

وزارة التعليم العالي والبحث العلمي

Université Badji Mokhtar  
Annaba  
Badji Mokhtar University-  
Annaba



جامعة باجي مختار  
عنابة

Faculté des Sciences  
Département de Mathématiques

## THESE

Présentée en vue de l'obtention du diplôme de  
Doctorat en Mathématiques  
**Option** : Mathématiques appliquées

Convergence des méthodes du gradient conjugué  
avec la recherche linéaire non monotone

**Par :**  
TAHAR Bouali

**Sous la direction de**  
Pr. YAMINA Laskri

**Devant le jury**

**PRÉSIDENT :** Benzine Rachid

Pr. EPST Annaba

**EXAMINATEUR :** Djebbar Bachir

Pr. USTO Oran

**EXAMINATEUR :** Ellaggoun Fateh

M.C.A. Université 8 Mai Guelma

**EXAMINATEUR :** Bouras M<sup>ed</sup> Cherif

M.C.A. Université Badji Mokhtar Annaba

**Année : 2014-2015**

## *Remerciements*

*Je tiens à adresser mes remerciements les plus chaleureux et ma profonde gratitude à mon encadreur Madame **Laskri Yamina** professeur à l'Université de Annaba pour m'avoir proposé le sujet de cette thèse. 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' Ecole Préparatoire aux Sciences et Techniques de Annaba, pour avoir bien voulu me faire l'honneur d'accepter de présider le jury.*

*De même je remercie Monsieur **Djabbar Bachir**, professeur à l'Université des Sciences et de la Technologie d'Oran, et Monsieur **Ellaggoun Fatah** maître de conférences classe A à l'Université de 8 Mai Guelma et **Bouras Mohammed Cherif**, maître de conférences classe A 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 au département au mathématiques qui ont rendu mon séjour agréable.*

---

# Résumé

Le but de cette thèse est de présenter des nouvelles classes de recherches linéaires non monotones lesquelles appliquées à la méthode du gradient conjugué donnent de bons résultats de convergence et assure la condition de descente suffisante. Plus particulièrement, appliquées à la méthode du gradient conjugué de Fletcher Reeves (FR) avec des fonctions fortement convexes assurent la convergence globale et donnent de bonnes performances numériques.

**Mots clés :** Gradient conjugué, Algorithme, Convergence globale, Recherche linéaire inexacte, Règle d'Armijo, Règle de Wolf, Méthode de Hestenes-Stiefel, Méthode de Fletcher-Reeves, Méthode de Polak-Ribière-Polyak, Méthode de la descente conjuguée, Méthode de Gradient conjugué non monotone, La recherche linéaire non monotone.

---

# Abstract

The purpose of this thesis is to present a new class of nonmonotone line search which applied to gradient conjugate method give a good results of convergence and assure the condition of sufficient descent. More particularly, applied to the method of conjugate gradient of Fletcher Reeves (FR) with strongly convex functions assure the global convergence and give a good numerical performance.

**Key words :** Conjugate gradient, Algorithm, Global convergence, Inexact line search, Armijo line search, Wolf line search, Hestenes-Stiefel Method, Fletcher-Reeves Method, Polak-Ribière-Polyak Method, Conjugate descent Method, Nonmonotone conjugate gradient method, nonmonotone line search.

## ملخص

الهدف من هذه الأطروحة هو تقديم أصناف جديدة من البحث الخطي غير الرتيب المطبقة على طريقة التدرج السلمي المشتق وتعطي نتائج جيدة للتقارب وتضمن الشرط اللازم للهبوط ، وبصورة خاصة تلك المطبقة على طريقة التدرج السلمي المشتق لفلتشر ريفز باستخدام دوال محدبة بقوة تضمن التقارب الشمولي وتعطي فعالية عددية جيدة.

## الكلمات المفتاحية:

التدرج السلمي المشتق، خوارزمية، التقارب الشمولي، البحث الخطي غير الدقيق، معيار أرميجو، معيار فولف، طريقة إستبان استيفال، طريقة فلتشر ريفز، طريقة بولاك ريبيار بولاياك، طريقة الهبوط المشتق، طرق التدرج السلمي المشتق غير الرتيب، البحث الخطي غير الرتيب.

# Table des matières

<b>1</b>	<b>Préliminaires : Optimisation sans contraintes</b>	<b>11</b>
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 . . . . .	13
1.2.1	Aspect général des algorithmes . . . . .	14
1.2.2	Fonctions multivoques et algorithmes . . . . .	14
1.3	Convergence globale des algorithmes . . . . .	16
1.4	Modes et vitesse de convergence . . . . .	16
1.5	Conditions d'optimalité des problèmes d'optimisation sans contraintes . . . . .	17
1.5.1	Conditions nécessaires d'optimalité . . . . .	18
1.5.2	Conditions suffisantes d'optimalité . . . . .	20
1.6	Fonctions convexes . . . . .	20
<b>2</b>	<b>Optimisation unidimensionnelle ou recherches linéaires non monotones</b>	<b>22</b>
2.1	Méthodes à directions de descente . . . . .	23
2.1.1	Principe général . . . . .	23
2.1.2	Direction de descente . . . . .	23
2.2	Recherche linéaire . . . . .	26
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
2.3.1	Les inconvénients des recherches linéaires exactes . . . . .	31
2.4	Recherches linéaires inexactes . . . . .	31
2.4.1	La règle d'Armijo . . . . .	33
2.4.2	La règle de Goldstein&Price. . . . .	37
2.4.3	La règle de Wolfe . . . . .	41
2.5	Recherches lineaires non monotones . . . . .	47
2.5.1	La recherche linéaire non monotone d'Armijo . . . . .	48
2.5.2	La recherche linéaire non monotone de Wolfe . . . . .	49
2.5.3	La recherche linéaire non monotone de Goldstein&Price . . . . .	50

**3 La convergence des méthodes du gradient conjugué avec la recherche linéaire non monotone** **52**

3.1	Condition de Zoutendijk . . . . .	53
3.2	La méthode du gradient conjugué . . . . .	56
3.3	Le principe général d'une méthode à directions conjuguées . . . . .	57
3.3.1	Description de la méthode . . . . .	57
3.4	Méthode du gradient conjugué dans le cas quadratique . . . . .	59
3.4.1	Algorithme de La méthode du gradient conjugué pour les fonctions quadratiques . . . . .	59
3.4.2	Les avantages de la méthode du gradient conjugué linéaire . . . . .	61
3.5	Synthèse des résultats de convergence des méthodes du gradient conjugué . . . . .	61
3.5.1	Résultats de Convergence du Gradient Conjugué, version Fletcher-Reves . . . . .	65
3.5.2	Résultats de Convergence du Gradient Conjugué, version Polyak Ribière et Polyak . . . . .	68
3.5.3	Résultats de convergence du gradient conjugué, version Descente conjuguée . . . . .	71

3.5.4	Résultats de convergence du gradient conjugué version(Dai-Yuan) . . . . .	74
3.6	Analyse de la convergence des méthodes du gradient conjugué avec la recherche linéaire non monotone . . . . .	77
3.6.1	Résultats de Convergence du Gradient Conjugué avec la recherche linéaire non monotone d'Armijo . . . . .	77
3.6.2	Résultats de convergence du gradient conjugué avec la recherche linéaire non monotone de Wolfe . . . . .	84
<b>4</b>	<b>Convergence de la méthode de FR avec une nouvelle classe de recherche linéaire non monotone</b>	<b>94</b>
4.1	Introduction . . . . .	95
4.2	convergence Globale . . . . .	105
4.3	Expériences numériques . . . . .	108



# Introduction

L'optimisation est une branche des mathématiques et de l'informatique en tant que disciplines, cherchant à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à déterminer quelles sont la ou les solution(s) satisfaisant un objectif quantitatif tout en respectant d'éventuelles contraintes.

L'optimisation joue un rôle important en recherche opérationnelle (domaine à la frontière entre l'informatique, les mathématiques et l'économie), dans les mathématiques appliquées (fondamentales pour l'industrie et l'ingénierie), en analyse et en analyse numérique, en statistique pour l'estimation du maximum de vraisemblance d'une distribution, pour la recherche de stratégies dans le cadre de la théorie des jeux, ou encore en théorie du contrôle et de la commande.

Aujourd'hui, tous les systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés. La qualité des résultats et des prédictions dépend de la pertinence du modèle, de l'efficacité de l'algorithme et des moyens pour le traitement numérique.

Les domaines d'applications sont extrêmement variés : optimisation d'un trajet, de la forme d'un objet, d'un prix de vente, d'une réaction chimique, du contrôle aérien, du rendement d'un appareil, du fonctionnement d'un moteur, de la gestion des lignes ferroviaires, du choix des investissements économiques, de la construction d'un navire, etc. L'optimisation de ces systèmes permet de trouver une configuration idéale, d'obtenir un gain d'effort, de temps, d'argent, d'énergie, de matière première, ou encore de satisfaction.

Très loin de constituer une liste exhaustive, ces quelques exemples attestent de la variété des formulations et préfigure la diversité des outils mathématiques susceptibles de résoudre ces problèmes.

Plus formellement, l'optimisation est l'étude des problèmes qui s'expriment de la manière suivante.

Étant donné une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , trouver un élément  $\bar{x}$  de  $\mathbb{R}^n$  tel que  $f(\bar{x}) \leq f(x)$  pour tout  $x \in \mathbb{R}^n$ .

On dit que l'on cherche à minimiser la fonction  $f$  sur l'ensemble  $\mathbb{R}^n$ . La fonction  $f$  porte divers noms : fonction-coût ou simplement coût, fonction-objectif ou simplement objectif, critère, etc.

Cela permet de varier le vocabulaire.

L'ensemble  $\Omega$  des points de  $\mathbb{R}$  qui satisfont cette condition est appelé l'ensemble admissible et les points de  $\Omega$  sont appelés les points admissibles du problème. On dit que le problème est réalisable si  $\Omega$  est non vide (l'ensemble admissible étant souvent défini de manière implicite, son caractère non vide n'est pas nécessairement évident, ce qui justifie le besoin de ce concept de réalisabilité).

Le point  $\bar{x}$  est appelé solution du problème d'optimisation (ou minimum ou minimiseur). On l'appelle aussi parfois une solution globale pour le distinguer des notions locales introduites ci-dessous.

L'optimisation est découpée en sous-disciplines qui se chevauchent, suivant la forme de la fonction objectif et celle des contraintes : l'optimisation en dimension finie ou infinie (on parle ici de la dimension de l'espace vectoriel des variables à optimiser), l'optimisation continue ou combinatoire (les variables à optimiser sont discrètes dans ce dernier cas), l'optimisation différentiable ou non lisse (on qualifie ici la régularité des fonctions définissant le problème), l'optimisation linéaire (fonctions affines), quadratique (objectif quadratique et contraintes affines), semi-définie positive (la variable à optimiser est une matrice dont on requiert la semi-définie positivité), copositive (la variable à optimiser est une matrice dont on requiert la copositivité), conique (généralisation des disciplines précédentes, dans laquelle on minimise une fonction linéaire sur l'intersection d'un cône et d'un sous-espace affine), convexe (fonctions convexes), non linéaire, la commande optimale, l'optimisation stochastique (en) et robuste (en) (présence d'aléas), l'optimisation multicritère (un compromis entre plusieurs objectifs contradictoires est recherché), l'optimisation algébrique (fonctions polynomiales), l'optimisation bi-niveaux, l'optimisation sous contraintes de complémentarité, l'optimisation disjonctive (l'ensemble admissible est une réunion d'ensembles), etc.

Cette abondance de disciplines provient du fait que pratiquement toute classe de problèmes modélisables peut conduire à un problème d'optimisation, pourvu que l'on y introduise des paramètres à optimiser. Par ailleurs, les conditions d'optimalité de ces problèmes d'optimisation apportent parfois des expressions mathématiques originales qui, par le mécanisme précédent, conduisent à leur tour à de nouveaux problèmes d'optimisation.

L'optimisation linéaire étudie le cas où la fonction objectif et les contraintes caractérisant l'ensemble  $A$  sont linéaires. C'est une méthode très employée pour établir les programmes des raffineries pétrolières, mais aussi pour déterminer la composition la plus rentable d'un mélange salé, sous contraintes, à partir des prix de marché du moment.

L'optimisation linéaire en nombres entiers étudie les problèmes d'optimisation linéaire dans lesquels certaines ou toutes les variables sont contraintes de prendre des valeurs entières. Ces problèmes peuvent être résolus par différentes méthodes : séparation et évaluation, méthode des plans sécants.

L'optimisation quadratique étudie le cas où la fonction objectif est une forme quadratique (avec contraintes linéaires)

L'optimisation non linéaire étudie le cas général dans lequel l'objectif ou les contraintes (ou les deux) contiennent des parties non linéaires, éventuellement non-convexes.

L'optimisation stochastique étudie le cas dans lequel certaines des contraintes dépendent de variables aléatoires. En optimisation robuste, les aléas sont supposés être situés dans des intervalles autour de positions nominales et on cherche à optimiser le système soumis à de tels aléas, dans le pire des cas.

La programmation dynamique utilise la propriété qu'une solution se compose nécessairement de sous-solutions optimales (attention : le contraire n'est pas vrai en général) pour décomposer le problème en évitant l'explosion combinatoire. Elle est utilisable lorsque la fonction objectif est une somme de fonctions monotones croissantes dont les arguments sont des inconnues distinctes. C'est la programmation dynamique qui permet par exemple :

- - aux avionneurs de trouver les plans de décollage optimaux de leurs engins,
- aux ingénieurs de bassin de répartir la production minière entre leurs différents puits,
- aux producteurs d'électricité de planifier la marche des usines hydroélectriques,
- aux media planners de répartir efficacement un budget de publicité entre différents supports.

Formellement on peut écrire ce problème noté  $(P)$  de la manière suivante :

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

Remarquons toute fois que comme on a

$$\sup_{x \in \mathbb{R}^n} f(x) = - \inf_{x \in \mathbb{R}^n} (-f(x)) \quad (0.1)$$

alors le problème de maximisation d'une fonction  $f$  est équivalent au problème de minimisation de  $-f$ . L'équivalence veut dire ici que les solutions sont les mêmes et que les valeurs optimales sont opposées. En particulier, une méthode pour analyser et résoudre un problème de minimisation pourra être utilisée pour analyser et résoudre un problème de maximisation.

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.

Cette méthode a été découverte en 1952 par Hestenes et Steifel [32] 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 Reeves [21] (*méthode de Fletcher-Reeves*) puis en 1969 par Polak, Ribière [45] et Ployak [46] (*méthode de Polak-Ribière-Ployak*). Une autre variante a été étudiée en 1987 par Fletcher [20] (*Méthode de la descente conjuguée*).

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

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

Le pas  $\lambda_k \in \mathbb{R}$  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  $\beta_k$  définissent les différentes formes du gradient conjugué.

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

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

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

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

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \text{ gradient conjugué variante de Dai-Yuan.} \quad (0.7)$$

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})}, \text{ gradient conjugué variante Hestenes-Stiefel.} \quad (0.8)$$

$$\beta_k^{HZ} = \frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}} - 2g_k^T d_{k-1} \frac{\|y_{k-1}\|^2}{(d_{k-1}^T g_{k-1})^2}, \text{ gradient conjugué variante de Hager-Zhang} \quad (0.9)$$

Le but de cette thèse est de faire une étude de la convergence des méthodes du gradient conjugué avec la recherche linéaire non monotone.

La technique de la recherche linéaire non monotone a été proposée en premier par Grippo, Lampariello, and Lucidi [30, 1986] et a connu de nombreuses applications aussi bien en optimisation sans contraintes qu'en optimisation avec contraintes [30], [62]. Bien que prometteuse, la recherche sur le thème de la recherche linéaire non monotone est toujours dans ses débuts. comme on le verra par la suite, les recherches linéaires non monotone ont la particularité de ne pas imposer la condition de décroissance suffisante à la fonction objectif.

Cette thèse comporte une introduction, quatre chapitres, des simulations numériques et une conclusion .

## Chapitre 1

On introduit dans ce chapitre les notions préliminaires de base concernant l'optimisation sans contraintes. On commence par citer quelques éléments de Topologie et de calcul différentiel puis on y explicite quelques notions sur les problèmes de minimisation sans contraintes et leurs algorithmes.

## Chapitre 2

Ce chapitre comporte les grandes lignes des méthodes d'optimisation sans contraintes basées sur les directions de descente et les recherches linéaires. Ces méthodes génèrent à partir d'un point initial  $x_0$  une suite de points  $\{x_k\}_{k \in \mathbb{N}}$  de la façon suivante :

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

On suppose connaître la direction de descente  $d_k$  au point  $x_k$ . La recherche linéaire consiste à trouver  $\lambda_k$  de façon à diminuer la fonction  $f$  suffisamment le long de cette direction. On insistera dans ce chapitre sur les recherches linéaires non monotones avec lesquelles nous allons établir des résultats de convergence de certaines méthodes du gradient conjugué.

## Chapitre 3

Ce chapitre est consacré aux différents résultats de convergence avec la recherche linéaire non monotone. On commence d'abord par faire une synthèse des résultats de convergence concernant les différentes variantes de la méthode du gradient conjugué avec la recherche linéaire inexacte . 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, de Fletcher et enfin celle de Dai et Yuan [12] et [17].

Ensuite on propose une synthèse sur les résultats de convergence des méthodes du gradient conjugué avec les recherches linéaires non monotones d'Armijo [7] , [10] , [11] , [30] ... et de Wolfe [31] , [36] , [37] , [62] .... Cette partie comporte le premier résultat original [53] de cette thèse dans lequel une nouvelle classe de recherche linéaire non monotone de Wolfe (RLN1) est présentée.

## Chapitre 4

Ce chapitre contient un résultat original, on propose une étude sur la convergence de la méthode de Fletcher Reeves avec une nouvelle classe de recherche linéaire non monotone de Wolfe (RLN2) pour des fonctions fortement convexes.

