

وزارة التعليم العالي والبحث العلمي

Université Badji Mokhtar
Annaba

Badji Mokhtar University -
Annaba



جامعة باجي مختار

عنابة

Faculté des Sciences

Année : 2013

Département de Mathématiques

MEMOIRE

Présenté en vue de l'obtention du diplôme

MAGISTER

Convergence globale d'une nouvelle méthode du gradient

Spécialité : Mathématique

Option: mathématique du développement

présenté Par:

BOUDIAR SALEM

DIRECTEUR DE MEMOIRE : LASKRI YAMINA professeur U.B.M. ANNABA

Devant le jury

PRESIDENT : Benzine Rachid Professeur **UBMA**

Rapporteur : Laskri Yamina Professeur **UBMA**

EXAMINATEUR : Ghanem Radouéne *MCA* **UBMA**

EXAMINATEUR : Bouras Med cherif *MCA* **UBMA**

Table des matières

1	Préliminaires :Optimisation sans contraintes	12
1.1	Différentiabilité	12
1.1.1	Gradient, Hessien	12
1.1.2	Dérivée directionnelle	13
1.2	les problèmes de minimisation sans contraintes et leurs algorithmes	14
1.2.1	Aspect général des algorithmes	15
1.2.2	Fonctions multivoques et algorithmes	16
1.2.3	Convergence globale des algorithmes	17
1.2.4	Modes et vitesse de convergence	17
1.2.5	Conditions d'optimalité des problèmes d'optimisation sans contraintes	18
2	Méthodes à directions de descente et recherches linéaires	23
2.1	Direction de descente	23
2.1.1	Schémas général des algorithmes d'optimisation sans contraintes	26
2.2	Recherche linéaire	27
2.2.1	Objectifs de la recherche linéaire	27
2.2.2	But de la recherche linéaire	28
2.2.3	Intervalle de sécurité	29
2.2.4	Algorithme de base	29
2.3	Recherches linéaires "exactes"	30
2.4	Recherches linéaires inexactes	31
2.4.1	Caractérisation de l'intervalle de sécurité	31
2.5	Convergence des méthodes à directions de descente	38
2.5.1	Condition de Zoutendijk	38
2.6	Méthodes itératives d'optimisation sans contraintes	40
2.6.1	Principe des méthodes de descente	41

2.6.2	Principe des méthodes de gradient (méthodes de la plus forte descente)	42
2.6.3	Les méthodes utilisant des directions conjuguées	46
3	Convergence des méthodes du Gradient Conjugué utilisant des recherches linéaires de Wolfe fortes.	48
3.1	Le principe général d'une méthode à directions conjuguées	48
3.1.1	Description de la méthode	49
3.2	La méthode du gradient conjugué	50
3.3	Algorithme de la méthode du gradient conjugué	51
3.3.1	Algorithme de La méthode du gradient conjugué pour les fonctions quadratiques (cas linéaire)	51
3.3.2	Algorithme de La méthode du gradient conjugué pour les fonctions quelconques (cas non linéaire)	56
3.4	Synthèse des résultats de convergence des méthodes du gradient conjugué non linéaire	58
3.5	avec la recherche linéaire de Wolfe forte	58
3.5.1	Méthode de Fletcher-Reeves	60
3.5.2	Méthode de Polak-Ribière-Polyak	64
3.5.3	Méthode de la descente conjuguée	67
3.5.4	Convergence d'une classe de méthodes de Gradient Conjugué dues à Dai et Yuan avec des recherches linéaires de Wolfe fortes.	69
3.5.5	Condition générale sur les scalaires β_k	72
3.5.6	Méthodes de Newton et quasi-Newton	73
3.5.7	Méthode de Newton	73
3.5.8	Avantages et inconvénients de l'algorithme de Newton .	75
3.5.9	Méthode de quasi-Newton	76
3.6	Méthode de Broyden-Fletcher-Goldfard-Shanno(BFGS). [1970]	78
	78	
3.6.2	Remarque :	79
4	Convergence globale d'une méthode du gradient à mémoire pour l'optimisation sans contrainte.	82
4.1	Introduction	82
4.2	Direction de descente	86
4.3	Convergence globale	89
4.3.1	Paramètre de classement par taille	92

TABLE DES MATIÈRES

4.4	Résultats numériques.	94
4.4.1	problème 4.1	94
4.5	Conclusion.	100

Remerciements

*Je tiens à adresser mes remerciements les plus chaleureux et ma profonde gratitude à mon encadreur **Madame Laskri Yamina**, professeur à l'Université d'Annaba pour m'avoir proposé le sujet de ce mémoire. C'est grâce à sa grande disponibilité, ses conseils, ses orientations et ses encouragements que j'ai pu mener à bien ce travail.*

*Mes remerciements vont également à **Monsieur Benzine Rachid**, professeur à l'Université d'Annaba, pour avoir bien voulu me faire l'honneur d'accepter de présider le jury.*

*De même je remercie **Monsieur Bouras mohamed Cherif**, Maître assistant à l'Université de Annaba, et **Monsieur Ghanem Radouene**, maître de Conférence à l'Université de Annaba pour l'honneur qu'ils m'ont fait de bien vouloir accepter de faire partie du jury.*

Enfin, je n'oublie pas de remercier toutes les personnes qui m'ont facilité la tâche et tous ceux que j'ai connus à l'institut de mathématiques qui ont rendu mes séjours au département agréables.

ملخص

في هذه الرسالة نقدم لكم طريقة التدرج الجديدة المستخدمة لتحسين الذاكرة وخاصة بالنسبة للمسائل ذات الحجم الكبير، وقد عرضت الفكرة الاولى لهذه الطريقة من طرف مياي وكنترال في سنة

1969 وكذا قراق ولوفي في نفس السنة، وقد بين في بادئ الامر ان هذه الطريقة الجديدة تولد احسن اتجاها تنازليا وتتقارب كليا خاصة عند تحقق شرطي وولف الضعيفين، لقد تبين من خلال تجارب التحليل العددي ان هذه الطريقة فعالة في كل المسائل لا سيما اذا تم اختيار الوسائط بشكل جيد.

الكلمات المفتاحية الاستمثال دون قيود، طريقة تدرج الذاكرة اتجاه البحث التنازلي، شروط وولف، التقارب الكلي.

Résumé

Dans ce mémoire nous présentons une nouvelle méthode du gradient à mémoire utilisée pour l'optimisation sans contrainte, en particulier pour les problèmes de grande taille.

La première idée de cette méthode fût proposée par Miele et Cantrell (1969) ainsi que Cragg et Levy (1969). On commence par montrer que cette nouvelle méthode génère une direction de descente et qu'elle converge globalement si les conditions de Wolfe faibles sont satisfaites. Nos résultats numériques démontrent que la méthode est efficace dans certains problèmes test standards si on y inclut un bon choix de paramètres.

Mots clés optimisation sans contrainte, méthode du gradient à mémoire, direction de recherche de descente, conditions de Wolfe , convergence globale

Abstract

In this paper, we propose a memory gradient method is used for unconstrained optimization, especially large scale problems. The first idea of memory gradient method was proposed by Miele and Cantrell (1969) and Cragg and Levy (1969). In this paper, we present a new memory gradient method which generates a descent search direction for the objective function at every iteration. We show that our method converges globally to the solution if the Wolfe conditions are satisfied within the framework of the line search strategy. Our numerical results show that the proposed method is efficient for given standard test problems, if we choose a good parameter included in the method.

Key words.

unconstrained optimization, memory gradient method, descent search direction, Wolfe conditions, global convergence.

Introduction

Dans la vie courante nous sommes fréquemment confrontés à des problèmes "d'optimisation" plus ou moins complexes. Cela peut commencer au moment où l'on tente de ranger son bureau, de placer son mobilier, et aller jusqu'à un processus industriel, par exemple pour la planification des différentes tâches. Ces problèmes peuvent être exprimés sous la forme générale d'un "problème d'optimisation".

L'optimisation peut être définie comme la science qui détermine la meilleure solution à certains problèmes mathématiquement définie, qui sont souvent des modèles de physique réel. C'est une technique qui permet de "quantifier" les compromis entre des critères parfois non commensurables ([31,2004]).

L'optimisation recouvre l'étude des critères d'optimalité pour les différents problèmes, la détermination des méthodes algorithmiques de solution, l'étude de la structure de telles méthodes et l'expérimentation de l'ordinateur avec ces méthodes avec vrais problèmes de la vie ([17,1987]).

D'un point de vue mathématique, l'optimisation consiste à rechercher le *minimum* ou le *maximum* d'une fonction avec ou sans contraintes.

L'optimisation possède ses racines au 18ième siècle dans les travaux de :

- Taylor, Newton , Lagrange, qui ont élaboré les bases des développements limités.

- Cauchy ([8, 1847]) fut le premier à mettre en œuvre une méthode d'optimisation, méthode du pas de descente, pour la résolution de problèmes sans contrainte.

Il faut attendre le milieu du vingtième siècle, avec l'émergence des calculateurs et surtout la fin de la seconde guerre mondiale pour voir apparaître des avancées spectaculaires en termes de techniques d'optimisation. A noter, ces avancées ont été essentiellement obtenues en Grande Bretagne.

De nombreuses contributions apparaissent ensuite dans les années soixante. G. Zoutendijk ([51, 1960]), C. W. Carroll ([7, 1961]), P. Wolfe ([45, 1961]), R. Fletcher et M. J. D. Powell ([18, 1963]), C. Reeves ([19, 1964]), A. A. Goldstein ([24, 1965]) et A. V. Fiacco et G. P. McCormick ([20, 1968]) pour la programmation non linéaire ainsi que E. Polak et G. Ribière([38, 1969]), B.T. Polyak ([39, 1969]) et J.F. Price ([25, 1969]).

Le problème que l'on étudie ici celui de la recherche du *minimum* (*maximum*) d'une fonction réelle $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Beaucoup de problèmes peuvent se formuler de cette manière .D'autre part, dans les problèmes où les variables x_1, \dots, x_n sont astreintes à vérifier des conditions supplémentaires (du type : $g_i(x_i) \leq 0, i = 1, \dots, m$) on peut dans certaines conditions se ramener à des problèmes d'optimisation sans contraintes

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$. On cherche à résoudre le problème de minimisation sans contraintes suivant :

$$(P) : \min \{f(x) : x \in \mathbb{R}^n\} \quad (0.1)$$

Parmi les plus anciennes méthodes utilisées pour résoudre les problèmes du type (P), on peut citer la méthode du Gradient conjugué. Cette méthode est sur tout utilisée pour les problèmes de grande taille.

Cette méthode a été découverte en 1952 par Hestenes et Steifel ([28, 1952]), pour la minimisation de fonctions quadratiques strictement convexes.

Plusieurs mathématiciens ont étendu cette méthode pour le cas non linéaire. Ceci a été réalisé pour la première fois, en 1964 par Fletcher et Reeves ([19, 1964]) (*méthode de Fletcher-Reeves*) puis en 1969 par Polak, Ribière ([38, 1969]) et Ployak ([39, 1969]) (*méthode de Polak-Ribière-Ployak*). Une autre variante a été étudiée en 1987 par Fletcher ([17, 1987]) (*Méthode de la descente conjuguée*).

Toutes ces méthodes génèrent une suite $\{x_k\}_{k \in \mathbb{N}}$ de la façon suivante :

$$x_{k+1} = x_k + \alpha_k d_k \quad (0.2)$$

Le pas $\alpha_k \in \mathbb{R}$ est déterminé par une optimisation unidimensionnelle ou recherche linéaire exacte ou inexacte.

Les directions d_k sont calculées de façon récurrente par les formules suivantes :

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases} \quad (0.3)$$

$g_k = \nabla f(x_k)$ et $\beta_k \in \mathbb{R}$.

Les différentes valeurs attribuées à β_k définissent les différentes formes du gradient conjugué.

Si on note $y_{k-1} = g_k - g_{k-1}$, on obtient les variantes suivantes :

$$\beta_k^{PRP} = \frac{g_k^T y_k}{\|g_{k-1}\|^2} \quad \text{Gradient conjugué variante Polak-Ribière-Polyak} \quad (0.4)$$

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad \text{Gradient conjugué variante Fletcher Reeves} \quad (0.5)$$

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}} \quad \text{Gradient conjugué variante descente conjugué} \quad (0.6)$$

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})}, \quad \text{Gradient conjugué variante Hestenes-Stiefel} \quad (0.7)$$

Le but de ce mémoire est de faire une synthèse des travaux suivants et qui concernent des développements récents de la méthode du gradient conjugué.

1) Dai et Yuan ([14,2002]), qui a étudié les problèmes de convergence de la méthode du gradient conjugué avec des recherches linéaires inexactes de Wolfe forte, à travers de posé une condition sur le scalaire β_k .

2) Dai et Yuan ([9,1999]), qui s'intéressé au problèmes de convergence de la méthode du gradient conjugué avec des recherches linéaires inexactes de Wolfe faible à travers de créer une nouvelle formule pour β_k (*formule Dai-Yuan*).

3) Dai et Yuan ([15,2001]), qui à considéré des méthodes hybride qui est des combinaisons du la *méthode de Dai-Yuan* et la *méthode de Hestenes-Stiefel*.

Le mémoire comporte quatre chapitres :

Chapitre 1

On introduit dans ce chapitre les notions préliminaires et de base. On commence par quelques éléments de Topologie et de calcul différentiel. On termine ce chapitre par quelques notions sur les problèmes de minimisation sans contraintes et leurs algorithmes.

Chapitre 2

On expose dans ce chapitre les grandes lignes des méthodes d'optimisation sans contraintes basées sur les directions de descente et les recherches linéaires. Ces méthodes generent des suites $\{x_k\}_{k \in \mathbb{N}}$ de la façon suivante :

$$x_{k+1} = x_k + \alpha_k d_k$$

On suppose connaitre la direction de descente d_k au point x_k . La recherche linéaire consiste à trouver α_k de façon à diminuer la fonction f *suffisamment* le long de cette direction.

On insistera dans ce chapitre sur les recherches linéaires inexactes dites de Wolfe, Armijo et Goldstein-Price.

Chapitre 3

On établit dans ce chapitre une synthèse des résultats du convergence concernant les différentes variantes de la méthode du gradient conjugué avec la recherche linéaire inexacte de **Wolfe forte**.

Cette synthèse comprend les plus importants résultats de convergence concernant les méthodes de Hestenes et Steifel, de Fletcher et Reeves, de Polak-Ribière et Ployak et enfin celle de Fletcher

On détaille aussi dans ce chapitre le travail de Y. H. Dai and Y. Yuan ([14,2002]). Les auteurs introduisent une classe de méthodes du gradient conjugué de la forme (0.2), (0.3) avec des **recherches linéaires inexactes**

de Wolfe fortes et donnent une condition sur les coefficients β_k pour obtenir la convergence globale. Ce résultat sera appliqué à la méthode de Fletcher Reeves et celle de Polak Ribière.

Chapitre 4

Ce chapitre est la partie la plus importante de ce travail, car il contient les résultats de descente et de convergence globale d'une nouvelle méthode du gradient à mémoire, avec les **conditions de Wolfe**.

Dans ce chapitre, nous avons présentée une nouvelle méthode du gradient à mémoire qui donne toujours une direction de recherche en descente, et nous avons prouvé qu'elle convergeait globalement vers la solution.

Chapitre 1

Préliminaires : Optimisation sans contraintes

1.1 Différentiabilité

On se place dans \mathbb{R}^n , $n < \infty$, considéré comme un espace vectoriel normé muni de la norme euclidienne notée $\|\cdot\|$.

Soit Ω un ouvert de \mathbb{R}^n .

1.1.1 Gradient, Hessien

Définition 1.1

- On note par

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

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

- On appelle *Hessien* de f la matrice symétrique de $M_n(\mathbb{R})$

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

Définition 1.2

- On dit que x^* est un point *stationnaire* de f si $\nabla f(x^*) = 0$.
- Soit \mathcal{M}_n l'anneau des matrices carrées $n \times n$ dans \mathbb{R} . Une matrice $A \in \mathcal{M}_n$ est dite *semi-définie positive* si

$$\forall x \in \mathbb{R}^n : x^T A x \geq 0$$

elle est dite *définie positive* si

$$\forall x \in \mathbb{R}_*^n : x^\top Ax > 0$$

1.1.2 Dérivée directionnelle

Définition 1.3[4, p.8]

On appelle *dérivée directionnelle* de f dans la direction d au point x , notée $\delta f(x, d)$, la limite (éventuellement $\pm\infty$) du rapport :

$$\frac{f(x + hd) - f(x)}{h} \text{ lorsque } h \text{ tend vers } 0.$$

Autrement dit :

$$\delta f(x, d) = \lim_{h \rightarrow 0} \frac{f(x + hd) - f(x)}{h} = \nabla^T f(x) d$$

- Si $\|d\| = 1$: la dérivée directionnelle est le taux d'accroissement de f dans la direction d au point x .

Remarque 1.1

- Le taux d'accroissement est maximal dans la direction du gradient
- Le gradient indique la direction de la plus grande pente.

Définition 1.4

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$. $y = f(x)$ définit une surface dans \mathbb{R}^{n+1} . $f(x) = r$, avec r constant, définissent des courbes de niveau ou des contours de cette surface.

Un *contour* de niveau r est l'ensemble des points $R(r) = \{x / f(x) = r\}$

Exemple 1.1

La fonction de Rosenbrock $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$.

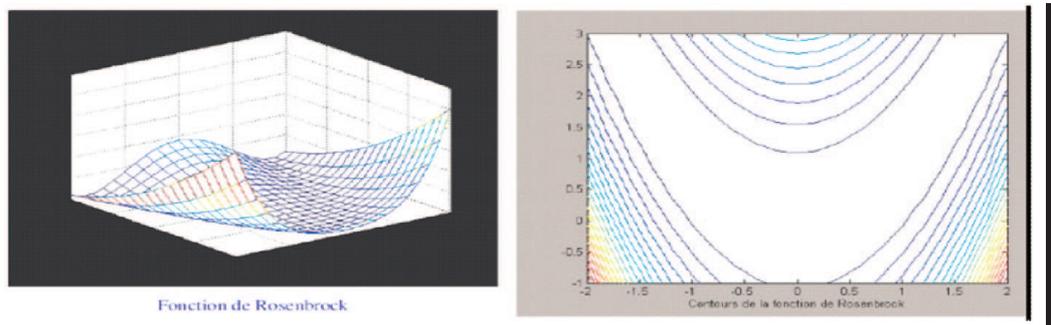


FIG 1.1 courbes de niveau ou des contours de surface.

Propriété 1.1

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et ∇f son gradient, R le contour, alors le gradient ∇f est orthogonal au contour.

Démonstration :

Soit le contour R de niveau $f(x_0)$ passant par $x_0 : R = \{x / f(x) = f(x_0)\}$
 Soit une courbure dans R paramétrée par $v(t) : \mathbb{R}^n \rightarrow \mathbb{R}$ avec $v(t_0) = x_0$
 et $v(t) = x \in R$.

La tangente à cette courbe en x_0 est de direction

$$\frac{dv}{dt}/_{t=t_0}$$

Nous Savons que

$$\begin{aligned} f(v(t)) &= f(v(t_0)) = f(x_0) \\ \Rightarrow \frac{d}{dt}f(v(t))/_{t=t_0} &= 0 \\ \Leftrightarrow \frac{\partial f}{\partial x_1}(v(t_0))\frac{dv_1}{dt}/_{t=t_0} + \dots + \frac{\partial f}{\partial x_n}(v(t_0))\frac{dv_n}{dt}/_{t=t_0} &= 0 \\ \Leftrightarrow \nabla^T f(x_0)\frac{dv}{dt}/_{t=t_0} &= 0 \end{aligned}$$

Ce qui veut dire que $\nabla f(x_0)$ est orthogonal à R en $x_0, \forall x_0 \quad \square$

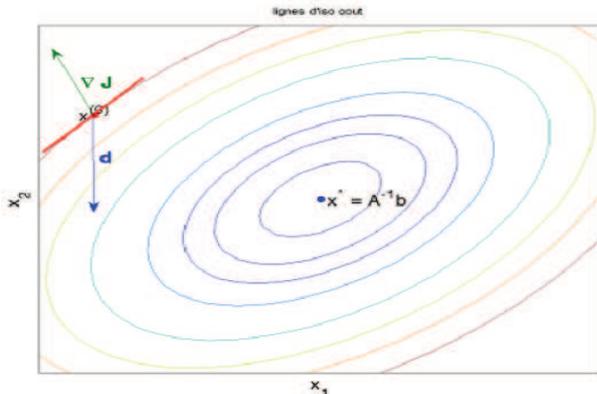


FIG 1.2

1.2 les problèmes de minimisation sans contraintes et leurs algorithmes

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$. On appelle problème de minimisation sans contraintes le problème suivant :

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

L'étude de ces problèmes est importante pour des raisons diverses. Beaucoup de problèmes d'optimisation avec contraintes sont transformés en des suites des problèmes d'optimisation sans contraintes (multiplicateur de Lagrange, méthodes des pénalités, ...). L'études des problèmes d'optimisation sans contraintes trouve aussi des applications dans la résolution des systèmes non linéaires.

Les méthodes numériques de résolution de divers problèmes de minimisation sans contraintes ont pris ces dernières années un bel essor si bien que la bibliographie correspondante contient des centaines d'ouvrages et d'articles. Cet intérêt n'est nullement fortuit, il reflète le rôle de premier plan que les problèmes d'optimisation jouent dans les applications.

La construction d'algorithmes consacrés à la recherche efficace de minimum d'une fonction sans contraintes est un problème complexe, car il ne suffit pas d'élaborer un algorithme il faut montrer de plus qu'il importe sur ceux connus.

On compare des algorithmes en se basant sur des plusieurs critères, par exemple, la précision du résultat, le nombre d'évaluations fonctionnelles et celui d'évaluations du gradient, la vitesse de la convergence, le temps de calcul, l'occupation de la mémoire nécessaire,

Même en se fixant des critères de comparaison, on ne peut pas classer les algorithmes ni en indiquant lequel et le meilleur ou le pire. Le faits et qu'on obtient les estimations de l'un des critères précédents pour des classes des problèmes, et un algorithme mauvais pour une vaste classe peut s'avérer efficace pour une autre, plus restreinte. L'utilisateur doit posséder tout un arsenal d'algorithmes pour être en mesure de faire face à chaque problème posé.

1.2.1 Aspect général des algorithmes

Pour construire des algorithmes de minimisation sans contrainte on fait appel à des processus itératifs du type

$$x_{k+1} = x_k + \alpha_k d_k \quad (1.1)$$

où d_k détermine la direction de déplacement à partir du point x_k et α_k est un facteur numérique dont la grandeur donne la longueur du pas dans la direction d_k .

Le type d'algorithme permettant de résoudre le problème (1.1) sera déterminé dès qu'on définit les procédés de construction du vecteur d_k et de calcul de α_k à chaque itération.

La façon avec laquelle on construit les vecteurs d_k et les scalaires α_k déterminent directement les propriétés du processus et spécialement en ce qui concerne la convergence de la suite $\{x_k\}$, la vitesse de la convergence,....

Pour s'approcher de la solution optimale du problème (1.1) (dans le cas général, c'est un point en lequel ont lieu peut être avec une certaine précision les conditions nécessaires d'optimalité de f), on se déplace naturellement à partir du point x_k dans la direction de la décroissance de la fonction f .

1.2.2 Fonctions multivoques et algorithmes

Une application multivoque est une application A qui à $x \in \mathbb{R}^n$ fait correspondre un sous ensemble $A(x)$ de \mathbb{R}^n .

Étant donné un point x_k . En appliquant les instructions d'un certain algorithme, on obtient un nouveau point x_{k+1} . Cette procédure peut être décrite par une application multivoque A appelée application algorithmique. Donc ; étant donné un point $x_1 \in \mathbb{R}^n$, l'application algorithmique génère une suite x_1, x_2, \dots , où $x_{k+1} \in \mathbb{R}^n$ pour tout k .

La notion d'application algorithmique fermée est directement liée à la convergence des algorithmes

Définition 1.5. Soient X et Y deux sous ensembles fermés non vides de \mathbb{R}^n et \mathbb{R}^m respectivement et $A : X \rightarrow Y$ une application

multivoque. A est dite fermée au point $x \in \mathbb{R}^n$ si :

$$\left. \begin{array}{l} x_k \in X, x_k \rightarrow x \\ y_k \in Y, y_k \rightarrow y \end{array} \right\} \text{ implique que } y \in A(x). \quad \square$$

Donnons un théorème de convergence qui utilise les fonctions multivoques et qui nous sera utile par la suite. Avant cela notons qu'à cause de la non convexité, de la taille du problème et d'autres difficultés on peut arrêter le processus itératif si on trouve un point appartenant à un ensemble spécifique qu'on appelle ensemble des solutions Ω .

Voici ci-dessous quelques exemples typiques de cet ensemble.

$$\Omega = \{x_* : \nabla f(x_*) = 0\}$$

$$\Omega = \{x_* : x_* \text{ est un solution optimale locale du problème (1.1)}\}$$

$\Omega = \{x_* : f(x_*) < \nu_* + \varepsilon\}$ où $\varepsilon > 0$ est une tolérance définie à l'avance et ν_* est la valeur minimale de la fonction objective.

Nous dirons que l'application algorithmique $A : X \rightarrow X$ converge dans $Y \subseteq X$ si commençant par n'importe quel point initial $x_1 \in Y$, la limite de toute sous suite convergente, extraite de la suite x_1, x_2, \dots , générée par l'algorithme, appartient à Ω .

Théorème 1.1. ([4]). Soient X un ensemble non vide fermé dans \mathbb{R}^n et $\Omega \subseteq X$ un ensemble des solutions non vide. Soit $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continue, on considère que l'application algorithmique $C : X \rightarrow$

X vérifie la propriété suivante : "étant donné $x \in X$ alors $\alpha(y) \leq \alpha(x)$ pour $y \in C(x)$ ". Soit $B : X \rightarrow X$ une application algorithmique fermée sur le complémentaire de Ω qui vérifie :

$\alpha(y) < \alpha(x)$ pour $y \in B(x)$ et $x \notin \Omega$. Maintenant considérons l'application algorithmique composée $A = CB$. Soit $x_1 \in X$, la suite $\{x_k\}$ est générée comme suit :

Si $x_k \in \Omega$, stop ; sinon poser $x_{k+1} \in A(x_k)$, remplacer k par $(k + 1)$ et répéter.

Supposons que $\Lambda = \{x \in X, \alpha(x) \leq \alpha(x_1)\}$ est compacte. Alors, ou bien l'algorithme s'arrête à un nombre fini d'itérations par un point $\in \Omega$, ou tous les points d'accumulation de $\{x_k\}$ appartenants à Ω .

