



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT DE MATHÉMATIQUES

Pour l'obtention du diplôme de :

MASTER

Spécialité : Analyse Fonctionnelle et Applications

Par :

**Chekroune Sofiane
Yahiaoui Abdelali
Zaich Amine**

Sur le thème

Résolution d'un problème d'optimisation les méthodes Quasi-Newton

Soutenu publiquement le 17 / 07 / 2021 à Tiaret devant le jury composé de :

Mr A.Larabi	MCA Université de Tiaret	Président
Mr N.Rezoug	MAA Université de Tiaret	Encadreur
Mr K.Maazouz	MCA Université de Tiaret	Examineur

2020-2021

★————— *Remerciement* —————★

Nous devond remercie tout d'abord "Allah" de nous
avoir donné
le courage d'entamer et de finir ce mémoire dans de
bonnes conditions.

Nous adressons nos vifs remerciements à notre encadreur
Mr N. Rezzoug qui par
ses conseils, ses recommandations, sa patience nous a
permis de réaliser ce mémoire avec un très grand plaisir.

Nos remerciements les plus chaleureux vont
à Mr A. Larabi, doyen de la Faculté de mathématiques
et informatique, pour nous avoir honoré de présider le
jury de notre mémoire. Nous également remercions
beaucoup Mr K. Maazouz, professeur au Département
de Mathématiques de l'Université Ibn Khaldoune de
Tiaret, pour l'honneur qu'il nous a fait en acceptant
d'examiner ce mémoire.

À tous les enseignants qui ont participés à notre
formation tout au long de notre cycle universitaire.

Enfin, nous ne voulons pas oublier tous les professeurs
que nous avons rencontrés tout au long de ces années de
licence et tous ceux qui ont contribué de près ou de loin
à la réalisation de ce travail, merci.



Dédicace

Je dédie ce modeste travail :

À ma mère qui m'a soutenu dans la réussite de mes études.

À mes frères.

À mes soeurs.

À tous mes professeurs.

À tous mes chers amis.

Yahiaoui Abdelali

Dédicace

Je dédie ce modeste travail :

À mes chers parents

Je vous remercie pour tout le soutien que vous me portez depuis mon enfance .
Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de
vos innombrables sacrifices, bien que je ne vous en acquitte jamais assez.

Puisse Dieu, le très haut, vous accorde santé, bonheur et longue vie.

À mes frères qui sont présents dans tous mes moments d'examens par leur
soutien moral et leurs belles surprises sucrées.

Je vous souhaite un avenir plein de joie, de bonheur, de réussite et de sérénité.

À tous mes adorables amis.

Cheikroune Sofiane

Dédicace

Tout d'abord, je remercie ALLAH de m'avoir donné la force de terminer ce
mémoire. Je dédie ce modeste travail :

À mes très chers parents qui m'ont soutenu dans la réussite de mes études.

À mes frères.

À mes soeurs.

À toute la famille Zaich.

À tous mes amis.

Zaich Amine

Table des matières

Notation	1
Introduction	2
1 Préliminaires	4
1.1 Dérivée partielle	4
1.2 Dérivée directionnelle	4
1.3 Différentiabilité	5
1.4 Gradient	6
1.5 Matrice Hessienne	6
1.6 Formules de Taylor	7
1.7 Convexité	9
1.7.1 Ensemble convexe	9
1.7.2 Fonction convexe	10
1.7.3 Fonction concave	11
2 Optimisation sans et avec contraintes	16
2.1 Classification des problèmes d'optimisation	16
2.1.1 Forme générale d'un problème d'optimisation	16
2.1.2 Problème d'optimisation sans contraintes	16

2.1.3	Problème d'optimisation avec contraintes	17
2.2	Direction de descente	18
2.3	Schémas général des algorithmes d'optimisation sans contraintes	19
2.4	Conditions d'optimalité	21
2.4.1	Conditions nécessaires d'optimalité du premier ordre	21
2.4.2	Conditions nécessaires d'optimalité du second ordre	21
2.4.3	Conditions suffisantes d'optimalité	22
3	Méthodes Quasi-Newton	24
3.1	Recherches linéaires D'Armijo et de Wolfe	24
3.1.1	Recherche linéaire d'Armijo	27
3.1.2	Recherche linéaire de Wolfe	28
3.2	Méthode de Newton	28
3.2.1	Algorithme	29
3.2.2	Etude de la convergence	30
3.2.3	Avantages et Inconvénients de la méthode de Newton	32
3.3	Méthodes Quasi-Newton	34
3.3.1	Dérivation des méthodes	35
3.3.2	Méthode de correction de rang un	36
3.3.3	Méthode de Davidon-Fletcher-Powell(D.F.P)	41
3.3.4	Méthode BFGS	45
3.4	Convergence des méthode Quasi-Newton	47
	Bibliographie	50

Notation

\mathbb{R}	Ensemble des nombres réels.
J	Jacobienne.
L	Lagrangienne.
H	Hessienne.
d_k	Direction de descente.
α_k	Pas de minimisation.
$f(\cdot)$	Fonction objective.
$\ \cdot\ $	Norme d'un vecteur.
$x = (x_1, \dots, x_n)^t$	Vecteur d'éléments en colonne.
\langle, \rangle	Produit scalaire.
x^t	Transposée d'un vecteur x .
∇f	Gradient de f .
$M_{m,n}$	Matrice de m ligne et n colonne.
$C(\mathbb{R}^n, \mathbb{R})$	L'espace des fonctions continues de \mathbb{R}^n dans \mathbb{R}
x^*	Solution optimale.
H^{-1}	Matrice inverse.
$\frac{\partial f}{\partial x}$	Dérivée de f suivant x .
$\frac{\partial^2 f}{\partial x_i \partial x_j}$	Dérivée partielle seconde de f .
DFP	Davidon-Fletcher-Powell.
$BFGS$	Broydon-Fletcher-Goldfarb-Shanno.

Introduction

L'optimisation mathématique, regroupe un ensemble de sujets dans l'étude des problèmes d'optimisation. Une fonction objectif est donnée, le problème consiste à trouver -caractériser- calculer un point minimisant (ou maximisant) cette fonction, tous les points de l'espace sont candidats, parfois des contraintes limitent le domaine de recherche.

Les fonctions utilisées sont continues, même différentiables. On utilise des résultats d'analyse mathématique pour caractériser les points candidats, un premier pas consiste donc à obtenir des conditions vérifiables satisfaites par les minimums ou maximums recherchés. Lorsqu'un point ne satisfait pas ces conditions d'optimalité, on en déduit une manière de calculer un point meilleur, et finalement un algorithme itératif réduisant (pour la minimisation) progressivement la fonction objectif.

Un effort important est consacré à relier l'étude des conditions d'optimalité au développement d'algorithmes de résolution. Ce lien entre la théorie et les algorithmes constitue le fil conducteur du texte, ainsi que l'originalité de la présentation.

Dans ce mémoire, on va étudier la résolution d'un problème d'optimisation en utilisant les méthodes Quasi-Newton, qui sont basées sur la méthode de Newton et les conditions nécessaires d'optimalité .

Ce travail est réparti en trois chapitres, le premier chapitre est consacré à quelques définitions

et résultats préliminaires qu'on va utiliser tout au long de ce mémoire .

Dans le deuxième chapitre, on va introduire les bases concernant l'optimisation en particulier, classification des problèmes d'optimisation et direction de descente, et schémas général des algorithmes d'optimisation sans contraintes

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

Dans le troisième chapitre, on va commencer par la Recherches linéaires d'Armijo et de Wolfe. Ensuite, on présentera la méthode de newton, puis on etendra à la nouvelle méthode nécessaire "Quasi-Newton" dans le reste de ce chapitre.

Préliminaires

Dans ce chapitre, on introduit les outils fonctionnels de base pour l'optimisation sans contraintes.

1.1 Dérivée partielle

Définition 1.1.1 (*Dérivées partielles en un point*) Soient $n \in \mathbb{N}^*$, U un ouvert de \mathbb{R}^n et f une fonction de \mathbb{R}^n vers \mathbb{R} définie sur U . La fonction f est dite d'avoir une dérivée partielle à $a = (a_1, \dots, a_n)$, par rapport à la i ème variable si la limite

$$\lim_{t \rightarrow a_i} \frac{f(a_1, \dots, a_{i-1}, t, a_{i+1}, \dots, a_n) - f(a)}{t - a_i},$$

existe. On note cette limite par

$$\frac{\partial f}{\partial x_i}(a) \quad \text{ou} \quad \partial_i f(a) \quad \text{ou} \quad D_i f(a).$$

1.2 Dérivée directionnelle

Les dérivées directionnelles sont des dérivées calculées en suivant une direction .

Définition 1.2.1 (*Dérivées directionnelles à un point*) Soient $n \in \mathbb{N}^*$, U un ouvert de \mathbb{R}^n , $f : U \rightarrow \mathbb{R}$, et $v \in \mathbb{R}^n$. La fonction f est dite d'avoir une dérivée directionnelle au point

$a \in U$ dans la direction v si la limite suivante existe

$$\lim_{h \rightarrow 0} \frac{1}{h} [f(a + hv) - f(a)].$$

et on la note $D_v f(a)$

Proposition 1.2.1 Soient $n \in \mathbb{N}^*$, $U \subset \mathbb{R}^n$ un ouvert de \mathbb{R}^n , $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable sur U . Alors

$$D_v f(a) = \nabla f(a) \cdot v.$$

1.3 Différentiabilité

Définition 1.3.1 1) On dit que la fonction $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est différentiable en $a \in U$ s'il existe une application L linéaire continue de \mathbb{R}^n dans \mathbb{R} telle que

$$\lim_{h \rightarrow 0} \frac{f(a + h) - f(a) - L(h)}{\|h\|} = 0, \quad (1.1)$$

où $h = (h_1, \dots, h_n)$.

2) Lorsque l'application L existe, elle s'appelle la différentielle de f en a , On la note $Df(a)$, ainsi (1.1) devient

$$\lim_{h \rightarrow 0} \frac{f(a + h) - f(a) - Df(a)(h)}{\|h\|} = 0.$$

3) Si f est différentiable en tout point de U on dit que f est différentiable sur U .

Remarque 1.3.1 La différentiabilité de f en a est équivalente à l'existence d'une forme linéaire L sur \mathbb{R}^n telle que

$$f(a + h) = f(a) + Df(a)(h) + o(h). \quad (1.2)$$

Proposition 1.3.1 Soient U un ouvert et $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$, si f de classe C^1 alors f est différentiable sur U .

1.4 Gradient

Si les dérivées partielles $\partial f/\partial x_i$ existent pour tout i , le gradient de f est défini de la façon suivante.

Définition 1.4.1 *On note par*

$$(\nabla f(x))^T = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)_{(x)},$$

le gradient de f au point $x = (x_1, \dots, x_n)$.