# Chapitre 1

## Préliminaires : Optimisation sans contraintes

- 
- 1- Différentiabilité.
  - 2- Les problèmes de minimisation sans contraintes et leurs algorithmes.
  - 3- Convergence globale des algorithmes.
  - 4- Modes et vitesse de convergence.
  - 5- Conditions d'optimalité des problèmes de minimisation sans contraintes.
  - 6- Fonctions convexes.
-



Ce chapitre sera consacré à certaines notions de base et théorèmes concernant les méthodes itératives d'optimisation non linéaire en général et les méthodes du gradient conjugué en particulier.

### 1.1 Différentiabilité

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

#### 1.1.1 Gradient, Hessien

##### Définition 1.1

- On note par

$$(\nabla f(x))^T = \frac{\partial f}{\partial x}(x) = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)(x) \quad (1.1)$$

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

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

$$H(x) = \nabla(\nabla^T f)(x) = \nabla^2 f(x) = \left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right)(x), \quad i = 1, \dots, n, \quad j = 1, \dots, n \quad (1.2)$$

##### 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

$$\forall x \in \mathbb{R}^n : x^\top A x \geq 0 \quad (1.3)$$

elle est dite *définie positive* si

$$\forall x \in \mathbb{R}_*^n : x^\top A x > 0 \quad (1.4)$$

### 1.1.2 Dérivée directionnelle

#### Définition 1.3 [5]

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 :

$$\frac{f(x + hd) - f(x)}{h} \text{ lorsque } h \text{ tend vers } 0. \quad (1.5)$$

Autrement dit :

$$\delta f(x, d) = \lim_{h \rightarrow 0} \frac{f(x + hd) - f(x)}{h} = \nabla^T f(x) d \quad (1.6)$$

• 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.

## 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 ( $P$ ) cité précédemment. L'étude de ces problèmes est importante pour des raisons diverses. Beaucoup de problèmes d'optimisation avec contraintes sont transformés en des suites de problèmes d'optimisation sans contraintes (multiplicateur de Lagrange, méthodes des pénalités, ...). L'étude des problèmes d'optimisation sans contraintes trouve aussi des applications dans la résolution des systèmes différentiels 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 mais il faut montrer de plus qu'il est plus performant que ceux déjà connus.

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

### 1.2.1 Aspect général des algorithmes

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

$$x_{k+1} = x_k + \lambda_k d_k \tag{1.7}$$

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

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

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

Pour s'approcher de la solution optimale du problème ( $P$ ) (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  $\mathbb{R}^n$ .

Etant 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  $A$  appliquée

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 1.4.** 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

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

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 a appelé ensemble des solutions  $\Omega$ .

Voici ci-dessous quelques exemples typiques de cet ensemble.

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

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

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

**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  $\Omega$  qui vérifie :

$\alpha(y) < \alpha(x)$  pour  $y \in B(x)$  et  $x \notin \Omega$ . 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 compact. Alors, ou bien l'algorithme s'arrête à un nombre fini d'itérations pour un point de  $\Omega$ , ou tous les points d'accumulation de  $\{x_k\}$  appartiennent à  $\Omega$ .

### 1.3 Convergence globale des algorithmes

#### Définition 1.5

Nous dirons qu'un algorithme décrit par une application multivoque  $A$ , est *globalement convergent* (ou encore : possède la propriété de convergence globale) si, *quelque 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.

### 1.4 Modes et vitesse de convergence

La convergence globale d'un 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  $\varepsilon$  près ( $\varepsilon$  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 petit 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 un indice d'efficacité appelé sa *vitesse de conver-*

gence.

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  $\{x_k\}$  convergent vers  $x^*$ .

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

On dit que la convergence est *linéaire* et  $\lambda$  est le taux *de convergence* associé.

$$\cdot \text{Si } \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \longrightarrow 0 \text{ quand } k \longrightarrow \infty$$

on dit que la convergence est *superlinéaire*.

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

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} < +\infty \quad (1.8)$$

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

En particulier si

$$\limsup_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} < +\infty \quad (1.9)$$

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

## 1.5 Conditions d'optimalité des problèmes d'optimisation sans contraintes

**Définition 1.6.** *Considérons le problème de minimisation sans contraintes (P).*

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

$$f(x_*) \leq f(x), \quad \forall x \in \mathbb{R}^n. \quad (1.10)$$

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

$$f(x_*) \leq f(x), \forall x \in V_\varepsilon(x_*). \quad (1.11)$$

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

$$f(x_*) < f(x), \forall x \in V_\varepsilon(x_*) \quad \text{et} \quad x \neq x_*. \quad (1.12)$$

### 1.5.1 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

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

et par conséquent on obtient :

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

**Théorème 1.3.** 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 :

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

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

**Théorème 1.4.** 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.**[41]

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 :

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

Comme  $\bar{x}$  est un minimum local alors  $f(\bar{x} + \alpha d) \geq f(\bar{x})$  pour  $\alpha$  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. ■



## 1.5.2 Conditions suffisantes d'optimalité

**Théorème 1.5.** *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 [41]**

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

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})^t(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^t H(\bar{x})(x - \bar{x}) + \|x - \bar{x}\|^2 \lambda(\bar{x}; x - \bar{x}) \quad (1.13)$$

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 (1.13) :

$$\frac{f(x_k) - f(\bar{x})}{\|x_k - \bar{x}\|^2} = \frac{1}{2} d_k^t H(\bar{x}) d_k + \lambda(\bar{x}; x_k - \bar{x}) \leq 0, \quad \forall k \quad (1.14)$$

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. ■

## 1.6 Fonctions convexes

**Définition 1.7.** *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 point  $x_1$  et  $x_2 \in \mathbb{R}^n$  et pour tout  $t \in [0, 1]$ .

b) Si l'inégalité précédente est stricte pour tous 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 :  $f(x_2) \geq f(x_1)$ .

**Théorème 1.6.** [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  $\nabla f(x_*) = 0$ .

**Théorème 1.7.** [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  $f$ .

**Remarque 1.2.** Les théorèmes 1.5 et 1.6 demeurent vrais si on remplace  $\mathbb{R}^n$  par un ouvert  $S$  de  $\mathbb{R}^n$ .

**Remarque 1.3.** Dans le cas où  $f$  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].

## Chapitre 2

# Optimisation unidimensionnelle ou recherches linéaires non monotones

- 
- 1- Méthodes à directions de descente.
  - 2- Recherche linéaire
  - 3- Recherches linéaires exactes.
  - 4- Recherches linéaires inexactes.
  - 5- Recherches linéaire non monotones.
-

## 2.1 Méthodes à directions de descente

### 2.1.1 Principe général

Considérons le problème d'optimisation sans contrainte :

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

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

On note aussi par  $\nabla f(x)$  et  $\nabla^2 f(x)$  le gradient et le hessien de  $f$  en  $x$  respectivement.

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

### 2.1.2 Direction de descente

**Définition 2.1.** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$   $d$  un vecteur non nul de  $\mathbb{R}^n$  est dit une direction de descente

si il existe un réel  $\delta > 0$  tel que pour tout  $\lambda$  dans l'intervalle  $]0, \delta[$ , on a

$$f(x + \lambda d) < f(x)$$

**Théorème 2.1.** Si  $f$  est différentiable en un point  $x \in \mathbb{R}^n$  et  $d \in \mathbb{R}^n$  telle que

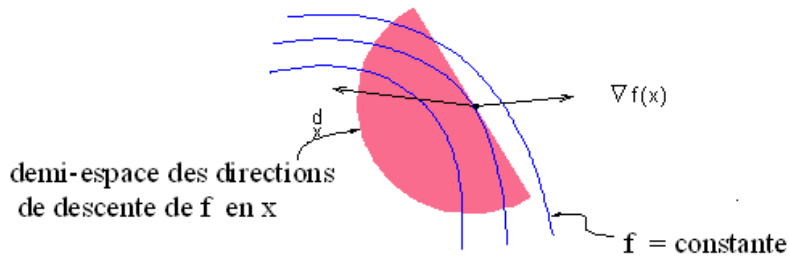
$$\nabla^T f(x).d < 0$$

Alors  $d$  est une direction de descente en  $x$

**Remarque 2.1.**  $d$  fait avec l'opposé du gradient  $-\nabla f(x)$  un angle  $\theta$  strictement plus petit que  $90^\circ$  :

$$\theta = \arccos \frac{-\nabla f(x)^T . d}{\|\nabla f(x)\| \|d\|} \in \left[ 0, \frac{\pi}{2} \right[.$$

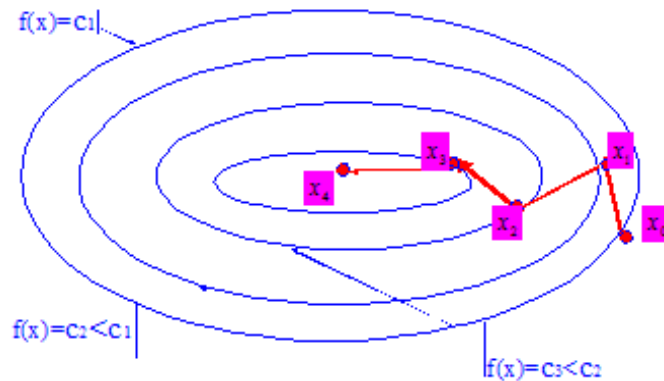
L'ensemble des directions de descente de  $f$  en  $x$ ,  $\{d \in \mathbb{R}^n : \nabla^T f(x).d < 0\}$  forme un demi-espace ouvert de  $\mathbb{R}^n$ .



Demi-espace (translaté) des directions de descente  $d$  de  $f$  en  $x$ . (Figure 2.1)

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  $d$ . De 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 une solution  $x_k$  du problème  $(P)$  par la formule de récurrence suivante :

$$x_{k+1} = x_k + \lambda_k d_k, \text{ pour } k \geq 1.$$



méthode à directions de descente (Figure 2.2)

Où  $\lambda_k$  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  $\lambda_k$ , c'est ce que l'on appelle la recherche linéaire.

**Remarque 2.2.** Par la suite l'angle  $\theta_k$  entre la direction  $d_k$  et  $-g_k$  ( $g_k = \nabla f(x_k)$ ) jouera un rôle important dans le processus de la convergence. Pour  $\theta_k$  on a la relation classique suivante

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

### Méthode à directions de descente- une itération

#### Algorithme 2.1

##### 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  $\lambda_k > 0$  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.

## 2.2 Recherche linéaire

Considérons le problème d'optimisation sans contraintes (P)

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

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

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

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

où  $\lambda_k$  est solution optimale du problème d'optimisation unidimensionnel suivant :

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

c'est à dire que  $\lambda_k$  vérifie

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

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

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

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

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

la recherche linéaire. Il existe deux grandes classes de 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

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

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

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

L'argument est le suivant.

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

#### Le second objectif

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

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

Or ceci peut entraîner une "fausse convergence", c'est-à-dire la convergence des itérés vers un point non stationnaire, comme le montre l'observation suivante [40].

Si on prend :

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

la suite  $\{x_k\}$  générée par (1.1) est de Cauchy, puisque pour  $1 < l < k$  on a :



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

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

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

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

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

### 2.2.2 But de la recherche linéaire

Dans le cas non-quadratique les méthodes de descente (0.2), nécessitent la recherche d'une valeur de  $\lambda_k > 0$  optimale ou non, vérifiant :

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

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

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

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

autrement dit,  $\lambda^*$  est un minimum local de  $h'$  qui assure de plus la décroissance de  $f$ .

En fait, dans la plupart des algorithmes d'optimisation modernes, on ne fait jamais de recherche linéaire exacte, car trouver  $\lambda^*$  signifie qu'il va falloir calculer un grand nombre de fois la fonction  $h'$  et cela peut être dissuasif du point de vue du temps de calcul. En pratique, on recherche plutôt une valeur de  $\lambda$  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  $\lambda_k$  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  $\lambda^*$  qui assure une décroissance suffisante de  $f$ . Cela conduit à la notion d'intervalle de sécurité.

**Définition 2.2.** *On dit que  $[\lambda_g, \lambda_d]$  est un intervalle de sécurité s'il permet de classer les valeurs de  $\lambda$  de la façon suivante :*

- Si  $\lambda < \lambda_g$  alors  $\lambda$  est considéré trop petit,
- Si  $\lambda_d \geq \lambda \geq \lambda_g$  alors  $\lambda$  est satisfaisant,
- Si  $\lambda > \lambda_d$  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  $\lambda_g$  et  $\lambda_d$ .

### 2.2.4 Algorithme de base

**Algorithme 2.2 (Algorithme de base)**

**Etape 0 : (initialisation)**

$\lambda_g = \lambda_d = 0$ , choisir  $\lambda_1 > 0$ , poser  $k = 1$  et aller à l'étape 1.

**Etape 1 :**

Si  $\lambda_k$  convient, poser  $\lambda^* = \lambda_k$  et on s'arrête.

Si  $\lambda_k$  est trop petit on prend  $\lambda_{g,k+1} = \lambda_k$ ,  $\lambda_d = \lambda_d$ .

et on va à l'étape 2 .

Si  $\lambda_k$  est trop grand on prend  $\lambda_{d,k+1} = \lambda_k$ ,  $\lambda_g = \lambda_g$ .

et on va à l'étape 2 .

**Etape 2 :**

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

si  $\lambda_{d,k+1} \neq 0$  déterminer  $\lambda_{k+1} \in ]\lambda_{g,k+1}, \lambda_{d,k+1}[$ .

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

## 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  $\lambda_k$  comme solution du problème.

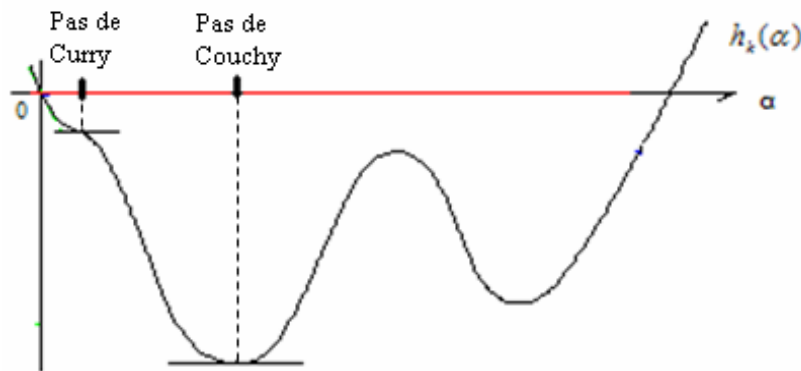
$$\min h_k(\lambda), \quad \lambda \geq 0 \tag{2.1}$$

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

(voir figure 2.3). Dans certains cas, on préférera le plus petit point stationnaire de  $h_k$  qui fait décroître cette fonction.

$$\lambda_k = \inf \{ \lambda \geq 0 : h'_k(\lambda) < h'_k(0) \} . \tag{2.2}$$

On parle alors de règle de Curry et le pas déterminé par cette règle est appelé pas de Curry (voir figure 2.3). De manière un peu imprécise, ces deux règles sont parfois qualifiées de recherche linéaire exacte.



Règles de Cauchy et de Curry (Figure 2.3)

**Remarque 2.3.** Ces deux règles ne sont utilisées que dans des cas particuliers, par exemple lorsque  $h_k$  est quadratique.

Le mot exact prend sa signification dans le fait que si  $f$  est quadratique la solution de la recherche linéaire s'obtient de façon exacte et dans un nombre fini d'itérations.

### 2.3.1 Les inconvénients des recherches linéaires exactes

Pour une fonction non linéaire arbitraire,

- Il peut ne pas exister de pas de Cauchy ou de Curry.
- La détermination de ces pas demande en général beaucoup d'observations et ne peut de toutes façons pas être faite avec précision.
- l'efficacité supplémentaire éventuellement apportée à un algorithme par une recherche linéaire exacte ne permet pas, en général, de compenser le temps perdu à déterminer un tel pas.
- les résultats de convergence autorisent d'autres types de règles (recherches linéaires inexactes), moins gourmandes en temps de calcul.

Au lieu de demander à  $\lambda_k$  de minimiser la fonction  $h_k$ , on préfère imposer des conditions moins restrictives, plus faciles à vérifier, 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.

## 2.4 Recherches linéaires inexactes

On considère la situation qui est typique pour l'application de la technique de recherche linéaire à l'intérieur de la méthode principale multidimensionnelle.

Sur une itération  $k$  de la dernière méthode nous avons l'itération courante  $x_k \in \mathbb{R}^n$  et la direction de recherche  $d_k \in \mathbb{R}^n$  qui est une direction de descente pour la fonction objectif  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , on a :

$$\nabla^T f(x_k).d_k < 0 \tag{2.3}$$

Le but est de réduire "de façon importante" la valeur de l'objectif par un pas  $x_k \mapsto x_{k+1} = x_k + \lambda_k d_k$  de  $x_k$  dans la direction  $d_k$ . Pour cela de nombreux mathématiciens (Armijo, Goldstein, Wolfe, Albaali, Lemaréchal, Fletcher...) ont élaboré plusieurs règles (tests).

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

D'abord présentons le schéma d'une recherche linéaire inexacte.

Elles reviennent à déterminer, par tâtonnement un intervalle  $[\lambda_g, \lambda_d]$ , où  $\lambda^* \in [\lambda_g, \lambda_d]$ , dans lequel :

$$h_k(\alpha_k) < h_k(0) \quad (f(x_k + \lambda_k d_k) < f(x_k)) \quad (2.4)$$

### **Algorithme 2.3 (Schéma général des recherches linéaires inexactes)**

#### **Etape 0 : (initialisation)**

$\lambda_{g,1} = \lambda_{d,1} = 0$ , choisir  $\lambda_1 > 0$ , poser  $k = 1$  et aller à l'étape 1.

#### **Etape 1 :**

si  $\lambda_k$  est satisfaisant (suivant un certain critère) : STOP( $\lambda^* = \lambda_k$ ).

si  $\lambda_k$  est trop petit (suivant un certain critère) : nouvel intervalle :  $[\lambda_{g,k+1} = \lambda_k, \lambda_{d,k+1} = \lambda_d]$

et aller à l'étape 2.

si  $\lambda_k$  est trop grand (suivant un certain critère) : nouvel intervalle :  $[\lambda_{g,k+1} = \lambda_g, \lambda_{d,k+1} = \lambda_k]$

et aller à l'étape 2.