1.2.3 Convergence globale des algorithmes

Définition 1.6

Nous dirons qu'un algorithme décrit par une application multivoque A , est *globalement convergent* (ou encore : possède la propriété de convergence globale) si, quel que soit le point de départ x_1 choisi, la suite $\{x_k\}$ engendrée par $x_{k+1} \in A(x_k)$ (ou une sous suite) converge vers un point satisfaisant les conditions nécessaires d'optimalité.

1.2.4 Modes et vitesse de convergence

La convergence globale d'une algorithme ayant été établie, nous nous intéressons maintenant à l'évaluation de son *efficacité*. D'un point de vue pratique, l'efficacité d'un algorithme dépend du nombre d'itérations nécessaires pour obtenir une approximation à ε près (ε fixé à l'avance) de l'optimum x^* . Si l'on compare entre eux plusieurs algorithmes, et si l'on admet que le temps de calcul par itération est sensiblement le même pour tous, le meilleur est celui qui nécessitera le plus petite nombre d'itérations.

Malheureusement, il se révèle impossible de dégager des conclusions générales de ce genre de comparaison

Suivant le point de départ choisi, la nature de la fonction à optimiser, la valeur de la tolérance choisie, la hiérarchie des algorithmes peut varier considérablement.

Si l'on veut dégager un critère ayant une certaine valeur d'absolu, il faut par conséquent recourir à un autre type d'analyse : c'est l'objet de l'étude de la *convergence asymptotique* c'est-à-dire du comportement de la suite $\{x_k\}$ au voisinage du point limite x_* .

Ceci conduit à attribuer à chaque algorithme une indice d'efficacité appelé *se vitesse de convergence*.

Nous introduisons ici les principales définitions de base qui seront abondamment utilisées par la suite.

Plaçons nous dans \mathbb{R}^n , où $\|\cdot\|$ désigne la norme euclidienne et considérons une suite $\{x_k\}$ convergeant vers x^* .

· Si $\limsup \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \lambda < 1$

On dit que la convergence est *linéaire* et λ est le *taux de convergence* associé.

· Si $\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \rightarrow 0$ quand $k \rightarrow \infty$

on dit que la convergence est *superlinéaire*.

Plus précisément si $\exists p > 1$ tel que :

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} < +\infty$$

on dit que la convergence est *superlinéaire* d'ordre p .

En particulier si

$$\limsup_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} < +\infty$$

on dit que la convergence est *quadratique* (superlinéaire d'ordre 2)

Définition 1.7

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$. On considère le problème d'optimisation (1.1)

1- $\tilde{x} \in \mathbb{R}^n$ est un optimum local ssi $\exists V_\epsilon(\tilde{x})$ tel que $\forall x \in V_\epsilon(\tilde{x}) : f(x) \geq f(\tilde{x})$

2- $\tilde{x} \in \mathbb{R}^n$ est un optimum local stricte ssi $\exists V_\epsilon(\tilde{x})$ tel que $\forall x \in V_\epsilon(\tilde{x})$

$x \neq \tilde{x} : f(x) > f(\tilde{x})$

3- $\tilde{x} \in \mathbb{R}^n$ est un optimum global ssi $\forall x \in \mathbb{R}^n \quad x \neq \tilde{x} : f(x) \geq f(\tilde{x})$

1.2.5 Conditions d'optimalité des problèmes d'optimisation sans contraintes

Définition 1.8. *Considérons le problème de minimisation sans contraintes* (1.1).

a) $x_* \in \mathbb{R}^n$ s'appelle *minimum global* du problème (1.1) si

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

b) x_* est un *minimum local* de (1.1) s'il existe un voisinage $V_\epsilon(x_*)$ de x_* tel que

$$f(x_*) \leq f(x), \quad \forall x \in V_\epsilon(x_*).$$

c) x_* est minimum local strict s'il existe un voisinage $V_\varepsilon(x_*)$ de x_* tel que

$$f(x_*) < f(x), \forall x \in V_\varepsilon(x_*) \quad \text{et} \quad x \neq x_*. \quad \square$$

Conditions nécessaires d'optimalité

Théorème 1.2. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ telle que f soit différentiable au point $\bar{x} \in \mathbb{R}^n$. Soit $d \in \mathbb{R}^n$ telle que $\nabla f(\bar{x})^t d < 0$. Alors il existe $\delta > 0$ tel que $f(\bar{x} + \alpha d) < f(\bar{x})$ pour tout $\alpha \in]0, \delta[$. La direction d s'appelle dans ce cas direction de descente.

Démonstration. ([4])

Comme f est différentiable en \bar{x} alors

$$f(\bar{x} + \alpha d) = f(\bar{x}) + \alpha \nabla f(\bar{x})^t d + \alpha \|d\| \lambda(\bar{x}; \alpha d)$$

où $\lambda(\bar{x}; \alpha d) \rightarrow 0$ pour $\alpha \rightarrow 0$. Ceci implique :

$$\frac{f(\bar{x} + \alpha d) - f(\bar{x})}{\alpha} = \nabla f(\bar{x})^t d + \|d\| \lambda(\bar{x}; \alpha d), \quad \alpha \neq 0$$

et comme $\nabla f(\bar{x})^t d < 0$ et $\lambda(\bar{x}; \alpha d) \rightarrow 0$ pour $\alpha \rightarrow 0$, il existe $\delta > 0$ tel que

$$\nabla f(\bar{x})^t d + \|d\| \lambda(\bar{x}; \alpha d) < 0 \quad \text{pour tout} \quad \alpha \in]0, \delta[$$

et par conséquent on obtient :

$$f(\bar{x} + \alpha d) < f(\bar{x}) \quad \text{pour tout} \quad \alpha \in]0, \delta[. \quad \square$$

Corollaire 1.1. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable en \bar{x} , si \bar{x} est un minimum local alors $\nabla f(\bar{x}) = 0$.

Démonstration. Par contre, on suppose que $\nabla f(\bar{x}) \neq 0$. Si on pose $d = -\nabla f(\bar{x})$, on obtient :

$$\nabla f(\bar{x})^t d = -\|\nabla f(\bar{x})\|^2 < 0$$

et par le théorème précédent, il existe $\delta > 0$ tel que :

$$f(\bar{x} + \alpha d) < f(\bar{x}) \quad \text{pour tout} \quad \alpha \in]0, \delta[$$

mais ceci est contradictoire avec le fait que \bar{x} est un minimum local, d'où $\nabla f(\bar{x}) = 0$. \square

Théorème 1.3. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable en \bar{x} . Supposons que \bar{x} soit minimum local. Alors $\nabla f(\bar{x}) = 0$ et $H(\bar{x})$ est semi définie positive.

Démonstration. ([35])

Le corollaire ci-dessus montre la première proposition, pour la deuxième proposition on a :

$$f(\bar{x} + \alpha d) = f(\bar{x}) + \alpha \nabla f(\bar{x})^t d + \frac{1}{2} \alpha^2 d^t H(\bar{x}) d + \alpha^2 \|d\|^2 \lambda(\bar{x}; \alpha d)$$

où $\lambda(\bar{x}; \alpha d) \rightarrow 0$ pour $\alpha \rightarrow 0$. Ceci implique :

$$\frac{f(\bar{x} + \alpha d) - f(\bar{x})}{\alpha^2} = \frac{1}{2} d^t H(\bar{x}) d + \|d\|^2 \lambda(\bar{x}; \alpha d), \quad \alpha \neq 0.$$

Comme \bar{x} est un minimum local alors $f(\bar{x} + \alpha d) \geq f(\bar{x})$ pour α suffisamment petit, d'où

$$\frac{1}{2} d^t H(\bar{x}) d + \|d\|^2 \lambda(\bar{x}; \alpha d) \geq 0 \text{ pour } \alpha \text{ petit.}$$

En passant à la limite quand $\alpha \rightarrow 0$, on obtient que $d^t H(\bar{x}) d \geq 0$, d'où $H(\bar{x})$ est semi définie positive. \square

Conditions suffisantes d'optimalité

Théorème 1.4. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable en \bar{x} . Si $\nabla f(\bar{x}) = 0$ et $H(\bar{x})$ est définie positive, alors \bar{x} est minimum local strict.

Démonstration. ([35])

f est deux fois différentiable au point \bar{x} . Alors $\forall x \in \mathbb{R}^n$, on obtient :

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})^t (x - \bar{x}) + \frac{1}{2} (x - \bar{x})^t H(\bar{x}) (x - \bar{x}) + \|x - \bar{x}\|^2 \lambda(\bar{x}; x - \bar{x})$$

où $\lambda(\bar{x}; x - \bar{x}) \rightarrow 0$ pour $x \rightarrow \bar{x}$. Supposons que \bar{x} n'est pas un minimum local strict. Donc, il existe une suite $\{x_k\}$ convergente vers \bar{x} telle que

$f(x_k) \leq f(\bar{x})$, $x_k \neq \bar{x}$, $\forall k$. Posons $d_k = (x_k - \bar{x}) / \|x_k - \bar{x}\|$. Donc, $\|d_k\| = 1$ et on obtient à partir de (1.3) :

$$\frac{f(x_k) - f(\bar{x})}{\|x_k - \bar{x}\|^2} = \frac{1}{2} d_k^t H(\bar{x}) d_k + \lambda(\bar{x}; x_k - \bar{x}) \leq 0, \quad \forall k$$

et comme $\|d_k\| = 1$, $\forall k$ alors $\exists \{d_k\}_{k \in N_1 \subset \mathbb{N}}$ telle que $d_k \rightarrow d$ pour $k \rightarrow \infty$ et $k \in N_1$. On a bien sûr $\|d\| = 1$. Considérons donc $\{d_k\}_{k \in N_1}$ et le fait que $\lambda(\bar{x}; x - \bar{x}) \rightarrow 0$ pour $k \rightarrow \infty$ et $k \in N_1$. Alors (1.4) donne : $d^t H(\bar{x}) d \leq 0$, ce qui contredit le fait que $H(\bar{x})$ est définie positive car $\|d\| = 1$ (donc, $d \neq 0$). Donc, \bar{x} est un minimum local strict. \square

Cas convexe **Définition 1.9.** Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

a) f est dite convexe sur \mathbb{R}^n si

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

pour tous les points x_1 et $x_2 \in \mathbb{R}^n$ et pour tout $t \in [0, 1]$.

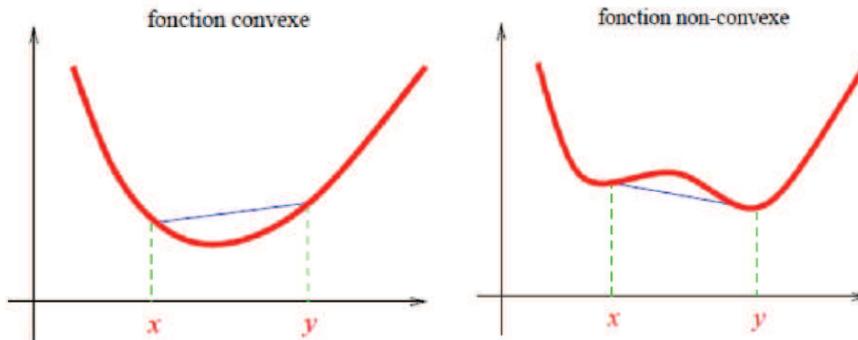


FIG 1.3

b) Si l'inégalité précédente est stricte pour tous les points x_1 et x_2 distincts et pour tout $t \in]0, 1[$, alors f est dite strictement convexe.

c) Supposons que f soit différentiable. f est dite fortement convexe si :

$$f((1-t)x_1 + tx_2) \leq (1-t)f(x_1) + tf(x_2) - \frac{\alpha}{2}t(1-t)\|x_1 - x_2\|^2$$

pour tous point x_1 et $x_2 \in \mathbb{R}^n$ et pour tout $t \in [0, 1]$.

d) Supposons que f soit différentiable. f est dite pseudo convexe si :

Pour tout x_1 et x_2 vérifiant $\nabla f(x_1)^t(x_2 - x_1) \geq 0$, on a : $f(x_2) \geq f(x_1)$. \square

Théorème 1.5. ([4]). Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ telle que f est convexe et différentiable. Alors x_* est un minimum global de f si et seulement si $\nabla f(x_*) = 0$. \square

Théorème 1.6. ([4]). Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ telle que est pseudo convexe. Soit $x_* \in \mathbb{R}^n$ tel que $\nabla f(x_*) = 0$, alors x_* est un minimum global de f . \square

Remarque 1.2. Les théorèmes 1.4 et 1.5 demeurent vrais si on remplace \mathbb{R}^n par un ouvert S de \mathbb{R}^n .

Remarque 1.3. Dans le cas ou f est convexe, alors tout minimum local est aussi global. De plus si f est strictement convexe, alors tout minimum local devient non seulement global mais aussi unique ([4]) \square

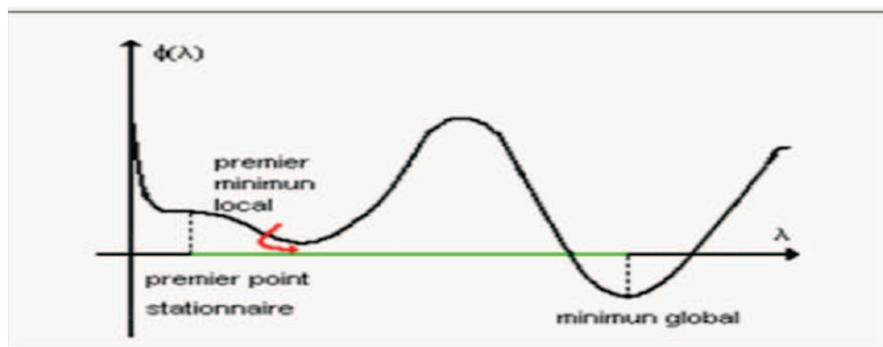


FIG 1.4 :point stationnaire, optimum local, optimum global

Chapitre 2

Méthodes à directions de descente et recherches linéaires

Ce chapitre introduit une classe importante d'algorithmes de résolution des problèmes d'optimisation sans contraintes.

Le concept central est celui de direction de descente. On le retrouvera dans des contextes variés, également pour résoudre des problèmes avec contraintes. Tous les algorithmes d'optimisation n'entrent pas dans ce cadre. Une autre classe importante de méthodes se fonde sur la notion de région de confiance.

Après avoir décrit comment fonctionne un algorithme à directions de descente, nous donnons quelques exemples d'algorithmes de ce type. Nous décrivons ensuite les principales règles de recherche linéaire

2.1 Direction de descente

Principes généraux

Considérons le problème d'optimisation sans contrainte :

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

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

On note aussi $\nabla f(x)$ et $\nabla^2 f(x)$ le gradient et le hessien de f en x pour ce produit scalaire.

On s'intéresse ici à une classe d'algorithmes qui sont fondés sur la notion de direction de descente.

On dit que d est une direction de descente de f en $x \in \mathbb{R}^n$ si

$$\nabla f(x)^T \cdot d < 0. \tag{2.1}$$

CHAPITRE 2. MÉTHODES À DIRECTIONS DE DESCENTE ET RECHERCHES LINÉAIRES

Ou encore que d fait avec l'opposé du gradient $-\nabla f(x)$ un angle θ strictement plus petit que 90° : $\theta = \arccos \frac{-\nabla f(x)^T \cdot d}{\|\nabla f(x)\| \|d\|} \in [0, \frac{\pi}{2}[$.

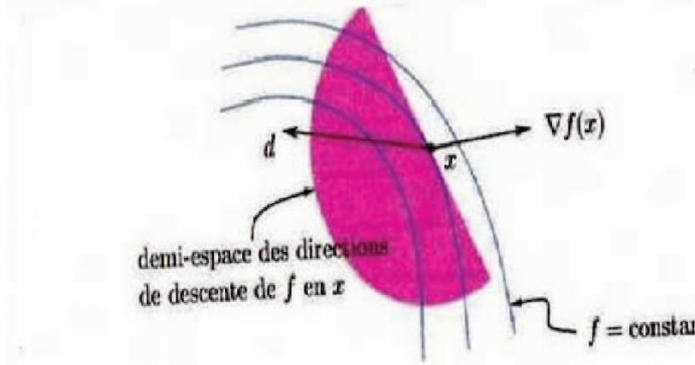


FIG 2.1. L'ensemble des directions de descente de f en x ,

$$\{d \in \mathbb{R}^n : \nabla^T f(x) \cdot d < 0\}$$

forme un demi-espace ouvert de \mathbb{R}^n .

On voit que si d est une direction de descente, $f(x + \lambda d) < f(x)$, pour tout $\lambda > 0$ suffisamment petit, et donc, que f décroît strictement dans la direction d . Des telles directions sont intéressantes en optimisation car, pour faire décroître f , il suffit de faire un déplacement le long de d . Les méthodes à directions de descentes utilisent cette idée pour minimiser une fonction. Elles construisent la suite des itérés $\{x_k\}_{k \geq 1}$, approchant un solution x_k du problème (1.1) par la récurrence : $x_{k+1} = x_k + \lambda_k d_k$, pour $k \geq 1$

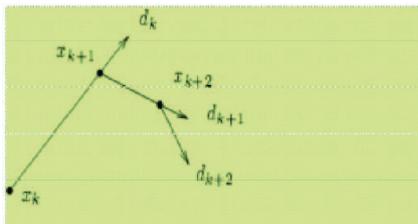


FIG 2.2 . La direction de descente

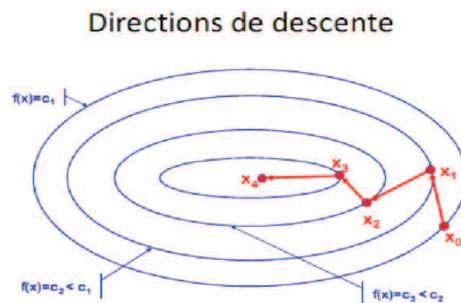


FIG 2.3 .

Où λ_k est appelé le pas et d_k la direction de descente de f en x_k .

Pour définir une direction de descente il faut donc spécifier deux choses :

♦ dire comment la direction d_k est calculée ; la manière de procéder donne le nom à l'algorithme ;

◆ dire comment on détermine le pas λ_k ; c'est ce que l'on appelle la recherche linéaire.

Remarque 1.3.

Par la suite l'angle θ_k entre la direction d_k et $-g_k$ jouera un rôle important dans le processus de la convergence. Pour θ_k on a la relation classique suivante :

$$-g_k^T d_k = \|g_k\| \|d_k\| \cos \theta_k$$

Définition 1.21

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

$$f(\hat{x} + \lambda d) < f(\hat{x}) : \quad \forall \lambda \in]0, \delta[, \quad (2.2)$$

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

Théorème 1.6.

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 :

$$f'(\hat{x}, d) = \nabla f(\hat{x})^T \cdot d < 0.$$

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

Preuve :

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

$$f(\hat{x} + \lambda d) = f(\hat{x}) + \lambda \nabla f(\hat{x})^T \cdot d + \lambda \|d\| \alpha(\hat{x}, d),$$

avec $\alpha(\hat{x}, d) \xrightarrow{\lambda \rightarrow 0} 0$ ceci implique que

$$f'(\hat{x}, d) = \lim_{\lambda \rightarrow 0} \frac{f(\hat{x} + \lambda d) - f(\hat{x})}{\lambda} = \nabla f(\hat{x})^T \cdot d < 0. \quad (2.3)$$

La limite étant strictement négative, alors il existe un voisinage de zéro $V(0)$ tel que

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

La relation (1.12) est particulièrement vraie pour tout $\lambda \in]-\delta, +\delta[$. On obtient le résultat cherché en multipliant la relation (1.12) par $\lambda > 0$.

Algorithme 2.1(méthode à directions de descente- une itération)

Etape 0 :(initialisation)

On suppose qu'au début de l'itération k , on dispose d'un itéré $x_k \in \mathbb{R}^n$

Etape1 :

Test d'arrêt : si $\|\nabla f(x_k)\| \simeq 0$, arrêt de l'algorithme ;

Etape2 :

Choix d'une direction de descente $d_k \in \mathbb{R}^n$;

Etape3 :

Recherche linéaire : déterminer un pas $\alpha_k > 0$ le long de d_k de manière à "faire décroître f suffisamment" ;

Etape4 :

Si la recherche linéaire réussie $x_{k+1} = x_k + \lambda_k d_k$;
remplacer k par $k + 1$ et aller à l'étape 1.

2.1.1 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 :

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

Vue la définition du direction de descente, on est assurée que

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

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

Exemple de choix de direction de descente :

Par exemple s'on choisit $d_k = -\nabla f(x_k)$ et si $\nabla f(x_k) \neq 0$, on obtient la méthode du gradient. Dans le cas $d_k = -(H(x_k))^{-1} \cdot \nabla f(x_k)$, on obtient La méthode de Newton, où la matrice Hessienne $H(x_k)$ est définie positive.

Exemple de choix de pas λ_k :

On choisit en général λ_k de façon optimale, c'est- a -dire que λ_k doit vérifier

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

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

2.2 Recherche linéaire

Considérons le problème d'optimisation sans contraintes P .

$$(P) : \min_{x \in \mathbb{R}^n} f(x), \quad (2.9)$$

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

Les algorithmes qu'on étudie par la suite suivant les schémas généraux suivants :

$$x_{k+1} = x_k + \lambda_k d_k \quad (2.10)$$

où λ_k est 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 suivant :

$$\lambda \rightarrow \varphi(\lambda) = f(x_k + \lambda d_k). \quad (2.11)$$

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

Dans ce chapitre nous allons décrire les différentes manières de déterminer un pas $\lambda_k > 0$ le long d'une direction de descente d_k . C'est ce que l'on appelle faire de la recherche linéaire. Il existe deux grandes classes des méthodes qui s'intéressent à l'optimisation unidimensionnelle :

- ▶ les recherches linéaires exactes.
- ▶ les recherches linéaires inexactes.

2.2.1 Objectifs de la recherche linéaire

Il s'agit de réaliser deux objectifs :

Le premier objectif, faire décroître f suffisamment, et pour cela on cherche à vérifier l'inégalité :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \text{“un terme négatif”} \quad (2.12)$$

Le terme négatif joue un rôle-clé dans la convergence de l'algorithme.

Le second objectif, on choisie le pas $\lambda_k > 0$ d'être trop petit, pour assurer la convergence d'algorithme au point stationnaire .

Le premier objectif n'est en effet pas suffisant car l'inégalité (2.4) est en général satisfaite par des 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

$$0 < \lambda_k \leq \frac{\epsilon}{2^k \|d_k\|},$$

la suite $\{x_k\}$ générée par (2.4) est de Cauchy, puisque pour $1 \leq l < k$ on a

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

lorsque $l \rightarrow \infty$.

Donc, $\{x_k\}$ converge, disons vers un point \bar{x} . En prenant $l = 1$ et $k \rightarrow \infty$ dans l'estimation ci-dessus, on voit que $\bar{x} \in \overline{B}(x_1, \epsilon)$ et donc \bar{x} ne saurait être solution s'il n'y a pas de solution donc $B(x_1, \epsilon)$. On a arbitrairement forcé la convergence de $\{x_k\}$ en prenant des pas très petits.

Pour simplifier les notations, on définit la restriction de f à la droite $\{x_k + \lambda d_k : \lambda \in \mathbb{R}\}$ comme la fonction

$$\phi : \mathbb{R}_+ \rightarrow \mathbb{R}; \quad \lambda \mapsto \phi(\lambda) = f(x_k + \lambda d_k).$$

2.2.2 But de la recherche linéaire

Dans le cas non-quadratique les méthodes de descente (2.2), nécessitent la recherche d'une valeur de $\lambda_k > 0$ optimale ou non, vérifiant :

$$f(x_k + \lambda_k d_k) \leq f(x_k). \tag{2.13}$$

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

$$\phi'(\lambda^*) = 0,$$

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

autrement dit, λ^* est un minimum local de ϕ qui assure de plus la décroissance de f . En fait, dans la plus-part 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 φ , 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.

2.2.3 Intervalle de sécurité

Dans la plupart des algorithmes d'optimisation modernes, on ne fait jamais de recherche linéaire exacte, car trouver α_k signifie qu'il va falloir calculer un grand nombre de fois la fonction h_k et cela peut être dissuasif du point de vue du temps de calcul.

En pratique, on recherche plutôt une valeur de α^* qui assure une décroissance suffisante de f .

Cela conduit à la notion d'intervalle de sécurité.

Définition 2.2. *On dit que $[\alpha_g, \alpha_d]$ est un intervalle de sécurité s'il permet de classer les valeurs de α de la façon suivant :*

- Si $\alpha < \alpha_g$ alors α est considéré trop petit,
- Si $\alpha_d \geq \alpha \geq \alpha_g$ alors α est satisfaisant,
- Si $\alpha > \alpha_d$ alors est considéré trop grand.

Le problème est de traduire de façon numérique sur h_k les trois conditions précédentes, ainsi que de trouver un algorithme permettant de déterminer α_g et α_d .

2.2.4 Algorithme de base

Algorithme 2.2 (Algorithme de base)

Etape 0 : (initialisation)

$\alpha_g = \alpha_d = 0$, choisir $\alpha_1 > 0$, poser $k = 1$ et aller à l'étape 1 ;

Etape 1 :

Si α_k convient, poser $\alpha^* = \alpha_k$ et on s'arrête.

Si α_k est trop petit on prend $\alpha_{g,k+1} = \alpha_k$, $\alpha_d = \alpha_d$ et on va à l'étape 2 .

Si α_k est trop grand on prend $\alpha_{d,k+1} = \alpha_k$, $\alpha_g = \alpha_g$ et on va à l'étape 2 .

Etape 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. □

Au lieu de demander que α_k minimise φ , on préfère imposer des conditions moins restrictives, plus facilement vérifiées, qui permettent toute fois de contribuer à la convergence des algorithmes. En particulier, il n'y aura plus un unique pas (ou quelques pas) vérifiant ces conditions mais tout un intervalle de pas (ou plusieurs intervalles), ce qui rendra d'ailleurs leur recherche plus aisée. C'est ce que l'on fait avec les règles d'Armijo, de Goldstein et de Wolfe décrites dans la prochaine section.

2.3 Recherches linéaires "exactes"

Comme on cherche à minimiser f , il semble naturel de chercher à minimiser le critère le long de d_k et donc, de déterminer le pas λ_k comme solution du problème

$$\min_{\lambda > 0} \varphi(\lambda)$$

C'est ce que l'on appelle la règle de *Cauchy* et le pas déterminé par cette règle est appelé : pas de *Cauchy* ou pas optimal. Dans certains cas, on préférera le plus petit point stationnaire de φ qui fait décroître par cette fonction :

$$\lambda_k = \inf \{ \lambda > 0 : \varphi'(\lambda) = 0, \varphi(\lambda) < \varphi(0) \}.$$

On parle alors de règle de *Curry* et le pas déterminé par cette règle est appelé "pas de *Curry*". De manière un peu imprécise, ces deux règles sont parfois qualifiées de recherche linéaire exacte. Ils ne sont utilisés que dans des cas particuliers, par exemple lorsque φ est quadratique, la solution de la recherche linéaire s'obtient de façon exacte et dans un nombre fini d'itérations.

