ . On doit donc prendre

$$0 < m_1 < \frac{1}{2} < m_2 < 1 \quad (2.9)$$

#### Remarque 2.6

- Nous avons définie plusieurs méthodes de recherche linéaire ce qui permet de calculer  $x_{k+1}$  à partir de  $x_k$  suivant le schéma itératif déjà défini, donc

$$x_{k+1} = x_k + t_k d_k$$

Où  $d_k$  est une direction. On peut se demander si la suite ainsi générée converge vers une solution du problème (1.12). Bien entendu que cette propriété ne peut pas avoir indépendamment du choix de  $d_k$ , la recherche linéaire sera impuissante si  $d_k$  est "trop orthogonale" à  $g(x_k)$ . Ainsi on définit l'angle  $\varphi_k$  entre  $-g(x_k)$  et  $d_k$  par

$$c_k = \cos(\varphi_k) = -\frac{g(x_k)^\top d_k}{\|g(x_k)\| \|d_k\|} \quad (2.10)$$

Et d'après le théorème 1.6  $d_k$  serait une direction de descente si et seulement si  $c_k > 0$ .

- Dans le chapitre 1, on a vu un théorème essentiel de convergence (théorème 1.1.), ce théorème couvre un domaine très large d'algorithmes. Sachant que notre but est l'étude des méthodes utilisant essentiellement des directions de descentes, avec des recherches linéaires qui sont dans la plus part du temps inexacts, on utilise souvent le théorème suivant

Donc

$$(1 - m_2)c_k \|g(x_k)\| \leq L \|d_k\|$$

Synthèse :

On déduit des deux arguments que (avec  $r = \frac{m_1(1 - m_2)}{L}$ )

$$rc_k^2 \|g(x_k)\|^2 \leq f(x_k) - f(x_{k+1})$$

Donc

$$r \sum_{i=1}^{i=n} c_i^2 \|g_i\|^2 \leq f(x_1) - f(x_{k+1})$$

si  $f$  est bornée inférieurement alors (et puisque  $\{f(x_k)\}$  ↘)

$$\sum c_i^2 \|g_i\|^2 < \infty$$

Et si de plus on suppose que  $\|g(x_k)\| \geq c > 0, \forall k \in \mathbb{N}$  implique

$$\sum c_i^2 < \infty$$

Contrairement aux hypothèses.

**Remarque 2.7**

- On démontre que le théorème 2.7 reste valable si on utilise une recherche linéaire vérifiant ce qu'on appelle condition de Zoutendijk : Il existe une constante  $L > 0$  telle que pour tout  $k \geq 1$  on ait

$$f(x_{k+1}) \leq f(x_k) - L \|g(x_k)\|^2 c_k \quad (2.11)$$

- On note que les recherches linéaires exactes ainsi que la recherche linéaire du type Armijo et Goldstein vérifie la condition (2.11).

**Exemple 2.1**

Nous allons tester l'algorithme "steepest descent" avec différentes recherches linéaires, comme fonction test on prend : la vallée - banane - de Rosenbrock. Il s'agit d'une fonction à deux variables du quatrième degré qui se présente comme une vallée en U dont le fond assez plat est incurvé suivant une parabole.



Tableau 2.2 : Steepest descent avec recherche linéaire d'Armijo

$k$	$t_k$	$x_k$	$f(x_k)$	$\ g(x_k)\ $
1	—	(-1.200, 1.000)	24.200	232.867
10	$2.499 \times 10^{-03}$	(-1.016, 1.035)	4.067	3.239
100	$4.999 \times 10^{-03}$	(0.360, 0.123)	0.413	1.399
200	$2.499 \times 10^{-03}$	(0.683, 0.466)	0.100	0.409
500	$2.499 \times 10^{-03}$	(0.930, 0.865)	$4.841 \times 10^{-03}$	0.124
1000	$1.250 \times 10^{-03}$	(0.976, 0.953)	$5.501 \times 10^{-04}$	$2.304 \times 10^{-02}$
1500	$2.499 \times 10^{-03}$	(0.988, 0.977)	$1.225 \times 10^{-04}$	$1.390 \times 10^{-02}$
2000	$4.999 \times 10^{-03}$	(0.994, 0.989)	$2.838 \times 10^{-05}$	$1.037 \times 10^{-02}$