#### **Etape 2 :**

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

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

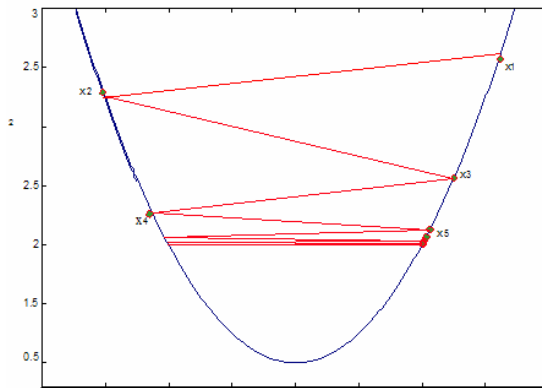
remplacer  $k$  par  $k + 1$  et aller à l'étape 1.

Il nous reste donc à décider selon quel(s) critère(s)  $\lambda$  est trop petit ou trop grand ou satisfaisant.

### 2.4.1 La règle d'Armijo

On bute à réduire "de façon importante" la valeur de l'objectif par un pas  $x_k \mapsto x_{k+1} = x_k + \lambda_k d_k$  de  $x_k$  dans la direction  $d_k$ , tel que  $f(x_k + \lambda_k d_k) < f(x_k)$ .

Or cette condition de décroissance stricte n'est pas suffisante pour minimiser  $h_k$  au moins localement. Par exemple, avec la fonction  $h : \mathbb{R} \rightarrow \mathbb{R}; x \mapsto h(x) = x^2$  et  $x_1 = 2$ , les choix  $d_k = (-1)^{k+1}; \lambda_k = 2 + 3 \times 2^{-(k+1)}$ , donnent :  $x_k = (-1)^k(1 + 2^{-k})$ .



Exemple 2.1 (Figure 2.4)

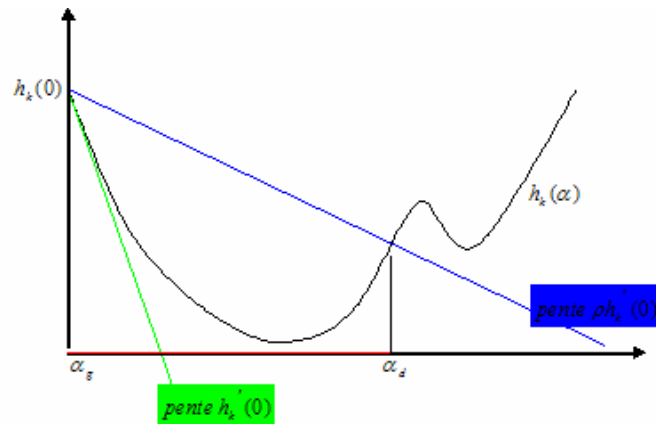
$(x_k)$  est bien strictement décroissante mais  $\{x_k\}_{k \geq 0}$  ne converge pas vers le minimum zéro mais vers 1 (dans cet exemple le pas est trop grand.).

La règle d'Armijo [2] impose une contrainte sur le choix de  $\lambda_k$  suffisante pour minimiser localement  $h$ .

Une condition naturelle est de demander que  $f$  décroisse autant qu'une portion  $\rho \in ]0, 1[$  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 :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \rho \lambda_k \nabla^T f(x_k) d_k \quad (2.5)$$

Elle est de la forme (2.5), car  $\rho$  devra être choisi dans  $]0, 1[$ .



Règle d'Armijo (Figure 2.5)

On voit bien à la figure 2.5 ce que signifie cette condition.

Il faut qu'en  $\lambda_k$ , la fonction  $h_k$  prenne une valeur plus petite que celle prise par la fonction  $\psi_\rho(\lambda)$  :

$$\lambda \mapsto h_k(0) + \rho h'_k(0)\lambda \quad \text{autrement dit } \lambda \mapsto f(x_k) + \rho \lambda_k \nabla^T f(x_k) d_k \quad (2.6)$$

**Test d'Armijo :**

◇ Si

$$h_k(\lambda) \leq h_k(0) + \rho h'_k(0)\lambda \quad (2.7)$$

$$\text{autrement dit } f(x_k + \lambda d_k) \leq f(x_k) + \rho \lambda \nabla^T f(x_k) d_k$$

alors  $\lambda$  convient.

◇ Si

$$h_k(\lambda) > h_k(0) + \rho h'_k(0)\lambda \quad (2.8)$$

$$\text{autrement dit } f(x_k + \lambda d_k) > f(x_k) + \rho \lambda \nabla^T f(x_k) d_k$$

alors  $\lambda$  est trop grand.

**Algorithme 2.4 (Règle d'Armijo)**

**Etape 0 : (initialisation)**

$\lambda_{g,1} = \lambda_{d,1} = 0$ , choisir  $\lambda_1 > 0$ ,  $\rho \in ]0, 1[$  poser  $k = 1$  et aller à l'étape 1.

**Etape 1 :**

si  $h_k(\lambda_k) \leq h_k(0) + \rho h'_k(0)\lambda_k$  : STOP ( $\lambda^* = \lambda_k$ ).

si  $h_k(\lambda_k) > h_k(0) + \rho h'_k(0)\lambda_k$ , alors

$\lambda_{d,k+1} = \lambda_d$ ,  $\lambda_{g,k+1} = \lambda_k$  et aller à l'étape 2.

**Etape 2 :**

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

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

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

**Remarque 2.4.** En pratique, la constante  $\rho$  est prise très petite, de manière à satisfaire (2.7) le plus facilement possible. Typiquement,  $\rho = 10^{-4}$ .

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.

**Remarque 2.5.** Il est clair d'après la figure 2.5 que l'inégalité (2.7) est toujours vérifiée si  $\lambda_k > 0$  est suffisamment petit.

En effet dans le cas contraire, on aurait une suite de pas strictement positifs  $\{\lambda_{k,i}\}_{i \geq 1}$  convergeant vers 0 lorsque  $i \rightarrow \infty$  et tels que (2.7) n'ait pas lieu pour  $\lambda_k = \lambda_{k,i}$ .

En retranchant  $f(x_k)$  dans les deux membres, en divisant par  $\lambda_{k,i}$  et en passant à la limite quand  $i \rightarrow \infty$ , on trouve :

$$\nabla^T f(x_k) d_k \geq \rho \nabla^T f(x_k) d_k \tag{2.9}$$

ce qui contredit le fait que  $d_k$  est une direction de descente ( $\rho < 1$ ).

Dans le théorème suivant on va assurer l'existence du pas d'Armijo en posant quelques conditions sur la fonction  $h_k$ .

**Théorème 2.2.** Si  $h_k : \mathbb{R}_+ \rightarrow \mathbb{R}$ ; définie par (2.2)  $h_k(\lambda) = f(x_k + \lambda d_k)$  est continue et bornée inférieurement, si  $d_k$  est une direction de descente en  $x_k$  ( $h'_k(0) < 0$ ) et si  $\rho \in ]0, 1[$ , alors l'ensemble des pas vérifiant la règle d'Armijo est non vide.

**Démonstration.** [24]



On a

$$\begin{aligned} h_k(\lambda) &= f(x_k + \lambda d_k) \\ \psi_\rho(\lambda) &= f(x_k) + \rho \lambda_k \nabla^T f(x_k) d_k \end{aligned}$$

Le développement de Taylor-Young en  $\lambda = 0$  de  $h_k$  est :

$$h_k(\lambda) = f(x_k + \lambda d_k) = f(x_k) + \rho \lambda \nabla^T f(x_k) d_k + \lambda \xi(\lambda) \text{ où } \xi(\lambda) \rightarrow 0, \lambda \rightarrow 0.$$

et comme  $\rho \in ]0, 1[$  et  $h'_k(0) = \nabla^T f(x_k) d_k < 0$  on déduit :

$$f(x_k) + \lambda_k \nabla^T f(x_k) d_k < f(x_k) + \rho \lambda_k \nabla^T f(x_k) d_k \text{ pour } \lambda > 0$$

On voit que pour  $\lambda > 0$  assez petit on a :

$$h_k(\lambda) < \psi_\rho(\lambda)$$

De ce qui précède et du fait que  $h_k$  est bornée inférieurement,

et  $\psi_\rho(\lambda) \rightarrow -\infty; \lambda \rightarrow +\infty$ , on déduit que la fonction  $\psi_\rho(\lambda) - h_k(\lambda)$  possède la propriété :

$$\begin{cases} \psi_\rho(\lambda) - h_k(\lambda) > 0 & \text{pour } \lambda \text{ assez petit} \\ \psi_\rho(\lambda) - h_k(\lambda) < 0 & \text{pour } \lambda \text{ assez grand} \end{cases}$$

donc s'annule au moins une fois pour  $\lambda > 0$ .

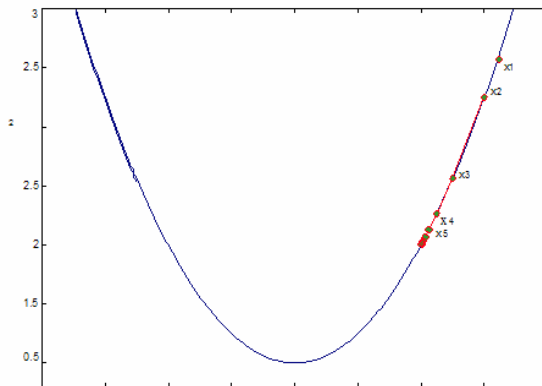
En choisissant le plus petit de ces zéros on voit qu'il existe  $\bar{\lambda} > 0$  tel que

$$h_k(\bar{\lambda}) = \psi_\rho(\bar{\lambda}) \text{ et } h_k(\alpha) < \psi_\rho(\alpha) \text{ pour } 0 < \alpha < \bar{\lambda}. \blacksquare$$

Décrivons maintenant la règle de Goldstein&Price.

### 2.4.2 La règle de Goldstein&Price.

Dans la règle d'Armijo on assure la décroissance de la fonction objectif à chaque pas, mais c'est ne pas suffisant ; reprenant l'exemple de la fonction  $h : \mathbb{R} \rightarrow \mathbb{R}; x \mapsto h(x) = x^2$  et  $x_1 = 2$ , et cette fois  $d_k = -1$  le choix  $\lambda_k = 2^{-(k+1)}$  donnent :  $x_k = (1 + 2^{-k})$



Exemple 2.2 (Figure 2.6)

$h_k$  est bien strictement décroissante mais  $\{x_k\}_{k \geq 0}$  ne converge pas vers le minimum zéro mais vers 1( voir figure 2.6).

Alors que la condition d'Armijo est satisfaite.

Les conditions de Goldstein & Price [28, 1969] suivant sont, comme on va le prouver, suffisante pour assurer la convergence sous certaines conditions et indépendamment de l'algorithme qui calcule le paramètre .

Etant données deux réels  $\rho$  et  $\delta$  tels que  $0 < \rho < \delta < 1$  ; ces conditions sont :

$$f(x_k + \lambda d_k) \leq f(x_k) + \rho \lambda \nabla^T f(x_k) d_k, \quad (2.10)$$

$$f(x_k + \lambda d_k) \geq f(x_k) + \delta \lambda \nabla^T f(x_k) d_k, \quad (2.11)$$

autrement dit

$$h_k(\lambda) \geq h_k(0) + \delta h'_k(0) \lambda. \quad (2.12)$$

$$h_k(\lambda) \geq h_k(0) + \rho h'_k(0) \lambda$$

Test de Goldstein&Price :

◇ Si

$$h_k(0) + \delta h'_k(0)\lambda \leq h_k(\lambda) \leq h_k(0) + \rho h'_k(0)\lambda$$

autrement dit  $f(x_k) + \delta \lambda \nabla^T f(x_k) d_k \leq f(x_k + \lambda d_k) \leq f(x_k) + \rho \lambda \nabla^T f(x_k) d_k$

alors  $\lambda$  convient.

◇ Si

$$h_k(\lambda) > h_k(0) + \rho h'_k(0)\lambda ,$$

autrement dit  $f(x_k + \lambda d_k) > f(x_k) + \rho \lambda \nabla^T f(x_k) d_k,$

alors  $\lambda$  est trop grand.

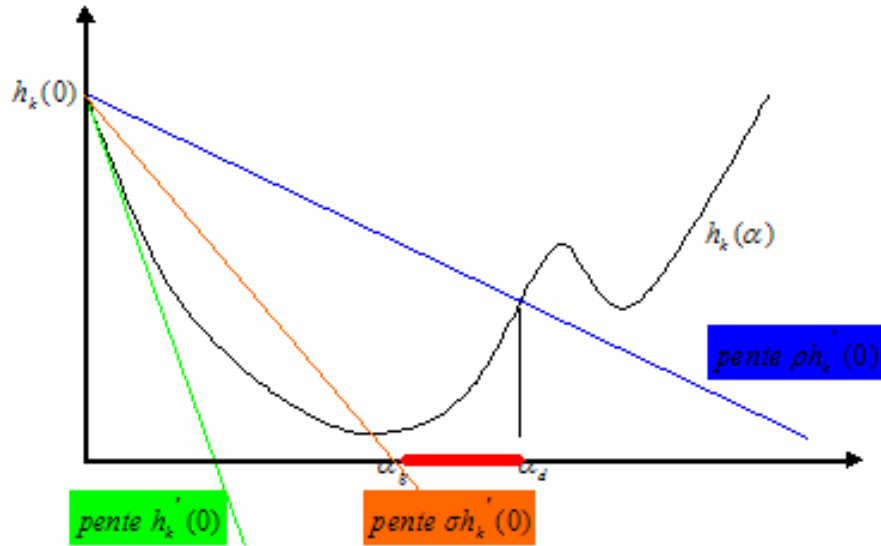
◇ Si

$$h_k(\lambda) < h_k(0) + \delta h'_k(0)\lambda ,$$

autrement dit  $f(x_k + \lambda d_k) < f(x_k) + \delta \lambda \nabla^T f(x_k) d_k$

alors  $\lambda$  est trop petit.

On voit bien dans la figure 2.7 ce que signifie cette condition.e cette condition.



Règle de Goldstein&Price (Figure 2.7)

**Algorithme 2.5 (Règle de Goldstein&Price)**

**Etape 0 : (initialisation)**

$\lambda_{g,1} = \lambda_{d,1} = 0$ , choisir  $\lambda_1 > 0$ ,  $\rho \in ]0, 1[$ ,  $\delta \in ]\rho, 1[$ , poser  $k = 1$  et aller à l'étape 1.

**Etape 1 :**

si  $h_k(0) + \delta h'_k(0)\lambda \leq h_k(\lambda_k) \leq h_k(0) + \rho h'_k(0)\lambda_k$  : STOP( $\lambda^* = \lambda_k$ ).

si  $h_k(\lambda_k) > h_k(0) + \rho h'_k(0)\lambda_k$ , alors

$\lambda_{d,k+1} = \lambda_k$ ,  $\lambda_{g,k+1} = \lambda_{g,k}$  et aller à l'étape 2.

si  $h_k(\lambda_k) < h_k(0) + \delta h'_k(0)\lambda_k$ , alors

$\lambda_{d,k+1} = \lambda_{d,k}$ ,  $\lambda_{g,k+1} = \lambda_k$  et aller à l'étape 2.

**Etape 2 :**

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

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

Dans le théorème suivant on va assurer l'existence du pas de Goldstein & price en posant quelques conditions sur la fonction  $h_k$ .

**Théorème 2.3.** Si  $h_k : \mathbb{R}_+ \rightarrow \mathbb{R}$  définie par (2.12)  $h_k(\lambda) = f(x_k + \lambda d_k)$  est continue et bornée inférieurement, si  $d_k$  est une direction de descente en  $x_k$  et si  $\rho \in ]0, 1[$ ,  $\delta \in ]\rho, 1[$ , alors l'ensemble des pas vérifiant la règle de Goldstein & Price (2.10)-(2.11) est non vide.

**Démonstration.** [24]

On a

$$\begin{aligned} h_k(\lambda) &= f(x_k + \lambda d_k) \\ \psi_\rho(\lambda) &= f(x_k) + \rho \lambda \nabla^T f(x_k) d_k \\ \psi_\delta(\lambda) &= f(x_k) + \delta \lambda \nabla^T f(x_k) d_k \end{aligned}$$

Le développement de Taylor-Yong en  $\lambda = 0$  de  $h_k$  est :

$$h_k(\lambda) = f(x_k + \lambda d_k) = f(x_k) + \rho \lambda \nabla^T f(x_k) d_k + \lambda \xi(\lambda) \text{ où } \xi(\lambda) \rightarrow 0, \lambda \rightarrow 0.$$

et comme  $\rho \in ]0, 1[$  et  $h'_k(0) = \nabla^T f(x_k) d_k < 0$  on déduit :

$$f(x_k) + \lambda \nabla^T f(x_k) d_k < f(x_k) + \delta \lambda \nabla^T f(x_k) d_k < f(x_k) + \rho \lambda \nabla^T f(x_k) d_k \text{ pour } \lambda > 0$$

On voit que pour  $\lambda > 0$  assez petit on a :

$$h_k(\lambda) < \psi_\delta(\lambda) < \psi_\rho(\lambda)$$

De ce qui précède et du fait que  $h_k$  est bornée inférieurement,

et  $\psi_\rho(\lambda) \rightarrow -\infty, \lambda \rightarrow +\infty$ , on déduit que la fonction  $\psi_\rho(\lambda) - h_k(\lambda)$  a la propriété :

$$\begin{cases} \psi_\rho(\lambda) - h_k(\lambda) > 0 & \text{pour } \lambda \text{ assez petit} \\ \psi_\rho(\lambda) - h_k(\lambda) < 0 & \text{pour } \lambda \text{ assez grand} \end{cases}$$

donc s'annule au moins une fois pour  $\lambda > 0$ .