On aurait aimé restreindre notre étude sur ce domaine sympathique, malheureusement, c'est très rare de rencontrer des problèmes quadratiques, et donc par exemple pour une fonction non linéaire arbitraire,

- ▶ la détermination de ces pas demande en général beaucoup de temps de calcul et ne peut pas de toute façon pas être faite avec une précision infinie,

- ▶ l'efficacité supplémentaire éventuellement apportée à un algorithme par une recherche linéaire exacte ne permet pas, en général, de compenser le temps perdue à déterminer un tel pas,

- ▶ les résultats de convergence autorisent d'autres types de règles, moins gourmandes en temps de calcul.

Au lieu de demander que λ_k minimise φ , on préfère imposer des conditions moins restrictives, plus facilement vérifiées, qui permettent toute fois de contribuer à la convergence des algorithmes. En particulier, il n'y aura plus

un unique pas (ou quelques pas) vérifiant ces conditions mais tout un intervalle de pas (ou plusieurs intervalles), ce qui rendra d'ailleurs leur recherche plus facile.

C'est ce que l'on fait avec les règles d'*Armijo*, de *Goldstein* et de *Wolfe* décrites ci-dessous.

2.4 Recherches linéaires inexactes

Les recherches linéaire exacte, malgré qu'elles n'aboutissent à une solution optimal qu'avec une tolérance fixée à l'avance, elles nécessitent beaucoup d'observation à chaque itération de l'algorithme principal. Des mathématiciens ont réussi (années 60,70,80) à élaborer des recherche linéaire qui demandent peu d'observations, mais respectent en même temps la descente de la fonction économique. On a même réussi à avoir une vitesse de convergence super linéaire.

L'objectif de cette section consiste à présenter les principales tests.

2.4.1 Caractérisation de l'intervalle de sécurité

la règle d'Armijo

Une condition naturelle est de demander que f décroisse autant qu'une portion $\omega_1 \in]0, 1[$ de ce que ferait le modèle linéaire de f en x_k . Cela conduit à l'inégalité suivante, parfois appelée condition d'Armijo ou condition de décroissance linéaire :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \omega_1 \lambda_k \nabla f(x_k)^T d_k. \quad (2.14)$$

Elle est de la forme (2.4), car ω_1 devra être choisie dans $]0, 1[$. On voit bien à la figure (2.1) ce que signifie cette condition. Il faut qu'en λ_k , la fonction prenne une valeur plus petite que celle prise par la fonction affine $\lambda \rightarrow f(x_k) + \omega_1 \lambda \nabla^T f(x_k) d_k$.

:/Users/USER/AppData/Local/Temp/graphics/swp0000_10.jpg

FIG 2.4 . Règle d'Armijo.

Règle d'Armijo

- ◆ Si $\varphi(\lambda) \leq \varphi(0) + \omega_1 \varphi'(0)\lambda$, alors λ convient.
- ◆ Si $\varphi(\lambda) > \varphi(0) + \omega_1 \varphi'(0)\lambda$, alors λ est trop grand.

On peut noter que l'on a

$$\varphi(\lambda) = f(x_k + \lambda d_k)$$

$$\varphi(0) = f(x_k),$$

$$\varphi'(0) = \nabla f(x_k)^T d_k.$$

Algorithme 2.1 (Règle d'Armijo)

Etape 0 : (initialisation)

$a_1 = b_1 = 0$, choisir $\lambda_1 > 0$, $\omega_1 \in]0, 1[$ poser $k = 1$ et aller à l'étape 1.

Etape 1 :

Si $\phi(\lambda_k) \leq \phi(0) + \omega_1 \phi'(0) \lambda_k$: STOP ($\lambda^* = \lambda_k$).

Si $\phi(\lambda_k) > \phi(0) + \omega_1 \phi'(0) \lambda_k$, alors

$b_{k+1} = b$, $a_{k+1} = \lambda_k$ et aller à l'étape 2.

Etape 2 :

Si $b_{k+1} = 0$ déterminer $\lambda_{k+1} \in] a_{k+1}, +\infty [$.

Si $b_{k+1} \neq 0$ déterminer $\lambda_{k+1} \in] a_{k+1}, b_{k+1}[$,

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

Remarque 2.1

1) En pratique, la constante ω_1 est prise très petite, de manière à satisfaire (2.6) le plus facilement possible. Typiquement, $\omega_1 = 10^{-4}$. Notons que cette constante ne doit pas être adaptée aux données du problème et donc que l'on ne se trouve pas devant un choix de valeur délicat.

2) Dans certains algorithmes, il est important de prendre $\omega_1 < \frac{1}{2}$ pour que le pas λ_k soit accepté lorsque x_k est proche d'une solution.

3) Il est clair d'après la figure (2.1) que l'inégalité (2.6) est toujours vérifiée si $\lambda_k > 0$ est suffisamment petit.

4) On a vu qu'il était dangereux d'accepter des pas trop petits, cela pouvait conduire à une fausse convergence. Il faut donc un mécanisme supplémentaire qui empêche le pas d'être trop petit. On utilise souvent la technique de rebroussement due à *Armijo* [9, 1966] ou celle de *Goldstein*.

La règle de Goldstein

La règle de *Goldstein* remédie à cet inconvénient (le pas λ_k doit être trop petit).

Dans celle-ci, en ajoutant une deuxième inégalité à la règle d'Armijo on obtient la règle de *Goldstein*.

$$f(x_k) + \omega_1 \lambda_k \nabla^T f(x_k) d_k \geq f(x_k + \lambda_k d_k) \geq f(x_k) + \omega_2 \lambda_k \nabla^T f(x_k) d_k, \quad (2.16)$$

où ω_1 et ω_2 sont deux constantes vérifiant $0 < \omega_1 < \omega_2 < 1$, cette inégalité qui empêche le pas d'être trop petit.

Règle de Goldstein

- ◆ Si $\varphi(\lambda) < \varphi(0) + \omega_2 \varphi'(0)\lambda$, alors λ est trop petit.
- ◆ Si $\varphi(\lambda) > \varphi(0) + \omega_1 \varphi'(0)\lambda$, alors λ est trop grand.
- ◆ Si $\varphi(0) + \omega_1 \varphi'(0)\lambda \geq \varphi(\lambda) \geq \varphi(0) + \omega_2 \varphi'(0)\lambda$, alors λ convient.

On peut noter que l'on a

$$\varphi(\lambda) = f(x_k + \lambda d_k)$$

$$\varphi(0) = f(x_k)$$

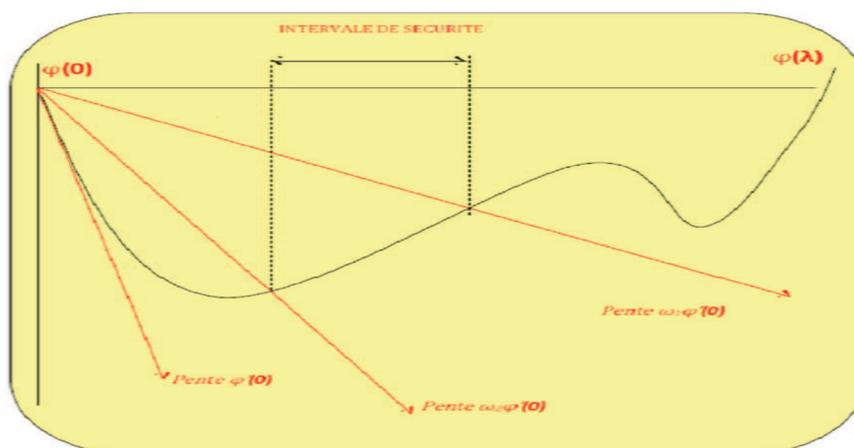


FIG 2.5. Règle de Goldstein

Algorithme 2.2 (Règle de Goldstein&Price)

Etape 0 : (initialisation)

$a_1 = b_1 = 0$, choisir $\lambda_1 > 0$, $\omega_1 \in]0, 1[$, $\omega_2 \in]\omega_1, 1[$, poser $k = 1$ et aller à l'étape 1.

Etape 1 :

Si $\phi(0) + \omega_2 \phi'(0)\lambda \leq \phi(\lambda_k) \leq \phi(0) + \omega_1 \phi'(0)\lambda_k$: STOP($\lambda^* = \lambda_k$).

Si $\phi(\lambda_k) > \phi(0) + \omega_1 \phi'(0)\lambda_k$, alors

$b_{k+1} = \lambda_k$, $a_{k+1} = a_k$, et aller à l'étape 2.

Si $\phi(\lambda_k) < \phi(0) + \omega_2 \phi'(0)\lambda_k$, alors

$b_{k+1} = b_k$, $a_{k+1} = \lambda_k$, et aller à l'étape 2.

Etape 2 :

- Si $b_{k+1} = 0$ déterminer $\lambda_{k+1} \in]b_{k+1}, +\infty[$
- Si $b_{k+1} \neq 0$ déterminer $\lambda_{k+1} \in]a_{k+1}, b_{k+1}[$.

Remarque 2.2

il n'est pas facile de voir comment on peut trouver un pas de *Goldstein* en un nombre fini d'étapes.

la règle de Wolfe :

La règle de *Wolfe* fait appel au calcul de $\varphi'(\lambda)$, elle est donc en théorie plus coûteuse que la règle de *Goldstein*. Cependant dans de nombreuses applications, le calcul du gradient $\nabla f(x)$ représente un faible coût additionnel en comparaison du coût d'évaluation de $f(x)$, c'est pourquoi cette règle est très utilisée.

Nous allons présenter les conditions de *Wolfe faibles* sur $\lambda > 0$:

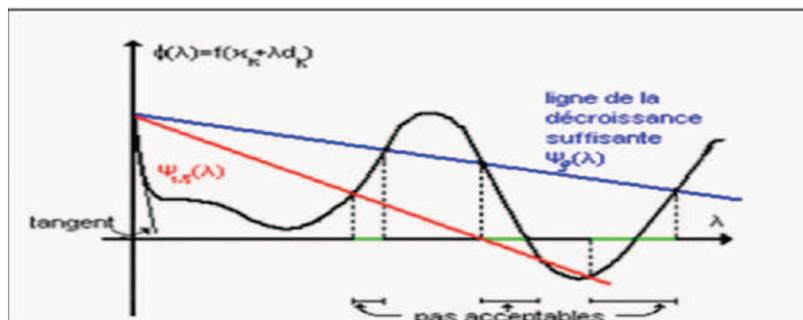


FIG 2.6. Règle de *Wolfe*..

Règle de wolf faible

$$f(x_k + \lambda d_k) \leq f(x_k) + \omega_1 \lambda \nabla f(x_k)^T \cdot d_k.$$

$$\nabla f(x_k + \lambda d_k)^T \cdot d_k \geq \omega_2 \nabla f(x_k)^T \cdot d_k.$$

avec $0 < \omega_1 < \omega_2 < 1$.

Pour expliquer ces conditions posons

$$l(\lambda) = \phi(0) + (\omega_2 \phi'(0))\lambda.$$

Règle de Wolfe faible

- ◆ Si $\phi(\lambda) \leq \phi(0) + \omega_1\phi'(0)\lambda$ et $\phi'(\lambda) \geq \omega_2\phi'(0)$, alors α convient.
- ◆ Si $\phi(\lambda) > \phi(0) + \omega_1\phi'(0)\lambda$, alors λ est trop grand.
- ◆ Si $\phi'(\lambda) < \omega_2\phi'(0)$, alors λ est trop petit.

On voit bien à la figure 2.3 ce que signifie cette condition.

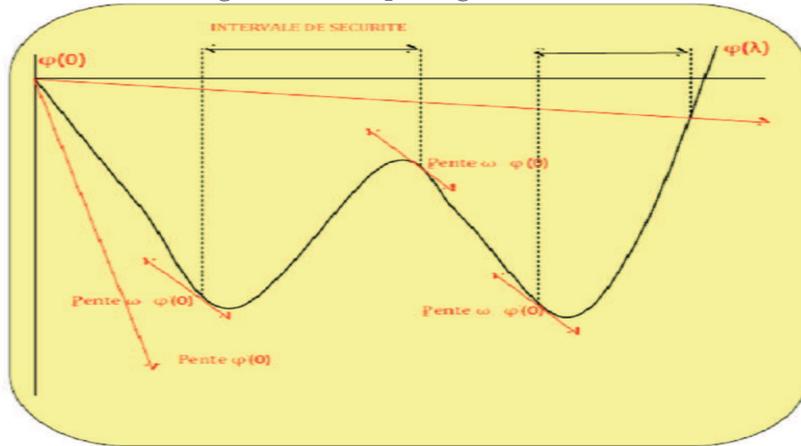


FIG 2.7. Règle de Wolfe faible..

On remarque que $\phi'(0) < 0$ car d_k est une direction de descente.

La condition (W1) impose que la réduction de f est proportionnelle à λ et $\phi'(0)$. On ne retiendra pas que les valeurs de λ pour lesquelles le graphe de ϕ est en dessous de la droite l , comme $0 < \omega_1 < 1$ ceci est possible au moins pour λ petit. La condition (W2) implique que $\phi'(\lambda) \geq \omega_2\phi'(0) \geq \phi'(0)$: si $\phi'(\lambda)$ est "très" négatif (i.e. $< \omega_2\phi'(0)$), on va chercher plus loin, sinon on peut s'arrêter. De cette façon le pas λ ne sera pas trop petit.

Algorithme 2.3 (Règle de Wolfe)

Etape 0 : (initialisation)

$a_1 = b_1 = 0$, choisir $\lambda_1 > 0$ $\omega_1 \in]0, 1[$, $\omega_2 \in]\omega_1, 1[$, poser $k = 1$ et aller à l'étape 1.

Etape 1 :

Si $\phi(\lambda_k) \leq \phi(0) + \omega_1\phi'(0)\lambda_k$ et $\phi'(\lambda) \geq \omega_2\phi'(0)$: STOP($\lambda^* = \lambda_k$).

Si $\phi(\lambda_k) > \phi(0) + \omega_1\phi'(0)\lambda_k$, alors

$b_{k+1} = \lambda_k$, $a_{k+1} = a_k$, et aller à l'étape 2.

Si $\phi'(\lambda) < \omega_2\phi'(0)$, alors

$b_{k+1} = b_k$, $a_{k+1} = \lambda_k$ et aller à l'étape 2.

Etape 2 :

Si $b_{k+1} = 0$ déterminer $\lambda_{k+1} \in]a_{k+1}, +\infty[$.

Si $b_{k+1} \neq 0$ déterminer $\lambda_{k+1} \in]a_{k+1}, b_{k+1}[$.

Règle de Wolfe Forte On obtient des contraintes plus fortes si l'on remplace (W2) par

$$|\nabla f(x_k + \lambda d_k)^T \cdot d_k| \leq -\omega_2 \nabla f(x_k)^T \cdot d_k$$

Les (W1) et (W3) sont les conditions de *Wolfe fortes*. La contrainte (W3) entraîne que $\omega_2 \phi'(0) \leq \phi'(\lambda) \leq -\omega_2 \phi'(0)$ c-à-d. $\phi'(\lambda)$ n'est pas "trop" positif.

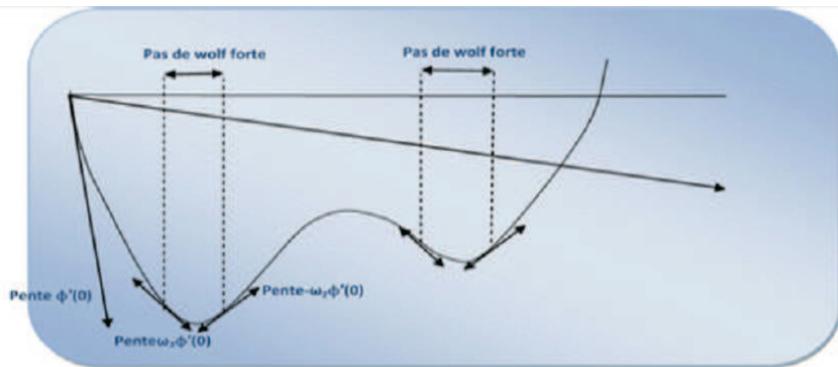


FIG 2.8. Règle de *Wolfe forte*.

Remarque 2.3.

1) On voit bien que les conditions de *Wolfe fortes* impliquent les conditions de *Wolfe faibles*, en effet :

$$\begin{aligned} |\nabla^T f(x_k + \lambda_k d_k) d_k| &\leq -\omega_2 \nabla^T f(x_k) d_k \\ \Leftrightarrow \omega_2 \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k \leq -\omega_2 \nabla^T f(x_k) d_k \\ \Rightarrow \omega_2 \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k. \end{aligned} \quad (W2)$$

2) L'existence de valeurs λ de vérifiant les conditions de *Wolfe faibles* et *fortes*. est donnée par la Proposition suivant.

Proposition 2.1.

[20]. Soit $f \in C^1$ et d une direction de descente de f en x , on suppose que f est minorée. Alors, si $0 < \omega_1 < \omega_2 < 1$, il existe des intervalles dans \mathbb{R}_+ qui vérifient les conditions de *Wolfe faibles* et *fortes*.

La règle de Wolfe relaxée Proposée par *Dai* et *Yuan* [1996], cette règle consiste à choisir le pas satisfaisant aux conditions :

$$\begin{aligned} f(x_k + \lambda d_k) &\leq f(x_k) + \omega_1 \lambda \nabla^T f(x_k) d_k \\ \omega_2 \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \alpha_k d_k) d_k \leq -\omega''_2 \nabla^T f(x_k) d_k, \end{aligned}$$

où $0 < \omega_1 < \omega_2 < 1$ et $\omega''_2 > 0$.

Remarque 2.4.

1) On voit bien que les conditions de *Wolfe relaxée* impliquent les conditions de *Wolfe fortes*. Effectivement (w4) est équivalente à (w1), tandis que pour le cas particulier $\omega_2 = \omega''_2 = \omega_2$, (w5) est équivalente à (w3). En effet :

$$\begin{aligned} \omega_2 \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k \leq -\omega''_2 \nabla^T f(x_k) d_k \\ \Rightarrow \omega_2 \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k \leq -\omega_2 \nabla^T f(x_k) d_k \\ \Rightarrow |\nabla^T f(x_k + \lambda_k d_k) d_k| &\leq -\omega_2 \nabla^T f(x_k) d_k. \end{aligned} \quad (w3)$$

2) Les conditions de *Wolfe relaxée* impliquent les conditions de *Wolfe faibles*. Effectivement (w4) est équivalente à (w1), tandis que pour le cas particulier $\omega_2 = \omega_2$ et $\omega''_2 = +\infty$, (w5) est équivalente à (w2). En effet :

$$\begin{aligned} \omega'_2 &= \nabla^T f(x_k) d_k \leq \nabla^T f(x_k + \lambda_k d_k) d_k \leq -\omega''_2 \nabla^T f(x_k) d_k \\ \Rightarrow \omega_2 \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k. \end{aligned} \quad (w2)$$

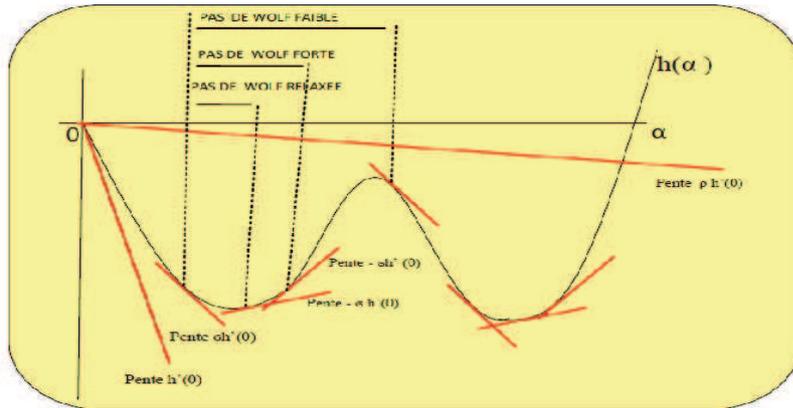


FIG 2.9 comparaison entre les trois règles de wolf

2.5 Convergence des méthodes à directions de descente

2.5.1 Condition de Zoutendijk

Dans cette section, on va étudier la *contribution* de la recherche linéaire inexacte à la convergence des algorithmes à directions de descente. Ce n'est pas qu'une contribution, parce que la recherche linéaire ne peut à elle seule assurer la convergence des itérés. On comprend bien que le choix de la direction de descente joue aussi un rôle. Cela se traduit par une condition, dite de *Zoutendijk*, donc on peut tirer quelques informations qualitatives intéressantes.

On dit qu'une règle de recherche linéaire satisfait la condition de Zoutendijk s'il existe une constante $C > 0$ telle que pour tout indice $k \geq 1$ on ait

$$f(x_{k+1}) \leq f(x_k) - C \|\nabla f(x_k)\|^2 \cos^2 \theta_k \quad (2.17)$$

où θ_k est l'angle que fait d_k avec $-\nabla f(x_k)$, défini par

$$\cos \theta_k = \frac{-\nabla^T f(x_k) d_k}{\|d_k\| \|d_k\|}$$

Voici comment on se sert de la condition de Zoutendijk.

Proposition 2.1. *Si la suite $\{x_k\}$ générée par un algorithme d'optimisation vérifie la condition de Zoutendijk (2.30) et si la suite $\{f(x_k)\}$ est minorée, alors*

$$\sum_{k \geq 1} \|\nabla f(x_k)\|^2 \cos^2 \theta_k < \infty \quad (2.18)$$

Démonstration. ([21])

En sommant les inégalités (2.17), on a

$$\sum_{k \geq 1} \|\nabla f(x_k)\|^2 \cos^2 \theta_k \leq \frac{1}{C} (f(x_1) - f(x_{l+1}))$$

Le série donc convergente puisqu'il existe une constant C' telle que pour tout k , $f(x_k) \geq C'$. \square

Les deux propositions suivantes précisent les circonstances dans lesquelles la condition de Zoutendijk (2.30) est vérifiée avec les règles d'Armijo et de Wolfe.

Proposition 2.2. *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continuellement différentiable dans un voisinage de $\Gamma = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$.*

CHAPITRE 2. MÉTHODES À DIRECTIONS DE DESCENTE ET
RECHERCHES LINÉAIRES

On considère un algorithme à directions de descente d_k , qui génère une suite $\{x_k\}$ en utilisant la recherche linéaire d'Armijo(2.17) avec $\alpha_1 > 0$.

Alors il existe une constante $C > 0$ telle que, pour tout $k \geq 1$, l'une des conditions

$$f(x_{k+1}) \leq f(x_k) - C \nabla^T f(x_k) d_k \quad (2.19)$$

ou

$$f(x_{k+1}) \leq f(x_k) - C \|\nabla f(x_k)\|^2 \cos^2 \theta_k \quad (2.20)$$

vérifiée.

Démonstration. ([21])

Si le pas $\alpha_k = \alpha_1$ est accepté, on a (2.30), car α_1 est uniformément positif.

Dans le cas contraire, (2.17) n'est pas vérifiée avec un pas $\alpha'_k \leq \frac{\alpha_k}{\tau}$, c'est-à-dire $f(x_k + \alpha'_k d_k) > f(x_k) + \rho \alpha'_k \nabla^T f(x_k) d_k$

Comme f est continuellement différentiable, on a pour tout $\alpha_k > 0$:

$$\begin{aligned} f(x_k + \alpha_k d_k) &= f(x_k) + \alpha_k \nabla^T f(x_k) d_k + \int_0^1 [\nabla f(x_k + t \alpha_k d_k) - \nabla f(x_k)]^T \alpha_k d_k dt \\ &\Rightarrow f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \nabla^T f(x_k) d_k + C \alpha_k^2 \|d_k\|^2 \end{aligned}$$

où $C > 0$ est une constante. Avec l'inégalité précédente, et le fait que , on obtient :

$$\begin{cases} f(x_k + \alpha'_k d_k) - f(x_k) > \rho \alpha'_k \nabla^T f(x_k) d_k \\ f(x_k + \alpha'_k d_k) - f(x_k) \leq \alpha'_k \nabla^T f(x_k) d_k + C \alpha_k'^2 \|d_k\|^2 \end{cases}$$

$$\begin{aligned} &\Rightarrow \rho \alpha'_k \nabla^T f(x_k) d_k \leq \alpha_k \nabla^T f(x_k) d_k + C \alpha_k'^2 \|d_k\|^2 \\ &\Rightarrow -C \alpha_k'^2 \|d_k\|^2 \leq (1 - \rho) \alpha'_k \nabla^T f(x_k) d_k \end{aligned}$$

or

$$\rho < 1 \Rightarrow 0 < 1 - \rho < 1 \Rightarrow \frac{1}{1 - \rho} > 1$$

d'où

$$\begin{aligned} \nabla^T f(x_k) d_k &\geq \frac{-C}{1 - \rho} \alpha'_k \|d_k\|^2 \Rightarrow -\nabla^T f(x_k) d_k \leq \frac{C}{1 - \rho} \alpha'_k \|d_k\|^2 \\ &\Rightarrow |\nabla^T f(x_k) d_k| = \|\nabla^T f(x_k)\| \|d_k\| \cos \theta_k \leq \frac{C}{1 - \rho} \alpha'_k \|d_k\|^2 \end{aligned}$$

ce qui permet de minorer $\alpha' \|d_k\|$ et donc aussi $\alpha \|d_k\|$ par une constante fois $\|\nabla^T f(x_k)\| \|d_k\|$. Cette minoration et l'expression suivante de (2.17)

$$f(x_k + \alpha_k d_k) \leq f(x_k) - \rho \alpha_k \|\nabla^T f(x_k)\| \|d_k\| \cos \theta_k$$

conduit à(2.30). \square

Proposition 2.3. *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction contiuement différentiable dans un voisinage de $\Gamma = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$.*

On considère un algorithme à directions de descente d_k , qui génère une suite $\{x_k\}$ en utilisant la recherche linéaire de Wolfe (2.22)-(2.23).

Alors il existe une constante $C > 0$ telle que, pour tout $k \geq 1$, la condition de Zoutendijk (2.30) est vérifiée.