Tableau 2.3 : Steepest descent avec recherche linéaire de Goldstein

$k$	$t_k$	$x_k$	$f(x_k)$	$\ g(x_k)\ $
1	—	(-1.200, 1.000)	24.200	232.867
10	$9.024 \times 10^{-03}$	(-0.939, 0.878)	3.765	5.702
100	$8.573 \times 10^{-04}$	(-0.483, 0.248)	2.221	2.956
200	$8.573 \times 10^{-04}$	(0.386, 0.149)	0.376	1.221
500	$8.573 \times 10^{-04}$	(0.737, 0.543)	$6.894 \times 10^{-02}$	0.428
1000	$9.024 \times 10^{-03}$	(0.906, 0.821)	$8.749 \times 10^{-03}$	0.142
1500	$8.573 \times 10^{-04}$	(0.989, 0.978)	$1.117 \times 10^{-04}$	$1.356 \times 10^{-02}$
2000	$8.573 \times 10^{-04}$	(0.998, 0.996)	$2.716 \times 10^{-06}$	$1.478 \times 10^{-03}$

Tableau 2.4 : Steepest descent avec recherche linéaire de Wolfe

$k$	$t_k$	$x_k$	$f(x_k)$	$\ g(x_k)\ $
1	—	(-1.200, 1.000)	24.200	232.867
10	$2.499 \times 10^{-03}$	(-1.016, 1.03)	4.067	3.239
100	$4.999 \times 10^{-03}$	(0.593, 0.347)	0.166	0.762
200	$2.499 \times 10^{-03}$	(0.716, 0.511)	$8.059 \times 10^{-02}$	0.455
500	$2.000 \times 10^{-02}$	(0.927, 0.859)	$5.380 \times 10^{-03}$	0.332
1000	$9.999 \times 10^{-03}$	(0.975, 0.951)	$5.919 \times 10^{-04}$	$8.178 \times 10^{-02}$
1500	$9.999 \times 10^{-03}$	(0.988, 0.977)	$1.279 \times 10^{-04}$	$1.259 \times 10^{-02}$
2000	$1.250 \times 10^{-03}$	(0.994, 0.989)	$2.887 \times 10^{-05}$	$6.108 \times 10^{-03}$

## Chapitre 3

# Méthodes Newtoniennes

### 3.1 Définition de l'algorithme

L'algorithme de Newton est une méthode très générale pour résoudre un système d'équations non linéaires de la forme

$$g(x) = 0 \tag{3.1}$$

où  $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$  est supposée régulière. Pour le problème (1.12) déjà posé, on utilise cette méthode avec  $g(x) = \nabla f(x)$ , donc (3.1) résume la condition nécessaire d'optimalité du problème.

L'algorithme de Newton génère une suite de vecteurs  $\{x_k\}$  de la manière suivante. Supposant connu l'itéré courant  $x_k$ , l'équation (3.1) linéarisée en  $x_k$  est :

$$g(x_k) + g'(x_k)(x - x_k) = 0 \tag{3.2}$$

si  $g$  est le gradient d'une fonction  $f$  à minimiser alors,  $g'$  est son hessien. L'équation (3.2) est réécrite :

$$g(x_k) + H(x_k)(x - x_k) = 0$$

Lorsque  $H(x_k)$  est inversible la solution de l'équation est  $x = x_{k+1}$ , la nouvelle itéré de l'algorithme de Newton est donc

$$x_{k+1} = x_k + d_k \tag{3.3}$$

où

$$d_k = -H^{-1}(x_k)g(x_k) \tag{3.4}$$

### 3.1. DÉFINITION DE L'ALGORITHME

43

- La suite de matrices  $\{B_k\}$  est symétrique et définie positive
- Il existe  $\mu > 0$  tel que  $\|B_k\| \|B_k^{-1}\| < \mu, \forall k \in \{1, 2, \dots\}$

Alors;  $\liminf_{k \rightarrow \infty} \|g(x_k)\| = 0$ .