En choisissant le plus petit de ces zéros on voit qu'il existe  $\bar{\lambda} > 0$  tel que

$$h_k(\bar{\lambda}) = \psi_\rho(\bar{\lambda}) \text{ et } h_k(\lambda) < \psi_\rho(\lambda) \text{ pour } 0 < \lambda < \bar{\lambda}$$

De la même manière, il existe  $\tilde{\lambda} > 0$  tel que :

$$h_k(\tilde{\lambda}) = \psi_\delta(\tilde{\lambda}) \text{ et } h_k(\lambda) < \psi_\delta(\lambda) \text{ pour } 0 < \lambda < \tilde{\lambda}$$

et comme  $\psi_\delta(\lambda) < \psi_\rho(\lambda)$  pour  $\lambda > 0$ , forcément  $\tilde{\lambda} < \bar{\lambda}$  et  $\lambda = \bar{\lambda}$  satisfait (2.10)-(2.11)

$$\left( \begin{array}{l} \psi_\delta(\tilde{\lambda}) = h_k(\tilde{\lambda}) < \psi_\rho(\alpha) \text{ n'est autre que} \\ f(x_k) + \delta\tilde{\lambda}\nabla^T f(x_k)d_k = f(x_k + \tilde{\lambda}d_k) < f(x_k) + \rho\tilde{\lambda}\nabla^T f(x_k)d_k. \blacksquare \end{array} \right)$$

### 2.4.3 La règle de Wolfe

Les conditions (2.10)-(2.11) "règle de Goldstein & Price" peuvent exclure un minimum ce qui est peut être un inconvénient.

Les conditions de Wolfe [57] n'ont pas cet inconvénient.

#### Conditions de Wolfe faibles

Etant donnés deux réels  $\rho$  et  $\sigma$  tel que  $0 < \rho < \sigma < 1$ , ces conditions sont :

$$f(x_k + \lambda d_k) \leq f(x_k) + \rho\lambda\nabla^T f(x_k)d_k \tag{2.13}$$

$$\nabla^T f(x_k + \lambda d_k)d_k \geq \sigma\nabla^T f(x_k)d_k \tag{2.14}$$

autrement dit :

$$h_k(\lambda) \leq h_k(0) + \rho h'_k(0)\lambda \tag{2.15}$$

$$h'_k(\lambda) \geq \sigma h'_k(0) \tag{2.16}$$

Test de Wolfe :

◇ Si

$$h_k(\lambda) \leq h_k(0) + \rho h'_k(0)\lambda \text{ et } h'_k(\lambda) \geq \sigma h'_k(0)$$

alors  $\lambda$  convient.

◇ Si

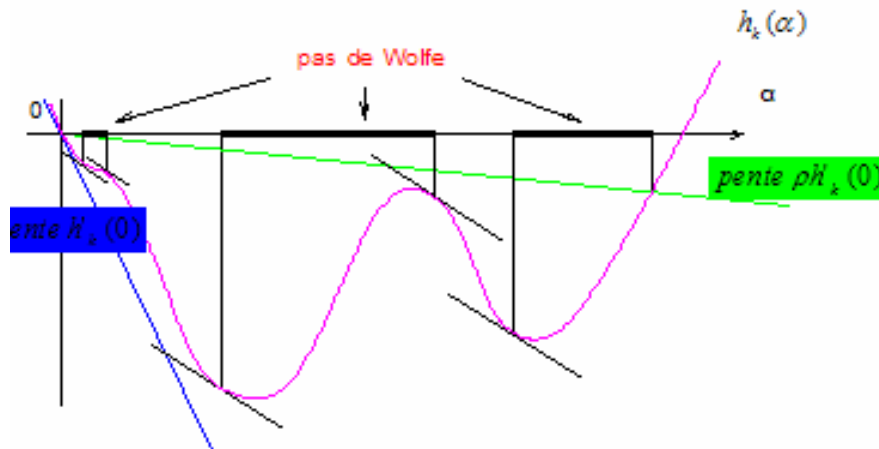
$$h_k(\lambda) > h_k(0) + \rho h'_k(0)\lambda$$

alors  $\lambda$  est trop grand.

$$h'_k(\lambda) < \sigma h'_k(0)$$

alors  $\lambda$  est trop petit.

On voit bien à la figure 2.8 ce que signifie cette condition.



Règle de Wolfe (Figure 2.8)

**Algorithme 2.6 (Règle de Wolfe)**

**Etape 0 : (initialisation)**

$\lambda_{g,1} = \lambda_{d,1} = 0$ , choisir  $\lambda_1 > 0$ ,  $\rho \in ]0, 1[$ ,  $\sigma \in ]\rho, 1[$ , poser  $k = 1$  et aller à l'étape 1.

**Etape 1 :**

si  $h_k(\lambda_k) \leq h_k(0) + \rho h'_k(0)\lambda_k$  et  $h'_k(\lambda) \geq \sigma h'_k(0)$  : STOP( $\lambda^* = \lambda_k$ ).

si  $h_k(\lambda_k) > h_k(0) + \rho h'_k(0)\lambda_k$ , alors

$\lambda_{d,k+1} = \lambda_k$  ,  $\lambda_{g,k+1} = \lambda_{g,k}$  et aller à l'étape 2.

si  $h'_k(\lambda) < \sigma h'_k(0)$  , alors

$\lambda_{d,k+1} = \lambda_{d,k}$  ,  $\lambda_{g,k+1} = \lambda_k$  et aller à l'étape 2.

**Etape 2 :**

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

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

**Remarque 2.6.** La règle de Wolfe fait appel au calcul de  $h'_k$ , elle est donc en théorie plus coûteuse que la règle de Goldstein & Price.

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.

**Conditions de Wolfe fortes**

Pour certains algorithmes (par exemple le gradient conjugué non linéaire) il est parfois nécessaire d'avoir une condition plus restrictive que (2.10).

Pour cela la deuxième condition (2.10) est remplacée par

$$|\nabla^T f(x_k + \lambda d_k) d_k| \leq \sigma |\nabla^T f(x_k) d_k| = -\nabla^T f(x_k) d_k.$$

On aura donc les conditions de Wolfe fortes [58] :

$$f(x_k + \lambda d_k) \leq f(x_k) + \rho \lambda \nabla^T f(x_k) d_k \tag{2.17}$$

$$|\nabla^T f(x_k + \lambda d_k) d_k| \leq -\sigma \nabla^T f(x_k) d_k \tag{2.18}$$

autrement dit :

$$h_k(\lambda) \leq h_k(0) + \rho h'_k(0) \lambda \tag{2.19}$$

$$|h'_k(\lambda)| \leq -\sigma h'_k(0) \tag{2.20}$$

où  $0 < \rho < \sigma < 1$ .

**Remarque 2.7.** La seconde condition (2.20) ou condition de courbure interdit le choix de pas



trop petit pouvant entrainer une convergence lente ou prématurée (section 2.21).

**Remarque 2.8.** On voit bien que les conditions de Wolfe fortes impliquent les conditions de Wolfe faibles. Effectivement (2.17) est équivalente à (2.13), tandis que (2.18) est équivalente à (2.14), en effet,

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

### La règle de Wolfe relaxée

Proposée par Dai et Yuan [14, 1996], cette règle consiste à choisir le pas  $\lambda_k$  satisfaisant aux conditions :

$$f(x_k + \lambda d_k) \leq f(x_k) + \rho \lambda \nabla^T f(x_k) d_k \quad (2.21)$$

$$\sigma_1 \nabla^T f(x_k) d_k \leq \nabla^T f(x_k + \lambda_k d_k) d_k \leq -\sigma_2 \nabla^T f(x_k) d_k \quad (2.22)$$

autrement dit :

$$h_k(\lambda) \leq h_k(0) + \rho h'_k(0) \lambda \quad (2.23)$$

$$\sigma_1 h'_k(0) \leq h'_k(\lambda) \leq -\sigma_2 h'_k(0) \quad (2.24)$$

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

**Remarque 2.9.** On voit bien que les conditions de Wolfe relaxées impliquent les conditions de Wolfe fortes. Effectivement (2.22) est équivalente à (2.17), tandis que pour le cas particulier  $\sigma_1 = \sigma_2 = \sigma$ , (2.24) est équivalente à (2.18). En effet,

$$\begin{aligned} \sigma_1 \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k \leq -\sigma_2 \nabla^T f(x_k) d_k \\ \Rightarrow \sigma \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k \leq -\sigma \nabla^T f(x_k) d_k \\ \Rightarrow |\nabla^T f(x_k + \lambda_k d_k) d_k| &\leq -\sigma \nabla^T f(x_k) d_k \end{aligned}$$

**Remarque 2.10.** Les conditions de Wolfe relaxées impliquent les conditions de Wolfe faibles. Effectivement (2.23) est équivalente à (2.13), tandis que pour le cas particulier  $\sigma_1 = \sigma$  et  $\sigma_2 = +\infty$ , (2.24) est équivalente à (2.14). En effet,

$$\begin{aligned} \sigma_1 \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k \leq -\sigma_2 \nabla^T f(x_k) d_k \\ \Rightarrow \sigma \nabla^T f(x_k) d_k &\leq \nabla^T f(x_k + \lambda_k d_k) d_k \end{aligned}$$

Dans le théorème suivant on va assurer l'existence du pas de Wolfe en posant quelques conditions sur la fonction  $h_k$ .

**Théorème 2.4.** *Si  $h_k : \mathbb{R}_+ \rightarrow \mathbb{R}$  définie par (2.12)  $h_k(\lambda) = f(x_k + \lambda d_k)$  est dérivable et bornée inférieurement, si  $d_k$  est une direction de descente en  $x_k$  et si  $\rho \in ]0, 1[$ ,  $\sigma \in ]\rho, 1[$ , alors l'ensemble des pas vérifiant la règle de Wolfe (faible) (2.13)-(2.14) est non vide.*

**Démonstration.** [24]

On a

$$\begin{aligned} h_k(\lambda) &= f(x_k + \lambda d_k) \\ \psi_\rho(\lambda) &= f(x_k) + \rho \lambda \nabla^T f(x_k) d_k \end{aligned}$$

Le développement de Taylor-Yong en  $\lambda = 0$  de  $h_k$  est :

$$h_k(\lambda) = f(x_k + \lambda d_k) = f(x_k) + \rho \lambda \nabla^T f(x_k) d_k + \alpha \xi(\lambda) \text{ où } \xi(\lambda) \rightarrow 0, \lambda \rightarrow 0.$$

et comme  $\rho \in ]0, 1[$  et  $h'_k(0) = \nabla^T f(x_k) d_k < 0$  on déduit :

$$f(x_k) + \lambda \nabla^T f(x_k) d_k < f(x_k) + \rho \lambda \nabla^T f(x_k) d_k \text{ pour } \lambda > 0$$

On voit que pour  $\lambda > 0$  assez petit on a

$$h_k(\lambda) < \psi_\rho(\lambda)$$

De ce qui précède et du fait que  $h_k$  est bornée inférieurement, et  $\psi_\rho(\lambda) \rightarrow -\infty, \lambda \rightarrow +\infty$ ,

on déduit que la fonction  $\psi_\rho(\lambda) - h_k(\lambda)$  a la propriété suivante

$$\begin{cases} \psi_\rho(\lambda) - h_k(\lambda) > 0 & \text{pour } \lambda \text{ assez petit} \\ \psi_\rho(\lambda) - h_k(\lambda) < 0 & \text{pour } \lambda \text{ assez grand} \end{cases}$$

donc s'annule au moins une fois pour  $\lambda > 0$ .

En choisissant le plus petit de ces zéros on voit qu'il existe  $\bar{\lambda} > 0$  tel que

$$h_k(\bar{\lambda}) = \psi_\rho(\bar{\lambda}) \text{ et } h_k(\lambda) < \psi_\rho(\lambda) \text{ pour } 0 < \lambda < \bar{\lambda}$$

La formule des accroissements finis fournit alors un nombre  $\hat{\lambda}, 0 < \hat{\lambda} < \bar{\lambda}$  tel que

$$\begin{aligned} h_k(\bar{\lambda}) - h_k(0) &= \bar{\lambda} h'_k(\hat{\lambda}) = \bar{\lambda} \nabla^T f(x_k + \hat{\lambda} d_k) d_k \\ &\Rightarrow \rho \bar{\lambda} \nabla^T f(x_k) d_k = \bar{\lambda} \nabla^T f(x_k + \hat{\lambda} d_k) d_k \\ &\Rightarrow \nabla^T f(x_k + \hat{\lambda} d_k) d_k = \rho \nabla^T f(x_k) d_k \geq \sigma \nabla^T f(x_k) d_k \end{aligned}$$

car  $0 < \rho < \sigma < 1$  et  $\nabla^T f(x_k) d_k < 0$ .

Donc  $\hat{\lambda}$  satisfait (2.23)

D'autre part,  $\lambda = \hat{\lambda}$  satisfait (2.22), en effet,

$\hat{\lambda}$  satisfait (\*)  $h_k(\hat{\lambda}) < \psi_\rho(\hat{\lambda})$  n'est autre que :

$$f(x_k + \hat{\lambda} d_k) < f(x_k) + \rho \hat{\lambda} \nabla^T f(x_k) d_k \blacksquare$$

En pratique, on utilise des algorithmes spécifiques pour trouver un pas de Wolfe.

On va présenter un algorithme simple, appelé de Fletcher-Lemaréchal, dont on peut montrer qu'il trouve un pas de Wolfe en un nombre fini d'étapes.

**Algorithme 2.7 (Règle de Fletcher-Lemaréchal)**

**Etape0 : (initialisation)**

$\lambda_{g,1} = 0, \lambda_{d,1} = +\infty, \rho \in ]0, 1[, \sigma \in ]\rho, 1[, v \in ]0, \frac{1}{2}[, \tau > 1$ , choisir  $\lambda_1 > 0$ , poser  $k = 1$  et aller à l'étape 1.

**Etape 1 :**

- 1.1 si  $h_k(\lambda_k) > h_k(0) + \rho h'_k(0)\lambda_k$ , alors  
 $\lambda_{d,k+1} = \lambda_k$ ,  $\lambda_{g,k+1} = \lambda_{g,k}$  et aller à l'étape 2.
- 1.3 Si non ( $h_k(\lambda_k) \leq h_k(0) + \rho h'_k(0)\lambda_k$ ), alors
- 1.3.1 si  $h'_k(\lambda) \geq \sigma h'_k(0)$  : STOP( $\lambda^* = \lambda_k$ ).
- 1.3.2 si non ( $h'_k(\lambda) < \sigma h'_k(0)$ )  
 $\lambda_{d,k+1} = \lambda_{d,k}$ ,  $\lambda_{g,k+1} = \lambda_k$  et aller à l'étape 2.

**Etape 2 :**

- si  $\lambda_{d,k+1} = +\infty$  déterminer  $\lambda_{k+1} \in ]\tau\lambda_{g,k+1}, +\infty[$   
 si non déterminer  $\lambda_{k+1} \in ](1-v)\lambda_{g,k+1} + v\lambda_{d,k+1}, v\lambda_{g,k+1} + (1-v)\lambda_{d,k+1}[$   
 poser  $k = k + 1$  et aller à l'étape 1.

**Théorème 2.5.** [24] *Si  $h_k : \mathbb{R}_+ \rightarrow \mathbb{R}$  définie par (2.12)  $h_k(\lambda) = f(x_k + \lambda d_k)$  est dérivable et bornée inférieurement, si  $d_k$  est une direction de descente en  $x_k$  et si  $\rho \in ]0, 1[$ ,  $\sigma \in ]\rho, 1[$ , alors l'algorithme de Fletcher-Lemaréchal trouve le pas de Wolfe(2.22)-(2.23) en un nombre fini d'étapes.*

## 2.5 Recherches linéaires non monotones

La technique de la recherche linéaire non monotone a été proposée en premier par Grippo, Lampariello, and Lucidi [30, 1986] et a connu de nombreuses applications aussi bien en optimisation sans contraintes qu'en optimisation avec contraintes [30], [62]. Bien que prometteuse, la recherche sur le thème de la recherche linéaire non monotone est toujours dans ses débuts, comme est souligné dans [54]. L'objectif de ce chapitre est d'explicitier les recherches linéaires et en particulier les recherches linéaire non monotone.

Soit le problème d'optimisation sans contraintes :

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

où  $f$  est une fonction différentiable et son gradient est noté par  $g = \nabla f(x)$ .

Supposons que le rapprochement actuel à la solution de (P) est  $x_k$ .

Si  $g_k = \nabla f(x_k) \neq 0$ , pour une direction de descente donnée  $d_k$  par une certaine méthode, on trouve le pas  $\lambda_k$  par la réalisation de certaines recherches linéaires tout au long de  $d_k$ ,

et on calcule le prochain rapprochement  $x_{k+1}$  comme suit

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

La recherche du pas  $\lambda_k$  doit assurer la décroissance suffisante de la fonction à chaque itération. c'est-à-dire que :

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j})$$

Par exemple, voir la recherche linéaire d'Armijo [2] et la recherche linéaire de Wolfe[57].

La recherche linéaire non monotone comme on va le voir n'impose pas la condition de décroissance suffisante de la fonction objectif comme est le cas des recherches linéaires précédentes. Nous allons expliciter dans ce qui suit deux recherches linéaires non monotones : la recherche linéaire non monotone d'Armijo et la recherche linéaire non monotone de Wolfe.

### 2.5.1 La recherche linéaire non monotone d'Armijo

Soit  $0 < n_1 < n_2$ ,  $\sigma \in (0, 1)$ ,  $\delta \in (0, 1)$ , et soit  $M$  est un entier positif, supposons que lors de la  $k$  ième itération, on a  $\bar{\lambda}_k \in (n_1, n_2)$  donné.

La recherche linéaire non monotone consiste à choisir dès la première itération un pas  $\lambda_k$  tel que :

$$\lambda_k = \bar{\lambda}_k \sigma^{n_k} \tag{2.25}$$

satisfaisant

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T d_k, \tag{2.26}$$

où

$$m(0) = 0 \text{ et } 0 \leq m(k) \leq \min[m(k-1) + 1, M - 1], \quad k \geq 1. \quad (2.27)$$

**Algorithme 2.8 (Règle nonmonotone d'Armijo )**

**Etape 0 : (initialisation)**

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

**Etape 1 :**

Si  $f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T d_k$ , : STOP ( $\lambda^* = \lambda_k$ ).

Si  $f(x_k + \lambda_k d_k) > \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T d_k$ , alors

$\lambda_{d,k+1} = \lambda_d$ ,  $\lambda_{g,k+1} = \lambda_k$  et aller à l'étape 2.