Démonstration. ([21])

D'après (2.22)

$$\nabla^T f(x_k + \alpha_k d_k) d_k \geq \sigma \nabla^T f(x_k) d_k$$

$$\begin{aligned} \Rightarrow & (\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k))^T d_k \geq (\sigma - 1) \nabla^T f(x_k) d_k \\ = & -(1 - \sigma) \nabla^T f(x_k) d_k = (1 - \sigma) |\nabla^T f(x_k) d_k| \\ \Leftrightarrow & (1 - \sigma) |\nabla^T f(x_k) d_k| \leq (\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k))^T d_k \end{aligned}$$

et du fait que f est contiuement différentiable :

$$\begin{aligned} (1 - \sigma) |\nabla^T f(x_k) d_k| &= (1 - \sigma) \|\nabla^T f(x_k)\| \|d_k\| \cos \theta_k \\ &\leq \|\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)\| \|d_k\| \\ \Rightarrow & (1 - \sigma) \|\nabla^T f(x_k)\| \cos \theta_k \leq L \alpha_k \|d_k\| \\ \Rightarrow & \alpha_k \|d_k\| \leq \frac{(1 - \sigma)}{L} \|\nabla^T f(x_k)\| \cos \theta_k \end{aligned}$$

en utilisant (2.22), on aura :

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \rho \alpha_k \nabla^T f(x_k) d_k \\ \Rightarrow & f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha \nabla^T f(x_k) d_k \leq f(x_k) + |\rho \alpha \nabla^T f(x_k) d_k| \\ \Rightarrow & f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha |\nabla^T f(x_k) d_k| \leq f(x_k) - \rho \alpha \nabla^T f(x_k) d_k \\ \Rightarrow & f(x_k + \alpha d_k) \leq f(x_k) - \rho \alpha \|\nabla^T f(x_k)\| \|d_k\| \cos \theta_k \\ \Rightarrow & f(x_k + \alpha d_k) \leq f(x_k) - \frac{\rho(1 - \sigma)}{L} \|\nabla^T f(x_k)\|^2 \cos^2 \theta_k \end{aligned}$$

On en déduit (2.30). \square

2.6 Méthodes itératives d'optimisation sans contraintes

A travers de ce chapitre et des éléments suivants, on s'attachera dorénavant à la description plus spécifique des algorithmes itératifs (ou méthodes

itératives) qui ont été implémentés et qui permettent la résolution des problèmes d'optimisation non linéaire. Il convient de souligner que la plupart des algorithmes d'optimisation, contrainte ou non, fonctionnent selon un schéma général consistant, à chaque itération, à se rapprocher du minimum par la résolution d'un sous-problème de minimisation.

Nous considérons ici les méthodes permettant de résoudre un problème d'optimisation sans contraintes (appelées aussi parfois méthodes d'optimisation directe), soit le problème

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

pour lequel nous commencerons par décrire les méthodes suivantes :

- ▶ Les méthodes de gradient
- ▶ Les méthodes de Newton
- ▶ Les méthodes utilisant des directions conjuguées.

Ces méthodes utilisent des dérivées (et donc la propriété de différentiabilité de f) à l'exception des méthodes de directions conjuguées (sauf dans le cas particulier de la méthode du gradient conjugué) basée, elle, sur des propriétés plus géométriques.

Parmi les plus anciennes méthodes utilisées pour résoudre les problèmes du type (P), on peut citer la méthode du Gradient conjugué. Cette méthode est surtout utilisée pour les problèmes de grande taille.

Après les avoir décrites, nous donnons la définition d'une forme quadratique et *Principe* des méthodes de descente et de gradient.

Définition 2.2.

Soit H une matrice symétrique $n \times n$ et $b \in \mathbb{R}^n$. On appelle forme quadratique la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ définie par

$$f(x) = \frac{1}{2}x^T H(x)x - b^T x. \quad (2.21)$$

Lorsque la matrice H est définie positive (resp. semi-définie positive), on dira que $f(x)$ est une forme quadratique définie positive (resp. semi-définie positive).

2.6.1 Principe des méthodes de descente

Le principe d'une méthode de descente consiste à faire les itérations suivantes

$$x_{k+1} = x_k + \lambda_k d_k \quad \lambda_k > 0,$$

tout en assurant la propriété

$$f(x_{k+1}) < f(x_k). \quad (2.22)$$

Le vecteur d_k est la direction de descente en x_k . Le scalaire λ_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.

2.6.2 Principe des méthodes de gradient (méthodes de la plus forte descente)

On cherche à déterminer la direction de descente qui fait décroître $\phi(\lambda) = f(x + \lambda d)$ le plus vite possible (au moins localement). Pour cela on va essayer de minimiser la dérivée de $\phi(\lambda)$ en 0. On a $\phi'(0) = \nabla f(x)^T d$, et on cherche d solution du problème

$$\min_{d \in \mathbb{R}^n, \|d\|=1} \phi'(0).$$

La solution est bien sûr

$$d = -\frac{\nabla f(x)}{\|\nabla f(x)\|}.$$

En vertu de l'inégalité de Schwartz. Il y a ensuite de nombreuses façon d'utiliser cette direction de descente. On peut par exemple utiliser un pas fixé a priori $\lambda_K = \alpha > 0, \forall k$. On obtient alors la méthode du gradient simple :

$$\begin{cases} d_k = -\nabla f(x_k), \\ x_{k+1} = x_k + \lambda d_k. \end{cases} \quad (2.23)$$

Sous certaines hypothèses de régularité (f deux fois différentiable) cette méthode converge si λ est choisi assez petit.

Où bien consiste à faire les itérations suivantes

$$\begin{cases} d_k = -\nabla f(x_k), \\ x_{k+1} = x_k + \lambda_k d_k. \end{cases} \quad (2.24)$$

Où λ_k est choisi de manière à ce que

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k), \forall \lambda > 0. \quad (2.25)$$

On obtient alors la méthode du gradient à pas optimal, cette méthode possède une propriété intéressante :

Proposition 2.4.

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Les directions de descente d_k générées par la méthode (2.15) et (2.16) vérifient

$$d_{k+1}^T d_k = 0. \quad (2.26)$$

Démonstration :

Si on introduit la fonction

$$\phi(\lambda) = f(x_k + \lambda d_k),$$

on a

$$\phi'(\lambda) = \nabla f(x_k + \lambda d_k)^T d_k,$$

et puisque ϕ est dérivable on a nécessairement

$$\phi'(\lambda) = 0,$$

donc

$$\nabla f(x_k + \lambda_k d_k)^T d_k = \nabla f(x_{k+1})^T d_k = -d_{k+1}^T d_k = 0.$$

Exemple 2.1.

► Calcul du pas optimal dans le cas quadratique :

On a $f(x) = \frac{1}{2}x^T Hx - b^T x$ avec $H > 0$ et on note $\phi(\lambda) = f(x_k + \lambda d_k)$.

Le pas optimal λ_k est caractérisé par

$$\phi'(\lambda) = 0,$$

on a donc

$$\nabla f(x_k + \lambda_k d_k)^T d_k = (H(x_k + \lambda_k d_k) - b)^T d_k = 0,$$

soit

$$(\nabla f(x_k) + \lambda_k H d_k)^T d_k = 0,$$

on obtient donc

$$\lambda_k = -\frac{\nabla f(x_k)^T d_k}{d_k^T H d_k}, \quad (2.27)$$

qui est bien positif car d_k est une direction de descente et

$$d_k^T H d_k > 0 \text{ (car } H > 0 \text{)}.$$

La méthode du gradient à pas optimal peut donc s'écrire (dans le cas quadratique)

$$\begin{aligned}d_k &= b - Hx_k, \\ \lambda_k &= \frac{d_k^T d_k}{d_k^T H d_k}, \\ x_{k+1} &= x_k + \lambda_k d_k.\end{aligned}$$

Exemple 2.2.

► Méthode du gradient simple dans le cas quadratique

Dans le cas où $f(x) = \frac{1}{2}x^T H(x)x - b^T x$ la méthode du gradient simple peut s'écrire

$$\begin{aligned}d_k &= b - Hx_k, \\ x_{k+1} &= x_k + \lambda d_k,\end{aligned}$$

où $\lambda > 0$ est fixé a priori. Il existe bien sûr des conditions sur pour que la méthode converge. Nous illustrons ici le fonctionnement de la méthode dans le cas $n = 2$ sur une petite simulation.

► Méthode du gradient à pas optimal dans le cas quadratique

Dans le cas où $f(x) = \frac{1}{2}x^T H(x)x - b^T x$ la méthode du gradient à pas optimal peut s'écrire :

$$\begin{aligned}d_k &= b - Hx_k, \\ \lambda_k &= \frac{d_k^T d_k}{d_k^T H d_k}, \\ x_{k+1} &= x_k + \lambda_k d_k.\end{aligned}$$

Nous illustrons ici le fonctionnement de la méthode dans le cas $n = 2$ sur une petite simulation

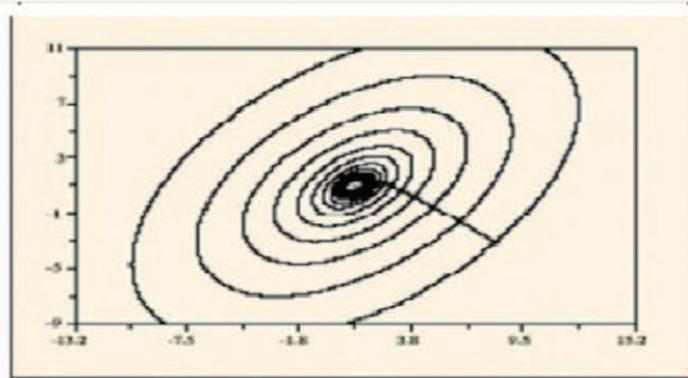


FIG 2.10 Illustration de la convergence plus rapide de la méthode de gradient

Algorithme de la méthode de la plus forte pente

Etape initiale :

- Choisir un $\varepsilon > 0$.
- Choisir un point initiale x_1 .
- Poser $k = 1$ et aller à l'étape principale.

Etape principale :

- Si $\|\nabla f(x)\| < \varepsilon$ stop.
- Sinon poser $d_k = -\nabla f(x_k)$ et soit λ_k la solution optimale de la recherche linéaire

$$\min \{f(x_k + \lambda d_k); \lambda > 0\}.$$

- Poser $x_{k+1} = x_k + \lambda_k d_k$.
- Remplacer k par $k + 1$ et répéter l'étape principale.

Inconvénients de la méthode de la plus forte pente

- Lenteur de la méthode au voisinage des points stationnaires.
- Cette méthode travaille de façon performante dans les premières étapes de l'algorithme. Malheureusement, dès qu'on s'approche du point stationnaire, La méthode devient très lente. On peut expliquer intuitivement ce phénomène par les considérations suivantes

$$f(x_k + \lambda d_k) = f(x_k) + \lambda \nabla f(x_k)^T d + \lambda \|d\| \alpha(x_k; \lambda d),$$

où $\alpha(x_k; \lambda d) \rightarrow 0$ quand $\lambda d \rightarrow 0$.

Si $d_k = -\nabla f(x_k)$, on obtient : $x_{k+1} = x_k - \lambda \nabla f(x_k)$ et par conséquent

$$f(x_{k+1}) - f(x_k) = \lambda \left[-\|\nabla f(x_k)\|^2 + \|\nabla f(x_k)\| \alpha(x_k; \lambda \nabla f(x_k)) \right].$$

D'après l'expression précédente, on voit que lorsque x_k s'approche d'un point stationnaire, et si f est continument différentiable, alors $\|\nabla f(x_k)\|$ est proche de zéro. Donc le terme à droite s'approche de zéro, indépendemment de λ , et par conséquent $f(x_{k+1})$ ne s'éloigne pas beaucoup de $f(x_k)$ quand on passe du point x_k au po

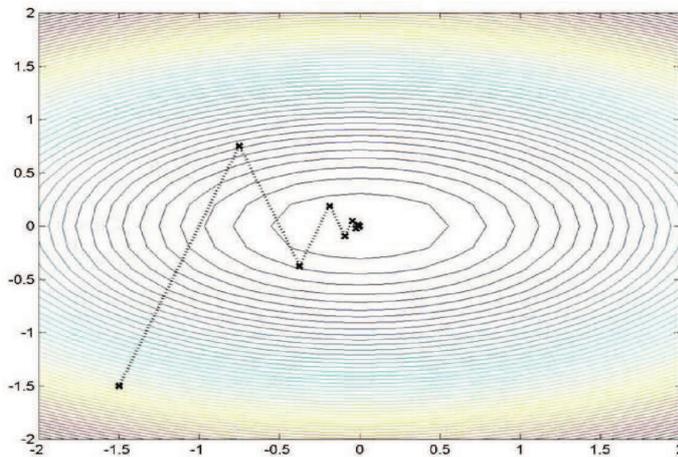


FIG 2.11 Exprime les directions de la méthode de la plus fort pente

2.6.3 Les méthodes utilisant des directions conjuguées

Description de la méthode

Ces méthode sont basées sur l'important concept de la conjugaison et ont été développées afin de résoudre le problème quadratique

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^T \cdot Q \cdot x + b^T \cdot x + c.$$

Où $Q \in \mathbb{R}^{n \times n}$ est symétrique et définie positive, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$. Les méthodes de direction conjuguées peuvent résoudre les problèmes de cette forme en au plus n itérations et, contrairement au méthodes présentées jusqu'à présent, elle n'utilisent pas de dérivées, sauf dans le cas particulier de la méthode du gradient conjugué. Donnons la définition de la notion de "conjugaison" :

Définition 2.3.

CHAPITRE 2. MÉTHODES À DIRECTIONS DE DESCENTE ET RECHERCHES LINÉAIRES

Soient Q une matrice $n \times n$ symétrique et définie positive et un ensemble de directions non nulles $\{d_1, d_2, \dots, d_k\}$. Ces directions sont dites Q -conjuguées si

$$d_i^T Q d_j = 0, \quad \forall i, j \text{ tels que } i \neq j. \quad (2.28)$$

Propriété 2.1.

Si d_1, \dots, d_k sont Q -conjuguées, alors elles sont linéairement indépendantes. ■

Propriété 2.2.

Comme des direction Q -conjuguées sont linéairement indépendantes, alors l'espace vectoriel engendré par un ensemble de n directions Q -conjuguées est de dimension n .

Etant donné un ensemble de n directions Q -conjuguées d_0, d_1, \dots, d_{n-1} , la méthode de directions conjuguées correspondante est donnée par

$$x_{k+1} = x_k + \lambda_k d_k, \quad k = 0, \dots, n-1,$$

où x_0 est un vecteur de départ choisi arbitrairement et où les λ_k sont obtenus par minimisation monodimensionnelle le long de d_k .

Le principal résultat concernant les méthodes utilisant des directions conjuguées est qu'à chaque itération k , la méthode minimise f sur le sous-espace généré par les k premières directions Q -conjuguées utilisées par l'algorithme. A la *nième* itération au plus tard, ce sous-espace inclura alors le minimum global de f , grâce à la propriété d'indépendance linéaire des direction Q -conjuguées qui assure qu'à l'itération n , l'espace vectoriel généré par les n directions Q -conjuguées ne sera autre que \mathbb{R}^n .

Remarque 2.5.

la notion de conjugaison n'a pas de sens dans le cas non quadratique.

Chapitre 3

Convergence des méthodes du Gradient Conjugué utilisant des recherches linéaires de Wolfe fortes.

Les méthodes du gradient conjugué sont utilisées pour résoudre les problèmes d'optimisation non linéaires sans contraintes de grandes tailles. On l'utilise aussi pour résoudre les grands systèmes linéaires.

Elles reposent sur le concept des directions conjuguées parce que les gradients successifs sont orthogonaux entre eux et aux directions précédentes.

L'idée initiale était de trouver une suite de directions de descente permettant de résoudre le problème

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

Où f est régulière (continûment différentiable)

Dans ce chapitre on va décrire toutes ces méthodes, mais avant d'accéder à ces derniers, on va d'abord donner le principe général d'une méthode à directions conjuguées

3.1 Le principe général d'une méthode à directions conjuguées

Donnons la définition de "conjugaison" :

Définition 3.1. Soit A une matrice symétrique $n \times n$, définie positive. On dit que deux vecteurs x et y de \mathbb{R}^n sont A -conjugués (ou conjugués par

rapport à A) s'ils vérifient

$$x^T Ay = 0 \quad (3.1)$$

3.1.1 Description de la méthode

Soit $\{d_0, d_1, \dots, d_n\}$ une famille de vecteurs A -conjugués. On appelle alors méthode de directions conjuguées toute méthode itérative appliquée à une fonction quadratique strictement convexe de n variables : $q(x) = \frac{1}{2}x^T Ax + b^T x + c$; avec $x \in \mathbb{R}^n$ et $A \in \mathcal{M}_{n \times n}$ est symétrique et définie positive, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$. conduisent à l'optimum en n étapes au plus. Et cette méthode de la forme :

$$\begin{aligned} x_0 & \text{ donné} \\ x_{k+1} & = x_k + \alpha_k d_k \end{aligned} \quad (3.2)$$

où α_k est optimal et d_1, d_2, \dots, d_n possédant la propriété d'être mutuellement conjuguées par rapport à la fonction quadratique

Si l'on note $g_k = \nabla q(x_k)$, la méthode se construit comme suit :

Calcul de α_k

Comme α_k minimise q dans la direction d_k , on a, $\forall k$:

$$q'(\alpha_k) = d_k^T \nabla q(x_{k+1}) = 0$$

$$d_k^T \nabla q(x_{k+1}) = d_k^T (Ax_{k+1} + b) = 0.$$

Soit :

$$d_k^T A(x_k + \alpha_k d_k) + d_k^T b = 0$$

d'où l'on tire :

$$\alpha_k = \frac{-d_k^T (Ax_k + b)}{d_k^T A d_k} \quad (3.3)$$

Comment construire les directions A -conjuguées ?

Des directions A -conjuguées d_0, \dots, d_k peuvent être générées à partir d'un ensemble de vecteurs linéairement indépendants ξ_0, \dots, ξ_k en utilisant la procédure dite de Gram-Schmidt, de telle sorte que pour tout i entre 0 et k , le sous-espace généré par d_0, \dots, d_i soit égale au sous-espace généré par ξ_0, \dots, ξ_i .

Alors d_{i+1} est construite comme suit :

$$d_{i+1} = \xi_{i+1} + \sum_{m=0}^i \varphi_{(i+1)m} d_m$$

Nous pouvons noter que si d_{i+1} est construite d'une telle manière, elle est effectivement linéairement indépendante avec d_0, \dots, d_i .

En effet, le sous-espace généré par les directions d_0, \dots, d_i est le même que le sous-espace généré par les directions ξ_0, \dots, ξ_i , et ξ_{i+1} est linéairement indépendant de ξ_0, \dots, ξ_i .

ξ_{i+1} ne fait donc pas partie du sous-espace généré par les combinaisons linéaires de la forme $\sum_{m=0}^i \varphi_{(i+1)m} d_m$, de sorte que d_{i+1} n'en fait

pas partie non plus et est donc linéairement indépendante des d_0, \dots, d_i .

Les coefficients $\varphi_{(i+1)m}$, eux sont choisis de manière à assurer la A -conjugaison des d_0, \dots, d_{i+1} .

3.2 La méthode du gradient conjugué

La méthode du gradient conjugué est obtenue en appliquant la procédure de Gram-Schmidt aux gradients $\nabla q(x_0), \dots, \nabla q(x_{n-1})$, c'est-à-dire en posant $\xi_0 = -\nabla q(x_0), \dots, \xi_{n-1} = -\nabla q(x_{n-1})$.

En outre, nous avons que

$$\begin{aligned} \nabla q(x) &= Ax + b \\ \text{et } \nabla^2 q(x) &= A. \end{aligned}$$

Notons que la méthode se termine si $\nabla q(x_k) = 0$.

La particularité intéressante de la méthode du gradient conjugué est que le membre de droite de l'équation donnant la valeur de d_{k+1} dans la procédure de Gram-Schmidt peut être grandement simplifié.

Notons que la méthode du gradient conjugué est inspirée de celle du gradient (plus profonde pente).

3.3 Algorithme de la méthode du gradient conjugué

3.3.1 Algorithme de La méthode du gradient conjugué pour les fonctions quadratiques (cas linéaire)

On suppose ici que la fonction à minimiser est quadratique sous la forme :
 $q(x) = \frac{1}{2}x^T Ax + b^T x + c$

Si l'on note $g_k = \nabla f(x_k)$, l'algorithme prend la forme suivante

Cet algorithme consiste à générer une suite d'itérés $\{x_k\}$ sous la forme :

$$x_{k+1} = x_k + \alpha_k d_k \quad (3.4)$$

L'idée de la méthode est :

1- construire itérativement des directions d_0, \dots, d_k mutuellement conjuguées

A chaque étape k la direction d_k est obtenue comme combinaison linéaire du gradient en x_k et de la direction précédente d_{k-1}
 c'est-à-dire

$$d_{k+1} = -\nabla q(x_{k+1}) + \beta_{k+1} d_k \quad (3.5)$$

les coefficients β_{k+1} étant choisis de telle manière que d_k soit conjuguée avec toutes les directions précédentes. autrement dit :

$$d_{k+1}^T Ad_k = 0,$$

on en déduit :

$$\begin{aligned} d_{k+1}^T Ad_k &= 0 \Rightarrow (-\nabla q(x_{k+1}) + \beta_{k+1} d_k)^T Ad_k = 0 \\ &\Rightarrow -\nabla^T q(x_{k+1}) Ad_k + \beta_{k+1} d_k^T Ad_k = 0 \\ &\Rightarrow \beta_{k+1} = \frac{\nabla^T q(x_{k+1}) Ad_k}{d_k^T Ad_k} = \frac{g_{k+1}^T Ad_k}{d_k^T Ad_k} \end{aligned} \quad (3.6)$$

2-déterminer le pas α_k :

en particulier, une façon de choisir α_k peut être de résoudre le problème d'optimisation (à une seule variable)

$$\alpha_k = \min f(x_k + \alpha d_k) \quad , \quad \alpha > 0 \quad (3.7)$$

on en déduit :

$$\alpha_k = \frac{-d_k^T g_k}{d_k^T Ad_k} = -\frac{1}{Ad_k} g_k \frac{d_k^T}{d_k^T} = \frac{-d_k^T g_k}{d_k^T Ad_k} \quad (3.8)$$

Le pas α_k obtenu ainsi s'appelle le pas optimal.

Algorithme 3.1 (Algorithme du gradient conjugué "linéaire")

Etape 0 : (initialisation)

Soit x_0 le point de départ, $g_0 = \nabla q(x_0) = Ax_0 + b$, poser $d_0 = -g_0$
poser $k = 0$ et aller à l'étape 1.

Etape 1 :

si $g_k = 0$: STOP ($x^* = x_k$). "Test d'arrêt"
si non aller à l'étape 2.

Etape 2 :

Définir $x_{k+1} = x_k + \alpha_k d_k$ avec :

$$\alpha_k = \frac{-d_k^T g_k}{d_k^T Ad_k}$$

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$$

$$\beta_{k+1} = \frac{g_{k+1}^T Ad_k}{d_k^T Ad_k}$$

Poser $k = k + 1$ et aller à l'étape 1. □

La validité de l'algorithme du gradient conjugué linéaire

On va maintenant montrer que l'algorithme ci-dessus définit bien une méthode de directions conjuguées.

Théorème 3.1. *A une itération k quelconque de l'algorithme où l'optimum de $q(x)$ n'est pas encore atteint (c'est-à-dire $g_i \neq 0$, $i = 0, 1, \dots, k$) on a :*

a)

$$\alpha_k = \frac{g_k^T g_k}{d_k^T Ad_k} \neq 0 \quad (3.9)$$

b)

$$\beta_{k+1} = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k} \quad (3.10)$$

$$= \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \quad (3.11)$$

CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.

c) Les directions d_0, d_1, \dots, d_{k+1} engendrées par l'algorithme sont mutuellement conjuguées.

Démonstration. ([35])

On raisonne par récurrence sur k en supposant que d_0, d_1, \dots, d_k sont mutuellement conjuguées.

a) Montrons d'abord l'équivalence de (3.8) et de (3.9)

On a : $d_k = -g_k + \beta_k d_{k-1}$.

Donc (3.8) s'écrit :

$$\begin{aligned} \alpha_k &= \frac{-d_k^T g_k}{d_k^T A d_k} \\ &= \frac{-[-g_k + \beta_k d_{k-1}]^T g_k}{d_k^T A d_k} \\ &= \frac{g_k^T g_k}{d_k^T A d_k} - \beta_k \frac{d_{k-1}^T g_k}{d_k^T A d_k} \end{aligned}$$

Comme $(d_0, d_1, \dots, d_{k-1})$ sont mutuellement conjuguées, x_k est l'optimum de $q(x)$ sur la variété ν^k passant par x_0 et engendrée par $(d_0, d_1, \dots, d_{k-1})$ (Théorème 3.1).

Donc $d_{k-1}^T g_k = 0$ d'où l'on déduit (3.9).

b) Pour démontrer (3.10) remarquons que :

$$\begin{aligned} g_{k+1} - g_k &= A(x_{k+1} - x_k) = \alpha_k A d_k \\ \Rightarrow A d_k &= \frac{1}{\alpha_k} [g_{k+1} - g_k] \end{aligned}$$

On a alors :

$$g_{k+1}^T A d_k = \frac{1}{\alpha_k} g_{k+1}^T [g_{k+1} - g_k]$$

et en utilisant (3.9)

$$\alpha_k = \frac{g_k^T g_k}{d_k^T A d_k}$$

il vient

$$\begin{aligned} g_{k+1}^T A d_k &= \frac{d_k^T A d_k}{g_k^T g_k} \cdot g_{k+1}^T [g_{k+1} - g_k] \\ \Rightarrow \frac{g_{k+1}^T A d_k}{d_k^T A d_k} &= \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k} \end{aligned}$$

Or de (3.6) on aura :

$$\beta_{k+1} = \frac{g_{k+1}^T A d_k}{d_k^T A d_k} = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k}$$

CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.

ce qui démontre (3.10).

(3.11) découle alors du fait que :

$$g_{k+1}^T g_k = 0$$

car

$$g_k = d_k - \beta_k d_{k-1}$$

Appartient au sous-espace engendré par (d_0, d_1, \dots, d_k) et que g_{k+1} est orthogonal à ce sous-espace .

c) Montrons enfin que d_{k+1} est conjuguée par rapport à (d_0, d_1, \dots, d_k) .