Le gradient jouera un rôle essentiel dans le développement et l'analyse des algorithmes d'optimisation.

Proposition 1.4.1 (*Gradient de la composée*) *Supposons qu'on a deux ouverts $\Omega \subset \mathbb{R}^n$ et $U \subset \mathbb{R}$ et deux fonctions $f : \Omega \rightarrow \mathbb{R}$ et $g : U \rightarrow \mathbb{R}$ avec en plus $f(\Omega) \subset U$ (on peut alors définir $g \circ f : \Omega \rightarrow \mathbb{R}$).*

Supposons que f et g sont de classe C^1 . Alors $g \circ f$ est aussi de classe C^1 avec

$$\nabla(g \circ f)(x) = g'(f(x))\nabla f(x), \quad \forall x \in \Omega.$$

1.5 Matrice Hessienne

Si les dérivées partielles $(\partial^2 f/\partial x_i \partial x_j)$ existent pour tout $i, j \in N^*$, la matrice hessienne définie comme suit

Définition 1.5.1 *On appelle Hessienne de f la matrice de $M_{n,n}(\mathbb{R})$*

$$H(x) = \nabla (\nabla^T f) (x) = \nabla^2 f(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right) (x), i = 1, \dots, n, j = 1, \dots, n.$$

Alors

$$H(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_2} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}.$$

Théorème 1.5.1 (Schwarz) soit $\Omega \subset \mathbb{R}^n$ un ouvert.

Si $f \in C^2(\Omega)$ alors $\nabla^2 f(x)$ est une matrice symétrique $\forall x \in \Omega$.

Exemple 1.5.1 Soit $f(x_1, x_2, x_3) = e^{x_1} - x_1 x_2 x_3$. L'hessienne de f est donné par

$$\nabla f(x) = \begin{pmatrix} e^{x_1} - x_2 x_3 \\ -x_1 x_3 \\ -x_1 x_2 \end{pmatrix}.$$

Alors

$$H(x) = \begin{pmatrix} e^{x_1} & -x_3 & -x_2 \\ -x_3 & 0 & -x_1 \\ -x_2 & -x_1 & 0 \end{pmatrix}.$$

1.6 Formules de Taylor

Formule de Taylor avec reste intégrale

Théorème 1.6.1 Soit U un ouvert de \mathbb{R}^n et $a, h \in \mathbb{R}^n$ tel que $[a, a+h] \subset U$.

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction de $C^2(\mathbb{R}^n)$. Alors on a

$$f(a+h) = f(a) + \sum_{i=1}^n h_i \frac{\partial f}{\partial x_i}(a) + \sum_{i,j=1}^n h_i h_j \int_0^1 (1-t) \frac{\partial^2 f}{\partial x_i \partial x_j}(a+th) dt.$$

Formule de Taylor-Lagrange

Théorème 1.6.2 Soit U un ouvert de \mathbb{R}^n et $a, h \in \mathbb{R}^n$ tel que $[a, a+h] \subset U$.

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable sur U , la formule de Taylor-Lagrange

d'ordre 2 est

$$f(a+h) = f(a) + \sum_{i=1}^n h_i \frac{\partial f}{\partial x_i}(a) + \frac{1}{2} \sum_{i,j=1}^n h_i h_j \frac{\partial^2 f}{\partial x_i \partial x_j}(a + \theta h) \quad \text{avec } \theta \in]0, 1[.$$

Formule de Taylor-young

Théorème 1.6.3 Soit U un ouvert de \mathbb{R}^n et $a \in U$. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction de classe C^2 de \mathbb{R}^n dans \mathbb{R} , la formule de Taylor-young se réécrit

$$f(a+h) = f(a) + \sum_{i=1}^n h_i \frac{\partial f}{\partial x_i}(a) + \frac{1}{2} \sum_{i,j=1}^n h_i h_j \frac{\partial^2 f}{\partial x_i \partial x_j}(a) + \varepsilon(h) \|h\|^2.$$

Théorème 1.6.4 Soit f une application continue sur un segment $[a, b]$ et dérivable sur $]a, b[$. S'il existe $M \in \mathbb{R}$ tel que $f'(x) \leq M$ pour tout $x \in]a, b[$ alors

$$f(b) - f(a) \leq M(b - a).$$

Preuve : Soit $\epsilon > 0$. Considérons la fonction h définie sur $[a, b]$ par

$$h(x) = f(x) - f(a) - M(x - a) - \epsilon(x - a).$$

Si l'on montre que $h(b) \leq \epsilon$ alors la proposition est démontrée. En effet

$$h(b) \leq \epsilon \Leftrightarrow \frac{f(b) - f(a) - M(b - a)}{b - a + 1} \leq \epsilon$$

d'où, l'inégalité étant vraie pour tout $\epsilon > 0$, $\frac{f(b) - f(a) - M(b - a)}{b - a + 1} \leq 0$. Où $b - a + 1 > 0$ et donc $f(b) - f(a) \leq M(b - a)$. Montrons donc que $h(b) \leq \epsilon$.

Soit $A = \{x \in [a, b] \mid h(x) \leq \epsilon\}$. Comme $h(a) = 0$, $a \in A$ et, la fonction h étant continue en a , il existe $d \in]a, b]$ tel que $h(\bar{d}) \leq \epsilon/2$. Donc $A \neq \{a\}$. Soit $c = \sup A \in]a, b]$. Il existe une suite (x_n) d'éléments de A qui converge vers c . Pour tout n , $h(x_n) \leq \epsilon$ et donc, la fonction h étant continue en c , $h(c) \leq \epsilon$ d'où $c \in A$.

supposons que $c < b$. Il existe $t \in]c, b]$ telque $\frac{f(t)-f(c)}{t-c} \leq f'(c) + \epsilon$, car

$$\lim_{x \rightarrow c, x > c} \frac{f(x) - f(c)}{x - c} = f'(c).$$

On a

$$f(t) - f(c) \leq (t - c)f'(c) + (t - c)\epsilon \leq M(t - c) + (t - c)\epsilon.$$

Comme $c \in A$,

$$f(c) - f(a) \leq M(c - a) + \epsilon(c - a) + \epsilon.$$

d'où, par addition,

$$f(t) - f(a) \leq M(t - a) + \epsilon(t - a) + \epsilon,$$

ce qui signifie $t \in A$. C'est absurde et donc $c = b$ et $h(b) \leq \epsilon$. ■

1.7 Convexité

1.7.1 Ensemble convexe

Définition 1.7.1 *un ensemble $U \subset \mathbb{R}^n$ est dit convexe si*

$$\forall x, y \in U, \quad \forall t \in [0, 1] \text{ on a } tx + (1 - t)y \in U.$$

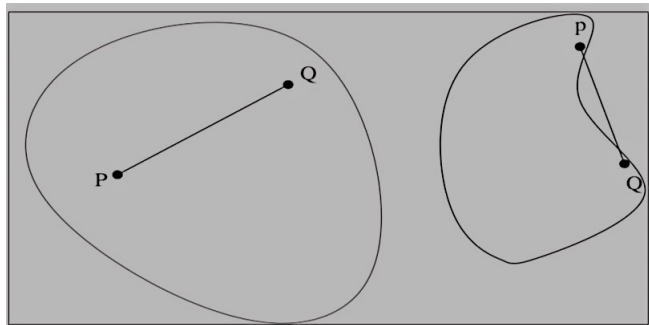
Autrement dit U est convexe si pour tous points x, y de U , le segment $[x, y]$ est inclus dans U (c'est à dire, $\forall x, y \in U$ on a $[x, y] \subset U$).

Définition 1.7.2 *Soit $x_1, x_2, \dots, x_n \in \mathbb{R}^n$ et $\lambda_j \geq 0$ tels que $\sum_{j=1}^k \lambda_j = 1$. Toute expression de la forme $\sum_{j=1}^k \lambda_j x_j$ s'appelle combinaison convexe des points x_j .*

Proposition 1.7.1 *i) L'intersection d'une famille quelconque de convexes est convexe.*

ii) Le produit cartésien de deux ensembles convexes est convexe.

iii) L'image directe d'un convexe par une application linéaire est convexe.



Ensemble convexe et ensemble non convexe

1.7.2 Fonction convexe

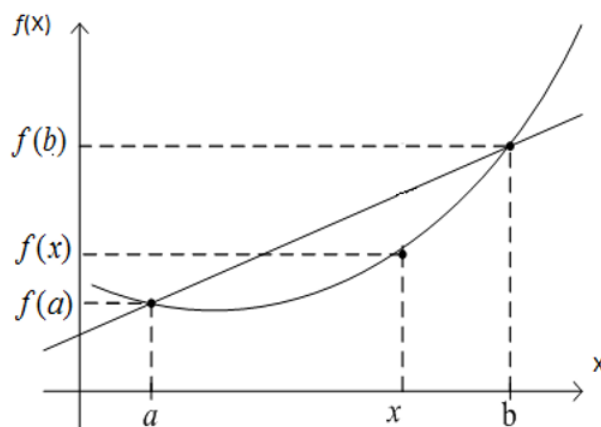
Définition 1.7.3 Soit $U \subset \mathbb{R}^n$ un ensemble convexe et $f : U \rightarrow \mathbb{R}$ une fonction. On dit que f est convexe sur U si

$$\forall x, y \in U, \quad \forall t \in [0, 1]$$

$$f(ty + (1 - t)x) \leq tf(y) + (1 - t)f(x).$$

Autrement dit pour tout $\alpha, \beta \in \mathbb{R}_+$ tel que $\alpha + \beta = 1$, on a

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y).$$



Courbe représentative d'une fonction convexe

Exemple 1.7.1 Les fonctions affines $f(x) = ax + b$ sont des fonctions convexes ou bien

concaves. En effet, soit $\alpha, \beta \in \mathbb{R}_+$ tel que $\alpha + \beta = 1$

on a

$$f(\alpha x + \beta y) = a(\alpha x + \beta y) + b = a(\alpha x + \beta y) + (\alpha + \beta)b = \alpha(ax + b) + \beta(ay + b) \leq \alpha f(x) + \beta f(y).$$

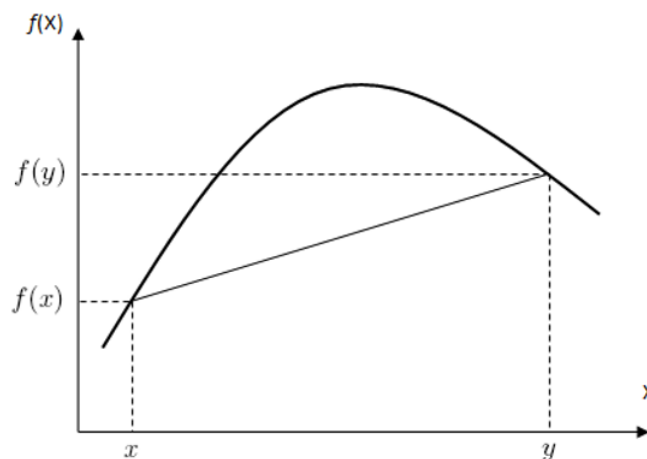
- La somme de deux fonctions convexes est convexe.