**Etape 2 :**

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

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

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

**Remarque 2.11**

Si  $m(k) \equiv 0$ , la recherche linéaire ci-dessus non monotone est réduite à la recherche linéaire inexacte d'Armijo classique.

## 2.5.2 La recherche linéaire non monotone de Wolfe

Déterminer  $\lambda_k$  tel que :

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \rho \lambda_k g_k^T d_k, \quad (2.28)$$

$$| \langle g(x_k + \lambda_k d_k), d_k \rangle | \leq -\delta g_k^T d_k, \quad (2.29)$$

où  $0 < \rho < \delta < \frac{1}{2}$  et où l'ordre du nombre entier  $\{m(k)\}$  est produit par la formule

$$m(0) = 0, m(k) \leq \min \{m(k-1) + 1, M_0\}, \quad (2.30)$$

avec  $M_0$  un nombre entier.

**Algorithme 2.9 (Règle non monotone de Wolfe)**

**Etape 0 : (initialisation)**

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

**Etape 1 :**

si  $f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \rho \lambda_k g_k^T d_k$  et  $|\langle g(x_k + \lambda_k d_k), d_k \rangle| \leq -\delta g_k^T d_k$  : STOP  
 $(\lambda^* = \lambda_k)$ .

si  $f(x_k + \lambda_k d_k) > \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \rho \lambda_k g_k^T d_k$ , alors

$\lambda_{d,k+1} = \lambda_d$ ,  $\lambda_{g,k+1} = \lambda_k$  et aller à l'étape 2.

si  $|\langle g(x_k + \lambda_k d_k), d_k \rangle| > -\delta g_k^T d_k$ , alors

$\lambda_{d,k+1} = \lambda_k$ ,  $\lambda_{g,k+1} = \lambda_g$  et aller à l'étape 2.

**Etape 2 :**

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

si  $\lambda_{d,k+1} \neq 0$  déterminer  $\lambda_{k+1} \in ]\lambda_{g,k+1}, \lambda_{d,k+1}[$ .

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

### 2.5.3 La recherche linéaire non monotone de Goldstein&Price

Soit  $\delta \in (0, \frac{1}{2})$ , le pas  $\lambda_k$  de la recherche linéaire non monotone de Goldstein&Price vérifient les conditions suivantes :

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T d_k$$

et

$$f(x_k + \lambda_k d_k) \geq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + (1 - \delta) \lambda_k g_k^T d_k$$

Une deuxième approche de la la recherche linéaire non monotone de Goldstein&Price a été proposé par [61] qui est la suivante :

Déterminer  $\lambda_k$  tels que

$$f(x_k + \lambda_k d_k) \leq C_k + \delta \lambda_k g_k^T d_k$$

et

$$f(x_k + \lambda_k d_k) \geq C_k + (1 - \delta) \lambda_k g_k^T d_k$$

**Algorithme 2.10 (Règle non monotone de Goldstein&Price )**

**Etape 0 : (initialisation)**

$\lambda_{g,1} = \lambda_{d,1} = 0$ , choisir  $\delta \in ]0, \frac{1}{2}[$  poson  $C_1 = f(x_1)$  et  $k = 1$  et aller à l'étape 1.

**Etape 1 :**

si  $f(x_k + \lambda_k d_k) \leq C_k + \delta \lambda_k g_k^T d_k$  et  $f(x_k + \lambda_k d_k) \geq C_k + (1 - \delta) \lambda_k g_k^T d_k$  : STOP ( $\lambda^* = \lambda_k$ ).

si  $f(x_k + \lambda_k d_k) > C_k + \rho \lambda_k g_k^T d_k$ , alors

$\lambda_{d,k+1} = \lambda_d$ ,  $\lambda_{g,k+1} = \lambda_k$  et aller à l'étape 2.

si  $f(x_k + \lambda_k d_k) < C_k + (1 - \delta) \lambda_k g_k^T d_k$ , alors

$\lambda_{d,k+1} = \lambda_k$ ,  $\lambda_{g,k+1} = \lambda_g$  et aller à l'étape 2.

**Etape 2 :**

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

si  $\lambda_{d,k+1} \neq 0$  déterminer  $\lambda_{k+1} \in ] \lambda_{g,k+1}, \lambda_{d,k+1} [$ .

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



## Chapitre 3

# La convergence des méthodes du gradient conjugué avec la recherche linéaire non monotone

- 
- 1- Condition de Zoutendijk.
  - 2- La méthode du gradient conjugué.
  - 3- Le principe général d'une méthode à directions conjuguées.
  - 4- Méthode de gradient conjugué dans le cas quadratique
  - 5- Synthèse des résultats de convergence des méthodes du gradient conjugué.
  - 6- Analyse de la convergence des méthodes du gradient conjugué avec la recherche linéaire non monotone.
-

### 3.1 Condition de Zoutendijk

Dans cette partie, on va étudier la *contribution* de la recherche linéaire non monotone à la convergence des algorithmes à directions de descente. Ce n'est 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*, dont 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

$$f(x_{k+1}) \leq f(x_k) - C \|\nabla f(x_k)\|^2 \cos^2 \theta_k, \quad (3.1)$$

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

$$\cos \theta_k = \frac{-\nabla^T f(x_k) d_k}{\|d_k\| \|d_k\|}. \quad (3.2)$$

Voici comment on se sert de la condition de Zoutendijk.

**Proposition 3.1.** *Si la suite  $\{x_k\}$  générée par un algorithme d'optimisation vérifie la condition de Zoutendijk (3.1) et si la suite  $\{f(x_k)\}$  est minorée, alors*

$$\sum_{k \geq 1} \|\nabla f(x_k)\|^2 \cos^2 \theta_k < \infty. \quad (3.3)$$

**Démonstration.** [24]

En sommant les inégalités (3.1), on a

$$\sum_{k \geq 1}^l \|\nabla f(x_k)\|^2 \cos^2 \theta_k \leq \frac{1}{C} (f(x_1) - f(x_{l+1})). \quad (3.4)$$

La série est donc convergente puisqu'il existe une constante  $C'$  telle que pour tout  $k$ ,  $f(x_k) \geq C'$ . ■

Les deux propositions suivantes précisent les circonstances dans lesquelles la condition de

Zoutendijk (3.1) est vérifiée avec les règles d'Armijo et de Wolfe.

**Proposition 3.2.** *Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction continûment 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 d'Armijo(2.5) avec  $\lambda_1 > 0$ .*

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

$$f(x_{k+1}) \leq f(x_k) - C \nabla^T f(x_k) d_k \quad (3.5)$$

où

$$f(x_{k+1}) \leq f(x_k) - C \|\nabla f(x_k)\|^2 \cos^2 \theta_k \quad (3.6)$$

est vérifiée.

**Démonstration.** [24]

Si le pas  $\lambda_k = \lambda_1$  est accepté, on a (3.1), car  $\lambda_1$  est uniformément positif.

Dans le cas contraire, (2.5) n'est pas vérifiée avec un pas  $\lambda'_k \leq \frac{\lambda_k}{\tau}$ , c'est-à-dire  $f(x_k + \lambda'_k d_k) > f(x_k) + \rho \lambda'_k \nabla^T f(x_k) d_k$

Comme  $f$  est continûment différentiable, on a pour tout  $\lambda_k > 0$  :

$$\begin{aligned} f(x_k + \lambda_k d_k) &= f(x_k) + \lambda_k \nabla^T f(x_k) d_k + \int_0^1 [\nabla f(x_k + t \lambda_k d_k) - \nabla f(x_k)]^T \lambda_k d_k dt \quad (3.7) \\ &\Rightarrow f(x_k + \lambda_k d_k) \leq f(x_k) + \lambda_k \nabla^T f(x_k) d_k + C \lambda_k^2 \|d_k\|^2 \end{aligned}$$

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

$$\begin{cases} f(x_k + \lambda'_k d_k) - f(x_k) > \rho \lambda'_k \nabla^T f(x_k) d_k \\ f(x_k + \lambda'_k d_k) - f(x_k) \leq \lambda'_k \nabla^T f(x_k) d_k + C \lambda_k'^2 \|d_k\|^2 \end{cases} \quad (3.8)$$

$$\Rightarrow \rho \lambda'_k \nabla^T f(x_k) d_k \leq \lambda_k \nabla^T f(x_k) d_k + C \lambda_k'^2 \|d_k\|^2 \quad (3.9)$$

$$\Rightarrow -C \lambda_k'^2 \|d_k\|^2 \leq (1 - \rho) \lambda'_k \nabla^T f(x_k) d_k$$

où

$$\rho < 1 \Rightarrow 0 < 1 - \rho < 1 \Rightarrow \frac{1}{1 - \rho} > 1 \quad (3.10)$$

d'où

$$\begin{aligned} \nabla^T f(x_k) d_k &\geq \frac{-C}{1 - \rho} \alpha'_k \|d_k\|^2 \Rightarrow -\nabla^T f(x_k) d_k \leq \frac{C}{1 - \rho} \lambda'_k \|d_k\|^2 \\ &\Rightarrow |\nabla^T f(x_k) d_k| = \|\nabla^T f(x_k)\| \|d_k\| \cos \theta_k \leq \frac{C}{1 - \rho} \lambda'_k \|d_k\|^2 \end{aligned} \quad (3.11)$$

ce qui permet de minorer  $\lambda' \|d_k\|$  et donc aussi  $\lambda \|d_k\|$  par une constante fois  $\|\nabla^T f(x_k)\| \|d_k\|$ . Cette minoration et l'expression suivante de (2.5)

$$f(x_k + \lambda_k d_k) \leq f(x_k) - \rho \lambda_k \|\nabla^T f(x_k)\| \|d_k\| \cos \theta_k \quad (3.12)$$

conduit à (3.6). ■

**Proposition 3.3.** *Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction continûment 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 (2.13)-(2.14).*

*Alors il existe une constante  $C > 0$  telle que, pour tout  $k \geq 1$ , la condition de Zoutendijk (3.1) est vérifiée.*

**Démonstration.** [24]

D'après (2.13)

$$\nabla^T f(x_k + \lambda_k d_k) d_k \geq \sigma \nabla^T f(x_k) d_k \quad (3.13)$$

$$\begin{aligned} &\Rightarrow (\nabla f(x_k + \lambda_k d_k) - \nabla f(x_k))^T d_k \geq (\sigma - 1) \nabla^T f(x_k) d_k \\ &= -(1 - \sigma) \nabla^T f(x_k) d_k = (1 - \sigma) |\nabla^T f(x_k) d_k| \\ &\Leftrightarrow (1 - \sigma) |\nabla^T f(x_k) d_k| \leq (\nabla f(x_k + \lambda_k d_k) - \nabla f(x_k))^T d_k \end{aligned} \quad (3.14)$$

et du fait que  $f$  est continûment différentiable :

$$\begin{aligned}
 (1 - \sigma) \left| \nabla^T f(x_k) d_k \right| &= (1 - \sigma) \left\| \nabla^T f(x_k) \right\| \|d_k\| \cos \theta_k & (3.15) \\
 &\leq \left\| \nabla f(x_k + \lambda_k d_k) - \nabla f(x_k) \right\| \|d_k\| \\
 &\Rightarrow (1 - \sigma) \left\| \nabla^T f(x_k) \right\| \cos \theta_k \leq L \lambda_k \|d_k\| \\
 &\Rightarrow \lambda_k \|d_k\| \leq \frac{(1 - \sigma)}{L} \left\| \nabla^T f(x_k) \right\| \cos \theta_k
 \end{aligned}$$

en utilisant (2.13), on aura :

$$\begin{aligned}
 f(x_k + \lambda_k d_k) &\leq f(x_k) + \rho \lambda_k \nabla^T f(x_k) d_k & (3.16) \\
 \Rightarrow f(x_k + \lambda d_k) &\leq f(x_k) + \rho \lambda \nabla^T f(x_k) d_k \leq f(x_k) + \left| \rho \lambda \nabla^T f(x_k) d_k \right| \\
 \Rightarrow f(x_k + \lambda d_k) &\leq f(x_k) + \rho \lambda \left| \nabla^T f(x_k) d_k \right| \leq f(x_k) - \rho \lambda \nabla^T f(x_k) d_k \\
 \Rightarrow f(x_k + \lambda d_k) &\leq f(x_k) - \rho \lambda \left\| \nabla^T f(x_k) \right\| \|d_k\| \cos \theta_k \\
 \Rightarrow f(x_k + \lambda d_k) &\leq f(x_k) - \frac{\rho(1 - \sigma)}{L} \left\| \nabla^T f(x_k) \right\|^2 \cos^2 \theta_k
 \end{aligned}$$

On en déduit (3.1). ■

## 3.2 La méthode du gradient conjugué

Les méthodes du gradient conjugué sont utilisées pour résoudre les problèmes d'optimisation non linéaires sans contraintes spécialement les problèmes 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 suivant :

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

Où  $f$  est régulière (continûment différentiable).

Dans cette partie on va décrire toutes ces méthodes, mais avant d'accéder à ces dernières, on va d'abord donner le principe général d'une méthode à directions conjuguées.

### 3.3 Le principe général d'une méthode à directions conjuguées

**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 :

$$x^T A y = 0$$

#### 3.3.1 Description de la méthode

Soit  $\{d_1, d_2, \dots, d_n\}$  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 définie par  $q(x) = \frac{1}{2}x^T A x + b^T x + c$ , avec  $x \in \mathbb{R}^n$  et  $A$  une matrice de  $Mn \times n$  symétrique et définie positive,  $b \in \mathbb{R}^n$  et  $c \in \mathbb{R}$ , conduisant à l'optimum en  $n$  étapes au plus. Cette méthode est de la forme suivante :

$$x_0 \text{ donné}$$

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

où  $k$  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  $g_k = \nabla q(x)$ , la méthode se construit comme suit :

**Calcul de  $\lambda_k$**

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

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

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

Soit :

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

d'où l'on tire :

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

### **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  $\xi_0, \dots, \xi_k$  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 égal au sous espace généré par  $\xi_0, \dots, \xi_i$ .

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  $\xi_0, \dots, \xi_i$ , et  $\xi_{i+1}$  est linéairement indépendant de  $\xi_0, \dots, \xi_i$ .

$\xi_{i+1}$  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  $d_0, \dots, d_i$ .

## 3.4 Méthode du gradient conjugué dans le cas quadratique

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

$$\nabla q(x) = Ax + b \quad (3.17)$$

$$\text{et } \nabla^2 q(x) = A. \quad (3.18)$$

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 forte pente).

### 3.4.1 Algorithme de La méthode du gradient conjugué pour les fonctions quadratiques

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 :

$$x_{k+1} = x_k + \lambda_k d_k \quad (3.19)$$

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 les coefficients  $\beta_{k+1}$  étant choisis de telle manière que  $d_k$  soit conjuguée avec toutes les directions précédentes. Autrement dit :

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

on en déduit que

$$\begin{aligned} d_{k+1}^T A d_k &= 0 \Rightarrow (-\nabla q(x_{k+1}) + \beta_{k+1} d_k)^T A d_k = 0, \\ &\Rightarrow -\nabla^T q(x_{k+1}) A d_k + \beta_{k+1} d_k^T A d_k = 0, \\ &\Rightarrow \beta_{k+1} = \frac{\nabla^T q(x_{k+1}) A d_k}{d_k^T A d_k} = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}. \end{aligned} \quad (3.20)$$

2- Déterminer le pas  $\lambda_k$

En particulier, une façon de choisir  $\lambda_k$  est de résoudre le problème d'optimisation (à une seule variable) suivant :

$$\lambda_k = \min f(x_k + \lambda d_k) \quad , \quad \lambda > 0. \quad (3.21)$$

On en déduit que

$$\lambda_k = \frac{-d_k^T g_k}{d_k^T A d_k} = -\frac{1}{d_k^T A d_k} g_k^T d_k = \frac{-d_k^T g_k}{d_k^T A d_k}. \quad (3.22)$$

Le pas  $\lambda_k$  obtenu ainsi s'appelle le pas optimal.

**Algorithme 3.1 (Algorithme du gradient conjugué "quadratique")**

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

$$\lambda_k = \frac{-d_k^T g_k}{d_k^T A d_k}. \quad (3.23)$$

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k. \quad (3.24)$$

$$\beta_{k+1} = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}. \quad (3.25)$$

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

### 3.4.2 Les avantages de la méthode du gradient conjugué linéaire

1- La consommation mémoire de l'algorithme est minimale : on doit stocker les quatre vecteurs  $x_k, g_k, d_k, A d_k$  ( bien sur  $x_{k+1}$  prend la place de  $x_k$  au niveau de son calcul avec des remarques analogues pour  $g_{k+1}, d_{k+1}, A d_{k+1}$ ) et les scalaires  $\lambda_k, \beta_{k+1}$ .

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.

## 3.5 Synthèse des résultats de convergence des méthodes du gradient conjugué

On présente dans cette section une synthèse concernant la convergence des différentes variantes du gradient conjugué avec la recherche linéaire inexacte de *Wolfe forte* et de *Wolfe faible*.