**Démonstration :**

Il suffit de démontrer que  $c_k = -\frac{g^t(x_k)d_k}{\|g(x_k)\| \|d_k\|} \geq \bar{\mu} > 0$  et on applique le théorème 2.8.

En effet; si  $\lambda_1 = \min(\sigma(B_k))$  et puisque  $B_k$  est symétrique définie positive

$\Rightarrow B_k^{-1}$  symétrique et définie positive alors  $\|B_k^{-1}\|_2 = \frac{1}{\lambda_1}$

Grâce à Cauchy-Schwarz et l'équivalence des normes;

$$\begin{aligned} c_k &= \frac{g(x_k)^T d_k}{\|g(x_k)\| \|d_k\|} = \frac{g(x_k)^t B_k g(x_k)}{\|g(x_k)\| \|B_k g(x_k)\|} \\ &\geq \frac{\lambda_1 \|g(x_k)\|^2}{\|g(x_k)\|^2 \|B_k\|} = \frac{1}{\|B_k^{-1}\|_2 \|B_k\|} \\ &\geq \bar{\mu} \end{aligned}$$

#### Théorème 3.1

Soit  $x^*$  une solution de (3.1). Supposant que  $g$  est de classe  $C^1$  dans un voisinage  $V_{x^*}$  de  $x^*$  et  $H(x^*)$  est inversible. S'il existe  $\varepsilon > 0$  et  $\delta > 0$  tels que

$$\|x_1 - x^*\| \leq \varepsilon \text{ et } \|B_k - H(x_k)\| \leq \delta$$

alors l'algorithme 3.1 avec  $d_k$  donné par (3.8) génère une suite  $\{x_k\}$  convergeant linéairement vers  $x^*$ . Si de plus  $\{B_k\}$  converge vers  $H(x^*)$ , alors la convergence est super-linéaire. Enfin, si  $H$  est lipschitzien et si  $B_k - H(x^*) = o(\|x_k - x^*\|)$ , alors la convergence est quadratique.

**Démonstration :**

$g \in C^1(V_{x^*}) \Rightarrow H$  est continue sur  $V_{x^*}$  et puisque  $H(x^*)$  est inversible  $\Rightarrow H(x_k)$  est inversible pour  $x_k \in V_{x^*}$ . Si de plus  $\delta$  est assez petit alors  $B_k$  sera aussi inversible donc la direction (3.4) est bien définie donc

$$\begin{aligned} x_{k+1} - x^* &= x_k + d_k - x^* = x_k - x^* - B_k^{-1}g(x_k) = \\ &= B_k^{-1}(B_k(x_k - x^*) - g(x_k)) \end{aligned}$$

En utilisant la formule (1.6) pour la fonction vectorielle  $g$

$$g(x_k) = g(x^* + (x_k - x^*)) = g(x^*) + \int_0^1 H(x^* + t(x_k - x^*))(x_k - x^*) dt$$



Ce théorème n'entraîne nullement la convergence globale de l'algorithme de Newton, mais il dit que si  $x_k$  est proche de la solution du problème (3.1), alors  $x_{k+1}$  est infiniment plus proche. En fait les inconvénients de la méthode de Newton sont bien connus :

1. L'algorithme n'est pas globalement convergent ;
2. L'algorithme n'est pas défini en un point  $x$  où  $H(x)$  est singulière ;
3. Si  $H(x_k)$  n'est pas définie positif ; la direction (3.4) n'est pas une direction de descente ;
4. Le point stationnaire obtenu peut être un minimum local, un maximum local ou un point de selle ;
5. Un système linéaire d'ordre  $n$  doit être résolu à chaque itération.