- Le produit d'une fonction convexe par un scalaire n'est pas toujours convexe.

1.7.3 Fonction concave

Définition 1.7.4 Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite concave si $(-f)$ est une fonction convexe, c'est à dire

$$\forall x, y \in \mathbb{R}^n, \quad \forall \lambda \in [0, 1], \quad f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y).$$



Courbe représentative d'une fonction concave

Définition 1.7.5 Soit $U \subset \mathbb{R}^n$ un ensemble convexe et $f : U \rightarrow \mathbb{R}$ une fonction.

1. On dit que f est strictement convexe sur U si

$$\forall x, y \in U \text{ avec } x \neq y, \forall t \in [0, 1], f(ty + (1 - t)x) < tf(y) + (1 - t)f(x).$$

2. On dit que f est fortement convexe sur U s'il existe $\alpha > 0$ tel que

$$\forall x, y \in U, \forall t \in [0, 1], \quad f(ty + (1-t)x) < tf(y) + (1-t)f(x) - \alpha t(1-t)\|y-x\|^2.$$

3. On dit que f est concave sur U (respectivement strictement concave, respectivement fortement concave) si $-f$ est convexe (respectivement strictement convexe, respectivement fortement convexe).

Remarque 1.7.1 Il est facile de voir que

fortement convexe \Rightarrow strictement convexe \Rightarrow convexe.

Mais la réciproque n'est pas vraie en général, par exemple une application affine

$f(x) = ax + b$, $a > 0$ est convexe mais elle n'est pas strictement convexe donc elle n'est pas fortement convexe.

Proposition 1.7.2 Soit $U \subset \mathbb{R}^n$ un ensemble convexe, $p \in \mathbb{N}^*$, $f_1, f_2, \dots, f_p : U \rightarrow \mathbb{R}$ des fonctions convexes et $\gamma_1, \gamma_2, \dots, \gamma_n$ des constantes strictement positives. Posons

$$f = \gamma_1 f_1 + \gamma_2 f_2 + \dots + \gamma_n f_n.$$

Alors

1. La fonction f est convexe (toute combinaison linéaire avec des coefficients strictement positifs de fonctions convexes est convexe).

2. Si au moins l'une des fonctions f_1, f_2, \dots, f_p est strictement convexe alors f est strictement convexe.

3. Si au moins l'une des fonctions f_1, f_2, \dots, f_p est fortement convexe alors f est fortement convexe.

Proposition 1.7.3 *Nous avons*

$$\nabla^2 f(x)h = \nabla \langle \nabla f(x), h \rangle, \quad \forall x \in U, \quad \forall h \in \mathbb{R}^n.$$

où le premier gradient dans le membre droite de l'égalité est considéré par rapport à la variable x .

Proposition 1.7.4 *Soit $\Omega \subset \mathbb{R}^n$ un ouvert, $U \subset \Omega$ avec U convexe et $f : \Omega \rightarrow \mathbb{R}$ une fonction de classe C^1 alors*

a) *Les trois assertions suivantes sont équivalentes*

1) *f est convexe sur U .*

2) *$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \quad \forall x, y \in U.$*

3) *∇f est monotone sur U , c'est à dire*

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \geq 0 \quad \forall x, y \in U.$$

b) *Si de plus f est de classe C^2 sur Ω alors f est convexe sur U si et seulement si*

$$\langle \nabla^2 f(x)(y - x), y - x \rangle \geq 0 \quad \forall x, y \in U. \tag{1.3}$$

Preuve : a) On montre l'équivalence .

1) \Rightarrow 2)

supposons que f est convexe, la définition de la convexité peut s'écrire

$$f(x + t(y - x)) - f(x) \leq t[f(y) - f(x)].$$

En fixant x, y , en divisant par t et en faisant t tend vers 0 (ce qui est possible car $t \in [0, 1]$),

on obtient 2).

2) \Rightarrow 3)

De 2) on déduit

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \quad \forall x, y \in U.$$

et on a aussi (prendre y en x et x en y)

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle, \quad \forall x, y \in U.$$

On faisant la somme de ces 2 inégalités on obtient 3).

3) \Rightarrow 1)

Soient $x, y \in U$ fixés. On introduit la fonction $g : I \rightarrow \mathbb{R}$ définie par

$$t \in I \longmapsto g(t) = f(ty + (1 - t)x) \in \mathbb{R}.$$

où I est un intervalle ouvert qui contient $[0, 1]$.

Il est facile de voir que g est de classe C^1 et on a

$$g'(t) = \langle \nabla f(ty + (1 - t)x), y - x \rangle \quad \forall t \in I.$$

Soient $t_1, t_2 \in [0, 1]$ avec $t_1 < t_2$, alors

$$\begin{aligned} g'(t_2) - g'(t_1) &= \langle \nabla f(x + t_2(y - x)) - \nabla f(x + t_1(y - x)), y - x \rangle \\ &= \langle \nabla f(x + t_2(y - x)) - \nabla f(x + t_1(y - x)), (t_2 - t_1)(y - x) \rangle \frac{1}{t_2 - t_1}. \end{aligned}$$

Par hypothèse 3) le dernier terme de l'égalité précédente supérieure ou égale à zéro, ce qui montre que la fonction g' est une fonction croissante. On déduit alors que g est une fonction convexe sur $[0, 1]$, ce qui nous donne pour tout $t \in [0, 1]$

$$g(t(1) + (1 - t)(0)) \leq tg(1) + (1 - t)g(0),$$

c'est à dire

$$f(ty + (1-t)x) \leq tf(y) + (1-t)f(x).$$

donc f est convexe.

b) On suppose ici $f \in C^2$.

\Rightarrow " Supposons que f est convexe et montrons (1.3) Soit $h \in \mathbb{R}^n$ fixé et notons $g : \Omega \mapsto \mathbb{R}$ la fonction donnée par $g(x) = \langle \nabla f(x), h \rangle, \forall x \in \Omega$. En utilisant aussi la proposition 1.7.3

$$\langle \nabla^2 f(x)h, h \rangle = \langle \nabla g(x), h \rangle = \frac{\partial g}{\partial h}(x) = \lim_{t \rightarrow 0} \frac{(\nabla f(x+th), h) - \langle \nabla f(x), h \rangle}{t},$$

ce qui nous donne

$$\langle \nabla^2 f(x)h, h \rangle = \lim_{t \rightarrow 0} \frac{\langle \nabla f(x+th) - \nabla f(x), th \rangle}{t^2}.$$

■

Optimisation sans et avec contraintes

2.1 Classification des problèmes d'optimisation

2.1.1 Forme générale d'un problème d'optimisation

Etant donné un ensemble $X \subset \mathbb{R}^n$ et une fonction $f : X \rightarrow \mathbb{R}$, la forme générale d'un problème d'optimisation est la suivante

$$\min\{f(x) : x \in X\}. \quad (2.1)$$

2.1.2 Problème d'optimisation sans contraintes

Si $X = \mathbb{R}^n$, le problème d'optimisation (2.1) s'appelle problème d'optimisation sans contraintes et aura donc la forme suivante

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

Définition 2.1.1 1) On dit que $\hat{x} \in \mathbb{R}^n$ est un minimum global de (P) si et seulement si

$$f(\hat{x}) \leq f(x) \quad \forall x \in \mathbb{R}^n.$$

2) on dit que $\hat{x} \in \mathbb{R}^n$ est un minimum local de (P) si et seulement s'il existe un voisinage $v_\varepsilon(\hat{x})$ tel que

$$f(\hat{x}) \leq f(x) \quad \forall x \in V_\varepsilon(\hat{x}).$$

3) on dit que $\hat{x} \in \mathbb{R}^n$ est un minimum local strict de (P) si et seulement s'il existe un voisinage $V_\varepsilon(\hat{x})$ tel que

$$f(\hat{x}) < f(x) \quad \forall x \in V_\varepsilon(\hat{x}), \quad x \neq \hat{x}.$$

Définition 2.1.2 Le problème (P) est dite problème d'optimisation sans contraintes non linéaire si f n'est pas affine.

Définition 2.1.3 Le problème (P) est dite problème d'optimisation sans contraintes non linéaire quadratique si

$$f(x) = x^t A x - b x. \quad (2.2)$$

où A est une matrice symétrique d'ordre (n, n) .

2.1.3 Problème d'optimisation avec contraintes

Si l'ensemble X est défini par des équations ou des inéquations, le problème (2.1) s'appelle dans ce cas problème d'optimisation avec contraintes. De façon plus précise le problème d'optimisation avec contraintes par exemple s'écrit

$$\left\{ \begin{array}{l} \min f(x) \\ c_j(x) \leq 0, \quad j \in I \\ c_j(x) = 0, \quad j \in E, \end{array} \right. \quad (2.3)$$

où I et E sont deux ensembles disjoints d'entiers.

2.2 Direction de descente

Définition 2.2.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\hat{x} \in \mathbb{R}^n$, $d \in \mathbb{R}^n$ est dite direction de descente au point \hat{x} si et seulement s'il existe $\delta > 0$ tel que

$$f(\hat{x} + \lambda d) < f(\hat{x}), \quad \forall \lambda \in]0, \delta[.$$

Donnons maintenant une condition suffisante pour que d soit une direction de descente.

Théorème 2.2.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable au point $\hat{x} \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$ une direction vérifiant la condition suivante

$$\nabla f(\hat{x})^t \cdot d < 0.$$

Alors d est une direction de descente au point \hat{x} .

Preuve : f est différentiable au point \hat{x} . Donc

$$f(\hat{x} + \lambda d) = f(\hat{x}) + \lambda \nabla f(\hat{x})^t \cdot d + \lambda \|d\| \alpha(\hat{x}, \lambda d), \quad \text{avec } \alpha(\hat{x}, \lambda d) \xrightarrow{\lambda \rightarrow 0} 0,$$

ceci implique que

$$f'(\hat{x}, d) = \lim_{\lambda \rightarrow 0} \frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} = \nabla f(\hat{x})^t \cdot d < 0.$$

La limite étant strictement négative, alors il existe un voisinage de zéro ,

$V(0) =] - \delta, +\delta[$ tel que

$$\frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} < 0, \quad \forall \lambda \in] - \delta, +\delta[, \quad (2.4)$$

la relation (2.4) est particulièrement vraie pour tout $\lambda \in]0, +\delta[$, en multipliant la relation (2.4) par $\lambda > 0$. On obtient le résultat cherché. ■

2.3 Schémas général des algorithmes d'optimisation sans contraintes

Supposons que d_k soit une direction de descente au point x_k , ceci nous permet de considérer le point x_{k+1} , successeur de x_k , de la manière suivante

$$x_{k+1} = x_k + \lambda_k d_k, \quad \lambda_k \in]0, +\delta[.$$

Vu la définition de direction de descente, on est assuré que

$$f(x_{k+1}) = f(x_k + \lambda_k d_k) < f(x_k).$$

Un bon choix de d_k et de λ_k permet ainsi de construire une multitude d'algorithmes d'optimisation.