Notre problème consiste à résoudre le problème  $(P)$  cité ci-dessus

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

où  $f$  est régulière (continûment différentiable) et  $g$  est son gradient. Notons par  $g_k$  le gradient de  $f$  au point  $x_k$ . Rappelons que les différentes variantes du gradient conjugué genèrent une suite  $\{x_k\}$  de la forme suivante :

$$x_{k+1} = x_k + \lambda_k d_k, \quad (3.26)$$

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

$$d_k = \begin{cases} -g_1 & \text{si } k = 1, \\ -g_k + \beta_k d_{k-1} \dots \dots \dots \text{si } k \geq 2, \end{cases} \quad (3.27)$$

le pas  $\lambda_k \in \mathbb{R}$  étant déterminé par une recherche linéaire. Rappelons les différentes variantes du gradient conjugué qui diffèrent selon les valeurs que prennent les constantes  $\beta_k$ . Particulièrement, on cite les variantes suivantes :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad (\text{Fletcher} - \text{Reeves}). \quad (3.28)$$

$$\beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2} \quad (\text{Polak} - \text{Ribière}). \quad (3.29)$$

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})}, \quad (\text{Hestenes} - \text{Stiefel}). \quad (3.30)$$

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}}, \quad (\text{Conjugate Descent Method}). \quad (3.31)$$

où  $\|\cdot\|$  est la norme Euclidienne. Récemment Dai et yuan ont aussi introduit la forme suivante :

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T (g_k - g_{k-1})}. \quad (3.32)$$

La convergence globale des méthodes (3.28)-(3.29)-(3.30)-(3.31)-(3.32) a été étudiée par beaucoup d'auteurs. Citons en particulier *Al-Baali* [3], *Gilbert et Nocedal* [25], *Hestenes et Stiefel* [32], *Hu et Story* [33], *Liu, Han et Yin* [37], *Powell* [47] *Touati-Ahmed* et *Story* [55], et *Zou-*

*tendijk* [63]. Un facteur clé dans l'étude de la convergence globale est comment sélectionner le paramètre  $\lambda_k$ . Le choix le plus naturel de  $\lambda_k$  est de faire une recherche linéaire exacte, c.-à-d. poser  $\lambda_k = \arg \min_{\lambda \geq 0} f(x_k + \lambda d_k)$ . Cependant, ce choix naturel pourrait ne pas donner de convergence comme est le cas de la méthode (PRP) et (HS), comme l'a montré *Powell* dans [47]. Inspiré par le travail de *Powell, Gilbert et Nocedal* dans [25] ont montré que la méthode (PRP) est globalement convergente si  $\beta_k^{PR}$  est non-négatif et  $\lambda_k$  est déterminé avec la recherche linéaire satisfaisant la condition suffisante de descente suivante :

$$g_k^T d_k < -c \|g_k\|^2, \quad \text{où } c > 0, \quad (3.33)$$

en plus des conditions standard de *Wolfe* :

$$\begin{cases} f(x_k + \lambda_k d_k) \leq f(x_k) + w_1 \lambda_k d_k^T g_k, \\ d_k^T g_{k+1} \geq \omega_2 d_k^T g_k. \end{cases} \quad (3.34)$$

ou'  $0 < \omega_1 < \omega_2 < 1$ .

Récemment, *Dai et Yuan* [10][11] ont montré que la méthode (CD) et la méthode (FR) sont globalement convergentes si les conditions de la recherche linéaire suivantes pour  $\lambda_k$  sont satisfaites :

$$\begin{cases} f(x_k + \lambda_k d_k) \leq f(x_k) + \omega_1 \lambda_k d_k^T g_k, \\ \omega'_2 d_k^T g_k \leq d_k^T g_{k+1} \leq -\omega''_2 d_k^T g_k. \end{cases} \quad (3.35)$$

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

Nous adoptons la supposition suivante sur fonction  $f$ .

**Supposition 3.1.** *Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ .*

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

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

$$\exists L > 0 \text{ tel que } \|g(x) - g(\tilde{x})\| \leq L \|x - \tilde{x}\|, \forall x, \tilde{x} \in \mathcal{N} \quad (3.36)$$

**Remarque 3.1** Ces suppositions impliquent qu'il existe  $\gamma > 0$  tel que

$$\|g(x)\| \leq \gamma, \forall x \in \mathcal{L} \quad (3.37)$$

Rappelons maintenant le Théorème suivant obtenu essentiellement par *Zoutendijk* [63] et *Wolfe* [57][58]. Ce théorème assure la satisfaction de la condition de *Zoutendijk*[63], pour toute méthode du type (3.27)-(3.28), dans laquelle le pas  $\lambda_k$  est déterminé par la règle de *Wolfe* faible (3.35).

**Théorème de Zoutendijk. 3.1.**

Considérons la suite  $(x_k)_k$ , définie par (3.26) où  $d_k$  est une direction de descente et  $\lambda_k$  vérifie les conditions (3.35). Considérons aussi que la supposition 3.1 soit satisfaite, alors :

$$\sum_{k \geq 1} \cos^2 \theta \|g_k\|^2 < \infty, \quad (3.38)$$

est vérifiée, avec

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

**Remarque 3.2.** (3.38)  $\iff$

$$\sum_{k \geq 1, \|d_k\| \neq 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \quad (3.39)$$

où  $\theta_k$  est l'angle que fait  $d_k$  avec  $-g_k$ .

**En effet**

D'après la 2<sup>eme</sup> condition de (3.35) nous avons

$$d_k^T (g_{k+1} - g_k) \geq (\omega_2 - 1) d_k^T g_k.$$

D'autre part

$$\begin{aligned} d_k^T (g_{k+1} - g_k) &\leq \|g_{k+1} - g_k\| \|d_k\| \\ &\leq \lambda_k L \|d_k\|^2, \end{aligned}$$

d'où

$$\lambda_k \geq \left( \frac{\omega_2 - 1}{L} \right) \frac{d_k^T g_k}{\|d_k\|^2}.$$

En remplaçant ceci dans la 1<sup>ère</sup> condition de on aura :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \mu \cos^2 \theta_k \|g_k\|^2,$$

où  $\mu = \frac{\omega_1(\omega_2 - 1)}{L}$ .

Or puisque  $f$  est bornée sur  $\mathcal{N}$  on a :

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

Ce qui achève la démonstration. ■

### 3.5.1 Résultats de Convergence du Gradient Conjugué, version Fletcher-Reeves

Avant d'exposer sans démonstration les différents résultats de convergence, donnons d'abord l'algorithme de la méthode de Fletcher Reeves :

**Algorithme 3.2 de la méthode de Fletcher-Reeves**

**Etape0 : (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"

Sinon aller à l'étape 2.

**Etape 2 :**

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

$$\lambda_k = \min_{\lambda > 0} f(x_k + \lambda d_k).$$

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

où

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

### 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  $n$ ).

### Convergence de la méthode de Fletcher-Reeves

Le premier résultat de convergence de la méthode du gradient conjugué non linéaire (version *Fletcher Reeves*) avec la recherche linéaire inexacte de *Wolfe forte* suivante

$$\begin{cases} f(x_k + \lambda_k d_k) \leq f(x_k) + \omega_1 \lambda_k d_k^T g_k, \\ |d_k^T g_{k+1}| \leq -\omega_2 d_k^T g_k. \end{cases} \quad (3.40)$$

où ( $\omega_2 < \frac{1}{2}$ ) a été démontré par *Al-Baali* [3].

- *Touati Ahmed* et *Story* [55] ont généralisé ce résultat pour

$$0 \leq \beta_k \leq \beta_k^{FR}. \quad (3.41)$$

- *Gilbert* et *Nocedal* [25] ont généralisé ce résultat pour

$$|\beta_k| \leq \beta_k^{FR}. \quad (3.42)$$

- Récemment, *Dai et Yuan* [13] a montré que la méthode (*FR*) est globalement convergente avec la recherche linéaire inexacte de *Wolfe relaxée*.

On donne dans ce paragraphe les principaux résultats de convergence de la méthode du gradient conjugué non linéaire avec la recherche linéaire inexacte de *Wolfe forte* [*Gilbert et Nocedal*], aussi avec la recherche linéaire inexacte de *Wolfe relaxée* [*Dai et Yuan*].

La propriété de descente de la méthode de Fletcher-Reeves

**proposition 3.4.** [25]

On considère que la supposition (3.1) est satisfaite. Considérons une méthode du type (3.26) et (3.27) avec  $\beta_k$  satisfaisant à (3.40) et le pas  $\lambda_k$  vérifiant la règle de Wolfe forte (3.40) où  $\omega_2 \in ]0, \frac{1}{2}[$ .

Alors cette méthode génère des directions de descente. De plus on a :

$$\frac{-1}{1 - \omega_2} \leq \frac{d_k^T g_k}{\|g_k\|^2} \leq \frac{2\omega_2 - 1}{1 - \omega_2}; \quad k = 1, \dots$$

**Théorème 3.1.** [25] On considère que la supposition (3.1) est satisfaite. Toute méthode du type (3.26) et (3.27) dans laquelle  $\beta_k$  vérifie

$$|\beta_k| \leq \beta_k^{FR}, \quad \forall k \geq 1,$$

et le pas  $\lambda_k$  est déterminé par la règle de Wolfe forte (3.40) avec  $0 < \omega_1 < \omega_2 < \frac{1}{2}$  est une méthode de descente ( $g_k^T d_k < 0, \forall k \geq 1$ ) convergente, dans le sens où

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \tag{3.43}$$

**Théorème 3.2** [25] Soit  $x_1$  un point de départ pour lequel la supposition (3.1) soit satisfaite. Considérons une méthode du type (3.26) et (3.27) avec  $\beta_k$  satisfaisant à (3.28). On suppose aussi que

◆ Chaque direction  $d_k$  vérifie

$$d_k^T g_k \leq 0 \tag{3.44}$$

◆ Le pas  $\lambda_k$  est déterminé par la règle de Wolfe relaxée (3.35) avec  $\omega''_2 < \infty$ , alors

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



### 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 à chaque  $n$  itérations par exemple.

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

### 3.5.2 Résultats de Convergence du Gradient Conjugué, version Polak Ribière et Polyak

La variante dite de *Polak-Ribière et Polyak (PRP)* consiste à définir  $\beta_k$  par la formule (3.29). Cette méthode fut découverte par *Polak, Ribière* [45] et *Polyak* [46]. L'algorithme de la méthode de (PRP) est le suivant

**Algorithme 3.3 de la méthode de Polak-Ribière-Polyak**

**Etape0 : (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"

Sinon aller à l'étape 2.

**Etape 2 :**

Définir  $x_{k+1} = x_k + \lambda > 0 d_k$  avec :

$$\lambda_k = \min_{\alpha > 0} f(x_k + \alpha d_k).$$

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

où

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

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

### 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 ([45, 1969]) ont démontré la convergence de la méthode (PRP) à travers le théorème suivant :

**Proposition 3.5.** [22] *Si  $f$  est fortement convexe, de classe  $C^1$  avec un gradient lipschitzien, alors la méthode de Polak Ribière et Polyac avec la recherche linéaire exacte génère une suite  $\{x_k\}$  convergeant vers l'unique point  $x_*$  réalisant le minimum de  $f$ .*

**Démonstration.** ([24])

Montrons dans un premier temps que

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

est 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

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

où  $\eta > 0$ .

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

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

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 3.1 et la recherche linéaire exacte, la condition de Zoutendijk est vérifié. Et comme  $\{f(x_k)\}$  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  $x^*$ .

**Cas où  $f$  n'est pas convexe** Powell ([48, 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([47, 1986]), a modifié la variante  $\beta_k^{PRP}$  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  $\lambda_k$  est déterminé par la règle de Wolfe forte.

Gilbert et Nocedal ([25, 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  $\beta_k$  comme suit :

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

### 3.5.3 Résultats de convergence du gradient conjugué, version Descente conjuguée

Cette méthode fut proposée en 1987 par Fletcher et Reeves([20, 1987]),

Rappelons que pour la méthode de la descente conjuguée la variante  $\beta_k$  est :

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

### La propriété de descente de la méthode de la descente conjuguée

Fletcher ([20, 1987]) a démontré que la méthode de la descente conjuguée est une méthode pour laquelle la condition de descente suffisante est assurée si le pas  $\lambda_k$  est déterminé par la règle forte de Wolfe (3.40) avec  $\sigma \leq \frac{1}{2}$ .

Dai et Yuan ([13, 1996]) ont démontré le théorème suivant :

**Théorème 3.3.** *Supposons que Supposition (3.1) est satisfaite.*

*Pour toute méthode du type (0.2) et (0.3) dont  $\beta_k$  satisfait à (3.31) et le pas  $\lambda_k$  satisfait aux conditions de Wolfe relaxée*

*(2.21) et (2.22) :*

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

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

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

**Démonstration.** ([13, 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 (2.22)

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

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 (3.41) de cette méthode avec un pas satisfaisant aux conditions (2.13)-(2.21) 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 3.4.** *Supposons que Supposition (3.1) est satisfaite.*

*Toute méthode du type (0.2) et (0.3) dans laquelle  $\beta_k$  vérifie (3.32) et le pas  $\lambda_k$  est déterminé par la règle de Wolfe relaxée (2.21)-(3.22) où  $0 < \rho < \sigma_1 < 1$  et  $\sigma_2 = 0$ ; la méthode converge dans le sens où*

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

**Démonstration.** ([13,1996])

Du théorème 3.3 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  $\beta_{k+1}^{CD}$  vérifie l'inégalité (3.41).

D'après le théorème 3.2 on a :

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

### 3.5.4 Résultats de convergence du gradient conjugué version(Dai-Yuan)

Récemment *Dai* et *yuan* ([12, 1998]) ont introduit une formule pour  $\beta_k$  sous la forme suivante :

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \quad (3.45)$$

où  $y_{k-1} = g_k - g_{k-1}$ .

Cette méthode possède plusieurs propriétés, par exemple elle possède la propriété de descente à chaque itération et elle converge si le pas est déterminé par la règle de (*Wolfe faible*, *Armijo* et *Goldstein*) lorsque  $f$  est strictement convexe.

### Descente de la méthode de Dai-Yuan

**Théorème 3.5.[16]** *Soit  $x_1$  un point de départ pour lequel la supposition (3.1) est satisfaite. Considérons une méthode du type (3.26) et (3.27) avec  $\beta_k$  satisfaisant à (3.45).*

◆ *Si  $f$  est strictement convexe sur l'ensemble convexe  $\mathcal{L}$  c- à- d :*

$$(g(x) - g(y))^T(x - y) > 0, \quad \text{pour tout } x, y \in \mathcal{L}. \quad (3.46)$$

Alors pour tout  $k \geq 1$  :

$$g_k^T d_k < 0 \quad (3.47)$$

En 1999 les auteurs ont généralisé ce résultat pour toute fonction régulière avec la recherche de Wolfe faible (3.35).

**Théorème 3.6.[12]** *Supposons que (3.26) soit satisfaite. Pour toute méthode du type (3.26) et (3.27) dont  $\beta_k$  satisfait à (3.45) et le pas  $\lambda_k > 0$  satisfaisant aux conditions de Wolfe faible (3.34), alors toutes les directions générées sont de descente, autrement dit :*

$$d_k^T g_k < 0, \quad \forall k \geq 1$$

### Convergence de la méthode de Dai-Yuan

Les résultats suivant sont dûs à *Dai* et *Yuan* [16]. Ils assurent la convergence de cette méthode pour une fonction strictement convexe avec une recherche linéaire inexacte d'*Armijo* et *Goldstein*.

**Théorème 3.7.[16]** *Soit  $x_1$  un point de départ pour lequel la supposition (3.1) soit satisfaite. Considérons une méthode du type (3.26) et (3.27) avec  $\beta_k$  satisfaisant à (3.45).*

◆ *Si  $f$  est strictement convexe sur l'ensemble  $\mathcal{L}$  et le pas  $\lambda_k$  satisfait aux conditions de Goldstein suivantes :*

$$w_2 \lambda_k \nabla^T f(x_k) d_k \leq f(x_k + \lambda_k d_k) - f(x_k) \leq \omega_1 \lambda_k \nabla^T f(x_k) d_k, \quad (3.48)$$

où  $g_k = \nabla f(x_k)$  et  $\omega_1$  et  $\omega_2$  sont deux constantes vérifiant  $0 < \omega_1 < 1/2 < \omega_2 < 1$ .



Alors :

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

**Théorème 3.8.**[16] *Soit  $x_1$  un point de départ pour lequel la supposition (3.1) est satisfaite. Considérons une méthode du type (3.26) et (3.27) avec  $\beta_k$  satisfaisant à (3.45).*

◆ *Si  $f$  est uniformément convexe sur l'ensemble convexe  $\mathcal{L}$  c-à-d s'il existe une constante  $\eta > 0$  tel que :*

$$(g(x) - g(y))^T(x - y) \geq \eta \|x - y\|^2, \quad \text{pour tout } x, y \in \mathcal{L}, \quad (3.49)$$

*et le pas  $\lambda_k$  satisfait aux conditions d'Armijo suivantes :*

$$f(x_k + \lambda_k d_k) - f(x_k) \leq \omega \lambda_k \nabla^T f(x_k) d_k, \quad (3.50)$$

*où  $g_k = \nabla f(x_k)$  et  $0 < \omega < 1$ , pour lequel la condition suffisante de descente (3.33) est assurée.*

Alors

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

**Algorithme 3.4 de la Méthode de Dai-Yuan avec la règle de Wolfe faible**

**Etape0 : (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  $x_{k+1} = x_k + \lambda_k d_k$  avec

$\lambda_k$  vérifie les conditions (3.10).