On a bien $d_{k+1}^T Ad_k$ car, en utilisant $d_{k+1} = -g_{k+1} + \beta_k d_k$ on aura :

$$\begin{aligned} d_{k+1}^T Ad_k &= (-g_{k+1} + \beta_{k+1} d_k)^T Ad_k \\ &= -g_{k+1}^T Ad_k + \beta_{k+1} d_k^T Ad_k \\ &= -g_{k+1}^T Ad_k + \frac{g_{k+1}^T Ad_k}{d_k^T Ad_k} d_k^T Ad_k \\ &= 0 \end{aligned}$$

Vérifions maintenant que :

$$d_{k+1}^T Ad_i = 0 \quad \text{pour } i = 0, 1, \dots, k-1.$$

On a :

$$d_{k+1}^T Ad_i = -g_{k+1}^T Ad_i + \beta_{k+1} d_k^T Ad_i$$

Le seconde terme est nul par l'hypothèse de récurrence ((d_0, d_1, \dots, d_k) sont mutuellement conjuguées).

Montrons qu'il en est de même du premier terme. Puisque $x_{i+1} = x_i + \alpha_i d_i$ et que $\alpha_i \neq 0$ on a :

$$\begin{aligned} Ad_i &= \frac{1}{\alpha_i} (Ax_{i+1} - Ax_i) \\ &= \frac{1}{\alpha_i} (g_{i+1} - g_i) \end{aligned}$$

En écrivant :

$$\begin{aligned} g_{i+1} &= d_{i+1} - \beta_i d_i \\ g_i &= d_i - \beta_{i-1} d_{i-1} \end{aligned}$$

on voit que Ad_i est combinaison linéaire de d_{i+1}, d_i et de d_{i-1} seulement).

Mais puisque (d_0, d_1, \dots, d_k) sont mutuellement conjuguées, on sait que le point x_{k+1} est l'optimum de $q(x)$ sur la variété ν^{k+1} , engendrée par (d_0, d_1, \dots, d_k) .

Donc g_{k+1} est orthogonal au sous-espace engendré par (d_0, d_1, \dots, d_k) et comme Ad_i appartient à ce sous-espace pour $i = 0, 1, \dots, k - 1$, on en déduit $g_{k+1}^T Ad_i = 0$ ce qui achève la démonstration. \square

Remarque 3.1.

dans ce cas d_k est une direction de descente puisque

$$\begin{aligned} d_k^T \nabla q(x_k) &= (-\nabla q(x_k) + \beta_{k+1} d_{k-1})^T \nabla q(x_k) \\ &= -\nabla q(x_k)^T \nabla q(x_k) + \beta_{k+1} d_{k-1}^T \nabla q(x_k) \\ &= -\|\nabla q(x_k)\|^2 \quad (\text{car } d_{k-1}^T \nabla q(x_k) = 0) \\ d_k^T \nabla q(x_k) &= -\|\nabla q(x_k)\|^2 < 0 \end{aligned}$$

Les avantages de la méthode du gradient conjugué linéaire

1- la consommation mémoire de l'algorithme est minimale : on doit stocker les quatre vecteurs x_k, g_k, d_k, Ad_k

(bien sur x_{k+1} prend la place de x_k au niveau de son calcul avec des remarques analogues pour $g_{k+1}, d_{k+1}, Ad_{k+1}$) et les scalaires α_k, β_{k+1} .

2- L'algorithme du gradient conjugué linéaire est surtout utile pour résoudre des grands systèmes creux, en effet il suffit de savoir appliquer la matrice A à un vecteur.

3- La convergence peut être assez rapide : si A admet seulement r ($r < n$) valeurs propres distincts la convergence a lieu en au plus r itération.

Différentes formules de β_{k+1} dans le cas linéaire

Formule de Hestenes-Stiefel Cette méthode été découverte en 1952 par Hestenes et Stiefels ([28, 1952]),

$$\beta_{k+1}^{HS} = \frac{g_{k+1}^T [g_{k+1} - g_k]}{d_k^T [g_{k+1} - g_k]} \tag{3.12}$$

Formule de Polak-Ribière-Polyak Cette méthode été découverte en 1969 par Polak, Ribière ([38, 1969]) et Ployak ([39, 1969]),

$$\beta_{k+1}^{PRP} = \frac{g_{k+1}^T [g_{k+1} - g_k]}{\|g_k\|^2} \tag{3.13}$$

Formule de Fletcher-Reeves Cette méthode a été découverte en 1964 par Fletcher et Reeves ([19, 1964]),

$$\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (3.14)$$

Formule de la descente conjuguée Cette méthode a été découverte en 1987 par Fletcher ([17, 1987]),

$$\beta_{k+1}^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k} \quad (3.15)$$

Remarque 3.2. Dans le cas quadratique on a vu que :

$$\beta_{k+1}^{HS} = \beta_{k+1}^{PRP} = \beta_{k+1}^{FR} = \beta_{k+1}^{CD} = \beta_{k+1}^{DY}.$$

Dans le cas non quadratique, ces quantités ont en général des valeurs différentes. \square

3.3.2 Algorithme de La méthode du gradient conjugué pour les fonctions quelconques (cas non linéaire)

On s'intéresse dans cette section à la minimisation d'une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$, non nécessairement quadratique :

$$\min f(x); \quad x \in \mathbb{R}^n \quad (3.16)$$

Les méthodes du gradient conjugué pour résoudre ce problème sont des méthodes itératives de la forme :

$$x_{k+1} = x_k + \alpha_k d_k \quad (3.17)$$

Le pas $\alpha_k \in \mathbb{R}$ étant déterminé par une recherche linéaire. la direction d_k est définie par la formule de récurrence suivante ($\beta_k \in \mathbb{R}$)

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases} \quad (3.18)$$

Ces méthodes sont des extensions de la méthode du gradient conjugué linéaire du cas quadratique, si β_k prend l'une des valeurs

$$\beta_k^{PRP} = \frac{g_k^T y_k}{\|g_{k-1}\|^2}$$

CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$$

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}}$$

où $y_{k-1} = g_k - g_{k-1}$.

Algorithme 3.2 des différentes méthodes du gradient conjugué non linéaire

Etape 0 : (initialisation)

Soit x_0 le point de départ, $g_0 = \nabla f(x_0)$, poser $d_0 = -g_0$
Poser $k = 0$ et aller à l'étape 1.

Etape 1 :

Si $g_k = 0$: STOP ($x^* = x_k$). "Test d'arrêt"
Si non aller à l'étape 2.

Etape 2 :

Définir $x_{k+1} = x_k + \alpha_k d_k$ avec :

α_k : calculer par la recherche linéaire

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$$

où

β_{k+1} : définir selon la méthode

Poser $k = k + 1$ et aller à l'étape 1. \square

Remarque 3.3. : Dans le cas quadratique avec recherche linéaire exacte, on a vu que $\beta_k^{FRP} = \beta_k^{FR} = \beta_k^{CD} = \beta_k^{DY}$.

La propriété de descente de la méthode du gradient conjugué non linéaire

Cas de recherche linéaire exacte Powell ([41, 1984]) a démontré la satisfaction de la propriété de descente de la fonction objectif pour la méthode de Fletcher-Reeves avec recherche linéaire exacte.

Soit $x_{k+1} = x_k + \alpha_k d_k$ la méthode itérative pour résoudre le problème, avec α_k : calculer par la recherche linéaire exacte, et

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases}$$

Alors : quelque soit $\beta_k \in \mathbb{R}$, d_k est une direction de descente, $\forall k \geq 1$

Explication : si le pas α_{k-1} est un point stationnaire de $\alpha \mapsto f(x_{k-1} + \alpha d_{k-1})$.

En effet, dans ce cas $d_{k-1}^T g_k = 0$ et on trouve lorsque $g_k \neq 0$:

$$\begin{aligned} d_k^T g_k &= (-g_k + \beta_k d_{k-1})^T g_k \\ &= -\|g_k\|^2 + \beta_k d_{k-1}^T g_k = -\|g_k\|^2 < 0 \end{aligned}$$

Cas de recherche linéaire inexacte Le premier théorème démontrant la descente d'une méthode du gradient conjugué non linéaire était établi par Albaali ([3, 1985]) pour la méthode de Fletcher-Reeves avec la recherche forte de Wolfe avec $\sigma \leq \frac{1}{2}$.

Gilbert et Nocedal ([22, 1992]) ont généralisé ce résultat pour tout algorithme du gradient conjugué dont

$$|\beta_k| \leq \beta_k^{FR}$$

Aucun théorème n'a été établi afin de démontrer la satisfaction de la propriété de descente pour la méthode de Polak-Ribière-Polyak non linéaire. Grippo et Lucidi ([26, 1997]) et P. Armand ([1, 2005]) ont suggéré des modifications dans le choix de α_k afin d'établir le résultat de la convergence, ainsi la propriété de descente.

Fletcher ([17, 1987]) a démontré que la méthode de la descente conjuguée est une méthode de descente si le pas α_k est déterminé par la règle forte de Wolfe (4.9)-(4.10) avec $\sigma \leq \frac{1}{2}$.

Dai et Yuan ([10, 1996]) ont démontré que cette méthode avec la règle de Wolfe relaxée (4.11)-(4.12) où $0 < \rho < \sigma_1 < 1$ et $0 \leq \sigma_2 \leq 1$, génère des directions de descente suffisante à chaque itération $k \geq 1$.

Dai et Yuan ([13, 1998]) ont démontré qu'à chaque itération $k \geq 1$, la direction recherchée par la méthode de Dai-Yuan avec la recherche de Wolfe faible (4.7)-(4.8), est de descente si la fonction objectif f est strictement convexe.

3.4 Synthèse des résultats de convergence des méthodes du gradient conjugué non linéaire

3.5 avec la recherche linéaire de Wolfe forte

Supposition 3.1.

CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.

(i) L'ensemble $L := \{x \in \mathbb{R}^n; f(x) \leq f(x_1)\}$ est borné; où $x_0 \in \mathbb{R}^n$ est le point initial.

(ii) Sur un voisinage N de L , la fonction objectif f est continûment différentiable et son gradient est lipschitzien i.e

$$\exists L > 0 \text{ tel que } \|g(x) - g(\tilde{x})\| \leq L \|x - \tilde{x}\|, \forall x, \tilde{x} \in \mathcal{N} \quad (3.19)$$

Remarque 3.4. Notons que ces suppositions impliquent qu'il existe $\lambda > 0$ tel que

$$\|g(x)\| \leq \lambda, \forall x \in \mathcal{L} \quad (3.20)$$

Définition 3.2. ([22, 1992]) On dit que d_k est une direction de descente suffisante si

$$d_k^T g_k \leq -C \|g_k\|^2; \text{ où } C > 0 \quad (3.21)$$

Rappelons les conditions de Wolfe faibles :

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha d_k^T g_k \quad (3.22)$$

$$d_k^T g_{k+1} \geq \sigma d_k^T g_k \quad (3.23)$$

où $0 < \rho < \sigma < 1$.

Les conditions de Wolfe fortes :

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \rho \alpha_k d_k^T g_k \\ |d_k^T g_{k+1}| &\leq -\sigma d_k^T g_k \end{aligned} \quad (3.24)$$

où $0 < \rho < \sigma < 1$.

Les conditions de Wolfe relaxées :

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \rho \alpha_k d_k^T g_k \\ \sigma_1 d_k^T g_k &\leq d_k^T g_{k+1} \leq -\sigma_2 d_k^T g_k \end{aligned} \quad (3.25)$$

où $0 < \rho < \sigma_1 < 1$ et $\sigma_2 > 0$.

Présentons maintenant un théorème fondamental qui assure la satisfaction de la condition de Zoutendijk (voir chapitre 2), pour toute méthode du type (3.17)-(3.18), dans laquelle le pas α_k est déterminé par la règle de Wolfe faible (3.22)-(3.23). Ce théorème était démontré par Zoutendijk ([47, 1970]) et Powell ([42, 1971]).

Théorème 3.2. *Considérons une méthode du type (3.17) et (3.18) dans laquelle d_k est une direction de descente et le pas α_k est déterminé par la règle de Wolfe faible (3.22)-(3.23) avec $\sigma \in]0, \frac{1}{2}[$. Considérons aussi que la supposition 3.1 soit satisfaite. Alors pour une telle méthode la condition de Zoutendijk suivante :*

$$\sum_{k \geq 1} \cos^2 \theta \|g_k\|^2 < \infty \quad (3.26)$$

est vérifiée.

3.5.1 Méthode de Fletcher-Reeves

La méthode de Fletcher-Reeves ([19, 1964]) est une extension directe de la méthode du gradient conjugué linéaire au cas des fonctions quelconques. Appliquée à une fonction quadratique, elle est identique au gradient conjugué linéaire.

Rappelons que pour la méthode de Fletcher-Reeves la variante β_k est :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad (3.27)$$

les avantages de la méthode de Fletcher-Reeves

Cette méthode est très intéressante, d'une part parce qu'elle nécessite le stockage de très peu d'informations (essentiellement trois vecteurs de dimension n).

La propriété de descente de la méthode de FR

Théorème 3.3. *Supposons que L'hypothèse 3.1 soit satisfaite. Considérons une méthode du type (3.17) et (3.18) dont*

$$|\beta_k| \leq \beta_k^{FR} \quad (3.28)$$

et le pas α_k satisfait à la règle de Wolfe (3.23) et (3.24) où $\sigma \in]0, \frac{1}{2}[$. Alors cette méthode génère des directions de descente. De plus on a :

$$\frac{-1}{1-\sigma} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma-1}{1-\sigma}; \quad k = 1, \dots \quad (3.29)$$

Démonstration. ([22, 1992])

La démonstration se fait par récurrence

*CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.*

1) Pour $k = 1$:

$$\frac{d_1^T g_1}{\|g_1\|^2} = \frac{-\|g_1\|^2}{\|g_1\|^2} = -1$$

D'autre part :

$$0 < \sigma < \frac{1}{2} \Rightarrow \begin{cases} \frac{-1}{1-\sigma} \leq -1 \\ \frac{2\sigma-1}{1-\sigma} \geq -1 \end{cases}$$

2) Supposons que (3.29) est satisfaite pour $k > 1$ et démontrons qu'elle le sera pour $k + 1$:

Supposons que :

$$\frac{-1}{1-\sigma} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma-1}{1-\sigma}; k = 1, \dots$$

On a :

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} = \frac{(-g_{k+1} + \beta_{k+1} d_k)^T g_{k+1}}{\|g_{k+1}\|^2} = -1 + \beta_{k+1} \frac{d_k^T g_{k+1}}{\|g_k\|^2}$$

D'autre part de (3.27) on aura :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \Rightarrow \frac{1}{\beta_{k+1}^{FR}} = \frac{\|g_k\|^2}{\|g_{k+1}\|^2}$$

d'où :

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} = -1 + \frac{\beta_{k+1}}{\beta_k^{FR}} \frac{d_k^T g_{k+1}}{\|g_k\|^2} \quad (3.30)$$

En utilisant la condition de recherche linéaire (3.24) on aura :

$$|d_k^T g_{k+1}| \leq -\sigma d_k^T g_k \Rightarrow \sigma |\beta_{k+1}| d_k^T g_k \leq \beta_{k+1} d_{k+1}^T g_{k+1} \leq -\sigma |\beta_{k+1}| d_k^T g_k$$

Remplaçant ceci dans (3.30) :

$$-1 + \sigma \frac{|\beta_{k+1}| d_k^T g_k}{\beta_{k+1}^{FR} \|g_k\|^2} \leq \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} \leq -1 - \sigma \frac{|\beta_{k+1}| d_k^T g_k}{\beta_{k+1}^{FR} \|g_k\|^2}$$

De (3.29) on aura :

$$-1 - \frac{|\beta_{k+1}| \sigma}{\beta_{k+1}^{FR} (1-\sigma)} \leq \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} \leq -1 + \frac{|\beta_{k+1}| \sigma}{\beta_{k+1}^{FR} (1-\sigma)}$$

et de (3.28) :

$$\frac{-\sigma}{1-\sigma} < \frac{|\beta_{k+1}|}{\beta_{k+1}^{FR}} < 1$$

On aura :

$$\frac{-1}{1-\sigma} \leq \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} \leq \frac{2\sigma-1}{1-\sigma}$$

Ce qui achève la démonstration. \square

Corollaire 3.1. D'après (3.29) on déduit que pour tout $k \geq 1$, d_k est une direction de descente suffisante avec $C = \frac{1-2\sigma}{1-\sigma} > 0$.

En effet :

$$\begin{aligned} \frac{-1}{1-\sigma} &\leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma-1}{1-\sigma} \\ \Rightarrow d_k^T g_k &\leq -C \|g_k\|^2 \quad \text{où } C = \frac{1-2\sigma}{1-\sigma} \quad \square \end{aligned}$$

Convergence de la méthode de Fletcher-Reeves

Le premier résultat de convergence de la méthode du gradient conjugué non linéaire (version Fletcher-Reeves) avec des recherches linéaires inexactes (recherche linéaire inexacte de Wolfe forte (3.22) et (3.24) où $\sigma < \frac{1}{2}$) a été démontré par Al-Baali ([3, 1985]). Touati Ahmed et Story ([44, 1990]) ont généralisé ce résultat pour

$0 \leq \beta_k \leq \beta_k^{FR}$. Gilbert et Nocedal ([22, 1992]) ont généralisé ce résultat pour $|\beta_k| \leq \beta_k^{FR}$.

Théorème 3.4. *Supposons que l'hypothèse 3.1 soit satisfaite. Considérons une méthode du type (3.17) et (3.18) avec β_k satisfaisant à (3.28) et le pas α_k satisfaisant aux conditions de Wolfe fortes (3.22) et (3.24) où $\sigma \in]0, \frac{1}{2}[$. Alors cette méthode est globalement convergente, dans le sens suivant :*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \tag{3.31}$$

Démonstration. ([22, 1992])

Puisque les conditions du théorème 4.2 sont satisfaites alors on a :

$$\frac{-1}{1-\sigma} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\sigma-1}{1-\sigma} \Rightarrow -\sigma d_{k-1}^T g_{k-1} \leq \frac{\sigma}{1-\sigma} \|g_{k-1}\|^2$$

D'autre part de (3.24)

$$|d_k^T g_{k+1}| \leq -\sigma d_k^T g_k \Rightarrow |d_{k-1}^T g_k| \leq -\sigma d_{k-1}^T g_{k-1}$$

d'où

$$|d_{k-1}^T g_k| \leq -\sigma d_{k-1}^T g_{k-1} \leq \frac{\sigma}{1-\sigma} \|g_{k-1}\|^2 \tag{3.32}$$

*CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.*

De (3.17), (3.28) et (3.32) :

$$\begin{aligned}\|d_k\|^2 &= \left| \|g_k\|^2 - 2\beta_k d_{k-1}^T g_k + \beta_k^2 \|d_{k-1}\|^2 \right| \\ &\leq \|g_k\|^2 + |2\beta_k d_{k-1}^T g_k| + \beta_k^2 \|d_{k-1}\|^2 \\ &\leq \left(\frac{1+\sigma}{1-\sigma} \right) \|g_k\|^2 + (\beta_k^{FR})^2 \|d_{k-1}\|^2\end{aligned}$$

Posons $\hat{\sigma} = \frac{1+\sigma}{1-\sigma} \geq 1$, on aura :

$$\begin{aligned}\|d_k\|^2 &\leq \hat{\sigma} \|g_k\|^2 + (\beta_k^{FR})^2 \|d_{k-1}\|^2 \\ &\leq \hat{\sigma} \|g_k\|^2 + (\beta_k^{FR})^2 \left[\hat{\sigma} \|g_{k-1}\|^2 + (\beta_{k-1}^{FR})^2 \|d_{k-2}\|^2 \right] \\ &\quad \cdot \\ &\leq \hat{\sigma} \|g_k\|^4 \sum_{j=2}^k \|g_j\|^{-2} + \hat{\sigma} \|g_k\|^4 \|g_1\|^{-2} = \hat{\sigma} \|g_k\|^4 \sum_{j=1}^k \|g_j\|^{-2}\end{aligned}$$

Supposons que g_k est borné en dehors du zéro ($\lim_{k \rightarrow \infty} \inf \|g_k\| \neq 0$), c'est-à-dire :

$$\|g_k\| \geq \omega > 0; \forall k \Rightarrow \|g_k\|^{-2} \leq \omega^{-2}$$

de (3.20) on a :

$$\begin{aligned}\|d_k\|^2 &\leq \hat{\sigma} \|g_k\|^4 \sum_{j=1}^k \|g_j\|^{-2} \leq \hat{\sigma} \frac{\lambda^4}{\omega^2} \sum_{j=1}^k 1 \\ &\Rightarrow \|d_k\|^2 \leq \hat{\sigma} \frac{\lambda^4}{\omega^2} k\end{aligned}$$

d'où

$$\sum_{k \geq 1} \frac{1}{\|d_k\|^2} \geq \frac{\omega^2}{\hat{\sigma} \lambda^4} \sum_{k \geq 1} \frac{1}{k} > \infty \quad (3.33)$$

Ce qui veut dire que $\sum_{k \geq 1} \frac{1}{\|d_k\|^2}$ est divergente.

D'autre part, puisque les conditions du Théorème 3.1, et du théorème 3.2 sont satisfaites on a :

$$\sum_{k \geq 1} \cos^2 \theta_k \|g_k\|^2 < \infty$$

et

$$c_1 \frac{\|g_k\|}{\|d_k\|} \leq \cos \theta_k \leq c_2 \frac{\|g_k\|}{\|d_k\|}$$

d'où

$$\begin{aligned}
 \sum_{k \geq 1} c_1^2 \frac{\|g_k\|^2}{\|d_k\|^2} \|g_k\|^2 &\leq \sum_{k \geq 1} \cos^2 \theta_k \|g_k\|^2 < \infty \\
 &\Rightarrow \sum_{k \geq 1} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty \\
 &\Rightarrow \sum_{k \geq 1} \frac{\omega^4}{\|d_k\|^2} < \infty \\
 &\Rightarrow \sum_{k \geq 1} \frac{1}{\|d_k\|^2} < \infty
 \end{aligned}$$

Ce qui contredit (3.33), d'où le résultat :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad \square$$

Remarque 3.5. Lui, Han et Yuan ([32, 1995]) ont prouvé le résultat (3.31) pour cette méthode si $\rho = \sigma = \frac{1}{2}$.

Inconvénients de la méthode de Fletcher-Reeves

La méthode de Fletcher-Reeves possède de bonnes propriétés théoriques, mais en pratique, elle converge parfois lentement, et même prématurément. En effet, si les pas sont trop petits, il se peut que ce comportement s'élargisse pour un grand nombre d'itérations, et c'est ce qu'il nous oblige à réinitialiser en posant $\beta_k^{FR} = 0$.

Powell ([43, 1977]) est le premier qui a observé ce comportement, ainsi il a donné un contre exemple avec une recherche linéaire exacte. On évite cet inconvénient en réinitialisant chaque n itérations par exemple.

Nemirovsky et Yudin ([36, 1983]) ont démontré par un contre exemple que la méthode de Fletcher-Reeves converge plus lentement que la méthode de Steepest descent. (cas d'une fonction quadratique avec recherche linéaire exacte).

3.5.2 Méthode de Polak-Ribière-Polyak

Cette méthode fut découverte par Polak, Ribière ([38, 1969]) et Polyak ([39, 1969]).

Rappelons que pour la méthode de Polak-Ribière-Polyak la variante β_k est :

$$\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2} \quad (3.34)$$

Avantages de la méthode de Polak-Ribière-Polyak

La méthode de Polak-Ribière-Polyak possède bonnes performances numériques

Convergence de la méthode de PRP

pour étudier la convergence de cette méthode on a deux cas

cas où f est fortement convexe Polak et Ribière ([38, 1969]) a démontré la convergence de la méthode de Polak-Ribière-Polyak à travers le théorème suivant :

Théorème 3.5. *Si f est fortement convexe, continûment différentiable avec un gradient lipschitzien, alors la méthode de Polak Ribière avec recherche linéaire exacte génère une suite $\{x_k\}$ convergeant vers l'unique point x^* réalisant le minimum de f .*

Démonstration. ([21])

Montrons dans un premier temps que

$$\cos \theta_k = \frac{-d_k^T g_k}{\|g_k\| \|d_k\|}$$

est uniformément positif.

Grâce à la recherche linéaire exacte, on a

$$\begin{aligned} d_{k-1}^T y_{k-1} &= d_{k-1}^T (g_k - g_{k-1}) \\ &= -d_{k-1}^T g_{k-1} = -(-g_{k-1} + \beta_{k-1}^{PRP} d_{k-2})^T g_{k-1} \\ &= \|g_{k-1}\|^2 \end{aligned}$$

La forte convexité de f implique que

$$d_{k-1}^T y_{k-1} = \frac{1}{\alpha_{k-1}} (x_k - x_{k-1})^T y_{k-1} \geq \frac{\eta}{\alpha_{k-1}} \|x_k - x_{k-1}\|^2$$

où $\eta > 0$ est le module de forte convexité de f .

On en déduit, en utilisant la constante de lipschitz L de g :

$$|\beta_k^{PRP}| = \frac{|g_k^T y_{k-1}|}{\|g_{k-1}\|^2} = \frac{|g_k^T y_{k-1}|}{d_{k-1}^T y_{k-1}} \leq \frac{\alpha_{k-1} L \|g_k\| \|x_k - x_{k-1}\|}{\eta \|x_k - x_{k-1}\|^2} = \frac{L}{\eta} \frac{\|g_k\|}{\|d_{k-1}\|}$$

*CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.*

On peut alors borner $\|d_k\|$ par :

$$\begin{aligned} \|d_k\| &\leq \|g_k\| + |\beta_k^{PRP}| \|d_{k-1}\| \\ &\leq \|g_k\| + \frac{L}{\eta} \frac{\|g_k\|}{\|d_{k-1}\|} \|d_{k-1}\| \\ &\leq \left(1 + \frac{L}{\eta}\right) \|g_k\| \end{aligned}$$

Ensuite

$$d_k^T g_k = -\|g_k\|^2 \leq -\left(1 + \frac{L}{\eta}\right) \|g_k\| \|d_k\|$$

ou encore

$$\cos \theta_k = \frac{-d_k^T g_k}{\|g_k\| \|d_k\|} \geq \left(1 + \frac{L}{\eta}\right)^{-1}$$

D'après la proposition 2.1 et la recherche linéaire exacte, la condition de Zoutendijk est vérifiée. Mais f et donc $\{f(x_k)\}$ est bornée inférieurement (car f est fortement convexe). On en déduit que $g_k \rightarrow 0$.

D'autre part, $\{x_k\}$ est bornée (f est fortement convexe) et possède donc des sous suites convergentes. La limite de celles-ci ne peut être que l'unique minimum x^* de f (car $g_k \rightarrow 0$).

Donc toute la suite $\{x_k\}$ converge vers x^* . \square

cas où f n'est pas convexe Powell ([41, 1984]) a donné un exemple de fonction (de 3 variables, deux fois continûment différentiable) pour laquelle l'algorithme génère une suite $\{x_k\}$ dont aucun des points d'adhérence n'est stationnaire.