Exemple 2.3.1 (*choix de directions de descente*) Par exemple, si on choisit $d_k = -\nabla f(x_k)$ et si $\nabla f(x_k) \neq 0$, on obtient la méthode du gradient. La méthode de Newton correspond à $d_k = -(H(x_k))^{-1} \cdot \nabla f(x_k)$.

Bien sur $-\nabla f(x_k)$ est une direction de descente car

$$\nabla f(x_k)^t \cdot d_k = -\nabla f(x_k)^t \cdot \nabla f(x_k) = -\|\nabla f(x_k)\|^2 < 0.$$

Pour la deuxième direction, si la matrice hessienne $H(x_k)$ d'une fonction est définie positive alors $d_k = -(H(x_k))^{-1} \cdot \nabla f(x_k)$ est aussi une direction de descente correspond à la méthode de Newton.

Exemple 2.3.2 (*choix de pas λ_k*) On choisit en général λ_k de façon optimale, c'est à dire que

λ_k doit vérifier

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k), \quad \forall \lambda \in [0, +\infty[.$$

En d'autres termes on est ramené à étudier à chaque itération un problème de minimisation d'une variable réelle. C'est ce qu'on appelle recherche linéaire.

Matrice définie positive

Soit M une matrice carrée symétrique d'ordre n . Soit x un vecteur colonne de \mathbb{R}^n . On note x^t sa transposée. M est dite définie positive si et seulement si

$$x^t M x > 0 \quad \forall x \neq 0.$$

Remarque 2.3.1 Les éléments diagonaux a_{ii} d'une matrice définie positive sont tous strictement positifs.

Matrice semi-définie positive

Soit M une matrice carrée symétrique d'ordre n . Soit x un vecteur de \mathbb{R}^n , on note x^t sa transposée. Une matrice M est dite semi-définie positive si et seulement si

$$x^t M x \geq 0 \quad \forall x \in \mathbb{R}^n.$$

Remarque 2.3.2 Les éléments diagonaux a_{ii} d'une matrice semi-définie positive sont tous positifs.

2.4 Conditions d'optimalité

2.4.1 Conditions nécessaires d'optimalité du premier ordre

Théorème 2.4.1 *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable au point $\hat{x} \in \mathbb{R}^n$. Si \hat{x} est un minimum local de (P) alors $\nabla f(\hat{x}) = 0$.*

Preuve : C'est une conséquence directe du théorème 2.2.1 et de la relation (2.4). En effet, supposons que $\nabla f(\hat{x}) \neq 0$. Puisque la direction $d = -\nabla f(\hat{x})$ est une direction de descente, alors il existe $\delta > 0$ tel que

$$f(\hat{x} + \lambda d) < f(\hat{x}), \quad \forall \lambda \in]0, \delta[.$$

Ceci est contradiction avec le fait que \hat{x} est une solution optimale locale de (P) . ■

2.4.2 Conditions nécessaires d'optimalité du second ordre

Théorème 2.4.2 *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable au point $\hat{x} \in \mathbb{R}^n$. Si \hat{x} est un minimum local de (P) alors $\nabla f(\hat{x}) = 0$ et la matrice hessienne de f au point \hat{x} , qu'on note $H(\hat{x})$, est semi définie positive.*

Preuve : Soit $x \in \mathbb{R}^n$ quelconque, f est une fonction deux fois différentiable au point \hat{x} , donc on a pour tout $\lambda \neq 0$

$$f(\hat{x} + \lambda x) = f(\hat{x}) + \frac{1}{2} \lambda^2 x^t H(\hat{x}) x + \lambda^2 \|x\| \alpha(\hat{x}, \lambda x), \quad \alpha(\hat{x}, \lambda x) \xrightarrow{\lambda \rightarrow 0} 0.$$

Ceci implique

$$\frac{f(\hat{x} + \lambda x) - f(\hat{x})}{\lambda^2} = \frac{1}{2} x^t H(\hat{x}) x + \|x\| \alpha(\hat{x}, \lambda x), \quad (2.5)$$

\hat{x} est un optimum local, il existe alors $\delta > 0$ tel que

$$\frac{f(\hat{x} + \lambda x) - f(\hat{x})}{\lambda^2} \geq 0, \quad \forall \lambda \in]-\delta, +\delta[.$$

Si on prend en considération (2.5) et on passe à la limite quand $\lambda \rightarrow 0, \lambda \neq 0$, on obtient

$$x^t H(\hat{x})x \geq 0, \quad \forall x \in \mathbb{R}^n.$$

■

2.4.3 Conditions suffisantes d'optimalité

Théorème 2.4.3 *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable au point $\hat{x} \in \mathbb{R}^n$. Si $\nabla f(\hat{x}) = 0$ et $H(\hat{x})$ est définie positive alors \hat{x} est un minimum local de (P) .*

Preuve : f est deux fois différentiable au point \hat{x} , on aura pour tout $x \in \mathbb{R}^n$

$$f(x) = f(\hat{x}) + \frac{1}{2}(x - \hat{x})^t H(\hat{x})(x - \hat{x}) + \|(x - \hat{x})\|^2 \alpha(\hat{x}, (x - \hat{x})), \quad \alpha(\hat{x}, (x - \hat{x})) \xrightarrow{x \rightarrow \hat{x}} 0, \quad (\nabla f(\hat{x}) = 0). \quad (2.6)$$

Supposons que \hat{x} n'est pas un optimum local strict. Alors il existe une suite $\{x_k\}_{k \in \mathbb{N}}$ telle que

$$x_k \neq \hat{x}, \forall k \in \mathbb{N}, \quad x_k \xrightarrow{k \rightarrow \infty} \hat{x} \text{ et } f(x_k) \leq f(\hat{x}). \quad (2.7)$$

Dans (2.6) prenons $x = x_k$, divisons le tout par $\|(x_k - \hat{x})\|^2$ et notons $d_k = \frac{(x_k - \hat{x})}{\|(x_k - \hat{x})\|}$, on obtient

$$\frac{f(x_k) - f(\hat{x})}{\|(x_k - \hat{x})\|^2} = \frac{1}{2} d_k^t H(\hat{x}) d_k + \alpha(\hat{x}, (x_k - \hat{x})), \quad \alpha(\hat{x}, (x_k - \hat{x})) \xrightarrow{k \rightarrow \infty} 0. \quad (2.8)$$

(2.7) et (2.8) impliquent

$$\frac{1}{2} d_k^t H(\hat{x}) d_k + \alpha(\hat{x}, (x_k - \hat{x})) \leq 0, \quad \forall k \in \mathbb{N}.$$

D'autre part la suite $\{d_k\}_{k \in \mathbb{N}}$ est bornée ($\|d_k\| = 1, \forall k \in \mathbb{N}$).

Donc il existe une sous suite $\{d_k\}_{k \in N_1 \subset \mathbb{N}}$ telle que

$$d_k \xrightarrow[k \rightarrow \infty, k \in N_1]{} \tilde{d}.$$

Finalement lorsque $k \rightarrow \infty, k \in N_1$, on obtient

$$\frac{1}{2} \tilde{d} H(\hat{x}) \tilde{d} \leq 0.$$

La dernière relation et le fait que $\tilde{d} \neq 0$ ($\|\tilde{d}\| = 1$) impliquent que la matrice hessienne $H(\hat{x})$ n'est pas définie positive. Ceci est contradiction avec l'hypothèse. ■

Méthodes Quasi-Newton

3.1 Recherches linéaires D'Armijo et de Wolfe

Recherche linéaire

Considérons le problème d'optimisation sans contraintes

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

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

Les algorithmes qu'on étudie par la suite suivent les schémas généraux de la forme

$$x_{k+1} = x_k + \lambda_k d_k.$$

où λ_k est une solution optimale du problème d'optimisation unidimensionnel suivant

$$\min_{\lambda > 0} f(x_k + \lambda d_k),$$

c'est à dire que λ_k vérifie

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k), \forall \lambda > 0.$$

x_k, d_k sont fixes et la fonction à minimiser est une fonction d'une variable réelle définie comme suit

$$\lambda \mapsto h_k(\lambda) = f(x_k + \lambda d_k).$$

Il faut noter que dans le problème d'optimisation sans contraintes on a besoin de résoudre à chaque itération x_k , un problème d'optimisation dans \mathbb{R} .

Objectifs de la recherche linéaire

Il s'agit de réaliser deux objectifs

Le premier objectif

Consiste à faire décroître f suffisamment. Cela se traduit le plus souvent par la réalisation d'une inégalité de la forme

$$f(x_k + \lambda d_k) \leq f(x_k) + \text{un terme négatif} \quad (**)$$

Le terme négatif, disons ν_k , joue un rôle-clé dans la convergence de l'algorithme utilisant cette recherche linéaire.

L'argument est le suivant.

Si $f(x_k)$ est minorée (il existe une constante C telle que $f(x_k) \geq C$ pour tout k), alors ce terme négatif tend nécessairement vers zéro : $\nu_k \rightarrow 0$. C'est souvent à partir de la convergence vers zéro de cette suite que l'on parvient à montrer que le gradient lui-même doit tendre vers zéro. Le terme négatif devra prendre une forme bien particulière si on veut pouvoir en tirer de l'information. En particulier, il ne suffit pas d'imposer $f(x_k + \lambda d_k) \leq f(x_k)$.

Le second objectif

Consiste d'empêcher le pas $\lambda_k > 0$ d'être trop petit, trop proche de zéro.

Le premier objectif n'est en effet pas suffisant car l'inégalité (***) est en général satisfaite par du pas $\lambda_k > 0$ arbitrairement petit.

Or ceci peut entraîner une "fausse convergence", c'est-à-dire la convergence des itérés vers un

point non stationnaire, comme le montre l'observation suivante. Si on prend

$$0 < \lambda_k \leq \frac{\varepsilon}{2^k \|d_k\|}.$$

la suite $\{x_k\}$ est de Cauchy, puisque pour $1 < l < k$ on a

$$\|x_k - x_l\| = \left\| \sum_{i=1}^{i=k-1} \lambda_i d_i \right\| \leq \sum_{i=1}^{i=k-1} \frac{\varepsilon}{2^i} \rightarrow 0, \quad l \rightarrow \infty$$

Donc $\{x_k\}$ converge, disons vers un point x^* . En prenant $l = 1$ et $k \rightarrow \infty$ dans l'estimation ci-dessus, on voit que $x^* \in \bar{B}(x_1, \varepsilon)$ et x^* ne soit pas une solution s'il n'y a pas de solution dans $\bar{B}(x_1, \varepsilon)$.