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

où

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T [g_{k+1} - g_k]} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}.$$

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

### 3.6 Analyse de la convergence des méthodes du gradient conjugué avec la recherche linéaire non monotone

Nous avons constaté que les résultats de convergence pour les différentes variantes du gradient conjugué ont été obtenus en utilisant des recherches linéaires inexactes . La question naturelle qui se pose est la suivante :

**Peut-on obtenir des résultats de convergence avec ces variantes du Gradient Conjugué ou d'autres en utilisant la recherche linéaire non-monotone ?**

La réponse a été donnée positivement dans les travaux de B. Tahar , L. Yamina, and G. Rafik [53], Y. H. Dai [11]

#### 3.6.1 Résultats de Convergence du Gradient Conjugué avec la recherche linéaire non monotone d'Armijo

**Lemme 3.1.**[11]

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction différentiable. On suppose que

(i) La fonction  $f$  est bornée inférieurement dans  $\mathbb{R}^n$

(ii) Son gradient est lipchitzien i.e

$$\exists L > 0 \text{ tel que } \|g(x) - g(\tilde{x})\| \leq L \|x - \tilde{x}\|, \forall x, \tilde{x} \in \mathcal{N} \quad (3.51)$$

Considérons une méthode de type (0.2) dans laquelle  $d_k$  est une direction de descente et le pas  $\lambda_k$  est déterminé par la recherche linéaire non monotone d'Armijo (2.25)-(2.26)

alors pour tout  $l \geq 1$ , on a

$$\max_{1 \leq i \leq M} f(x_{Ml+i}) \leq \max_{1 \leq i \leq M} f(x_{Ml(l-1)+i}) + \delta \max_{0 \leq i \leq M-1} [\lambda_{Ml+i} g_{Ml+i}^T d_{Ml+i}], \quad (3.52)$$

$$\sum_{l \geq 1} \min_{0 \leq i \leq M-1} \left\{ \left| g_{Ml+i}^T d_{Ml+i} \right|, \left( g_{Ml+i}^T d_{Ml+i} \right)^2 / \|d_{Ml+i}\|^2 \right\} < +\infty \quad (3.53)$$

**preuve**

Pour prouver (3.52) il suffit de montrer que l'inégalité suivante est satisfaite pour  $j = 1, \dots, M$ .

$$f(x_{Ml+i}) \leq \max_{1 \leq i \leq M} f(x_{M(l-1)+i}) + \delta \lambda_{Ml+j-1} g_{Ml+j-1}^T d_{Ml+j-1}. \quad (3.54)$$

Puisque les conditions de la recherche linéaire (2.25)-(2.26) impliquent

$$f(x_{Mi+1}) \leq \max_{0 \leq i \leq m(MI)} f(x_{M-i}) + \delta \lambda_{MI} g_{MI}^T d_{MI} \quad (3.55)$$

il s'ensuit de ceci que :

$$m(MI) \leq M - 1$$

et que (3.54) est satisfaite pour  $j = 1$ . Supposons que (3.54) est satisfaite pour tout  $1 \leq j \leq M - 1$ . En vertu de la propriété de la descente de  $d_k$ , on a

$$\max_{1 \leq i \leq j} f(x_{Ml+i}) \leq \max_{1 \leq i \leq M} f(x_{Ml(l-1)+i}). \quad (3.56)$$

D'après les conditions de la recherche linéaire, il suit que :

$$m(MI + j) \leq M - 1$$

et de (3.56), on obtient que :

$$\begin{aligned}
 f(x_{MI+j+1}) &\leq \max_{0 \leq i \leq m(MI+j)} f(x_{MI+j+i}) + \delta \lambda_{MI+j} g_{MI+j}^T d_{MI+j} \\
 &\leq \max \left\{ \max_{1 \leq i \leq M} f(x_{M(l-1)+i}), \max_{1 \leq i \leq j} f(x_{MI+i}) \right\} + \delta \lambda_{MI+j} g_{MI+j}^T d_{MI+j} \\
 &\leq \max_{1 \leq i \leq M} f(x_{M(l-1)+i}) \delta \lambda_{MI+j} g_{MI+j}^T d_{MI+j}.
 \end{aligned}$$

ainsi (3.54) est aussi juste pour  $j + 1$ . Par induction (3.54) est satisfaite pour  $1 \leq j \leq M$ . Par conséquent on a (3.52).

Puisque  $f(x)$  est bornée inférieurement, il s'ensuit que :

$$\max_{1 \leq i \leq M} f(x_{M(j+i)}) > -\infty.$$

En sommant (3.52) par rapport  $l$ , on peut obtenir

$$\sum_{l \geq 1} \min_{0 \leq i \leq M-1} [-\lambda_{MI+i} g_{MI+i}^T d_{MI+i}] < +\infty. \quad (3.57)$$

supposons que (2.26) est fautive pour le pas  $\bar{\lambda}_k$ . Alors, nous avons

$$f(x_k + (\lambda_k/\sigma)d_k) > \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta(\lambda_k/\sigma)g_k^T d_k \geq f(x_k) + \delta(\lambda_k/\sigma)g_k^T d_k \quad (3.58)$$

D'après le théorème des valeurs moyennes et la continuité de Lipschitz du  $\nabla f$ , on peut montrer que

$$\begin{aligned}
 f(x_k + \lambda d_k) - f(x) &= \lambda g_k^T d_k + \int_0^\alpha [\nabla f(x_k + t d_k) - g_k]^T d_k dt \leq \lambda g_k^T d_k + \int_0^\alpha t L \|d_k\|^2 dt \\
 &= \lambda g_k^T d_k + (1/2) L \lambda^2 \|d_k\|^2 \leq \delta \lambda g_k^T d_k
 \end{aligned}$$

pour tout

$$\lambda \in (0, [2(1 - \delta)/L] \cdot |g_k^T d_k| / \|d_k\|^2) \quad (3.59)$$

Il s'ensuit de (3.58) - (3.59) que

$$\lambda_k / \sigma \geq [2(1 - \delta)/L] |g_k^T d_k| / \|d_k\|^2 \quad (3.60)$$

Si (2.26) est vraie pour le pas  $\bar{\lambda}_k$ , alors  $\bar{\lambda}_k = \lambda_k \geq \lambda_1$ . Donc, la relation suivante est satisfaite pour certain constant  $c > 0$  :

$$\lambda_k \geq \min \{ \lambda_1, c |g_k^T d_k| / \|d_k\|^2 \} \quad (3.61)$$

en vertu de (3.57) implique (3.53). ■

**Remarque 3.1 :** De la relation (3.52), on voit que pour n'importe quelle méthode de recherche linéaire non monotone , l'ordre  $\left\{ \max_{1 \leq i \leq M} f(x_{M(l-1)+i}) \right\}$  est strictement décroissant. En conséquence, cette relation joue un rôle important dans les analyses de convergence R-linéaires.

Supposons qu'il existe deux constantes  $C_1$  et  $C_2$  telles que, pour tout  $k$ , on ait :

$$g_k^T d_k \leq -c_1 \|g_k\|^2 \quad (3.62)$$

$$\|d_k\| \leq c_2 \|g_k\| \quad (3.63)$$

Nous pouvons montrons facilement le résultat de convergence suivant en utilisant le lemme 3.1.

**Théorème 3.9.**[11] Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction différentiable. On suppose que :

- (i) La fonction  $f$  est bornée inférieurement dans  $\mathbb{R}^n$
- (ii) Le gradient est lipchitzien i.e

$$\exists L > 0 \text{ tel que } \|g(x) - g(\tilde{x})\| \leq L \|x - \tilde{x}\|, \forall x, \tilde{x} \in \mathcal{N} \quad (3.64)$$

Considérons une méthode de type (0.2) dans laquelle  $d_k$  satisfait les deux conditions (3.62)-

(3.63) et le pas  $\lambda_k$  est déterminé par la recherche linéaire non monotone d'Armijo (2.25)-(2.26).

Alors il existe une constante  $C_3$  tel que

$$\|g_{k+1}\| \leq c_3 \|g_k\| \text{ pour tout } k \quad (3.65)$$

en plus nous avons que

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \quad (3.66)$$

**Preuve**

Notons que  $\lambda_k \leq n_2$  donc (3.61) et (3.63) nous donne :

$$\|x_{k+1} - x_k\| \leq \lambda_k \|d_k\| \leq c_2 n_2 \|g_k\|, \quad (3.67)$$

D'après le (ii) on a que :

$$\|g_{k+1} - g_k\| \leq c_2 n_2 L \|g_k\|. \quad (3.68)$$

Ainsi (3.65) est satisfaite pour

$$c_3 = 1 + c_2 n_2 L$$

En outre, d'après (3.53), (3.62), (3.63) il s'en suit que

$$\lim_{k \rightarrow \infty} \|g_{MI+\Phi(I)}\| = 0, \quad (3.69)$$

où

$$0 \leq \Phi(I) \leq M - 1.$$

D'après (3.65) on a que

$$\|g_{M(I+1)+i}\| \leq c_3^{2M} \|g_{MI+\phi(I)}\| \text{ for } i = 0, \dots, M - 1. \quad (3.70)$$

Par conséquent, il s'en suit de (3.69) - (3.70) que (3.66) est satisfaite.

Dans le cas où l'ensemble

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

est bornée, la relation (3.66) implique que chaque point de  $\{x_k\}$  est un point stationnaire de  $f(x)$ . Il s'en suit de (3.66) - (3.67) que  $x_{k+1} - x_k$  tend vers zéro quand  $k \rightarrow \infty$ . Ceci montre que, si le nombre des points stationnaires de  $f$  est fini, la suite  $\{x_k\}$  est convergente. Ainsi, nous prouvons encore la convergence globale de ces méthodes avec la recherche linéaire non monotone pour des fonctions générales sous les conditions (3.62) - (3.63). Cependant, la relation utile (3.53) peut être établie, et par conséquent elle rend possible d'affaiblir les conditions (3.62) et (3.63) en utilisant la direction de recherche  $d_k$ . Par exemple, si pour  $\|d_k\|^2$  il existe deux constantes positives  $\beta$  et  $\gamma$  telles que :

$$\|d_k\|^2 \leq \beta + \gamma \text{ pour tout } k \quad (3.71)$$

et si la condition suffisante (3.62) est satisfaite, alors on peut montrer que la méthode converge dans le sens

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \blacksquare \quad (3.72)$$

**Théorème 3.10** [11]. Supposons que  $f(x)$  est bornée inférieurement sur  $\mathbb{R}^n$  et que son gradient  $\nabla f(x)$  est continu Lipschitzien . Considérons n'importe quelle méthode itérative (0.2), où  $d_k$  satisfait (3.62) et (3.71), et où  $\lambda_k$  est obtenu par la recherche linéaire non monotone (2.25)-(2.26). Alors, la méthode converge dans le sens de (3.72).

**Preuve**

On suppose le contraire donc la relation (3.72) n'est pas satisfaite Alors, il existe une certaine constante  $\tau > 0$  telle que :

$$\|g_k\| \geq \tau \text{ pour tout } k \geq 1, \quad (3.73)$$

de la relation (3.62) on a :

$$g_k^T d_k \geq c_1 \tau^2. \quad (3.74)$$

Alors, il s'en suit de (3.74) et (3.71) que :

$$\begin{aligned} & \sum_{l \geq 1} \min_{0 \leq i \leq M-1} \left\{ \left| g_{Ml+i}^T d_{Ml+i} \right|, \left( g_{Ml+i}^T d_{Ml+i} \right)^2 / \|d_{Ml+i}\|^2 \right\} \\ & \geq \sum_{l \geq 1} \min_{0 \leq i \leq M-1} \{ c_1 \tau^2, c_1 \tau^4 / [\beta + \gamma (MI + i)] \} \\ & \geq \sum_{l \geq 1} \min \{ c_1 \tau^2, c_1 \tau^4 / [\beta + \gamma (MI + M - 1)] \} = +\infty \end{aligned} \quad (3.75)$$

La relation (3.75) contredit (3.53).■

[11] a détendu plus loin la condition de la descente suffisante (3.62). Dans [11], on a proposé un algorithme du gradient conjugué non monotone pour résoudre (1), pour lequel seulement les relations suivantes sont montrées pour tout  $k$  :

$$-g_k d_k \geq \min \{ \beta, \gamma / \sqrt{k} \} \quad (3.76)$$

$$(g_k^T d_k)^2 / \|d_k\|^2 \geq \tau / k \quad (3.77)$$

où  $\beta, \gamma, \tau$  sont des constantes positives. Il est évident que les relations ci-dessus sont plus faibles que les conditions (3.62) et (3.63). Cependant, pareillement au théorème 3.10, nous pouvons prouver sans difficultés par les relations (3.53), (3.76), (3.77) que l'algorithme converge dans le sens de (3.72).■



### 3.6.2 Résultats de convergence du gradient conjugué avec la recherche linéaire non monotone de Wolfe

Dans cette partie, on étudie la convergence globale de la méthode du gradient conjugué en utilisant une nouvelle classe de recherche linéaire non monotone de Wolfe. L'idée vient des techniques de recherche de F-role.

Dans l'analyse de convergence et implémentation de la méthode du gradient conjugué. La règle de wolfe pour la recherche linéaire non monotone classique notée (RLN) élargie est :

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \gamma_1 \lambda_k g_k^T d_k \quad (3.78)$$

$$\gamma_2 g_k^T d_k \leq g(x_{k+\lambda_k d_k})^T d_k \quad (3.79)$$

où  $0 < \gamma_1 \leq \gamma_2 \leq 1, \lambda_1 > 0$  et  $M \in \mathbb{N}$  ,

$$0 \leq m(k) \leq \min \{m(k-1) + 1, M\}, m(0) = 0 \quad (3.80)$$

Nous allons voir dans cette partie comment on introduit la nouvelle classe de recherche linéaire non monotone de Wolfe que l'on notera par (RLN1)

#### Technique de la recherche linéaire non monotone de Wolfe

Etant donné d'abord, les suppositions générales de cette partie.

**Supposition 3.2.** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction différentiable. On suppose que :

(A<sub>1</sub>) L'ensemble de niveau  $L_0 = \{x \mid f(x) \leq f(x_0), x \in \mathbb{R}^n\}$  est borné, où  $x_0$  et le point de départ.

(A<sub>2</sub>)  $f$  est fortement convexe et différentiable dans l'ensemble de niveau  $L_0$  et son gradient  $g_k = \nabla f(x_k)$  est continue de lipschitz . ie, il existe une constante  $L > 0$  tel que :

$$\|g(x) - g(y)\| \leq L \|x - y\|, \forall x, y \in \mathcal{N} \quad (3.81)$$

**Définition 3.2.** La fonction  $\sigma : [0, \infty) \rightarrow [0, \infty)$  est une F-fonction, si pour toute suite  $\{t_i\} \subset [0, \infty)$

$$\lim_{k \rightarrow \infty} \sigma(t_i) = 0 \Rightarrow \lim_{k \rightarrow \infty} t_i = 0 \quad (3.82)$$

**Définition 3.3.** Soit

$$\eta = \sup\{\|g(x) - g(y)\| / x, y \in L_0\} > 0.$$

Alors l'application  $\delta : [0, \infty) \rightarrow [0, \infty)$  définie par

$$\delta(t) = \begin{cases} \inf \left\{ \frac{\|x-y\|}{\|g(x)-g(y)\|} \geq t \right\}, & t \in [0, \eta) \\ \lim_{s \rightarrow \eta^-} \delta(s), & t \in [\eta, \infty) \end{cases}$$

est appelé la fonction reverse modulus du gradient  $g(x)$ .

Maintenant on donne la F-rule de la recherche linéaire non monotone (RLN1) comme suit :

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) - \sigma(t_k) \quad (3.83)$$

où  $\sigma$  est une fonction différentiable et  $t_k = -\frac{g_k^T d_k}{\|d_k\|}$ . Posons  $x_{k+1} = x_k + \lambda_k d_k$

Evidemment, si  $M = 0$ , la F-rule non monotone est justement la règle de décroissance suffisante dans [44].

Notons aussi que toute fonction non décroissante  $\sigma : [0, \infty) \rightarrow [0, \infty)$  telle que  $\sigma(0) = 0$  et  $\sigma(t_i) > 0$  pour  $t > 0$  est nécessairement une F-fonction. Par convention dans la suite on pose :

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j})$$

où

$$k - m(k) \leq l(k) \leq k$$

En utilisant la définition 3.3 on a

$$\lambda_k \|d_k\| \geq \delta \left[ (\gamma_2 - 1) \frac{g_k^T d_k}{\|d_k\|} \right]$$

qui désigne

$$\lambda_k \|d_k\| \geq \delta \left[ (1 - \gamma_2) \left( -\frac{g_k^T d_k}{\|d_k\|} \right) \right] \quad (3.84)$$

où  $\delta(\cdot)$  est la fonction reverse modulus du gradient. donc, il s'en suit de (3.78) et (3.84) que :

$$f(x_k + \lambda_k d_k) \leq f(x_{l(k)}) + \gamma_1 \lambda_k g_k^T d_k$$

$$f(x_k + \lambda_k d_k) = f(x_{l(k)}) - \gamma_1 \lambda_k \|d_k\| \left( -\frac{g_k^T d_k}{\|d_k\|} \right)$$

$$f(x_k + \lambda_k d_k) \leq f(x_{l(k)}) - \gamma_1 \left( -\frac{g_k^T d_k}{\|d_k\|} \right) \delta \left[ (1 - \gamma_2) \left( -\frac{g_k^T d_k}{\|d_k\|} \right) \right]$$

$$f(x_k + \lambda_k d_k) \leq f(x_{l(k)}) - \sigma \left( -\frac{g_k^T d_k}{\|d_k\|} \right) \quad (3.85)$$

où  $\sigma(t) = \gamma_1 t \delta[(1 - \gamma_2)t]$ ,  $t \geq 0$ . clairement,  $\sigma(t)$  est une fonction différentiable. Ceci indique que la règle (3.78) - (3.79) satisfait la F-rule non monotone (3.83)

### Algorithme 3.5

Etant donné  $\rho > 0$ ,  $\beta \in [0, 1]$ ,  $\gamma_2 \in [0, 1]$ ,  $M$  un entier non négatif, faire l'étape initiale du test

$$r_k = \frac{\rho g_k^T d_k}{\|d_k\|^2}$$

Prendre  $\lambda_k = \beta^{h(k)} r_k$ ,  $h(k) = 0, 1, 2, \dots$ ,  $h(k)$  est faire

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) - \sigma(t_k)$$

où  $m(0) = 0$ ,  $0 \leq m(k) \leq \min\{m(k-1) + 1, M\}$ ,  $k \geq 1$ , évidemment si la direction de

la descente  $d_k$  est supposée être  $d_k^T g_k$ , alors si  $m(k)$  est suffisamment grand, l'inégalité (3.83) est toujours satisfaite, donc elle satisfait l'existence de  $\lambda_k$ .

Dans la recherche linéaire ci-dessus à chaque itération il est recommandé que l'étape initiale du test  $r_k$  reste la même mais peut être ajusté automatiquement.

**Remarque 3.2**

1- Afin d'utiliser une recherche linéaire non monotone le calcul du pas  $\lambda_k$  doit se faire suivant la direction de descente, on prouve que cette étude du gradient conjugué préserve la direction de recherche  $d_k$ .

2- Afin de calculer un grand pas  $\lambda_k$  on doit avoir :

$$\sum_{k=0}^{\infty} \sigma(t_k) < \infty \tag{3.86}$$

**Analyse de convergence**

Dans cette section, on établit les propriétés de la convergence globale des méthodes du gradient conjugué avec F-role non monotone (RLN1). Notons que pour établir notre résultat, on a besoin de quelques conditions.

**Lemme 3.2.** Soit  $d_k$  une direction de descente et  $\lambda_k$  le pas trouvé par la recherche linéaire non monotone alors  $\{x_k\} \subset L_0$ .