### 3.2 Modification de l'algorithme de Newton et globalisation

On note que, contrairement aux méthodes du premier ordre (Chapitre 2) l'algorithme (3.3), (3.4) fournit non seulement une direction, mais aussi une longueur de pas le long de cette direction, en d'autres termes, le pas  $t = 1$  est privilégié (pour avoir une convergence super-linéaire vue le théorème 3.1). Ceci est valable seulement pour  $x_1$  proche de la solution, loin de là on n'a pas de convergence globale, pire encore la direction (3.4) n'assure pas la descente ; c'est-à-dire

$$f(x_{k+1}) = f(x_k + d_k) \leq f(x_k)$$

Pour cela on peut interpréter (3.4) comme une direction, le long de laquelle on effectue une recherche linéaire pour forcer la descente entraînant la convergence globale.

Une telle recherche est possible que si  $H(x_k)$  est définie positif (pour avoir  $g(x_k)d_k < 0$ ). Sinon, on peut modifier le hessien ; car ce cas est facilement repérable (revenant aux remarques 3.1) pendant la décomposition de  $H(x_k)$  (par la méthode de Choleski), on s'aperçoit très vite qu'on doit extraire une racine carrée négative, on peut alors ajouter un terme positif là où il le faut, de façon à obtenir la définie positivité de la matrice hessienne. Il faut mentionner que cette technique est très gourmande en nombre d'opérations effectuées ; donc elle est peut-être efficace.

### 3.2. MODIFICATION DE L'ALGORITHME DE NEWTON ET GLOBALISATION 47

#### Exemple 3.1

On applique l'algorithme 3.2 sur le problème test de l'exemple 2.1, les résultats sont illustrés dans le tableau 3.1

Tableau 3.1

$k$	$t_k$	$x_k$	$f(x_k)$	$\ g(x_k)\ $
1	—	(-1.200, 1.000)	24.201	232.867
2	1.000	(-1.170, 1.380)	4.731	4.639
3	0.117	(-0.947, 0.844)	4.067	26.020
4	1.000	(-0.778, 0.576)	3.243	13.716
5	1.000	(-0.513, 0.193)	2.779	22.312
6	1.000	(-0.412, 0.160)	2.005	4.9507
7	0.490	(-0.185, -0.022)	1.729	13.165
8	1.000	(-0.089, -0.001)	1.193	3.101
9	0.490	(0.099, -0.030)	0.973	8.064
10	1.000	(0.198, 0.029)	0.651	2.137
11	0.700	(0.387, 0.111)	0.523	9.039
12	1.000	(0.457, 0.204)	0.296	1.007
13	0.490	(0.591, 0.329)	0.208	5.692
14	1.000	(0.671, 0.444)	0.111	1.703
15	1.000	(0.814, 0.642)	$7.565 \times 10^{-02}$	7.439
16	1.000	(0.850, 0.722)	$2.239 \times 10^{-02}$	0.314
17	0.700	(0.933, 0.863)	$9.590 \times 10^{-03}$	2.912
18	1.000	(0.960, 0.922)	$1.606 \times 10^{-03}$	0.260
19	1.000	(0.994, 0.988)	$1.634 \times 10^{-04}$	0.511
20	1.000	(0.999, 0.998)	$9.929 \times 10^{-07}$	$6.173 \times 10^{-03}$
21	1.000	(0.999, 0.999)	$1.030 \times 10^{-10}$	$4.210 \times 10^{-04}$
22	1.000	(1.000, 1.000)	$4.459 \times 10^{-19}$	$4.157 \times 10^{-09}$

#### 3.2.1 Algorithme de Newton tronqué

A l'étape 2 de l'algorithme 3.2, et vue les remarques précédentes, on voit bien qu'il est pratiquement très coûteux (ou impossible) de calculer exactement la direction  $d_k$ , on peut alors résoudre l'équation de Newton (3.5) de manière approchée. Dans la pratique cette résolution se fait par "un processus itératif interne" (le plus souvent il s'agit de l'algorithme du gradient conjugué linéaire), ce processus sera stopper avant convergence d'où le nom d'algorithme de Newton tronqué. Voici un exemple de tel méthodes.