On a donc arbitrairement forcer la convergence de $\{x_k\}$ en prenant des pas très petits.

Pour simplifier les notations, on définit $h_k : \mathbb{R}_+ \rightarrow \mathbb{R}$ comme suit

$$\lambda \mapsto h_k(\lambda) = f(x_k + \lambda d_k)$$

But de la recherche linéaire

Dans le cas non-quadratique les méthodes de descente, nécessitent la recherche d'une valeur de $\lambda_k > 0$ optimale ou non, vérifiant

$$f(x_k + \lambda_k d_k) \leq f(x_k).$$

Rappelons que si f est différentiable, le pas optimal λ^* peut être caractérisé par

$$h^*(\lambda^*) = 0,$$

$$h(\lambda^*) \leq h(\lambda), \text{ pour } 0 < \lambda \leq \lambda^*.$$

autrement dit, λ^* est un minimum local de h qui assure de plus la décroissance de f .

En fait, dans la plupart des algorithmes d'optimisation modernes, on ne fait jamais de recherche linéaire exacte, car trouver λ^* signifie qu'il va falloir calculer un grand nombre de fois

la fonction f et cela peut être dissuasif du point de vue du temps de calcul. En pratique, on recherche plutôt une valeur de λ qui assure une décroissance suffisante de f . Cela conduit à la notion d'intervalle de sécurité.

Dans ce qui suit nous introduisons deux types de recherches linéaires.

3.1.1 Recherche linéaire d'Armijo

Définition 3.1.1 (condition d'Armijo)

Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable, $x_k \in \mathbb{R}^n$, une direction (de descente) $d_k \in \mathbb{R}^n$ telle que $\nabla f(x_k)^T d_k < 0$. On dit que le pas $\alpha_k > 0$ vérifie la condition d'Armijo, si on a

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \beta_1 \nabla f(x_k)^T d_k, \quad 0 < \beta_1 < 1. \quad (\text{Armijo})$$

Algorithme (règle d'Armijo)

Étape 0 (initialisation)

$\alpha_{g,1} = \alpha_{d,1} = 0$, choisir $\alpha_1 > 0, \rho \in]0, 1[$ poser $k = 1$ et aller à l'étape 1.

Étape 1

si $\varphi_k(\alpha_k) \leq \varphi_k(0) + \rho \varphi'_k(0) \alpha_k$: STOP ($\alpha^* = \alpha_k$)

si $\varphi_k(\alpha_k) > \varphi_k(0) + \rho \varphi'_k(0) \alpha_k$, alors

$\alpha_{d,k+1} = \alpha_d, \alpha_{g,k+1} = \alpha_k$ et aller à l'étape 2.

Étape 2

si $\alpha_{d,k+1} = 0$ déterminer $\alpha_{k+1} \in]\alpha_{g,k+1}, +\infty[$

si $\alpha_{d,k+1} \neq 0$ déterminer $\alpha_{k+1} \in]\alpha_{g,k+1}, \alpha_{d,k+1} |$

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

3.1.2 Recherche linéaire de Wolfe

α_k est acceptable par la recherche linéaire inexacte de Wolfe si α_k vérifie les deux conditions (Wolfe1) et (Wolfe2) suivantes

$$\varphi_k(\alpha_k) \leq \varphi_k(0) + \rho\alpha_k\varphi'_k(0), \quad 0 < \rho < \frac{1}{2}. \quad (\text{Wolfe 1})$$

$$\varphi'_k(\alpha_k) \geq \sigma\varphi'_k(0), \quad 0 < \sigma < 1, \quad \sigma > \rho. \quad (\text{Wolfe 2})$$

ou en remplaçant $\varphi_k(\alpha_k)$, $\varphi_k(0)$, $\varphi'_k(0)$, $\varphi'_k(\alpha_k)$ par leurs expressions, on a

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k \nabla f(x_k)^T d_k, \quad 0 < \rho < \frac{1}{2}. \quad (\text{Wolfe 3})$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma \nabla f(x_k)^T d_k, \quad 0 < \sigma < 1, \quad \sigma > \rho. \quad (\text{Wolfe 4})$$

3.2 Méthode de Newton

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$. On supposera que $f \in C^2(\mathbb{R}^n)$.

Soit maintenant (P) le problème d'optimisation sans contraintes que nous avons déjà rencontré dans le chapitre 2

$$\text{Minimiser } \{f(x) : x \in \mathbb{R}^n\}$$

La méthode de Newton est attribuée au mathématicien, physicien et astronome anglais Isaac Newton (1642 – 1727).

Le principe de la méthode de Newton consiste à minimiser successivement les approximations du second-ordre d'une fonction f , plus précisément si

$$f(x) \simeq f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^t H(x_k)(x - x_k) = q(x), \quad \forall x \in V(x_k).$$

$H(x_k)$ étant la matrice hessienne de f au point x_k , alors une condition nécessaire de minimum pour q est que $\nabla q(x) = 0$, ou

$$\nabla f(x_k) + H(x_k)(x - x_k) = 0. \quad (3.1)$$

Supposons que H est inversible, alors le successeur x_{k+1} de x_k est donné par

$$x_{k+1} = x_k - H^{-1}(x_k) \nabla f(x_k). \quad (3.2)$$

Cette équation donne la forme générale des points générés par l'algorithme de Newton.

Supposons que $\nabla f(\bar{x}) = 0$ et que $H(\bar{x})$ est définie positive (\bar{x} un minimum local), alors $H(x_k)$ reste définie positive en tout point voisin de \bar{x} . Ceci assure que le successeur x_k de x_k est bien défini.

3.2.1 Algorithme

Etape initiale

Soit ε un paramètre qui détermine le critère d'arrêt. Choisir un point initial. Poser $k = 1$ et aller à l'étape principale.

Etape principale

Si $\|\nabla f(x_k)\| < \varepsilon$, stop ; sinon, poser $x_{k+1} = x_k - H^{-1}(x_k) \nabla f(x_k)$, remplacer k par $k + 1$ et répéter l'étape principale.

Exemple 3.2.1 *Nous allons appliquer maintenant cet algorithme sur un problème test très classique du à Rosenbrock. La fonction de rosenbrock à minimiser dans \mathbb{R}^n est la suivante*

$$f(x_1, x_2) = (x_1 - 1)^2 + p(x_1^2 - x_2)^2.$$

où p est un paramètre positif, ici égal à 100. Dans les lignes de niveau de cette fonction il y'a une vallée très étroite, en forme de banane, d'où le surnom "Rosenbrock banana", conduisant au minimum global $x^* = (1, 1)$, avec $f(x^*) = 0$. Donc seule une méthode efficace permettra de

trouver ce minimum ([4]).

Les résultats obtenus par un programme en fortran 90 sont illustrés dans le tableau 3.1, le point initial est $(-1.2, 1)$ et $\varepsilon = 10^{-5}$.

tableau 3.2.1

k	$f(x)$	$\ \nabla f(x)\ $
1	0.242000E + 02	0.232868E + 03
2	0.473188E + 01	0.463943E + 01
3	0.141185E + 04	0.137079E + 04
4	0.559655E - 01	0.473110E + 00
5	0.313189E + 00	0.250274E + 02
6	0.185274E - 10	0.860863E - 05

Pour atteindre la précision ε donnée, il nous a fallu 6 itérations.

la solution approchée est $x_6 = (x_1, x_2)$.

$$x_1 = 9.999956956536786E - 1, \quad x_2 = 9.999913913257368E - 1.$$

$$\text{avec } f(x_6) = f(x_1, x_2) = 1.852739725430225E - 11.$$

3.2.2 Etude de la convergence

En général la méthode de Newton ne converge pas à cause de la possibilité que $H(x_k)$ soit singulière et donc que le point x_{k+1} ne soit pas bien défini. Et même si $H^{-1}(x_k)$ existe, la décroissance de $f(x)$ n'est pas toujours assurée. Cependant ces problèmes peuvent être évités si le point initial est assez voisin du point \bar{x} , tel que $\nabla f(\bar{x}) = 0$ et $H^{-1}(\bar{x})$ existe. Dans ce cas la méthode de Newton est bien définie et elle converge vers le point \bar{x} . C'est ce que nous prouverons dans ce théorème.

Théorème 3.2.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ trois fois continuellement différentiable. Considérons l'algorithme de Newton défini par l'application $A(x) = x - H^{-1}(x)\nabla f(x)$. Soit \bar{x} tel que $\nabla f(\bar{x}) = 0$ et supposons que $H(\bar{x})^{-1}$ existe. Soit x_1 un point initial assez proche de \bar{x} de sorte que cette proximité implique l'existence de $k_1, k_2 > 0$, avec $k_1 k_2 \|x_1 - \bar{x}\| < 1$ et vérifiant les deux propriétés suivantes

1. $\|H(x)^{-1}\| \leq k_1$.
2. $\|\nabla f(\bar{x}) - \nabla f(x) - H(x)(x - \bar{x})\| \leq k_2 \|x - \bar{x}\|^2$.

pour tout x satisfaisant

$$\|x - \bar{x}\| \leq \|x_1 - \bar{x}\|.$$

Alors, l'algorithme converge de façon superlinéaire vers \bar{x} , avec une vitesse de convergence quadratique.

Preuve : Soit l'ensemble de solutions $\Omega = \{\bar{x}\}$ et l'ensemble

$$X = \{x : \|x - \bar{x}\| \leq \|x_1 - \bar{x}\|\}.$$

Pour prouver la convergence, on utilise le corollaire 1 du théorème 5. Notons que X est compact et que l'application A définie par

$$y \in A(x) \Leftrightarrow y = x - H^{-1}(x)\nabla f(x),$$

est fermée sur X .

Montrons maintenant que $\alpha(x) = \|x - \bar{x}\|$ est une fonction de descente. Soit $x \in X$, et supposons que $x \neq \bar{x}$. Soit $y \in A(x)$, donc

$$\begin{aligned} y - \bar{x} &= (x - \bar{x}) - H^{-1}(x)[\nabla f(x) - \nabla f(\bar{x})] \\ &= H^{-1}(x)[\nabla f(\bar{x}) - \nabla f(x) - H(x)(\bar{x} - x)]. \end{aligned}$$

D'après (1), (2), on a

$$\begin{aligned}
 \|y - \bar{x}\| &= \|H^{-1}(x)[\nabla f(\bar{x}) - \nabla f(x) - H(x)(\bar{x} - x)]\| \\
 &\leq \|H^{-1}(x)\| \|\nabla f(\bar{x}) - \nabla f(x) - H(x)(\bar{x} - x)\| \\
 &\leq \|H^{-1}(x)\| \|\nabla f(\bar{x}) - \nabla f(x) - H(x)(\bar{x} - x)\| \\
 &< \|x - \bar{x}\|.
 \end{aligned}$$

Donc on a prouvé que $\alpha(y) < \alpha(x)$, $\forall y \in A(x)$ et par suite α est une fonction de descente, on conclut que l'algorithme converge vers \bar{x} . De plus, pour tout $x_k \in X$, son successeur x_{k+1} satisfait

$$\begin{aligned}
 \|x_{k+1} - \bar{x}\| \leq k_1 k_2 \|x - \bar{x}\|^2 &\Rightarrow \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|^2} \leq k_1 k_2 \\
 &\Rightarrow \limsup_{k \rightarrow +\infty} \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|^2} \leq k_1 k_2.
 \end{aligned}$$

puisque $\{x_k\} \rightarrow \bar{x}$, donc la vitesse de convergence est quadratique. ■

3.2.3 Avantages et Inconvénients de la méthode de Newton

Avantages

Comme nous l'avons vu au théorème précédent, la méthode de Newton bénéficie d'une convergence quadratique, et c'est son principal avantage.