Donc Si f n'est pas convexe, la méthode de Polak-Ribière-Polyak peut ne pas converger.

En 1986 Powell([40, 1986]), a modifié la variante β_k^{PRP} en évitant les valeurs négatives, autrement dit si à l'itération k on a : si $\beta_k^{PRP} < 0$, on redémarre en posant $\beta_k^{PRP} = 0$ (prendre la direction de la plus profonde pente)

$$\beta_k = \max \{0, \beta_k^{PRP}\}$$

Ce choix assure la convergence si le pas α_k est déterminé par la règle de Wolfe forte. \square

Gilbert et Nocedal ([22, 1992]) ont assuré la convergence avec une recherche linéaire exacte ou inexacte, en hybridant les méthodes de FR et de PRP.

Donc cette nouvelle méthode consiste à prendre la variante β_k comme suit :

$$\beta_k = \begin{cases} -\beta_k^{FR} & \text{si } \beta_k^{PRP} < -\beta_k^{FR} \\ \beta_k^{PRP} & \text{si } |\beta_k^{PRP}| \leq \beta_k^{FR} \\ \beta_k^{FR} & \text{si } \beta_k^{PRP} > \beta_k^{FR} \end{cases}$$

Grippo et Lucidi ([26, 1997]) et P. Armand ([1, 2005]) ont suggéré des modifications dans le choix de α_k afin d'établir le résultat de la convergence.

3.5.3 Méthode de la descente conjuguée

Cette méthode fut proposée en 1987 par Fletcher et Reeves ([17, 1987]), Rappelons que pour la méthode de la descente conjuguée la variante β_k est :

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_k} \quad (3.35)$$

La propriété de descente de la méthode de la descente conjuguée

Fletcher ([17, 1987]) a démontré que la méthode de la descente conjuguée est une méthode de descente si le pas α_k est déterminé par la règle forte de Wolfe (3.22) et (3.24) avec $\sigma \leq \frac{1}{2}$.

Dai et Yuan ([11, 1996]) ont démontré le théorème suivant :

Théorème 3.6. *Supposons que l'hypothèse 4.1 est satisfaite.*

Pour toute méthode du type (3.17) et (3.18) dont β_k satisfait à (3.35) et le pas α_k satisfait aux conditions de Wolfe relaxée (3.22) et (3.25) :

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \rho d_k^T g_k \\ \text{et } \sigma_1 d_k^T g_k &\leq d_k^T g_{k+1} \leq -\sigma_2 d_k^T g_k \end{aligned}$$

où $0 < \rho < \sigma_1 < 1$ et $0 \leq \sigma_2 \leq 1$

Alors notre méthode génère des directions de descente suffisante à chaque itération $k \geq 1$.

Démonstration. ([11, 1996])

On a

$$\begin{aligned} -d_k^T g_k &= -(-g_k + \beta_k^{CD} d_{k-1})^T g_k \\ &= \|g_k\|^2 \left[1 + \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} \right] \\ \Rightarrow \frac{-d_k^T g_k}{\|g_k\|^2} &= 1 + \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} \end{aligned}$$

D'autre part de (3.25)

$$\begin{aligned}\sigma_1 d_k^T g_k &\leq d_k^T g_{k+1} \leq -\sigma_2 d_k^T g_k \\ \Rightarrow 1 - \sigma_2 &\leq 1 + \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} \leq 1 + \sigma_1\end{aligned}$$

d'où

$$1 - \sigma_2 \leq \frac{-d_k^T g_k}{\|g_k\|^2} \leq 1 + \sigma_1$$

Donc si $\|g_k\| \neq 0$, on a :

$$d_k^T g_k \leq -C \|g_k\|^2 \quad \text{où } C = 1 - \sigma_2 > 0$$

et donc d_k est une direction de descente suffisante

Convergence de la méthode de la descente conjuguée

Yuan ([49, 1993]) a démontré la convergence au sens (3.31) de cette méthode avec un pas satisfaisant aux conditions (3.22)-(3.25) si $\sigma_1 < \frac{1}{2}$ et $\sigma_2 = 0$.

Dai et Yuan ([11,1996]) ont démontré ce résultat pour $\sigma_1 < 1$ et $\sigma_2 = 0$,

Théorème 3.7. *Supposons que la l'hypothèse 4.1 est satisfaite.*

Toute méthode du type (3.17) et (3.18) dans laquelle β_k vérifie (3.35) et le pas α_k est déterminé par la règle de Wolfe relaxée (3.22)-(3.25) où $0 < \rho < \sigma_1 < 1$ et $\sigma_2 = 0$; est de descente convergente, dans le sens où

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

Démonstration. ([11,1996])

Du théorème 3.4 on a :

$$\begin{aligned}1 - \sigma_2 &\leq \frac{-d_k^T g_k}{\|g_k\|^2} \leq 1 + \sigma_1 \\ \Rightarrow 1 &\leq \frac{-d_k^T g_k}{\|g_k\|^2} \leq 1 + \sigma_1 \\ \Rightarrow (1 + \sigma_1)^{-1} &\leq \frac{\|g_k\|^2}{-d_k^T g_k} \leq 1 \\ \Rightarrow (1 + \sigma_1)^{-1} &\leq \frac{\|g_{k+1}\|^2 \|g_k\|^2}{-d_k^T g_k \|g_{k+1}\|^2} \leq 1 \\ \Rightarrow (1 + \sigma_1)^{-1} &\leq \frac{\beta_{k+1}^{CD}}{\beta_{k+1}^{FR}} \leq 1 \\ \Rightarrow \beta_{k+1}^{CD} &\leq \beta_{k+1}^{FR}\end{aligned}$$

Donc β_{k+1}^{CD} vérifie l'inégalité (3.28).
D'après le théorème 3.2 on a :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad \square$$

Nous terminons ce chapitre concernant la convergence des méthodes du gradient conjugué utilisant les recherches linéaires de Wolfe forte par un résultat récent de Dai et Yuan ([14,2002]).

3.5.4 Convergence d'une classe de méthodes de Gradient Conjugué dues à Dai et Yuan avec des recherches linéaires de Wolfe fortes.

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, non nécessairement quadratique. Considérons toujours notre problème de minimisation sans contraintes

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

Pour trouver un point stationnaire ou une solution optimale à notre problème, les différentes méthodes du gradient conjugué generent des suites $\{x_k\}_{k \in \mathbb{N}}$ de la forme :

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.36}$$

Le pas $\alpha_k \in \mathbb{R}$ étant déterminé par une recherche linéaire exacte ou inexacte. Nous supposons dans cette partie, comme c'est le cas dans tout le chapitre, que notre recherche est une **recherche linéaire inexacte de Wolfe Forte**, c'est à dire que α_k vérifie :

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \rho \alpha_k d_k^T g_k \\ |d_k^T g_{k+1}| &\leq -\sigma d_k^T g_k, \quad 0 < \rho < \sigma < 1. \end{aligned}$$

La direction d_k est définie par la formule de récurrence suivante ($\beta_k \in \mathbb{R}$)

$$d_k = \begin{cases} -g_1 & \text{si } k = 1 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2 \end{cases} \tag{3.37}$$

Nous avons vu dans les paragraphes précédents comment les résultats de convergences des différentes méthodes de gradient conjugué dépendaient essentiellement des scalaires β_k et ces résultats ont été étudiés séparément suivant les différentes valeurs de β_k , voir El-Baali ([3], 1985), Gilbert et Nocdall

CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.

([22], 1992), Grippo et Lucidi ([26], 1997), Powell ([40], 1986), Touati-Ahmed et Storey ([44], 1990), Powell ([41], 1984).

En 2002 Dai et Yuan ([14], 2002) ont étudié le problème suivant :

Quelles conditions assez générales doivent vérifier les coefficients β_k pour assurer la convergence globale en utilisant des recherches linéaires inexacts de Wolfe fortes.

Ce paragraphe détaille cette étude. Avant de donner le résultat essentiel de cette étude, considérons d'abord les deux lemmes suivants dus aux mêmes auteurs.

Lemme 3.1 ([13]) *Supposons que l'hypothèse 3.1 est satisfaite. Pour toute méthode du type (3.17) et (3.18) dans laquelle :*

d_k : est une direction de descente et le pas α_k est déterminé par la règle fort de Wolfe (3.22) et (3.24), Si

$$\sum_{k \geq 1} \frac{\|g_k\|^t}{\|d_k\|^2} = \infty \quad \text{pour } t \in [0, 4] \quad (3.38)$$

Alors cette méthode est globalement convergente, dans le sens suivant :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

Lemme 3.2 ([14]) *Supposons que la l'hypothèse 3.1 est satisfaite. Toute méthode du type (3.17) et (3.18) dans laquelle :*

d_k : est une direction de descente et le pas α_k est déterminé par la règle fort de Wolfe forte (3.22) et (3.24). Si

$$\sum_{k \geq 1} \frac{|g_k^T d_k|^r}{\|d_k\|^2} = \infty \quad \text{pour } r \in [0, 2] , \quad (3.39)$$

alors cette méthode est globalement convergente, dans le sens suivant

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

Démonstration.

Pour tout $r \in [0, 2]$, si $|g_k^T d_k| > 1 \implies (g_k^T d_k)^2 \geq |g_k^T d_k|^r$. On a

$$|g_k^T d_k|^r \leq 1 + (g_k^T d_k)^2$$

Alors

CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.

$$\sum_{k \geq 1} \frac{1}{\|d_k\|^2} \geq \sum_{k \geq 1} \frac{|g_k^T d_k|^r}{\|d_k\|^2} - \sum_{k \geq 1} \frac{|g_k^T d_k|^2}{\|d_k\|^2} \quad (3.40)$$

Du lemme 3.2, et en utilisant (3.39) et (3.40), on peut remarquer que (3.33) est vérifiée pour $t = 0$

Donc du lemme 3.2 on trouve que (3.31) est vérifiée aussi. Ce qui achève la démonstration. \square

Corollaire 3.2. *Supposons que la l'hypothèse 3.1 est satisfaite. Toute méthode du type (3.17) et (3.18) dans laquelle :*

d_k : est une direction de descente et le pas α_k est déterminé par la règle fort de Wolfe forte (3.22) et (3.24)

Si

$$\sum_{k \geq 1} \frac{\|g_k\|^t |g_k^T d_k|^r}{\|d_k\|^2} = \infty \quad (3.41)$$

pour $t > 0$ et $r > 0$ avec $t + 2r \leq 4$. La méthode est globalement convergente, dans le sens suivant :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

Démonstration.

Supposons que cela n'est pas vrai. On obtient par le lemme 3.1

$$\sum_{k \geq 1} \frac{\|g_k\|^{t+2r}}{\|d_k\|^2} < \infty \quad \forall t, r > 0 \quad (3.42)$$

Si : $|g_k^T d_k| \leq \|g_k\|^2$ on a :

$$\|g_k\|^t |g_k^T d_k|^r \leq \|g_k\|^{t+2r}$$

Sinon, on aurait

$$\|g_k\|^t |g_k^T d_k|^r \leq |g_k^T d_k|^{\frac{t}{2}+r}.$$

Donc la relation suivante est vérifiée

$$\|g_k\|^t |g_k^T d_k|^r \leq \|g_k\|^{t+2r} + |g_k^T d_k|^{\frac{t}{2}+r}$$

De (3.41) et (3.42) on a

$$\sum_{k \geq 1} \frac{|g_k^T d_k|^{\frac{t}{2}+r}}{\|d_k\|^2} = \infty$$

Si on prend en considération le lemme 3.1. et le fait que $\frac{t}{2} + r \in [0, 2]$, on en conclut que :

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

Ainsi on obtient une contradiction. Ce qui achève la démonstration. \square

3.5.5 Condition générale sur les scalaires β_k

Le but de cette partie est le théorème 3.8 qui suit dans lequel Dai et Yuan ([14, 2002]) ont posé une condition sur les scalaires β_k qui assure la convergence globale de la méthode du gradient conjugué dans le cas de la recherche linéaire de Wolfe forte.

Théorème 3.8. *Supposons que la l'hypothèse 3.1 est satisfaite. Toute méthode du type (3.17) et (3.18) dans laquelle d_k : est une direction de descente et le pas α_k est déterminé par la règle de Wolfe forte (3.22) et (3.24).*

S'il existe une sous suite infinie $\{k_i\}$ et une constante positive $c > 0$ telle que les valeurs de β_j vérifient :

$$\prod_{j=\ell}^{k_i} |\beta_j| \leq c \quad \forall i \geq 1, \quad \ell \leq k_i, \quad (3.43)$$

alors la méthode converge dans le sens suivant

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

Démonstration. ([14])

pour $k \geq 2$, (3.18) implique que

$$\begin{aligned} d_k + g_k &= \beta_k d_{k-1} \Rightarrow \|d_k\|^2 \leq -2g_k^T d_k + \beta_k^2 \|d_{k-1}\|^2 \\ &\Rightarrow \|d_k\|^2 \leq -2g_k^T d_k - 2 \sum_{j=2}^k \prod_{i=j}^k \beta_i^2 g_{j-1}^T d_{j-1} \\ &\Rightarrow \|d_k\|^2 \leq (1 + c^2) \sum_{j=1}^{k_i} (-2g_j^T d_j) \end{aligned}$$

Si

$$\liminf_{k \rightarrow \infty} \|d_{k_i}\| < \infty$$

Donc

$$\sum_{k \geq 1} \frac{1}{\|d_k\|^2} = \infty.$$

Soit en utilisant le lemme (3.2) on a

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

Sinon on obtiendrait la relation suivante :

$$\lim_{i \rightarrow \infty} \|d_{k_i}\| = \infty.$$

Dans ce cas, on aura : obtient d'après (3.31) vérifier qui implique

$$\lim \sum_{j=1}^{k_i} (-g_j^T d_j) = \infty.$$

Cette dernière relation donne :

$$\lim \sum_{j=1}^{k_i} \frac{(-g_j^T d_j)}{\|d_j\|^2} = \infty.$$

Par conséquent, en utilisant le lemme (3.1) nous avons

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

Ce qui achève la démonstration. \square

3.5.6 Méthodes de Newton et quasi-Newton

3.5.7 Méthode de Newton

La méthode de *Newton* est attribuée au mathématicien, physicien et astronome anglais *Issac Newton* (1642-1727). Toutefois, comme le dit Dirand c'est Raphson qui publiait, en 1690, la formule itérative utilisée actuellement. C'est la raison pour la quelle certains auteurs l'appellent méthode

de Raphson-Newton. L'algorithme de Newton est la généralisation multi dimensionnelle de la méthode de Newton Raphson, appliquée à la recherche des racines de $\nabla f(x)$. La méthode de Newton n'est pas une méthode d'optimisation proprement parler. C'est une méthode de recherche des zéros d'une fonction $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ selon : $g(x) = 0$. L'idée de cette méthode ici est de résoudre l'équation $\nabla f(x) = 0$, condition nécessaire de premier ordre pour la détection d'extrema d'une fonction. L'équation $\nabla f(x) = 0$ est donnée par un système $n \times n$ d'équations non linéaires. Nous allons d'abord la décrire puis montrer comment on peut l'appliquer à la recherche de minimum.

Description de la méthode

Autour de x_k , une approximation quadratique de la fonction f supposée deux fois différentiable est :

$$q(x) = f(x_k) + \nabla f^t(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^t H(x_k)(x - x_k) \quad (3.44)$$

Si la matrice $H(x_k)$ est définie positive alors la fonction $q(x)$ est strictement convexe et a un minimum unique définie par :

$$\nabla q(x_{k+1}) = 0 \quad (3.45)$$

c'est à dire

$$\nabla f(x_k) + H(x_k)(x_{k+1} - x_k) = 0 \quad (3.46)$$

qui donne

$$x_{k+1} = x_k - [H(x_k)]^{-1} \nabla f(x_k) \quad (3.47)$$

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

supposons que $f(\bar{x}) = 0$ 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 de x_k est bien défini.

Newton

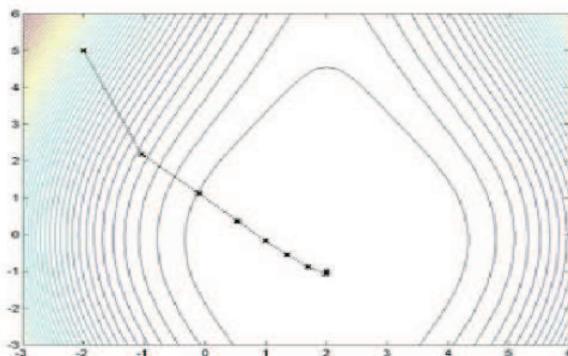


FIG 3.1

Algorithme Etape initiale : soit $\varepsilon > 0$, critère d'arrêt. Choisir x_1 point initial, poser $k = 1$ et aller à l'étape principale.

Etape principale : Si $\|\nabla f(x_k)\| \leq \varepsilon$ stop, sinon poser

$$x_{k+1} = x_k - [H(x_k)]^{-1} \nabla f(x_k) \quad (3.48)$$

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

Il est important d'observer que l'algorithme de Newton construit des suites convergeant vers des points stationnaires, sans faire de distinction entre les minima où le maxima, par exemple. Ceci est dû au fait qu'il est conçu pour trouver des racines de $\nabla f(x) = 0$. Par conséquent, sans modification adéquate, si le premier itéré est proche d'un point stationnaire, la suite générée convergera vers ce point stationnaire. On comprend que, si l'on cherche à minimiser f , converger vers un maximum local n'est pas une propriété satisfaisante. Il sera donc nécessaire de modifier l'algorithme de Newton de manière à le contraindre à éviter les points stationnaires qui ne sont pas des minima.

3.5.8 Avantages et inconvénients de l'algorithme de Newton

Avantages

Si le point x_1 est assez proche de la solution optimale locale x_* telle que $H(x_*)$ soit définie positive, alors l'algorithme de Newton converge de façon quadratique vers la solution x_* . c'est à dire que l'on a

$$\|x_{k+1} - x_*\| \leq \gamma \|x_k - x_*\|^2 \quad \gamma \geq 0 \quad (3.49)$$

Inconvénients

Les inconvénients de la méthode de Newton pour résoudre des systèmes d'équations non linéaires (resp des problèmes d'optimisation) sont bien connus :

- Cette méthode fonctionne très bien pour les problèmes de petite dimension ($1 \leq n \leq 10$), lorsqu'on peut calculer facilement la matrice hessienne H et son inverse. Ce calcul nécessite des itérations plus nombreuses et coûteuse dans les problèmes de grandes tailles.

L'algorithme n'est pas défini aux points x où la matrice hessienne H est singulière.

Pour les problèmes d'optimisation, si f n'est pas strictement convexe, l'algorithme ne génère pas nécessairement des directions de descente de f .

L'algorithme n'est pas globalement convergent (si le premier itéré est éloigné d'une solution, le comportement des itérés suivants est erratique).

Un système linéaire d'ordre n doit être résolu à chaque itération.

A cause de ces inconvénients, il existe des modifications qui permettent d'obtenir une convergence superlinéaire, pour approximer l'étape de Newton asymptotiquement. C'est le principe de Dennis et Moré. Comment peut-on y aboutir sans évaluer la matrice Hessienne dans chaque itération. La réponse a été découverte par *Davidon* en 1959 et a été développée et popularisée par *Fletcher* et *Powell* en 1963. Elle consiste à commencer par n'importe quelle approximation de la matrice hessienne et à chaque itération, on améliore la matrice en introduisant la courbure du problème mesurée tout au long de l'étape. Si cette amélioration est faite correctement, on obtient quelques méthodes remarquablement robustes et efficaces, qu'on appelle les méthodes de la variable métrique ou quasi-Newton. Ils ont libéré l'optimisation non linéaire en procurant une alternative à la méthode de Newton, qui est très coûteuse pour plusieurs applications. Il y a plusieurs méthodes de quasi-Newtoniennes, on s'étalera particulièrement sur les trois plus importantes, la méthode de correction de rang un, la méthode de DFP (Davidon, Fletcher, Powell), et la méthode de BFGS (Broyden, Fletcher, Goldfarb, Shanno).

3.5.9 Méthode de quasi-Newton

Cette classe de méthodes cherche à tirer le meilleur profit possible de l'information de premier ordre (gradient) fournie par les itérés. A chaque itération, on résout un système linéaire dont la matrice B_k est une approximation de la matrice hessienne $H(x_k)$.

Cette méthode de type :

$$\begin{cases} x_{k+1} = x_k + \alpha_k d_k \\ d_k = -B_k g_k \end{cases} \quad (3.50)$$

ou bien

$$\begin{cases} x_{k+1} = x_k + \alpha_k d_k \\ d_k = -S_k^{-1} g_k \end{cases} \quad (3.51)$$

Où B_k (respectivement S_k) est une matrice destinée à approcher l'inverse du Hessien de f (respectivement le hessien) de f en x_k . Le problème posé

*CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.*

est de trouver une stratégie à adopter pour faire cette approximation. On peut commencer par poser $B_1 = I$. mais comment ensuite mettre à jour l'approximation B_k au cours des itérations? L'idée est la suivante : prenons $f \in C^2(\mathbb{R}^n)$, et faisons un développement de ∇f au voisinage de x_k .

$$\nabla f(x) = \nabla f(x_k) + H(x_k)(x - x_k) + o(\|x - x_k\|) \quad (3.52)$$

$$\simeq \nabla f(x_k) + H(x_k)(x - x_k) \quad (3.53)$$

ce qui implique

$$[H(x_k)]^{-1} [\nabla f(x) - \nabla f(x_k)] \simeq x - x_k \quad (3.54)$$

les approximations sont exacte si f est quadratique.

En particulier avec $x = x_{k+1}$ et si B_k était une bonne approximation de $[H(x_k)]^{-1}$ alors

$$B_k [g(x_{k+1}) - g(x_k)] \simeq x_{k+1} - x_k \quad (3.55)$$

on peut imposer que B_{k+1} satisfait cette equation exactement d'ou

$$B_{k+1} [g(x_{k+1}) - g(x_k)] = x_{k+1} - x_k \quad (3.56)$$

posons

$$s_k = x_{k+1} - x_k \quad (3.57)$$

$$y_k = g(x_{k+1}) - g(x_k) \quad (3.58)$$

On obtient l'équation de sécante

$$B_{k+1} y_k = s_k \quad (3.59)$$

On souhaite que la matrice B_k possède les propriétés suivantes :

- Facile à mettre à jour,
- symétrique,
- "converge" vers la matrice hessienne,
- généralise la méthode de sécante, c'est à dire possède la propriété de sécante (exacte dans le cas quadratique)
- résolve un problème quadratique de dimension n à n itération
- soit définie positive

3.6 Méthode de Broyden-Fletcher-Goldfarb-Shanno (BFGS). [1970]

Méthode de Broyden, Fletcher, Goldfarb et Shanno (BFGS) Cet algorithme, développé indépendamment par Broyden (1970), Fletcher (1970), Goldfarb (1970) et Shanno (1969). La formule de mise à jour de Broyden, Fletcher, Goldfarb et Shanno est une formule de correction de rang deux, qui s'obtient à partir de la formule DFP en intervertissant les rôles de s_k et y_k . La formule obtenue permet de mettre à jour une approximation B_k de Hessien lui même et non de son inverse comme dans le cas de la méthode DFP. On exigera que posée dans les mêmes propriétés, à savoir B_{k+1} reste définie positive si M_k l'est et bien sur l'équation d'approximation de quasi Newton doit être vérifiée, c'est à dire :

3.6.1

$$M_{k+1}s_k = y_k \quad (3.60)$$

on obtient donc

$$M_{k+1} = M_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{M_k s_k s_k^T M_k}{s_k^T M_k s_k} \quad (3.61)$$

BFGS avec $H_0=I$

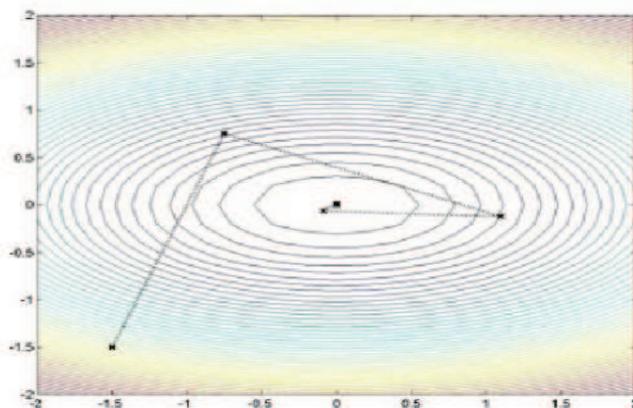


FIG 3.2

L'algorithme associé est le suivant :

Algorithme2. (Méthode de BFGS)

Etape initiale :

Soit $\varepsilon > 0$, déterminer le critère d'arrêt. Choisir \varkappa_1 point initial et M_1 définie positive quelconque (par exemple $M_1 = I$).

Poser $k = 1$ et aller aux étapes principales

Étapes principales.

Etape 1 :

Si $\|\nabla f(\varkappa_k)\| < \varepsilon$ STOP ; sinon, poser $d_k = -M_k g_k$ et déterminer le pas

optimal λ_k solution optimale du problème $\min f(\varkappa_k + \lambda d_k)$, $\lambda \geq 0$ et poser

$$\varkappa_{k+1} = \varkappa_k + \lambda_k d_k$$

Etape 2 :

Construire M_{k+1} comme suit :

$$M_{k+1} = M_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{M_k s_k s_k^\top M_k}{s_k^\top M_k s_k}$$

avec

$$\begin{aligned} s_k &= \varkappa_{k+1} - \varkappa_k \\ y_k &= \nabla f(\varkappa_{k+1}) - \nabla f(\varkappa_k) \end{aligned}$$

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

3.6.2 Remarque :

1-Notons que la direction d_k est obtenue par une résolution d'un système linéaire. En particulier la mise à jour de M_k est faite directement sur le facteur de Cholesky C_K où $M_K = C_K C_K^\top$ ce qui ramène le calcul de d_k au même coût que pour la formule de DFP.

2-La méthode BFGS possède les mêmes propriétés que la méthode DFP dans le cas quadratique. Les directions engendrées sont conjuguées. Cette méthode est reconnue comme étant beaucoup moins sensible que la méthode DFP aux imprécisions dans la recherche linéaire, du point de vue de vitesse de convergence. Elle est donc tout à fait adaptée quand la recherche linéaire est faite de façon économique, avec par exemple la règle de Goldstein ou la règle de Wolfe et Powell.