#### Algorithme 3.3



### 3.2. MODIFICATION DE L'ALGORITHME DE NEWTON ET GLOBALISATION 49

#### Exemple 3.2

On applique l'algorithme de Newton tronqué sur le problème vue dans l'exemple 3.1, les résultats sont présentés dans le tableau 3.2

Tableau 3.2

$k$	$t_k$	$x_k$	$f(x_k)$	$\ g(x_k)\ $
1	—	(-1.200, 1.000)	24.200	232.867
2	1.000	(-1.175, 1.380)	4.731	4.639
3	0.117	(-0.947, 0.844)	4.067	26.020
4	1.000	(-0.778, 0.576)	3.243	13.716
5	1.000	(-0.513, 0.193)	2.779	22.312
6	1.000	(-0.412, 0.160)	2.005	4.950
7	0.490	(-0.185, -0.022)	1.729	13.165
8	1.000	(-0.089, -0.001)	1.195	3.101
9	0.490	(0.099, -0.030)	0.973	8.064
10	1.000	(0.198, 0.029)	0.651	2.137
11	0.700	(0.387, 0.111)	0.523	9.039
12	1.000	(0.457, 0.204)	0.296	1.007
13	0.490	(0.591, 0.329)	0.208	5.692
14	1.000	(0.671, 0.444)	0.111	1.706
15	1.000	(0.814, 0.642)	$7.565 \times 10^{-02}$	7.438
16	1.000	(0.850, 0.722)	$2.239 \times 10^{-02}$	0.314
17	0.700	(0.933, 0.863)	$9.59 \times 10^{-03}$	2.912
18	1.000	(0.960, 0.922)	$1.606 \times 10^{-03}$	0.260
19	1.000	(0.994, 0.988)	$1.634 \times 10^{-04}$	0.511
20	1.000	(0.999, 0.998)	$9.931 \times 10^{-07}$	$6.174 \times 10^{-03}$
21	1.000	(0.999, 0.999)	$1.031 \times 10^{-10}$	$4.211 \times 10^{-04}$
22	1.000	(1.000, 1.000)	$4.462 \times 10^{-19}$	$4.159 \times 10^{-09}$

#### Remarque 3.2

L'algorithme de Newton (ainsi ces modifications) est très efficace, seulement pour le mettre en oeuvre, il nécessite à chaque itération de l'algorithme, le calcul de la matrice hessienne ainsi que son inverse, ce qui demande un temps de calcul considérable et un investissement humain trop important. Dans ce cas l'algorithme de Quasi-Newton, offre une alternative.

# Bibliographie

- [1] **W. Alt**, The Lagrange-Newton method for infinite-dimensional optimization problems. *Numerical Functional Analysis and Optimization*. 11, (1990), 201-224.
- [2] **D. P. Bertsekas**, Nonlinear programming, *Athena Scientific*, 1995.
- [3] **M. Bierlaire**, Introduction à l'optimisation différentiable, *PPUR, Lausanne*, 2006.
- [4] **J.F. Bonnans**, Local analysis of Newton-type methods for variational inequalities and nonlinear programming, *Applied Mathematics and Optimization*. 29, (1994), 161-186.
- [5] **J. M. Borwein and A. S. Lewis**, *Convex Analysis and Nonlinear Optimization*. Springer, New York, 2000.
- [6] **P.G. Ciarlet**, Introduction à l'Analyse Numérique Matricielle et à l'Optimisation, (Second édition). *Masson, Paris*, 1988.
- [7] **J.C. Culioli**, Introduction à l'optimisation, ellipes, 1994.
- [8] **J-B. Hiriart-Urruty and C. Lemaréchal**, Fundamentals of convex analysis. coll. «*Grundlehren Text Editions*», *Springer-Verlag, Berlin*, 2001.
- [9] **S.Mottelet**, optimisation non linéaire, Université de technologie de compiegne, Printemps, 2003.
- [10] **C. Nicolescu and L.-E. Persson**, *Convex Functions and their Applications : A Contemporary Approach*, coll. «*Ouvrages de mathématiques de la Société mathématique du Canada*», vol. 23, *Springer*, 2006.
- [11] **R.T. Rockafellar**, *Convex Analysis*. *Princeton Mathematics Ser. 28*. *Princeton University Press, Princeton, New Jersey*, 1970.