Inconvénients

1. Cette méthode fonctionne très bien pour les problèmes de petite dimension, lorsqu'on peut calculer facilement la matrice hessienne et son inverse. Cette procédure nécessite des itérations plus nombreuses et coûteuses dans les problèmes de grand taille.

2. Comme

$$x_{k+1} = x_k - H^{-1}(x_k) \nabla f(x_k),$$

on voit que le successeur x_{k+1} de x_k n'est pas toujours bien défini.

3. Même si $H^{-1}(x_k)$ est inversible, la direction $d_k = -H^{-1}(x_k) \nabla f(x_k)$ n'est pas toujours une

direction de descente. (Si $H(x_k)$ est définie positive, alors d_k est une direction de descente).

Conclusion : Ceci nous ramène à conclure que la méthode de Newton ne génère pas en général une suite qui converge vers le minimum de la fonction. Mais sous certaines conditions elle devient très intéressante (Hessien défini positif, point initial assez proche de la solution optimale,...).

Exemple 3.2.2

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 101[(x_2 - 1)^2 + (x_4 - 1)^2] \\ + 198(x_2 - 1)(x_4 - 1)$$

cette fonction est la fonction de Wood, elle admet le point $(1, 1, 1, 1)$ comme un minimum et le point $(-1, 1, -1, 1)$ comme un point selle.

Nous allons appliquer la méthode de Newton pour minimiser cette fonction en prenant $x_0 = (-3, -1, -3, -1)$ et $\varepsilon = 10^{-4}$, que la méthode de Newton à converge vers le point $(-1, 1, -1, 1)$ qui est le point selle de cette fonction.

le nombre d'itérations est $s = 15$

la solution est $x = (-9.679741\text{E} - 01, 9.471393\text{E} - 01, -9.695163\text{E} - 01, 9.512478\text{E} - 01)$

la valeur de $f(x)$ est 7.876967.

tableau 3.2.2

itération	$f(x)$	$\ \nabla f(x)\ $
1	1291.438	16397.13
2	295.9513	1679.932
3	67.68565	1003.095
4	17.33662	285.0620
5	8.689081	106.9116
6	7.892798	26.45835
7	7.876516	3.768905
8	7.877190	0.150338
9	7.876882	0.650068
10	7.876977	0.198463
11	7.876966	0.186190
12	7.876967	0.015349
13	7.876967	0.007221
14	7.876967	0.000112

A cause de ces inconvénients, il existe des modifications qui permettent d'assurer la convergence globale de la méthode de Newton, alliant les avantages de la vitesse de convergence quadratique près de la solution, et ceux d'une méthode de descente. Parmi ces méthodes, on notera par exemple les méthodes dites quasi-Newton discutées plus tard.

3.3 Méthodes Quasi-Newton

Ce chapitre est consacré aux méthodes dites Quasi-Newton pour la résolution des problèmes d'optimisation non linéaires sans contraintes. Parmi ces méthodes, on s'étalera particulièrement sur les trois plus importantes, la méthode de correction de rang un, la méthode *DFP* (Davidon, Fletcher, Powell) et la méthode *BFGS* (Broyden, Fletcher, Goldfarb, Shano).

3.3.1 Dérivation des méthodes

Ces méthodes s'inspirent de l'algorithme de Newton, mais sans calculer la matrice hessienne de f , ni son inverse. L'itération de la méthode de Newton étant définie par

$$x_{k+1} = x_k - H^{-1}(x_k) \nabla f(x_k),$$

l'idée est de remplacer cette itération par

$$x_{k+1} = x_k - \lambda_k S_k \nabla f(x_k),$$

où λ_k est un paramètre fourni par une recherche linéaire le long de la direction $d_k = -S_k \nabla f(x_k)$, S_k est une approximation symétrique, définie positive, de $H^{-1}(x_k)$. Bien sur, plus S_k sera proche de $H^{-1}(x_k)$, plus l'algorithme convergera rapidement. L'objectif, est donc de trouver une bonne suite de matrices S_k , faciles à construire, c'est à dire utilisant des informations seulement sur le gradient et qui converge vite vers des approximations de plus en plus précises, de l'inverse du hessien de f .

Pour réaliser cela, prenons $f \in C^2(\mathbb{R}^n)$, et faisons un développement de $\nabla f(x)$ au voisinage de x_k

$$\nabla f(x) = \nabla f(x_k) + H(x_k)(x - x_k) + o(\|x - x_k\|),$$

$$\nabla f(x) \simeq \nabla f(x_k) + H(x_k)(x - x_k), \quad x \in V(x_k).$$

ou encore

$$H^{-1}(x_k) [\nabla f(x) - \nabla f(x_k)] \simeq (x - x_k).$$

Les approximations sont exactes si f est quadratique. En particulier, avec $x = x_{k+1}$ et si S_k était une bonne approximation de $H^{-1}(x_k)$, on devrait avoir

$$S_k [\nabla f(x_{k+1}) - \nabla f(x_k)] \simeq (x_{k+1} - x_k). \quad (3.3)$$

mais comme x_{k+1} est calculé après S_k , il est peu probable que cette équation soit satisfaite, même approximativement. En revanche, on peut toujours imposer que S_{k+1} satisfasse cette équation exactement, d'où

$$S_{k+1} [\nabla f(x_{k+1}) - \nabla f(x_k)] = (x_{k+1} - x_k). \quad (3.4)$$

cette équation est dite "équation de la sécante" ou "condition quasi Newton".

À l'étape k , la mise à jour de la matrice S_k se fait avec une formule simple,

$$S_{k+1} = S_k + C_k,$$

C_k étant une matrice de correction qui intègre au mieux la nouvelle information fournie par x_{k+1} et $\nabla f(x_{k+1})$ de telle manière que S_{k+1} satisfasse la condition (3.4). Basées sur ce principe, les méthodes quasi-Newton diffèrent l'une par rapport à l'autre, d'après la définition de la matrice C_k .

Commençons par la méthode de correction de rang un qui est considérée comme une introduction élémentaire aux deux autres méthodes (DFP et *BFGS*) décrites par la suite.

3.3.2 Méthode de correction de rang un

Étant donné que $H^{-1}(x_k)$ est symétrique, il est naturel de construire des approximations successives S_k symétriques. Par exemple, on peut exprimer S_{k+1} en fonction de S_k de façon très simple, en rajoutant à cette dernière une matrice de rang un de la forme suivante $a_k u_k u_k^t$, où u_k est un vecteur de \mathbb{R}^n et a_k est une constante. On obtiendra alors

$$S_{k+1} = S_k + a_k u_k u_k^t.$$

Montrons qu'il est facile de calculer a_k et u_k de telle sorte que la condition (3.4) soit satisfaite.

Posons

$$\begin{aligned} p_k &= x_{k+1} - x_k \\ q_k &= \nabla f(x_{k+1}) - \nabla f(x_k), \end{aligned}$$

la condition (3.4) s'écrit donc

$$S_{k+1}q_k = p_k,$$

Soit en remplaçant par l'expression de S_{k+1} ,

$$(S_k + a_k u_k u_k^t) q_k = p_k,$$

ou encore

$$a_k u_k (u_k^t q_k) = p_k - S_k q_k.$$

d'où l'on déduit que u_k est proportionnel à $p_k - S_k q_k$ avec un facteur qui peut être pris en compte dans a_k . En particulier en prenant $u_k = p_k - S_k q_k$ et a_k tel que $a_{ij} (u_k^t q_k) = 1$, on obtient

$$S_{k+1} = S_k + \frac{(p_k - S_k q_k) (p_k - S_k q_k)^t}{(p_k - S_k q_k)^t q_k}.$$

Algorithme

Etape initiale

Soit $\varepsilon > 0$, déterminant le critère d'arrêt. Choisir un point initial x_1 et une matrice symétrique définie positive S_1 . Poser $y_1 = x_1, k = 1$ et aller aux étapes principales.

Etapes principales

Etape1 Si $\|\nabla f(y_k)\| < \varepsilon$. stop; sinon, poser $d_k = -S_k \nabla f(x_k)$ et soit λ_k la solution optimale du problème $\min f(y_k + \lambda d_k), \lambda \geq 0$.

Etape2 Construire S_{k+1} comme suit

$$S_{k+1} = S_k + \frac{(p_k - S_k q_k) (p_k - S_k q_k)^t}{(p_k - S_k q_k)^t q_k},$$

avec

$$p_k = \lambda_k d_k \equiv y_{k+1} - y_k$$

$$q_k = \nabla f(y_{k+1}) - \nabla f(y_k),$$

remplacer k par $k + 1$, et répéter l'étape 1.

Il est utile de citer le théorème suivant qui montre que lorsque H est constante, la condition (3.4) est non seulement vérifiée pour $i = k$, mais aussi pour tout $i < k$, et que dans ce cas la convergence est obtenue dans au plus n étapes.