3-La relation

$$M_{k+1} = M_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{M_k s_k s_k^\top M_k}{s_k^\top M_k s_k} \quad (3.62)$$

*CHAPITRE 3. CONVERGENCE DES MÉTHODES DU GRADIENT
CONJUGUÉ UTILISANT DES RECHERCHES LINÉAIRES DE WOLFE
FORTES.*

permet de construire une approximation de la matrice Hessienne elle même (et non pas son inverse).

En effet : Posons

$$C_K = \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{M_k s_k s_k^\top M_k}{s_k^\top M_k s_k} \quad (3.63)$$

Nous avons

$$H_{k+1} = [M_{k+1}]^{-1} = [M_k + C_K]^{-1} \quad (3.64)$$

Par application de la formule de Sherman-Morrison-Woodbury suivante

$$(A + ab^\top)^{-1} = A^{-1} - \frac{A^{-1} ab^\top A^{-1}}{1 + b^\top A^{-1} a} \quad (3.65)$$

Où A est une matrice inversible, et b est un vecteur de \mathbb{R}^n , et en supposant que $b^\top A^{-1} a \neq -1$ alors on a

$$s_{k+1} = [M_{k+1}]^{-1} = [M_k + C_K]^{-1} = \left[M_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{M_k s_k s_k^\top M_k}{s_k^\top M_k s_k} \right]^{-1} \quad (3.66)$$

Posons

$$A = M_k + \frac{y_k y_k^\top}{y_k^\top s_k}, a = \frac{M_k s_k}{s_k^\top M_k s_k}, b^\top = s_k^\top M_k \quad (3.67)$$

donc

$$H_{k+1} = \left[M_k + \frac{y_k y_k^\top}{y_k^\top s_k} \right]^{-1} - \frac{\left[M_k + \frac{y_k y_k^\top}{y_k^\top s_k} \right]^{-1} \frac{M_k s_k}{s_k^\top M_k s_k} s_k^\top M_k \left[M_k + \frac{y_k y_k^\top}{y_k^\top s_k} \right]^{-1}}{1 + s_k^\top M_k \left[M_k + \frac{y_k y_k^\top}{y_k^\top s_k} \right]^{-1} \frac{M_k s_k}{s_k^\top M_k s_k}} \quad (3.68)$$

On doit calculer $\left[M_k + \frac{y_k y_k^\top}{y_k^\top s_k} \right]^{-1}$, pour cela on applique la formule de Sherman-Morrison-Woodbury une deuxième fois, on pose

$$A = M_k, a = \frac{y_k}{y_k^\top s_k}, b^\top = y_k^\top \quad (3.69)$$

$$\left[M_k + \frac{y_k y_k^\top}{y_k^\top s_k} \right]^{-1} = [M_k]^{-1} - \frac{[M_k]^{-1} \frac{y_k}{y_k^\top s_k} y_k^\top [M_k]^{-1}}{1 + y_k^\top [M_k]^{-1} \frac{y_k}{y_k^\top s_k}} = [M_k]^{-1} - \frac{[M_k]^{-1} y_k y_k^\top [M_k]^{-1}}{y_k^\top s_k + y_k^\top [M_k]^{-1} y_k}$$

remplaçons cette dernière dans la formule et d'après un calcul on obtient

$$\begin{aligned}
 H_{k+1} &= [M_{k+1}]^{-1} & (3.70) \\
 &= [M_k]^{-1} + \left[1 + y_K^T [M_k]^{-1} \frac{y_k}{s_k^T y_k} \right] - \frac{s_k y_K^T [M_k]^{-1} + [M_k]^{-1} y_K s_K^T}{s_k^T y_k} \\
 &= H_K + \left[1 + \frac{y_k^T H_K y_k}{s_k^T y_k} \right] \frac{s_k s_k^T}{s_k^T y_k} - \frac{s_k y_K^T H_K + H_K y_K s_K^T}{s_k^T y_k}
 \end{aligned}$$

Propriétés

- La condition $y_k^T s_k > 0$ assure la défini-positivité de M_k .
- La condition $y_k^T s_k > 0$ est satisfaite lorsqu' on utilise la recherche linéaire de Wolfe.
- Les directions s_1, s_2, \dots, s_n engendrées par la formule de BFGS sont conjuguées.
- Dans le cas quadratique la méthode de BFGS possède les mêmes propriétés que DFP.
- Dans le cas non quadratique, il faut procéder à des réinitialisations périodiques pour assurer la convergence globale.

Chapitre 4

Convergence globale d'une méthode du gradient à mémoire pour l'optimisation sans contrainte.

4.1 Introduction

Considérons le problème d'optimisation sans contrainte (P) suivant :

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

où f est une fonction différentiable.

Les méthodes du gradient conjugué sont des méthodes importantes pour résoudre le problème (P). Ces méthodes sont de type itératif et génèrent à partir d'un point initial une suite de points $\{x_k\}$ de la façon suivante :

$$x_{k+1} = x_k + \lambda_k d_k, \quad (4.1)$$

la direction d_k est définie par la formule de récurrence suivante:

$$d_k = \begin{cases} -g_1 & \text{si } k = 1, \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 2, \end{cases} \quad (4.2)$$

où $g_k = \nabla f(x_k)$ et λ_k est la longueur de direction obtenu par une recherche linéaire unidimensionnelle et β_k est un scalaire.

Les conditions de Wolfe fortes sont données par :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \omega_1 \lambda_k d_k^T g_k. \quad (4.3)$$

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.

$$|d_k^T g_{k+1}| \leq -\omega_2 d_k^T g_k. \quad (4.4)$$

où $0 < \omega_1 < \omega_2 < 1$.

Il existe plusieurs formules célèbres pour β_k , on peut citer celle de Fletcher-fletcher-Reeves (*FR*), de Polak-Ribiere-Polyak (*PRP*) et de Hestenes-Stiefel (*HS*) (voir [15 ;27, 28 ;18]). leurs formules respectives sont données par :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}. \quad (4.5)$$

$$\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2} \quad (4.6)$$

$$\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} \quad (4.7)$$

où $\|\cdot\|$ est la norme euclidienne et $y_{k-1} = g_k - g_{k-1}$.

Les propriétés de convergence des méthodes *FR*, *PRP* et *HS* ont été étudiées dans beaucoup de références, par exemple [1, 5, 7, 16, 17, 21, 25, 28, 31].

Pendant, si la condition imposée à ω_2 à savoir $\omega_2 < 1$, ni l'une ni l'autre des trois méthodes du gradient conjugué non-linéaires célèbres ci-dessus ne peuvent assurer la descente.

La méthode de descente conjuguée (*CD*) dû à Fletcher [14], où :

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}} \quad (4.8)$$

assure une direction de descente généralement, avec la recherche linéaire de Wolfe forte (4.3), (4.4) avec $\omega_2 < 1$.

Mais la convergence globale de la méthode de (*CD*) est prouvée (voir [8]) seulement dans le cas où la recherche linéaire satisfait la recherche (4.3) et :

$$\omega_2 d_k^T g_k \leq g(x_k + \lambda_k d_k) d_k \leq 0 \quad (4.9)$$

Pour n'importe quelle constante positive ω_2 satisfaisant (4.3). Un exemple est construit dans [8] prouvant que la méthode de la descente conjuguée produit une direction de descente à chaque itération avec λ_k satisfaisant :

$$\omega_2 d_k^T g_k \leq g(x_k + \lambda_k d_k) d_k \leq -\omega_2 d_k^T g_k, \quad (4.10)$$

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.

Récemment, Dai et Yuan [6] ont proposé une nouvelle méthode du gradient conjugué dans laquelle :

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \quad (4.11)$$

Une propriété remarquable de la méthode de *DY* est qu'elle fournit une direction de recherche de descente à chaque itération et converge globalement à condition que λ satisfait la condition de Wolfe (4.3), à savoir :

$$g(x_k + \lambda_k d_k) d_k \geq \omega_2 d_k^T g_k \quad (4.12)$$

Dans [11], Dai et Yuan ont proposé une famille globalement convergente des méthodes du gradient conjuguées, dans la quelle :

$$\beta_k = \frac{\|g_k\|^2}{\lambda \|g_{k-1}\|^2 + (1 - \lambda) d_{k-1}^T y_{k-1}} \quad (4.13)$$

où $\lambda \in [0, 1]$ est un paramètre. Le cas $\lambda \in (-\infty, +\infty)$ a aussi été étudié dans [12], il s'est avéré que la famille des méthodes employant la recherche linéaire qui satisfont (4.3) et (4.10) convergent globalement si les paramètres $\omega_2, \acute{\omega}_2$ et λ sont tels que :

$$\omega_2 - 1 \leq (\omega_2 + \acute{\omega}_2) \lambda \leq 1 \quad (4.14)$$

Dans [23], Nazareth a considéré les formules de *FR*, *PRP*, *HS*, et *DY* comme les quatre principaux compétiteurs pour le scalaire β_k et a proposé une famille de deux paramètres des méthodes de gradient conjugué :

$$\beta_k = \frac{\lambda_k \|g_k\|^2 + (1 - \lambda_k) g_k^T y_{k-1}}{\mu_k \|g_{k-1}\|^2 + (1 - \mu_k) d_{k-1}^T y_{k-1}} \quad (4.15)$$

λ_k et $\mu_k \in [0, 1]$.

Il existe plusieurs types de méthodes itératives telles la méthode de Newton et les méthodes quasi-Newton. Ces méthodes sont connues comme les plus efficaces parmi les méthodes itératives pour la résolution du problème (4.1) car elles sont très performantes.

Cependant, leur inconvénient est qu'elles sont très coûteuse car on doit calculer des matrices Hessienne et leurs inverses. Il est donc difficile d'appliquer ces méthodes pour les problèmes à grande taille.

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.

La méthode (*BFGS*) est une méthode quasi Newton, elle est la plus utilisée pour surmonter cette difficulté. Il existe aussi certains types de méthodes du gradient conjugué qui incluent des méthodes du gradient, la méthode à trois termes [8] qui génère des directions de la façon suivante :

$$d_k = y_{k-1} + \beta_{k1}d_{k-1} + \beta_{k2}d_{k-2}$$

telle que

$$\begin{aligned}\beta_{k1} &= \frac{y_{k-1}^T y_{k-1}}{y_{k-1}^T d_{k-1}} \\ \beta_{k2} &= \frac{y_{k-1}^T y_{k-2}}{y_{k-2}^T d_{k-2}}\end{aligned}$$

où la direction initiale de recherche est choisie arbitrairement pour être une direction arbitraire de descente.

Miele et all [3], Cantrell et Cragg [6] ont proposé pour la première fois, la méthode du gradient à mémoire mais ils n'ont pas montré sa convergence globale. Cette méthode produit une suite $\{x_k\}$ à partir de la formule suivante :

$$x_{k+1} = x_k + S_k$$

et on a à chaque itération on à

$$f(x_{k+1}) < f(x_k).$$

c'est-à-dire la décroissance suffisante.

En utilisant le développement de Taylor on a :

$$f(x_{k+1}) \approx f(x_k) + g_k^T S_k$$

S_k est choisi afin de satisfaire la contrainte suivante :

$$K = (S_k - \sum_{i=1}^m \eta_{ki} S_{k-i})^T (S_k - \sum_{i=1}^m \eta_{ki} S_{k-i})$$

où m dénote le nombre d'itérations, k est une constante donnée et $\eta_{ki} \in \mathbb{R}$ ($i = 1, \dots, m$). l'algorithme résultant s'appelle l'algorithme du gradient à mémoire. Pour résoudre ce problème, on fait appel à la fonction de Lagrange :

$$L(S_k, \gamma_k) = g_k^T S_k + \frac{1}{2\gamma_k} \left\{ (S_k - \sum_{i=1}^m \eta_{ki} S_{k-i})^T (S_k - \sum_{i=1}^m \eta_{ki} S_{k-i}) - K \right\}$$

où $\frac{1}{2\gamma_k}$ est un multiplicateur. La condition nécessaire d'optimalité du premier ordre est donnée par :

$$\nabla_{S_k} L(S_k, \gamma_k) = 0$$

Par conséquent on a

$$S_k = -\gamma_k g_k + \sum_{i=1}^m \eta_{ki} S_{k-i}$$

Les paramètres γ_k, η_{ki} sont choisis tels que la valeur de la fonction objectif diminue.

Nous allons dans la suite montrer que cette nouvelle méthode du gradient à mémoire produit toujours une direction de recherche de descente qui vérifie :

$$g_k^\top d_k < 0 \quad \text{pour tout } k \tag{4.16}$$

et qu'elle converge globalement.

4.2 Direction de descente

Nous définissons une direction de recherche de la forme :

$$d_k = \gamma_k g_k + \frac{1}{m} \sum_{i=1}^m \beta_{ki} d_{k-i} \quad , \quad (k \geq 1) \tag{4.17}$$

Où $\beta_{ki} \in \mathbb{R} (i = 1, \dots, m)$ et $\gamma_k > 0$ sont des paramètres. Normalement la direction de recherche à la première itération est la direction de descente la plus raide avec le paramètre de classement par taille $\gamma_0 > 0$, à savoir $d_0 = -\gamma_0 g_0$. Notons que ces paramètres sont différents de ceux donnés par Miele et al. Notre but principal est de proposer une méthode du gradient à mémoire qui produit une direction de recherche de descente. Nous définissons alors,

$$\begin{aligned} \psi_{k1} &> \text{Max} \left\{ \frac{g_k^\top}{\gamma_k} d_{k-1}; 0 \right\} \quad , i = 1 \\ \psi_{ki} &> \text{Max} \left\{ \frac{g_k^\top}{\gamma_k} d_{k-i}; 0 \right\} \quad , i = 2, \dots, m \end{aligned} \tag{4.18}$$

et

$$\beta_{ki} = \|g_k\|^2 \psi_{ki}^* \tag{4.19}$$

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.

où ψ^* est défini par $\psi^* = \begin{cases} 0 & \text{si } \psi = 0 \\ \frac{1}{\psi} & \text{si } \psi \neq 0 \end{cases}$

Notons que $\psi\psi^* \leq 1$, $\beta_{k_1} > 0$ et $\beta_{k_i} \geq 0$ pour $i = 2, \dots, m$. On a alors le théorème suivant :

Théorème 4.1

soit d_k défini par la méthode du gradient à mémoire (4.17). Si ψ_{k_i} et β_{k_i} satisfont (4.18) et (4.19) pour tout k Alors cette méthode satisfait la condition de descente pour tout k .

Preuve.

De (4.18) et (4.19) , On peut écrire pour $i = 1$

$$\begin{aligned} -\gamma_k \|g_k\|^2 + \beta_{k_1} g_k^T d &\leq -\gamma_k \|g_k\|^2 + \beta_{k_1} \max \{d_{k-1}^T g_k, 0\} \\ &= -\gamma_k \|g_k\|^2 + \|g_k\|^2 \psi_{k_1}^* \max \{d_{k-1}^T g_k, 0\} \\ &\leq -\gamma_k \|g_k\|^2 + \gamma_k \|g_k\|^2 \psi_{k_1}^* \psi_{k_1} \\ &\leq 0 \end{aligned}$$

et pour $i = 2, \dots, m$ on a

$$\begin{aligned} -\gamma_k \|g_k\|^2 + \beta_{k_i} g_k^T d &\leq -\gamma_k \|g_k\|^2 + \beta_{k_i} \max \{d_{k-i}^T g_k, 0\} \quad , i = 2, \dots, m \\ &= -\gamma_k \|g_k\|^2 + \|g_k\|^2 \psi_{k_i}^* \max \{d_{k-i}^T g_k, 0\} \quad , i = 2, \dots, m \\ &\leq -\gamma_k \|g_k\|^2 + \gamma_k \|g_k\|^2 \psi_{k_i}^* \psi_{k_i} \quad , i = 2, \dots, m \\ &\leq 0 \quad , i = 2, \dots, m \end{aligned}$$

Alors nous obtenons

$$\begin{aligned} g_k^T d_k &= -\gamma_k g_k g_k^T + g_k^T \frac{1}{m} \sum_{i=1}^m \beta_{k_i} d_{k-i} \\ &= -\gamma_k \|g_k\|^2 + \frac{1}{m} \sum_{i=1}^m \beta_{k_i} g_k^T d_{k-i} \\ &= \frac{1}{m} \sum_{i=1}^m (-\gamma_k \|g_k\|^2 + \beta_{k_i} g_k^T d_{k-i}) \\ &< 0 \end{aligned}$$

Par conséquent la condition de descente (4.17) est satisfaisant. ■

A présent nous nous intéressons à la condition suffisante de descente, c'est.-à-dire :

$$g_k^T d_k \leq -C \|g_k\|^2 \quad \text{tel que } k \geq 1 \quad (4.20)$$

pour une certaine constante C . Le théorème suivant montre que la condition

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.

(4.20) est réalisée si on pose des conditions plus fortes sur γ_k et ψ_{ki} ($i = 1, \dots, m$).

Théorème 4.2

soit d_k définie par la méthode du gradient à mémoire (4.17). On suppose qu'il existe une constante $\bar{\gamma} > 0$ tels que $\gamma_k \geq \bar{\gamma}$ et ψ_{ki} ($i = 1, \dots, m$) sont choisies telles que

$$\begin{aligned} g_k^\top d_{k-1} + \|g_k\| \|d_{k-1}\| &< \gamma_k \psi_{k1} & (i = 1) \\ g_k^\top d_{k-i} + \|g_k\| \|d_{k-i}\| &< \gamma_k \psi_{ki} & (i = 2, \dots, m) \end{aligned} \quad (4.21)$$

Alors la condition suffisante de descente (4.20) est satisfaite .

Preuve :

nous définissons l'ensemble d'index suivant :

$$I = \{i \mid g_k^\top d_{k-i} > 0\} \quad \text{et } |I| = t \quad (0 \leq t \leq m)$$

Il suit de (4.17) , (4.19) et de (4.21) que

$$\begin{aligned} g_k^\top d_k &= -\gamma_k \|g_k\|^2 + \frac{1}{m} \sum_{i=1}^m g_k^\top d_{k-i} \psi_{ki}^* \|g_k\|^2 \\ &\leq -\gamma_k \|g_k\|^2 + \frac{1}{m} \sum_{i \in I} \frac{g_k^\top d_{k-i}}{\psi_{ki}} \|g_k\|^2 \\ &\leq -\gamma_k \|g_k\|^2 + \frac{1}{m} \sum_{i \in I} \frac{\gamma_k g_k^\top d_{k-i}}{g_k^\top d_{k-i} + \|g_k\| \|d_{k-i}\|} \|g_k\|^2 \\ &\leq -\gamma_k \|g_k\|^2 + \frac{1}{m} \sum_{i \in I} \frac{\gamma_k g_k^\top d_{k-i}}{2 g_k^\top d_{k-i}} \|g_k\|^2 \\ &= -\gamma_k \|g_k\|^2 + \frac{1}{m} \sum_{i \in I} \frac{y_k}{2} \|g_k\|^2 \\ &= -\gamma_k \|g_k\|^2 \left(1 - \frac{t}{2m}\right) \\ &\leq -\frac{y_k}{2} \|g_k\|^2 \\ &\leq -\frac{\bar{\gamma}}{2} \|g_k\|^2 \end{aligned}$$

Ce qui implique que la condition suffisante de descente (4.20) est satisfaite avec $c = \frac{\bar{\gamma}}{2}$.

Notons que la condition (4.21) est plus forte que la condition (4.18). ■

4.3 Convergence globale

Dans cette section nous établissons le théorème de convergence global pour cette nouvelle méthode. On commence par présenter son algorithme en utilisant les conditions de Wolfe fortes.

$$f(x_k) - f(x_k + \lambda_k d_k) \geq -\delta \lambda_k g_k^T d_k \quad (4.22)$$

où $0 < \delta < \sigma < 1$.

$$g(x_k + \lambda_k d_k)^T d_k \geq \sigma g_k^T d_k \quad (4.23)$$

ALGORITHME 4.1 (une nouvelle méthode du gradient)

Etape 0 : Etant donné $x_0 \in \mathbb{R}^n$, $\gamma_0 > 0$ et m , posons $d_0 = -\gamma_0 g_0$ et $k = 0$, si $g_0 = 0$ puis stop.

sinon, passez à l'étape 2.

Etape 1 : Calculer $\gamma_k > 0$ et ψ_{ki} qui satisfont (4.21), définissons β_{ki} par (4.19) et considérons d_k par (4.17)

Etape 2 : Calculer $\lambda_k > 0$ satisfaisant (4.22) et (4.23)

Etape 3 : $x_{k+1} = x_k + \lambda_k d_k$, si $g_{k+1} = 0$ stop.

Etape 4 : $k = k + 1$ allez à l'étape 1.

En choisissant ψ_{ki} et γ_k , nous pouvons obtenir plusieurs variantes des méthodes du gradient à mémoire.

Pour établir le théorème global de convergence pour cette méthode, nous posons les hypothèses suivantes :

Hypothèse 4.1

(A1) f est bornée inférieurement sur \mathbb{R}^n et continûment différentiable dans un voisinage N de l'ensemble $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ au voisinage du point initial x_0 .

(A2) Le gradient $g(x)$ est Lipschitz continu dans N , c.-à-d, il existe une constante $L > 0$ tels que :

$$\|g(x) - g(y)\| \leq L \|x - y\| \quad (4.24)$$

pour tout $x, y \in N$.

Nous avons aussi le lemme bien connu suivant qui a été prouvé par Zoutendijk [12].

Lemme 4.2

supposons que x_0 est un point de départ pour lequel l'hypothèse (4.1) est satisfaite.

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.

Considérons n'importe quelle méthode de la forme (4.18), où le d_k est une direction de recherche de descente et le λ_k remplit les conditions de Wolfe fortes (4.22) et (4.23). Alors

$$\sum_{k \geq 1} \frac{(d_k^T g_k)^2}{\|d_k\|^2} < +\infty \quad (4.25)$$

Ce lemme permet d'établir le théorème suivant :

Théorème 4.3

supposons que x_0 est un point de départ pour lequel l'hypothèse 4.1 est satisfaite. Soit la suite $\{x_k\}$ générée par l'algorithme 4.1. Alors, l'algorithme de la méthode du gradient à mémoire converge dans le sens où

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (4.26)$$

Preuve.

Si l'algorithme ne se termine pas après un nombre fini d'itérations, nous avons $\|g_k\| > 0$ pour tout k .

Puisque toutes les hypothèses du Théorème (4.1) sont satisfaites, les directions de recherche sont de descente, à savoir, $g_k^T d_k < 0$ pour tout k . De (4.18) et (4.19), nous avons :

$$\begin{aligned} \sum_{i=1}^m (\gamma_k \psi_{ki} - g_k^T d_{k-i}) \beta_{ki} & \geq (\gamma_k \psi_{k1} - \gamma_k \psi_{k1}) + \sum_{i=2}^m (\gamma_k \psi_{ki} - g_k^T d_{k-i}) \beta_{ki} \geq \sum_{i=2}^m (\gamma_k \psi_{ki} - g_k^T d_{k-i}) \beta_{ki} \\ & = 0 \end{aligned}$$

Par conséquent il suit de (4.16) et (4.17) que

$$|g_k^T d_k| = -g_k^T d_k \quad (4.27)$$

$$= \gamma_k \|g_k\|^2 - \frac{1}{m} \sum_{i=1}^m \beta_{ki} g_k^T d_{k-i} \quad (4.28)$$

$$= \frac{1}{m} \sum_{i=1}^m (\gamma_k \|g_k\|^2 - \beta_{ki} g_k^T d_{k-i}) \quad (4.29)$$

$$\geq \frac{1}{m} \sum_{i=1}^m (\gamma_k \psi_{ki} - g_k^T d_{k-i}) \beta_{ki} \quad (4.30)$$

$$> 0 \quad (4.31)$$

de (4.17) on peut avoir

$$d_k + \gamma_k g_k = \frac{1}{m} \sum_{i=1}^m \beta_{ki} d_{k-i}$$

ceci implique que

$$\|d_k\|^2 = \left\| \frac{1}{m} \sum_{i=1}^m \beta_{ki} d_{k-i} \right\|^2 - 2 \gamma_k g_k^T d_k - \gamma_k^2 \|g_k\|^2$$

Divisant les deux côtés par $(g_k^T d_k)^2$ et appliquons (4.35), on obtient alors :

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} = \frac{\left\| \frac{1}{m} \sum_{i=1}^m \beta_{ki} d_{k-i} \right\|^2}{(g_k^T d_k)^2} - 2 \gamma_k \frac{g_k^T d_k}{(g_k^T d_k)^2} - \gamma_k^2 \frac{\|g_k\|^2}{(g_k^T d_k)^2} \quad (4.32)$$

$$= \frac{\left\| \frac{1}{m} \sum_{i=1}^m \beta_{ki} d_{k-i} \right\|^2}{(g_k^T d_k)^2} - \gamma_k \frac{2}{g_k^T d_k} - \gamma_k^2 \frac{\|g_k\|^2}{(g_k^T d_k)^2}$$

$$= \frac{\left\| \frac{1}{m} \sum_{i=1}^m \beta_{ki} d_{k-i} \right\|^2}{(g_k^T d_k)^2} - \left(\frac{1}{\|g_k\|} + \gamma_k \frac{\|g_k\|}{g_k^T d_k} \right)^2 + \frac{1}{\|g_k\|^2} \quad (4.33)$$

$$\leq \frac{\left\| \frac{1}{m} \sum_{i=1}^m \beta_{ki} d_{k-i} \right\|^2}{(g_k^T d_k)^2} + \frac{1}{\|g_k\|^2} \quad (4.34)$$

$$\leq \left(\frac{\frac{1}{m} \sum_{i=1}^m \beta_{ki} \|d_{k-i}\|}{|g_k^T d_k|} \right)^2 + \frac{1}{\|g_k\|^2}$$

$$\leq \left(\frac{\frac{1}{m} \sum_{i=1}^m \beta_{ki} \|d_{k-i}\|}{\frac{1}{m} \sum_{i=1}^m (\gamma_k \psi_{ki} - g_k^T d_{k-i}) \beta_{ki}} \right)^2 + \frac{1}{\|g_k\|^2} \quad (4.35)$$

Notons que

$$\gamma_k \psi_{ki} \geq g_k^T d_{k-i} + \|g_k\| \|d_{k-i}\|$$

et multipliant ceci par $\beta_{ki} > 0$ nous avons

$$\beta_{ki} (\gamma_k \psi_{ki} - g_k^T d_{k-i}) \geq \beta_{ki} \|g_k\| \|d_{k-i}\|$$

Sommons l'inégalité ci-dessus, nous obtenons alors

$$\sum_{i=1}^m \beta_{ki} (\gamma_k \psi_{ki} - g_k^T d_{k-i}) \geq \|g_k\| \sum_{i=1}^m \beta_{ki} \|d_{k-i}\|$$

et on a

$$\sum_{i=1}^m \beta_{ki} (\gamma_k \psi_{ki} - g_k^T d_{k-i}) > 0$$

donc

$$\frac{\sum_{i=1}^m \beta_{ki} \|d_{k-i}\|}{\sum_{i=1}^m \beta_{ki} (\gamma_k \psi_{ki} - g_k^T d_{k-i})} \leq \frac{1}{\|g_k\|} \quad (4.36)$$

D'après (4.35) et (4.36) nous avons

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \frac{1}{\|g_k\|} + \frac{1}{\|g_k\|} = \frac{2}{\|g_k\|} \quad \text{pour tout } k. \quad (4.37)$$

Si le théorème n'est pas vrai, il existe une constante $C_1 > 0$ tel que

$$\|g_k\| \geq C_1 \quad \text{pour tout } k. \quad (4.38)$$

Par conséquent il suit de (4.37) et (4.38) que

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \frac{2}{C_1^2}$$

ce qui implique que

$$\frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \frac{C_1^2}{2}$$

Par conséquent nous obtenons

$$\sum_{k \geq 0} \frac{(d_k^T g_k)^2}{\|d_k\|^2} = +\infty$$

Cela contredit la condition de Zoutendijk [12]. ■

Ce théorème implique que si nous choisissons γ_k et ψ_{ki} ($i = 1, \dots, m$) qui remplissent la condition (4.21), alors la méthode converge globalement.