Théorème 3.3.1 *Si f est quadratique, de matrice hessienne H définie positive et si p_1, \dots, p_n sont des vecteurs indépendants, alors la méthode de correction de rang un converge au plus dans $(n + 1)$ itérations et $(S_{n+1})^{-1} = H$.*

Preuve : La preuve se fait par récurrence. Montrons tout d'abord que

$$p_i = S_{k+1} q_i, \quad \text{pour } i \leq k. \quad (3.5)$$

Pour $k = 1$ elle est évidente d'après la définition de S_2 . Supposons qu'elle est vraie pour $k > 1$ et montrons qu'elle vraie pour $k + 1$. On a donc

$$S_k q_i = p_i, \quad \forall i \leq k - 1.$$

Pour exploiter la définition de S_{k+1} , calculons

$$(p_k - S_k q_k)^t q_i = p_k^t q_i - q_k^t S_k q_i = p_k^t q_i - q_k^t p_i. \quad (3.6)$$

En remarquant que $q_k = H p_k$, on obtient

$$(p_k - S_k q_k)^t q_i = p_k^t q_i - p_k^t H p_i = p_k^t q_i - p_k^t q_i = 0. \quad (3.7)$$

Donc

$$S_{k+1} q_i = S_k q_i + 0 = p_i, \quad \forall i \leq k - 1.$$

Reste le cas $i = k$. La relation est vraie d'après la condition (3.4). Ce qui établit le résultat voulu au rang $k + 1$. Pour terminer la preuve du théorème et puisque on a

$$p_i = S_{n+1} q_i = S_{n-1} H p_i, \quad i = 1, \dots, n \quad (3.8)$$

et comme les vecteurs p_i sont indépendants, alors

$$S_{n+1} H = I \text{ ou encore, } S_{n+1} = H^{-1}.$$

■

Exemple 3.3.1 *Considérons le problème suivant*

$$(P1) \begin{cases} \min \frac{1}{2} x^t Q x \\ x \in \mathbb{R}^n \end{cases}$$

où

$$Q = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 3 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix},$$

Appliquons la méthode de correction de rang un sur cet exemple, en prenant $\varepsilon = 10^{-5}$ et $y_0 = (0, 1, 2, 3)$. Les résultats obtenues sont illustrés au tableau 3.3.1, et la recherche linéaire considérée est exacte.

tableau 3.3.1

itération	x_k	$f(x_k)$	$\ \nabla f(x_k)\ $	λ_k
1	(0.000, 1.000, 2.000, 3.000)	15.000	13.000	0.250
2	(-2.005, -1.005, 0.746, 1.997)	6.187	4.357	0.842
3	(0.665, -0.548, 0.004, 0.113)	0.766	0.196	0.444
4	(0.000, -0.001, 0.004, -0.002)	0.002	0.087	0.003
5	(0.000, 0.000, 0.000, 0.000)	0.000	0.000	—

D'après le tableau, on voit bien que le minimum de f est atteint en 4 itérations.

Avantages et Inconvénients

Avantages

Cette méthode présente l'avantage que le point x_{k+1} n'a pas besoin d'être choisi comme le minimum exact, c'est à dire qu'on n'a pas besoin d'effectuer des recherches linéaires exactes.

Inconvénients

Les inconvénients de cette méthode sont

1. Même si la fonction est quadratique, et même si son hessien est défini positif, il se peut que la matrice S_k ne soit pas définie positive.

2. Le dénominateur $(p_k - S_k q_k)^t p_k$ peut très bien devenir nul ou très petit, rendant le procédé instable.

Dans ce qui suit, on présente les méthodes quasi-Newton, basées sur des formules de correction de rang 2 (DFP, BFGS), qui évitent ces inconvénients.

3.3.3 Méthode de Davidon-Fletcher-Powell(D.F.P)

Cette méthode a été proposée par Davidon en 1959 ([1]) et développée plus tard en 1963 par Fletcher & Powell ([2]). C'est une méthode assez remarquable pour plusieurs raisons. Tout d'abord, quand elle est appliquée à une fonction quadratique, non seulement les différentes itérations aboutissent à l'inverse du hessien, mais en plus elle engendrent des directions conjuguées. La mise à jour de S_{k+1} en fonction de S_k se fait en ajoutant cette fois deux matrices de rang un, d'où le nom "correction de rang deux". De façon plus précise construisons S_{k+1} en fonction de S_k , de la forme

$$S_{k+1} = S_k + A_k + B_k,$$

avec A_k et B_k deux matrices de rang un chacune et de la forme suivante

$$A_k = a_k u_k u_k^t, \quad B_k = b_k v_k v_k^t,$$

a_k, b_k sont des constantes, u_k, v_k sont deux vecteurs de \mathbb{R}^n . S_{k+1} doit satisfaire la condition (3.4), c'est à dire

$$(S_k + a_k u_k u_k^t + b_k v_k v_k^t) q_k = p_k$$

$$S_k q_k + a_k u_k u_k^t q_k + b_k v_k v_k^t q_k = p_k$$

$$a_k u_k u_k^t q_k + b_k v_k v_k^t q_k = p_k - S_k q_k.$$

un choix évident pour satisfaire l'équation est de prendre

$$u_k = p_k, \quad v_k = S_k q_k, \quad a_k u_k^t q_k = 1 \text{ et } b_k v_k^t q_k = -1, \text{ d'où}$$

$$S_{k+1}^{DFP} = S_k + \frac{p_k p_k^t}{p_k q_k^t} - \frac{S_k q_k q_k^t S_k}{q_k^t S_k q_k}. \quad (3.9)$$

Remarque 3.3.1 *Il y'a aussi une formule DFP, pour approcher le hessien lui même, au lieu de son inverse. Cette formule est donnée par*

$$B_{k+1}^{DFP} = B_k + \left[1 + \frac{p_k^t B_k p_k}{p_k^t q_k} \right] \frac{q_k q_k^t}{p_k^t q_k} - \left[\frac{q_k p_k^t B_k + B_k p_k q_k^t}{p_k^t q_k} \right].$$

La procédure est la même que celle de correction de rang un, mais en général il est recommandable d'implémenter les méthodes quasi-Newton avec une réinitialisation périodique pour des raisons de convergence. L'algorithme suivant est donné sous cette forme.

Algorithme DFP

Etape initiale

Soit ε un scalaire d'arrêt, choisir un point initial x_1 et une matrice symétrique définie positive S_1 . Poser $y_1 = x_1, k = j = 1$, et aller aux étapes principales.

Etapes principales

Etape1

Si $\|\nabla f(y_j)\| < \varepsilon$. stop ; sinon, poser $d_j = -S_j \nabla f(y_j)$ et soit λ_j la solution optimale du problème

$$\min f(y_j + \lambda d_j) \text{ telque } \lambda \geq 0. \text{ Soit } y_{j+1} = y_j + \lambda_j d_j.$$

Si $j < n$ aller à l'étape 2)

Si $j = n$, poser $y_1 = x_{k+1} = y_{n+1}$, remplacer k par $k + 1$, poser $j = 1$ et aller à l'étape 1)

Etape2

Construire S_{j+1} comme suit

$$S_{j+1} = S_j + \frac{p_j p_j^t}{p_j q_j^t} - \frac{S_j q_j q_j^t S_j}{q_j^t S_j q_j}.$$

avec

$$p_j = \lambda_j d_j \equiv y_{j+1} - y_j$$

$$q_j = \nabla f(y_{j+1}) - \nabla f(y_j)$$

remplacer j par $j + 1$, et répéter l'étape 1

Définition 3.3.1 Soit H une matrice $(n \times n)$ symétrique. Les vecteurs d_1, \dots, d_n sont dits H -conjugués s'ils sont linéairement indépendants et si $d_i^t H d_j = 0$ pour $i \neq j$.

Théorème 3.3.2 Considérons la méthode DFP décrite par la relation

$$S_{k+1}^{DFP} = S_k + \frac{p_k p_k^t}{p_k^t q_k^t} - \frac{S_k q_k q_k^t S_k}{q_k^t S_k q_k}$$

1. A l'étape k , si S_k est symétrique définie positive et si la recherche linéaire est exacte (ou bien si $p_k^t q_k^t > 0$), alors la matrice S_{k+1} est symétrique définie positive aussi.

2. Si f est quadratique de hessien H , avec en plus H définie positive, alors la méthode DFP vérifie

i) $p_i^t H p_j = 0 \quad 1 \leq i < j \leq k.$

ii) $S_{k+1} H p_i = p_i \quad 1 \leq i \leq k.$

Théorème 3.3.3 Ces propriétés impliquent, dans le cas quadratique, la convergence de la méthode en n itérations et la propriété $S_{n+1} = H^{-1}$.

Preuve : La méthode engendre des directions conjuguées, d'après le point 2. De plus, les directions p_1, \dots, p_k sont des vecteurs propres (associés à la valeur propre 1) de la matrice $S_{n+1} H$. Elles forment une famille libre, puisque ces vecteurs sont des directions conjuguées.

Notons P la matrice telle que ses colonnes sont ces directions, alors P est inversible et d'après ii)

$$S_{n+1} H P = P \Rightarrow S_{n+1} H = I \Rightarrow S_{n+1} = H^{-1}.$$

■

Exemple 3.3.2 *Considérons le même problème que l'exemple 3.3.1 et appliquant la méthode DFP, on trouve les résultats suivants*

tableau 3.3.2

itération	x_k	$f(x_k)$	$\ \nabla f(x_k)\ $	λ_k
1	(0.000, 1.000, 2.000, 3.000)	15.000	13.000	0.250
2	(-2.005, -1.005, 0.746, 1.997)	6.187	4.357	0.812
3	(0.665, -0.548, 0.004, 0.113)	0.766	1.969	0.483
4	(0.000, -0.001, 0.004, -0.002)	0.002	0.087	0.602
5	(0.000, 0.000, 0.000, 0.000)	0.000	0.000	—

(3.10)

Les matrices S_k obtenues à chaque itération sont :

$$S_1 = \begin{pmatrix} 1.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 0.947 & -0.136 & 0.117 & 0.134 \\ -0.140 & 0.730 & 0.227 & 0.261 \\ -0.123 & 0.227 & 0.811 & -0.213 \\ -0.149 & -0.261 & -0.213 & 0.761 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 0.176 & -0.495 & 0.129 & 0.567 \\ -0.453 & 0.515 & -0.101 & 0.112 \\ 0.129 & -0.101 & 0.722 & -0.469 \\ 0.567 & 0.112 & -0.469 & 0.002 \end{pmatrix}, \quad S_4 = \begin{pmatrix} -0.129 & -0.213 & 0.160 & 0.485 \\ -0.213 & 0.328 & -0.126 & 0.176 \\ 0.160 & -0.126 & 0.720 & -0.462 \\ 0.485 & 0.176 & -0.462 & 0.002 \end{pmatrix}$$

Avantages et Inconvénients

Avantages

1. Pour des fonctions quadratique (avec une recherche linéaire exacte)

i) elle termine au plus en n étapes avec $S_{n+1} = H^{-1}$,

ii) elle engendre des directions conjuguées.

2. Pour des fonctions quelconques

iii) les matrices S_K restent définies positives. Ceci entraîne que les propriétés de descente de la fonction sont vérifiées,

iv) une convergence superlinéaire.

Inconvénients

Cette méthode est assez sensible à la précision de la recherche linéaire (la convergence dépend du fait que la recherche linéaire est exacte ou non).

3.3.4 Méthode BFGS

Cette méthode a été développée de façon indépendante par Broyden ([6],1970), Fletcher ([9],1970), Goldfarb ([7],1970) et Shanno ([8],1969). Pour construire une approximation de l'inverse du hessien, elle utilise une formule de correction de rang deux qui dérive directement de la formule (3.9) .

Plus précisément

$$S_{k+1} = S_k + \frac{p_k p_k^t}{p_k^t q_k} + \frac{q_k^t S_k q_k p_k p_k^t}{(p_k q_k^t)^2} - \frac{p_k q_k^t S_k + S_k q_k p_k^t}{p_k^t q_k}. \quad (3.11)$$

Cette formule est moins sensible aux erreurs dans les recherches linéaire, et elle est considérée comme la meilleure actuellement. On peut se demander ce qui justifie l'introduction d'une telle formule. Une des réponses est que si l'on note $B_k = (S_k)^{-1}$, c.à.d, B_k approxime H_k et non pas son inverse, la condition quasi-Newton (voir formule (3.4)) sera dans ce cas

$$B_{k+1} p_k = q_k, \quad (3.12)$$

et

$$B_{k+1} = B_k + \frac{q_k q_k^t}{q_k^t p_k} - \frac{B_k p_k p_k^t B_k}{p_k^t B_k p_k}. \quad (3.13)$$

Comme l'a remarqué Fletcher ([9]), si l'on interverti les rôles de p_k et de q_k dans (3.9), on obtient les matrices vérifiant (3.12), qui est la relation inverse de (3.11). On voit alors que la formule (3.13) permet de construire une approximation de la matrice hessienne elle même (et

non pas son inverse). Posons

$$C_k = \frac{q_k q_k^t}{q_k^t p_k} - \frac{B_k p_k p_k^t B}{p_k^t B p_k},$$

d'après la relation

$$S_{k+1} = S_k + C_k = B_{k+1}^{-1} = (B_k + C_k)^{-1}, \quad (3.14)$$

et par application de la formule de Sherman - Morrison - woodbury suivante

$$(A + a^t b)^{-1} = A^{-1} - \frac{A^{-1} a b^t A^{-1}}{1 + b^t A^{-1} a}, \quad (3.15)$$

où A est une matrice ($n \times n$), b est un vecteur de \mathbb{R}^n , en supposant que

$$(1 + b^t A^{-1} a) \neq 0,$$

on obtient

$$[B_{k+1}]^{-1} = [B_k]^{-1} + \left[1 + \frac{q_k^t [B_k]^{-1} q_k}{p_k^t q_k} \right] \frac{p_k p_k^t}{p_k^t q_k} - \frac{p_k q_k^t [B_k]^{-1} + [B_k]^{-1} q_k p_k^t}{p_k^t q_k}.$$

Ce qui montre que $[B_{k+1}]^{-1}$ est exprimé directement en fonction de $[B_k]^{-1}$ et d'où la formule de correction de rang (3.11).

Remarques importantes

1. la formule (3.11) a des propriétés analogues à celles de la formule (3.9). En particulier

i) Si $p_k^t q_k > 0$, alors la définie positivité des matrices S_k est préservée.

ii) Appliquée à une fonction quadratique (matrice hessienne H définie positive), cette formule permet d'obtenir en au plus n itérations, l'inverse H^{-1} du hessien de f . De plus les directions p_k engendrées par l'algorithme BFGS sont mutuellement conjuguées par rapport à H .

2. Lorsque l'algorithme BFGS ou DFP est appliqué à une fonction non linéaire quelconque (non quadratique), il faut procéder à des réinitialisations périodiques, pour assurer la convergence globale.

3. Théoriquement, la condition $p_k^t q_k > 0$ assure la définie positivité des matrices S_{k+1} ou B_{k+1} , mais sur le plan pratique rien n'est garanti, a cause des erreurs d'arrondi, S_{k+1} ou B_{k+1} peuvent devenir singulières ou non définies.

4. La majorité des auteurs affirme la supériorité de l'algorithme BFGS sur celui de DFP, soit sur le plan théorique ou pratique (il est plus stable).

Exemple 3.3.3 *Si on considère le même exemple traité par DFP et la méthode de correction de rang un pour illustrer la remarque importante ii), on trouve les résultats si dessous*

tableau 3.3.3

itération	x_k	$f(x_k)$	$\ \nabla f(x_k)\ $	λ_k
1	(0.000, 1.000, 2.000, 3.000)	15.000	13.000	0.250
2	(2 : 005; 1 : 005; 0 : 746; 1 : 997)	6.187	4.357	0.732
3	(0.665, -0.548, 0.004, 0.113)	0.766	0.196	0.393
4	(0.000, -0.001, 0.004, -0.002)	0.002	0.087	0.601
5	(0.000, 0.000, 0.000, 0.000)	0.000	0.000	—

3.4 Convergence des méthode Quasi-Newton

Cas où la recherche linéaire est inexacte

Dans ce cas, le théorème (3.3.2), de convergence finie, n'est plus valable pour les méthodes DFP et BFGS car sa preuve nécessite les propriétés de la recherche linéaire exacte.

Pour les fonctions non quadratiques, sans utiliser une recherche linéaire exacte, et sous certaines conditions, Powell, en 1976 ([3]) a montré qu'une version de la méthode metric-variable (méthodes DFP et BFGS) converge vers la solution optimale, si la fonction objective f était convexe. Il montra aussi dans le même contexte que si la matrice hessienne de f était définie positive au point x^* (x^* solution optimale), alors la convergence était superlinéaire.

Dans notre mémoire on utilisé les méthodes DFP et BFGS avec les recherches linéaire inexactes de Wolf et d'Armijo. Donc il est nécessaire de donner les théorème qui caractérisent la convergence de ces méthodes.

Théorème 3.4.1 *Soit l'algorithme $x_{k+1} = x_k - \lambda_k S_k \nabla f(x_k)$, où $\{S_k\}$ est une suite de matrices définies positives dont les spectres sont uniformément bornes*

$$(0 < c_1 \leq \min m_i(S_k) \leq \max m_i(S_k) \leq c_2).$$

si $f \in C^1(\mathbb{R}^n)$, minorée, et à gradient Lipschitisien de constante L , et si les λ_k sont choisis selon la méthode de sélection d'Amijo (voir recherche linéaire d'Armijo), alors cet algorithme converge vers un minimum local de f .

Preuve : L'algorithme est bien une méthode de descente, car

$$\nabla f(x_k)^t d_k = - \|\nabla f(x_k)\|_{S_k}^2 \leq -c_1 \|\nabla f(x_k)\|^2.$$

par ailleurs, en appliquant la formule de la moyenne à la fonction $\lambda \mapsto f(x_k + \lambda d_k)$, on obtient

$$f(x_k + \lambda d_k) = f(x_k) + \nabla f(\xi_k)^t (\lambda d_k), \xi_k \in [x_k, x_k + \lambda d_k].$$

On peut donc écrire

$$f(x_k + \lambda d_k) - f(x_k) = (\nabla f(\xi_k) - \nabla f(x_k))^t (\lambda d_k),$$

ce qui, en utilisant le caractère lipschitisien du gradient de f et la définition de d_k .

$$\begin{aligned} f(x_k + \lambda d_k) - f(x_k) &\leq L \|\lambda d_k\|^2 + (\nabla f(x_k))^t (\lambda d_k) \\ &\leq L \lambda^2 \|S_k \nabla f(x_k)\|^2 - \lambda \nabla f(x_k)^t S_k \nabla f(x_k). \end{aligned}$$

Le premier terme du membre de droite étant majeure par $L\lambda^2 c_2 \|\nabla f(x_k)\|_{S_k}^2$, on peut conclure que

$$f(x_k + \lambda d_k) - f(x_k) \leq \lambda(L\lambda c_2 - 1) \|\nabla f(x_k)\|_{S_k}^2. \quad (3.16)$$

Rappelons que la méthode de sélection d'armijo, dans le cas d'une direction de descente $d_k = -S_k \nabla f(x_k)$ est

$$f(x_k + \lambda d_k) - f(x_k) \leq -\lambda_\varepsilon \|\nabla f(x_k)\|_{S_k}^2.$$

La constante ε étant fixée, il suffit de choisir λ_k tel que $L\lambda c_2 - 1 \leq -\varepsilon$, c-à-d, $\lambda_k \leq \frac{1-\varepsilon}{c_2 L}$ donc (3.16) pour remplir la condition d'Armijo. Par ailleurs, compte-tenu du principe de sélection, on aura $\lambda_k \geq \frac{1-\varepsilon}{2c_2 L}$.

On en déduit d'une part qu'avec ce choix, on a $f(x_{k+1}) = f(x_k + \lambda_k d_k) < f(x_k)$, et donc que la suite $\{f(x_k)\}_{k \in \mathbb{N}}$ converge, puisqu'elle est minorée, et d'autre part que

$$\|\nabla f(x_k)\|_{S_k}^2 \leq \frac{f(x_k) - f(x_{k+1})}{\varepsilon}. \text{ Le terme de droite convergeant vers } 0, \text{ donc } \|\nabla f(x_k)\| \rightarrow 0,$$

puisque tous les S_k ont un spectre strictement positif. La méthode proposée converge donc vers un minimum local de f . ■

Bibliographie

- [1] Y. H. Dai and Y. Yuan, *An efficient hybrid conjugate gradient méthode for unconstrained optimisation*, Ann. Oper. Res., 103 (2001), pp. 33–47.
- [2] Y. H. Dai and Y. Yuan, *A class of globally convergent conjugate gradient méthodes*, Sci. China Ser. A, 46 (2003), pp. 251–261.
- [3] Y. H. Dai and Y. Yuan, *A class of globally convergent conjugate gradient méthodes*, Research report ICM-98-030, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1998.
- [4] Y. H. Dai and Y. Yuan, *Convergence properties of the Fletcher-Reeves méthode*, IMA J. Numer. Anal., 16 (1996), pp. 155–164.
- [5] R. Danchin " Cours de calcul différentiel Licence de mathématique, 3ème année" (23 novembre 2010).
- [6] R. Fletcher, *Practical Méthodes of Optimisation vol. 1 : Unconstrained Optimisation*, John Wiley Sons, New York, 1987.
- [7] R. Fletcher and C. Reeves, *Function minimisation by conjugate gradients*, Comput. J., 7(1964), pp. 149–154.
- [8] R. Fletcher and M. J. D. Powell, *A rapidly convergent descent méthode for minimisation*, Computer. J., 6 (1963), pp. 163-168.

-
- [9] G. Fasano, *Planar conjugate gradient algorithm for large-scale unconstrained optimization*. Part 1 : Applications, *Journal of Optimization Theory and Applications* 125(2005) 543–558.
- [10] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming*, John Wiley, New York, (1968).
- [11] A.A. Goldstein and J.F. Price, *An effective algorithm for minimisation*, *Num.Math.*, 10 (1969), pp. 184-189.
- [12] A. A. Goldstein, *On steepest descent*, *SIAM J. on Control A*, Vol. 3, No. 1 (1965), pp. 147-151.
- [13] W. W. Hager and H. Zhang, *A new conjugate gradient méthode with guaranteed descent and an efficient line search*, Novembre 17, (2003) (to appear in *SIAM J. Optima.*).
- [14] B. Rachid "Optimisation Convexe, Optimisation Sans Contraites", (2007).