4.3.1 Paramètre de classement par taille

Dans cette section, nous discutons le choix du paramètre γ_k . Comme indiqué dans le théorème 4.3, la condition (4.21) est suffisante pour montrer la propriété de la convergence globale théoriquement. Mais dans la pratique le choix de γ_k affecte l'exécution numérique. Dans notre expérience numérique, nous définissons d'abord $\gamma_k > 0$ puis on donne les ψ_{ki} ($i = 1, \dots, m$) qui

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.

satisfont(4.21). Nous proposons deux choix de γ_k suivant la technique de classement par taille des méthodes quasi-newtoniennes.

Du développement de Taylor, nous avons

$$\nabla^2 f(x_k)S_{k-1} \approx y_{k-1} \text{ ou } S_{k-1} \approx \nabla^2 f(x_k)^{-1}y_{k-1} \quad (4.39)$$

Afin de choisir γ_k tels que le spectre d'intervalle de $(\gamma_k I)$ recouvre l'intervalle de $\nabla^2 f(x_k)^{-1}$, nous employons le quotient de Rayleigh suivant :

$$\begin{aligned} \frac{y_{k-1}^T \nabla^2 f(x_k)^{-1} y_{k-1}}{y_{k-1}^T y_{k-1}} &\approx \frac{y_{k-1}^T S_{k-1}}{y_{k-1}^T y_{k-1}} \\ &= \frac{y_{k-1}^T (\gamma_k I) y_{k-1}}{y_{k-1}^T y_{k-1}} \end{aligned}$$

ici I est la matrice unité. Nous définissons $\gamma_0^{(1)} = 1$ et pour $k \geq 1$ on a

$$\gamma_k^{(1)} = \begin{cases} 1 & \text{si } \frac{y_{k-1}^T S_{k-1}}{y_{k-1}^T y_{k-1}} < \gamma_\varepsilon \\ \frac{y_{k-1}^T S_{k-1}}{y_{k-1}^T y_{k-1}} & \text{si non} \end{cases} \quad (4.40)$$

où γ_ε est un nombre positif très petit. Notons que $\gamma_k^{(1)}$ vaut 1 quand $\frac{y_{k-1}^T S_{k-1}}{y_{k-1}^T y_{k-1}}$ est négatif.

En second lieu nous définissons un autre paramètre de classement par taille proposé par Zhang et al [11]. Ils ont défini Z_{k-1} comme suit

$$Z_{k-1} = y_{k-1} + \frac{\theta_{k-1}}{S_{k-1}^T U_{k-1}} U_{k-1} \quad (4.41)$$

Où U_{k-1} est n'importe quel vecteur tel que $S_{k-1}^T U_{k-1} \neq 0$ et

$$\theta_{k-1} = 6(f(x_{k-1}) - f(x_k)) + 3(g_{k-1} + g_k)^T S_{k-1}$$

On a aussi le théorème suivant

théorème 4.4 (Zhang et Xu [11])

Si le S_{k-1} est suffisamment petit, alors pour n'importe quel vecteur U_{k-1} tel que $S_{k-1}^T U_{k-1} \neq 0$ nous avons

$$\begin{aligned} S_{k-1}^T(\nabla^2 f(x_k)S_{k-1} - y_{k-1}) &= O(\|S_{k-1}\|^3) \\ S_{k-1}^T(\nabla^2 f(x_k)S_{k-1} - Z_{k-1}) &= O(\|S_{k-1}\|^4) \end{aligned}$$

Ce théorème suggère que $S_{k-1}^T Z_{k-1}$ se rapproche de la deuxième courbure de $f(x_k)$ mieux que de $S_{k-1}^T y_{k-1}$. D'une manière semblable au premier cas, nous proposons $\gamma_0^{(2)} = 1$, et pour $k \geq 1$ on a

$$\gamma_k^{(2)} = \begin{cases} 1 & \text{si } \frac{Z_{k-1}^T S_{k-1}}{Z_{k-1}^T Z_{k-1}} < \gamma_\varepsilon \\ \frac{Z_{k-1}^T S_{k-1}}{Z_{k-1}^T Z_{k-1}} & \text{si non} \end{cases} \quad (4.42)$$

4.4 Résultats numériques.

Dans cette section, nous étudions comment le nombre m affecte l'exécution numérique. Nous examinons notre méthode avec divers stockage m qui est changé de 1 à 50.

Le théorème 4.3, nous permet de faire n'importe quel choix de γ_k et de ψ_{ki} qui remplissent la condition (4.21) afin de montrer la convergence globale. Dans notre expérience, nous avons d'abord choisi γ_k et avons considéré ψ_{ki} ($i = 1, \dots, m$) qui remplissent la condition (4.21). Spécifiquement, nous avons choisi deux γ_k comme suit :

Méthode 1. $\gamma_k = \gamma_k^{(1)}$ k choisi par (4.2)

Méthode 2. aucun classement par taille, c.-à-d. $\gamma_k = 1$.

Pour les deux méthodes, nous avons posé $\gamma_\varepsilon = 10^{-15}$ et $\delta = 0,01$, $\sigma = 0,1$. Pour γ_k donné nous avons considéré ψ_{ki} ($i = 1, \dots, m$) défini par

$$\psi_{ki} = \frac{\|g_k\| \|d_{k-i}\| + g_k^T d_{k-i} + n}{\gamma_k} \quad (4.43)$$

Chaque problème a été examiné avec de diverses valeurs de n changeant de $n = 100$ à $n = 100000$. La condition d'arrêt était $\|g_k\|_2 \leq \varepsilon$.

4.4.1 problème 4.1

considérons le problème suivant

*CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.*

$\left\{ \begin{array}{l} \text{minimiser } f(x; y) , f(x; y) = (1 - x)^2 + 100(y - x^2)^2 \text{ [fonction de Rosenbrock]} \\ \text{avec } (x; y) \in \mathbb{R}^2 \end{array} \right. \quad \parallel \parallel \parallel$

la solution du problème 4.1 est $(x; y) = (1, 1)$

Le graphe de $f(x; y)$

Les résultats numériques dans les tableaux 4.1, 4.2, ; ; ; 4.7 et la figure 4.1 et 4.2

Tableau 4. 1 Les résultats numériques pour $\varepsilon = 10^{-5}$ [Méthode 1]

m	n	le nombre d'itérations
2	100	52
2	10000	64
5	1000	64
5	10000	54
5	100000	58
10	100	61
10	100000	60
15	1000	60
15	10000	55
15	100000	56
19	10000	60
19	100000	61
27	10000	54
50	10000	66

Tableau 4. 2 Les résultats numériques pour $\varepsilon = 10^{-6}$ [Méthode 1]

*CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.*

m	n	le nombre d'itérations
2	100	57
2	10000	67
5	1000	58
5	10000	59
5	100000	63
10	100	66
10	100000	65
15	1000	65
15	10000	60
15	100000	61
19	10000	65
19	100000	64
27	10000	59
50	100000	71

Tableau 4. 3 Les résultats numériques pour $\varepsilon = 10^{-7}$ [Méthode 1]

m	n	le nombre d'itérations
2	100	62
2	10000	73
5	1000	64
5	10000	67
5	100000	68
10	100	70
10	100000	72
15	1000	71
15	10000	67
15	100000	67
19	10000	72
19	100000	71
27	10000	65
50	10000	77

Tableau 4.4 Les résultats numériques pour $\varepsilon = 10^{-9}$ [Méthode 1]

*CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.*

m	n	le nombre d'itérations
2	100	68
2	10000	76
5	1000	69
5	10000	70
5	100000	73
10	100	73
10	100000	75
15	1000	78
15	10000	70
15	100000	72
19	10000	76
19	100000	74
27	10000	70
50	100000	80

Tableau 4. 5 Les résultats numériques pour $\varepsilon = 10^{-5}$ [Méthode 1]

m	n	le nombre d'itérations
1	1000	49
1	10000	60
1	100000	54
5	1000	53
5	10000	54
5	100000	58
10	1000	64
10	10000	55
10	100000	60
19	1000	62
19	10000	60
19	100000	61
37	100000	57
50	100000	66

Tableau 4.6 Les résultats numériques pour $\varepsilon = 10^{-7}$ [Méthode 1]

*CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.*

m	n	le nombre d'itérations
1	1000	57
1	10000	70
1	100000	62
5	1000	61
5	10000	64
5	100000	65
10	1000	72
10	10000	63
10	100000	70
19	1000	72
19	10000	69
19	100000	67
37	100000	64
50	100000	63

Tableau 4.7 Les résultats numériques pour $\varepsilon = 10^{-9}$ [Méthode 1]

m	n	le nombre d'itérations
1	1000	66
1	10000	76
1	100000	76
5	1000	69
5	10000	70
5	100000	73
10	1000	78
10	10000	72
10	100000	75
19	1000	78
19	10000	76
19	100000	74
37	100000	69
50	100000	80

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS CONTRAINTE.

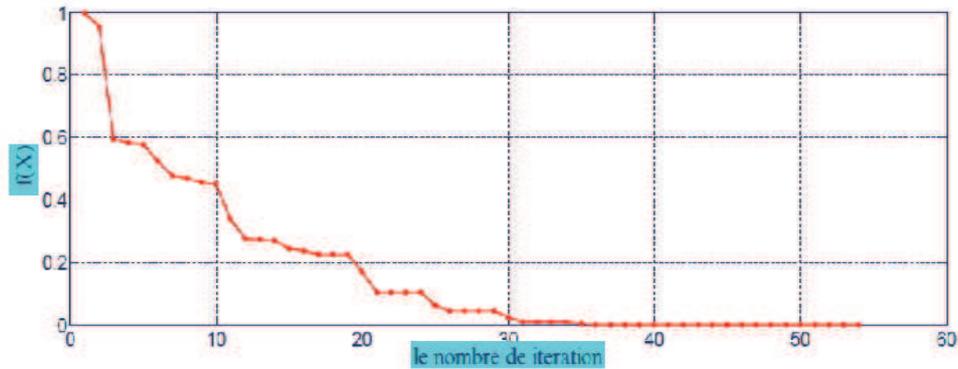


figure 4.1 : Les résultats numériques pour $m = 8$, $n = 10000$ et $\varepsilon = 10^{-5}$

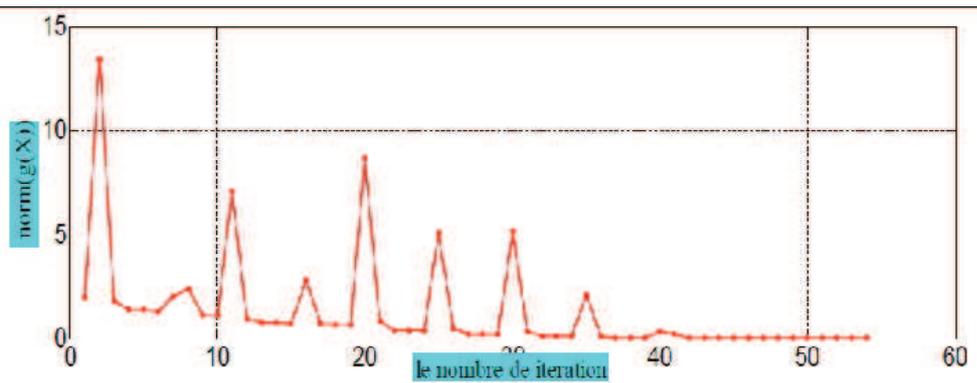


figure 4. 2 : Les résultats numériques pour $m = 8$, $n = 10000$ et $\varepsilon = 10^{-5}$

problème 4.2

considérons le problème suivant

$$\begin{cases} \text{minimiser } f(x; y) , f(x; y) = (3x - 3)^2 + (10y - 1)^2 + x^2y^2 \\ \text{avec } (x; y) \in \mathbb{R}^2 \end{cases} \parallel$$

la solution du problème 4.2 est $(x; y) = (0.9989; 0.0990)$

Les résultats numériques [par la **Méthode 2**] dans les tableaux 4.8 4.9 et 4.10 et la figure 4.3

CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS CONTRAINTE.

tableau 4.8 $\varepsilon = 10^{-5}$ tableau 4.9 $\varepsilon = 10^{-8}$ tableau 4.10 $\varepsilon = 10^{-9}$

m	n	I
2	100	52
2	10000	64
5	1000	64
5	10000	54
5	100000	58
10	100	61
10	100000	60
15	1000	60
15	10000	55
15	100000	56
19	10000	60
19	100000	61
27	10000	54
50	100000	66

m	n	I
2	100	62
2	10000	73
5	1000	64
5	10000	67
5	100000	68
10	100	70
10	100000	72
15	1000	71
15	10000	67
15	100000	67
19	10000	72
19	100000	71
27	10000	65
50	100000	77

m	n	l
2	100	68
2	10000	76
5	1000	69
5	10000	70
5	100000	73
10	100	73
10	100000	75
15	1000	78
15	10000	70
15	100000	72
19	10000	76
19	100000	74
27	10000	70
50	10000	80

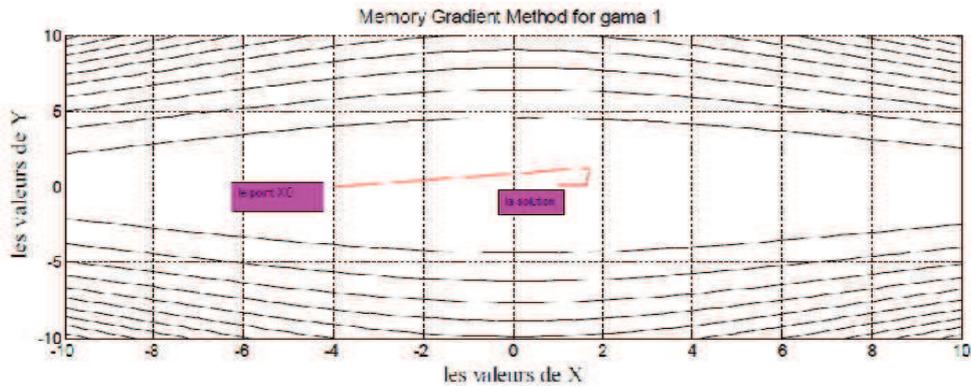


figure 4.3 : Les résultats numériques pour $m = 10$, $n = 10^5$ et $\varepsilon = 10^{-5}$

4.5 Conclusion.

Dans ce travail, nous avons présenté une nouvelle méthode du gradient à mémoire qui donne toujours une direction de recherche de descente, et nous avons prouvé qu'elle converge globalement vers la solution. Nous avons aussi discuté la condition de descente suffisante. Nous avons proposé un choix

*CHAPITRE 4. CONVERGENCE GLOBALE D'UNE MÉTHODE DU
GRADIENT À MÉMOIRE POUR L'OPTIMISATION SANS
CONTRAİNTE.*

concret de paramètres ;. Au cours de l'expérience numérique, nous avons étudié comment les choix de m et y_k pouvaient influencer la performance numérique et nous avons comparé trois types de nos méthodes avec les méthodes *CG* en changeant la valeur de m de 1 à 50. Nous remarquons que le choix des paramètres a un grand effet et que nos méthodes dépassaient en efficacité les autres méthodes à condition de bien choisir la valeur de m . Il reste néanmoins un terrain pour des recherches futures pour trouver théoriquement le bon choix de m .

Bibliographie

- [1] P. Armand (2005), Modification of the Wolfe Line Search Rules to Satisfy the Descent Condition in the Polak-Ribière-Polyak Conjugate Gradient Method, *Journal of Optimization Theory and Applications*, Vol. 131, pp. 103-115.
- [2] L. Armijo (1966), Minimization of function having Lipschitz continuous first partial derivatives, *Pacific Journal of Mathematics*, Vol. 16(1), pp.1-3.
- [3] M. Al-Baali (1985), Descent property and global convergence of the Fletcher-Reeves method with inexact line search. *IMA J. Num. Anal.*, Vol. 5, pp.121-124.
- [4] M. S. Bazaraa, H. D. Sherali, et C. M. Shetty (1993), Nonlinear Programming, Theory and Algorithms, *Wiley-Interscience*.
- [5] M. Barrault et C. Le. Bris (1999), Optimisation Numérique et Différentiation Automatique pour un Problème Industriel, *Notes de cours, Ecole Nationale de Ponts et Chaussées, Paris*.
- [6] M. Bergounioux (2001), Optimisation et Contrôle des Systèmes Linéaires, *Dunod*.
- [7] C. W. Carroll (1961), The created response surface technique for optimizing nonlinear restrained systems, *Operations Res.*, Vol. 9, pp. 169-84.
- [8] A. Cauchy (1847), Analyse mathématique, Méthode générale pour la résolution des systèmes d'équations simultanées, *Comptes Rendus de l'Académie des Sciences de Paris*, t-25, pp. 536-538.
- [9] Y.H. Dai and Y. Yuan (1999), A non linear conjugate gradient with a strong global convergence property, *SIAM J. Optimization*, Vol. 10(1) pp.177-182.
- [10] Y. H. Dai and Y. Yuan (1996), Convergence properties of the Fletcher-Reeves method, *IMA J Numer. Anal.*, Vol.16(2), pp. 155-164.
- [11] Y. H. Dai and Y. Yuan (1996), Convergence properties of the conjugate descent method, *Advances in Mathematics*, 6, pp.552-562.Y.

-
- [12] H. Dai and Y. Yuan (2001), New properties of a new conjugate gradient method, *Numer. Math.*, Vol. 89(2), pp. 83-98.
- [13] Y. H. Dai and Y. Yuan (1998), Some properties of a new conjugate gradient method, in : *Advances in Nonlinear Programming*, ed . Kluwer Academic, Boston, pp. 251-262.
- [14] Y. H. Dai and Y. Yuan (2002), A note on the nonlinear conjugate gradient method Engineering Computing, *Chinese Academy of Sciences*, P. O. Box 2719, Beijing 100080, China)
- [15] Y. H. Dai and Y. Yuan (2001), An Efficient Hybrid Conjugate Gradient Method for Unconstrained Optimization, *Chinese Academy of Sciences*, P.O. Box 2719, Beijing 100080, P.R. China
- [16] T. Diallo (2006), Travail pratique de master, Sujet : Etude et Illustration De Méthodes Itératives D'optimisation Non Linéaire, *Responsable : Benjamin Leroy-Beaubeir, Ecole polytechnique Fédérale de Lausanne.*
- [17] R. Fletcher (1987), Practical methods of optimization, *John Wiley&Sons, Chichster.*
- [18] R. Fletcher and M. J. D. Powell (1963), A rapidly convergent descent method for minimization, *Computer. J.*, 6, pp. 163-168.
- [19] R. Fletcher and C. Reeves (1964), Function minimization by conjugate gradients. *Comput. J.*, 7, pp.149-154.
- [20] A. V. Fiacco and G. P. McCormick (1968), Nonlinear Programming, *John Wiley, New York.*
- [21] J. C. Gilbert (2007), Eléments d'Optimisation Différentiable : Théorie et Algorithmes, *Notes de cours, École Nationale Supérieure de Techniques Avancées, Paris.*
- [22] J.C. Gilbert and J. Nocedal (1992), Global convergence properties of conjugate gradient methods for optimization, *SIAM J. Optimization*. Vol. 3, No.1, pp. 21-42.
- [23] A.A. Goldstein (1967), Constructive Real Analysis, *A Harper International Edition.*
- [24] A. A. Goldstein (1965), On steepest descent, *SIAM J. on Control A*, Vol. 3, No. 1, pp. 147-151.
- [25] A.A. Goldstein and J.F. Price (1969), An effective algorithm for minimization, *Num.Math.*, 10, pp. 184-189.
- [26] L. Grippo and S. Lucidi (1997), A Globally Convergent Version of the Polak-Ribière Conjugate Gradient Method, *Mathematical Programming*, Vol. 78, pp. 375-391.

-
- [27] W.W. Hager et H. Zhang (2005), A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM J. Optim.*, 16, pp. 170-192.
- [28] M. R. Hestenes and E.L. Stiefel (1952), Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards Sect.*, 5(49), pp. 409-436.
- [29] Y.F. Hu and C. Story (1991), Global convergence result for conjugate gradient methods, *JOTA*, 71(2), pp. 399-405.
- [30] Z. Li, J. Chen and N. Deng (1998), A New Conjugate Gradient Method and its Global Convergence Properties, *Systems Science and Mathematical Sciences*, Vol. 11, pp. 53-60.
- [31] Laetitia ANDRIEU (2004) Optimisation sous contrainte en probabilité, THESE présentée pour l'obtention du titre de Docteur de l'École Nationale des Ponts et Chaussées, décembre
- [32] G.H. Liu and J.Y. Han and H.X. Yin (1995), Global convergence of the Fletcher-Reeves algorithm with an inexact line search, *Appl. math. J. Chinese Univ. Ser. B*, 10, pp. 75-82.
- [33] D. G. Luenberger (1969), Optimization by Vector Space Methods, *John Wiley and Sons, Inc. New York*.
- [34] M. Minoux (1983), Programmation Mathématique, Théorie et Algorithmes, tome 1, *Dunod*.
- [35] J.J.Moré, B.S.Garbow and K.E. Hillstrom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* 7 (1981) 17-41.
- [36] A.S. Nemirovsky et D.B. Yudin (1983), Problem Complexity and Method Efficiency, *New York, Wiley*.
- [37] J. Nocedal (1991), Theory of algorithm for unconstrained optimization, *Acta Numerica*, pp. 199-242.
- [38] E. Polak and G. Ribière (1969), Note sur la convergence de directions conjuguées, *Rev. Française Informat. Recherche Operationelle*, 3e année, 16, pp. 35-43.
- [39] B.T. Polyak (1969), The conjugate gradient method in extremem problems, *Comput. Math. Math. Phys.*, 9, pp. 94-112.
- [40] M.J.D. Powell (1986), Convergence properties of algorithms for nonlinear optimization, *SIAM rev.*, 28, pp. 487-500.
- [41] M.J.D. Powell (1984), Non convex minimzation calculation and the conjugate gradient method, in : *Lecture Notes in Mathematics 1066 (Springer, Berlin)*, pp. 122-141.

-
- [42] M.J.D. Powell (1971), On the convergence of the variable metric algorithm; *J. Inst. Math. Appl*, 7, pp. 21-36.
- [43] M.J.D. Powell (1977), Restart procedures for the conjugate gradient method, *Math Programming*, 2, pp 241-254.
- [44] D. Touati-Ahmed and C. Storey (1990), Efficient hybrid conjugate gradient techniques, *JOTA*, 64, pp. 379-397.
- [45] P. Wolfe (1961), A duality theorem for nonlinear programming, *Quart. Appl. Math.*, 19, pp. 239-244.
- [46] P. Wolfe (1969), Convergence conditions for ascent methods, *SIAM Review*, 11, pp. 226-235.
- [47] P. Wolfe (1971), Conditions for ascent methods some corrections 2, *SIAM Review*, 13, pp.185-188.
- [48] P. Wolfe (1963), Methods of Nonlinear Programming, in *Recent Advances in Mathematical Programming (Eds R. L. Graves and P. Wolfe)*, McGraw-Hill, New York.
- [49] Y.X.Yuan (1993), Numerical Methods for Non linear Programming, *Shanghai scientific & Technical Publishers (in Chinese)*.
- [50] G. Zoutendijk (1970), Nonlinear Programming Computational Methods, *Integer and Nonlinear Programming*, North Holland, Amsterdam.
- [51] G. Zoutendijk (1960), Methods of Feasible Directions, *Elsevier, Amsterdam*
- [52] L. Adams and J. L. Nazareth, eds., Linear and Nonlinear Conjugate Gradient-Related Methods, SIAM, 1996.
- [53] M.Al-Baali, "Descent property and global convergence of the Fletcher-Reeves method with inexact line search," *IMA Journal of Numerical Analysis*, vol.5, pp.121-124, 1985.
- [54] E.E Cragg and A.V.Levy, "Study on a Supermemory Gradient Method for the minimization of function," *Journal of Optimization Theory and Application*, vol.4, pp.191- 205, 1969.
- [55] Y.H.Dai and Y.Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol.10, pp.177-182, 1999.
- [56] J.C.Gilbert and J.Nocedal, "Global convergence properties of conjugate gradient method for optimization," *SIAM Journal on Optimization*, vol.2, pp.21-42, 1992.
- [57] A.Miele and J.W.Cantrell, "Study on aMemory GradientMethod for the minimization of function," *Journal of Optimization Theory and Application*, vol3, pp. 459-470, 1969.

- [58] J.J.Moré, B.S.Garbow, and K. E. Hillstom, “Testing unconstrained optimization software,” *ACM Transactions on Mathematical Software*, vol.7, pp.17-41, 1981.
- [59] J.L. Nazareth, “A conjugate direction algorithm for unconstrained minimization without line searches,” *Journal of Optimization Theory and Application*, vol.23, pp. 373-387, 1977.
- [60] J.Nocedal, “Updating quasi-Newton matrices with limited storage, ” *Mathematics of Computation*, vol.35, pp. 773-782, 1980.
- [61] J.Nocedal and S.J.Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer Verlag :New York, 1999.
- [62] J. Zhang and C. Xu, “Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations,” *Journal of Computational and Applied Mathematics*, vol.137, pp. 269–278, 2001.
- [63] G.Zoutendijk, *Nonlinear Programming, Computational Methods*, in *Integer and Nonlinear Programming*, J.Abadie,ed., North-Holland, Amsterdam, (1970), pp. 37- 86

BIBLIOGRAPHIE