**Preuve.** La recherche linéaire non monotone (RLN1) (3.83) montre que :

$$f(x_1) \leq f(x_0) - \sigma(t_0) < f(x_0)$$

$$f(x_2) \leq \max_{0 \leq j \leq m(1)} f(x_{1-j}) - \sigma(t_1) < f(x_0)$$

$$f(x_3) \leq \max_{0 \leq j \leq m(2)} f(x_{2-j}) - \sigma(t_2) < f(x_0)$$

$$f(x_{k+1}) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) - \sigma(t_k) < f(x_0)$$

On alors  $\{x_k\} \subset L_0$ . ■

**Lemme 3.3** [34]. Soit  $\lambda_k$  le pas obtenu par la recherche linéaire non monotone (RLN) et  $d_k$  une direction de descente. Soit  $g_k$  le gradient de  $f$ . Alors :

$$g_k^T d_k \leq -\frac{7}{8} \|g_k\|^2, k = 0, 1, 2, \dots \quad (3.87)$$

**Preuve.** Puisque  $d_0 = -g_0$ , on a  $g_0^T d_0 = -\|g_0\|^2$

$$g_k^T d_k = -\|g_k\|^2 + \beta_k g_k^T d_{k-1}$$

où  $\beta_k = \beta_k^{HZ}$

$$g_k^T d_k = -\|g_k\|^2 + \beta_k^N g_k^T d_{k-1}$$

$$\begin{aligned} -g_k^T d_k &= \|g_k\|^2 - g_k^T d_{k-1} \left( \frac{g_k^T y_{k-1}}{-g_{k-1}^T d_{k-1}} - 2g_k^T d_{k-1} \frac{\|y_{k-1}\|^2}{(-g_{k-1}^T d_{k-1})^2} \right) \\ &= \frac{\|g_k\|^2 (-g_{k-1}^T d_{k-1})^2 - g_k^T d_{k-1} g_k^T y_{k-1} (-g_{k-1}^T d_{k-1}) + 2(g_k^T d_{k-1})^2 \|y_{k-1}\|^2}{(-g_{k-1}^T d_{k-1})^2} \end{aligned} \quad (3.88)$$

on applique l'inégalité

$$u^T v \leq \frac{1}{2} (\|u\|^2 + \|v\|^2)$$

au second terme dans (3.88) avec

$$u = \frac{1}{2} g_k (-g_{k-1}^T d_{k-1}), \quad v = 2(g_k^T d_{k-1}) y_{k-1}$$

Donc, on peut obtenir (3.87) pour tout  $k \in N$ .

Pour la simplification, on introduit la notation suivante :

$$l(k) = \max \left\{ i \mid 0 \leq k - i \leq m(k), f(x_i) = \max_{0 \leq j \leq m(k)} f(x_k - j) \right\} \quad (3.89)$$

i.e  $l(k)$  est un entier non négatif qui satisfait les deux formules suivantes :

$$k - m(k) \leq l(k) \leq k \quad (3.90)$$

$$f(x_{l(k)}) = \max_{0 \leq j \leq m(k)} f(x_{k-j}) \quad (3.91)$$

D'où la recherche linéaire non monotone (RLN) (3.78) peut être réécrite comme suit :

$$f(x_k + \lambda_k d_k) \leq f(x_{l(k)}) - \sigma(t_k). \blacksquare \quad (3.92)$$

**Lemme 3.4.** Sous les conditions de la suppositions  $(A_1)$ , la suite  $f(x_{l(k)})$  est croissante.

**Preuve.** D'après (3.83), pour tout  $k$  on a :

$$f(x_{k+1}) \leq f(x_{l(k)})$$

d'après la recherche linéaire non monotone (RLN),  $0 \leq m(k) \leq m(k-1) + 1$ , on a :

$$\begin{aligned} f(x_{l(k)}) &= \max_{0 \leq j \leq m(k)} f(x_{k-j}) \leq \max_{0 \leq j \leq m(k-1)+1} f(x_{k-j}) \\ &= \max \left\{ \max_{0 \leq j \leq m(k-1)} f(x_{k-1-j}), f(x_k) \right\} \\ &= \max \{ f(x_{l(k-1)}), f(x_k) \} \\ &= f(x_{l(k-1)}). \blacksquare \end{aligned}$$

**Lemme 3.5.** En supposant que  $(A_1)$  soit vérifiée, alors  $f(x_{l(k)})$  converge i.e  $\lim_{k \rightarrow \infty} f(x_{l(k)})$  existe.

et

$$\lim_{k \rightarrow \infty} \sigma(t_{l(k)-1}) = 0 \quad (3.93)$$

**Preuve.** On a que  $f(x)$  est bornnée inférieurement. Puisque  $f(x_k) \leq f(x_0)$ ,  $\forall k$ ,  $\{x_k\} \subset L_0$

(voir la preuve du Lemme 3.2 ) et  $\{f(x_{l(k)})\}$  est croissante (voir la preuve du Lemme 3.4 ) , alors

$\lim_{k \rightarrow \infty} f(x_{l(k)})$  existe. D'après (3.92), on a

$$f(x_{l(k)}) \leq f(x_{l(k)-1}) - \sigma(t_{l(k)-1})$$

quand  $k \rightarrow \infty$ ,

$$\lim_{k \rightarrow \infty} \sigma(t_{l(k)-1}) = 0. \blacksquare$$

**Theorème 3.11.** Soit la fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  satisfaisant la supposition 3.1. Soit la suite  $\{x_k\}$  définie par (0.2) où le pas  $\lambda_k$  défini par la F-role non monotone (RLN1) . Si la direction  $d_k$  satisfait

$$-\frac{g_k^T d_k}{\|d_k\|} \geq \sigma(\|g_k\|), k = 0, 1, \dots, \quad (3.94)$$

et

$$\|d_k\| \leq m_1 \|g_k\| \quad (3.95)$$

où  $\sigma(\cdot)$  est une fonction différentiable et  $m_1 > 0$ . Alors la suite  $\{x_k\} \subset L_0$  et

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

**Preuve.** D'après la définition 3.2 on a que :

$$\lim_{k \rightarrow \infty} t_{l(k)-1} = -\frac{g_{l(k)-1}^T d_{l(k)-1}}{\|d_{l(k)-1}\|} = 0 \quad (3.96)$$

En utilisant la condition (3.94), on déduit :

$$\lim_{k \rightarrow \infty} \sigma(\|g_{l(k)-1}\|) = 0$$

ce qui implique que :

$$\lim_{k \rightarrow \infty} \|g_{l(k)-1}\| = 0$$

d'après la définition 3.2.

Alors il s'en suit d'après (3.95) que

$$\lim_{k \rightarrow \infty} \|d_{l(k)-1}\| = 0 \quad (3.97)$$

Soit

$$l_1(k) = l(k + M + 2) \quad (3.98)$$

On prouve par récurrence que pour tout  $j \geq 1$

$$\lim_{k \rightarrow \infty} \|d_{l_1(k)-j}\| = 0 \quad (3.99)$$

et

$$\lim_{k \rightarrow \infty} f(x_{l_1(k)-j}) = \lim_{k \rightarrow \infty} f(x_{l(k)}) \quad (3.100)$$

Si  $j = 1$ , puisque  $\{l_1(k)\} \subset \{l(k)\}$ , (3.99) et (3.100) donne (3.97). Supposons que (3.99) et (3.100) sont vérifiées pour un  $j$  donné. On considère le cas de  $j + 1$ . Puisque

$$f(x_{l_1(k)-j}) \leq f(x_{l_1(k)-(j+1)}) - \sigma(\gamma_{l_1(k)-(j+1)})$$

en utilisant le même argument pour dériver (3.97), on obtient :

$$\lim_{k \rightarrow \infty} \|d_{l_1(k)-(j+1)}\| = 0 \quad (3.101)$$

Maintenant (3.99) est établi. Mais

$$x_{l_1(k)-j} - x_{l_1(k)-(j+1)} = \lambda_{l_1(k)-(j+1)} d_{l_1(k)-(j+1)}$$

Notons que  $L_0 = \{x \mid f(x) \leq f(x_0), x \in R^n\}$  est borné,  $x_{k+1} = x_k + \lambda_k d_k \in L_0$ , pour tout  $k$  et



que  $\lambda_k$  reste borné, on a

$$\lim_{k \rightarrow \infty} \left\| x_{l_1(k)-j} - x_{l_1(k)-(j+1)} \right\| = 0$$

$f(x)$  est uniformément continue sur le niveau  $L_0$ , telle que :

$$\begin{aligned} \lim_{k \rightarrow \infty} f(x_{l_1(k)-(j+1)}) &= \lim_{k \rightarrow \infty} f(x_{l_1(k)-j}) \\ &= \lim_{k \rightarrow \infty} f(x_{l_1(k)}) = \lim_{k \rightarrow \infty} f(x_{l(k)}) \end{aligned} \quad (3.102)$$

Ceci montre que pour  $j \geq 1$ , (3.99) est aussi établi.

Par la définition de  $L_1$  et (3.90) on a :

$$L_1(k) = L(k + M + 2) \leq k + M + 2$$

c.à.d

$$L_1(k) - k - 1 \leq M + 1 \quad (3.103)$$

Donc, pour tout  $k$

$$\begin{aligned} x_{k+1} &= x_{L_1(k)} - \sum_{j=1}^{L(k)-k-1} (x_{l_1(k)-j+1} - x_{l_1(k)-j}) \\ &= x_{L_1(k)} - \sum_{j=1}^{L(k)-k-1} \lambda_{l_1(k)-j} d_{l_1(k)-j} \end{aligned} \quad (3.104)$$

d'après (3.103) on a :

$$\left\| x_{k+1} - x_{L_1(k)} \right\| = \left\| - \sum_{j=1}^{L(k)-k-1} \lambda_{l_1(k)-j} d_{l_1(k)-j} \right\| \leq \sum_{j=1}^{M+1} \left\| \lambda_{l_1(k)-j} d_{l_1(k)-j} \right\| \quad (3.105)$$

Et quand  $k \rightarrow \infty$ , il vient que

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_{L_1(k)}\| = 0 \quad (3.106)$$

Donc, d'après la continuité uniforme de  $f(x)$ ,  $\lim_{k \rightarrow \infty} f(x_{l_1(k)}) = \lim_{k \rightarrow \infty} f(x)$ , alors d'après (3.105), on peut voir que

$$\lim_{k \rightarrow \infty} f(x_{l(k)}) = \lim_{k \rightarrow \infty} f(x_K) \quad (3.107)$$

D'où, pour

$$f(x_{k+1}) \leq f(x_{l(k)}) - \sigma \left( \frac{-g_k^T d_k}{\|d_k\|} \right)$$

en prenant la limite pour  $k \rightarrow \infty$ , on obtient

$$\lim_{k \rightarrow \infty} \sigma \left( -\frac{g_k^T d_k}{\|d_k\|} \right) = 0.$$

qui signifie

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \blacksquare$$

# Chapitre 4

## Convergence de la méthode de FR avec une nouvelle classe de recherche linéaire non monotone

- 
- 1- Introduction.
  - 2- Convergence Globale.
  - 3- Expériences numérique.
-

## 4.1 Introduction

Dans ce chapitre une nouvelle classe de recherche linéaire non monotone de Wolfe notée (RLN2) est présentée pour prouver la descente suffisante et la convergence globale de la méthode du gradient conjugué de Fletcher Reeves (FR).

Cette méthode possède la propriété de descente suffisante et converge globalement.

Le premier objectif de ce travail est d'étudier la convergence globale et la performance pratique de la méthode de gradient conjugué de (FR) avec la nouvelle recherche linéaire non monotone de Wolfe forte (RLN2).

Considérons le problème d'optimisation sans contraintes suivant :

$$\min \{f(x) : x \in \mathbb{R}^n\} \quad (4.1)$$

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est continûment différentiable. La forme générale de la méthode du gradient conjugué est donnée par :

$$x_{k+1} = x_k + \lambda_k d_k \quad (4.2)$$

$$d_k = \begin{cases} -g_k & , k = 0 \\ -g_k + \beta_k d_{k-1} & , k \geq 1 \end{cases} \quad (4.3)$$

où  $g_k = \nabla f(x_k)$ ,  $\lambda_k$  est le pas obtenu par une certaine recherche linéaire, et  $\beta_k$  [12, 21, 32, 46] est un scalaire. Il existe plusieurs manières pour déterminer  $\beta_k$ , rappelons que celui qui correspond à la méthode du gradient conjugué version Fletcher Reeves est donné par :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \text{ Gradient conjugué variante Fletcher Reeves.} \quad (4.4)$$

Dans l'analyse de la convergence et de l'implémentation de la méthode du gradient conjugué, la recherche linéaire non monotone de Wolfe forte (RLN) est la suivante :

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T d_k \quad (4.5)$$

$$\sigma_1 g_k^T d_k \leq g(x_k + \lambda_k d_k)^T d_k \leq -\sigma_2 g_k^T d_k, \quad (4.6)$$

où  $0 < \delta \leq \sigma_1 \leq 1, 0 \leq \sigma_2 \leq 1, 0 \leq \alpha \leq 1$  et  $M_0 \in \mathbb{N}$ ,

$$0 \leq m(k) \leq \min \{m(k-1) + 1, M_0\}, m(0) = 0$$

De plus, la condition de la descente suffisante est comme suit :

$$g_k^T d_k \leq -c \|g_k\|^2 \quad (4.7)$$

Pour  $f$  fortement convexe, le pas  $\lambda_k$  satisfait la nouvelle condition de recherche linéaire non monotone de Wolfe (RLN2) suivante :

$$f(x_k + \lambda_k d_k) - \alpha \max_{0 \leq j \leq m(k)} f(x_{k-j}) - (1 - \alpha) \min_{0 \leq j \leq m(k)} f(x_{k-j}) \leq \delta \lambda_k g_k^T d_k \quad (4.8)$$

Lucidi et Roma [38] ont présenté un algorithme avec une recherche linéaire non monotone de (FR) en 1995. G. H. Liu, L. L. Jing, L. X. Han, And D. Han, et Han dans [36] ont montré que les méthodes (PRP) et (HS) pour une fonction objective convexe avec la recherche linéaire de Wolfe non monotone convergent globalement avec la recherche linéaire de Wolfe forte non monotone (RLN). En 2002, Dai Yu Hong [10] approfondi l'étude de la recherche linéaire non monotone fondée initialement par Grippo, Lampariello et Lucidi et l'a utilisé pour montrer la convergence de la méthode du gradient conjugué version (DY) et (CD).

#### **Remarque 4.1**

La nouvelle recherche linéaire non monotone de Wolfe (RLN2) peut être vu en quelques sorte comme une combinaison convexe de la recherche linéaire non monotone de Wolfe forte (RLN) et la recherche linéaire inexacte de Wolfe.

Lorsque  $\alpha = 1$ , la nouvelle recherche linéaire non monotone de Wolfe (RLN2) se réduit à la recherche linéaire non monotone de Wolfe forte (RLN), et lorsque  $\alpha = 0$  la nouvelle recherche linéaire non monotone se réduit à la recherche linéaire inexacte de Wolfe.

Dans cette partie, on montre la convergence globale de la méthode du gradient conjugué de

(FR) avec la recherche linéaire non monotone de Wolfe (RLN2).

**Supposition 4.1.** On suppose que

(A<sub>1</sub>) L'ensemble de niveau  $L_1 = \{x \mid f(x) \leq f(x_1), x \in R^n\}$  est borné, où  $x_1$  est le point de départ

(A<sub>2</sub>)  $f$  est fortement convexe et différentiable dans l'ensemble de niveau  $L_1$  et son gradient  $g_k = \nabla f(x_k)$  est continu et lipschitzien. *ie*, il existe une constante  $L > 0$  et  $\eta > 0$  :

$$\|g(x) - g(y)\| \leq L \|x - y\|, \forall x, y \in \mathcal{N} \quad (4.9)$$

et

$$(g(x) - g(y))^T(x - y) \geq \eta \|x - y\|^2 \quad (4.10)$$

Si  $f$  satisfait les suppositions (A<sub>1</sub>) et (A<sub>2</sub>), on peut obtenir que :

$$\|g(x)\| \leq \gamma \quad (4.11)$$

pour tout  $x \in L_1$ .

A présent, on donne le théorème suivant qui illustre que la formule (4.5) possède la condition de descente suffisante sans aucune recherche linéaire.

**Théorème 4.1.** Considérons les méthodes (4.2) et (4.3), où  $\beta_k = \beta_k^{FR}$ . Alors pour tout  $k \geq 1$

$$g_k^T d_k = -\|g_k\|^2 \quad (4.12)$$

**Preuve :** Comme  $d_0 = -g_0$ , on a  $g_0^T d_0 = -\|g_0\|^2$

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

$$\begin{aligned}
 &= -\frac{d_{K-1}^T y_{k-1}}{\|g_{k-1}\|^2} \|g_k\|^2 + \frac{\|g_k\|^2}{\|g_{k-1}\|^2} g_k^T d_{K-1} \\
 &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} (-d_{k-1}^T y_{k-1} + g_k^T d_{K-1}) \\
 &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} (-d_{K-1}^T y_{k-1} + d_{K-1}^T g_k) \\
 &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} [-d_{K-1}^T (g_k - g_{k-1}) + d_{K-1}^T g_k] \\
 &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} d_{k-1}^T g_{k-1}
 \end{aligned}$$

Ainsi , on trouve la relation de récurrence suivante :

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

donc

$$\begin{aligned}
 g_k^T d_k &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} d_{k-1}^T g_{k-1} \\
 &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \frac{\|g_{k-1}\|^2}{\|g_{k-2}\|^2} d_{k-2}^T g_{k-2} = \frac{\|g_k\|^2}{\|g_{k-2}\|^2} d_{k-2}^T g_{k-2} \\
 &= \dots \\
 &= \frac{\|g_k\|^2}{\|g_0\|^2} d_0^T g_0 \\
 &= \frac{\|g_k\|^2}{\|g_0\|^2} (-\|g_0\|^2) = -\|g_k\|^2 < 0
 \end{aligned}$$

**Lemme 4.1.** Supposons que la supposition 4.1 soit satisfaite et  $\lambda_k$  est obtenu par la nouvelle

recherche linéaire non monotone de Wolfe (RLN2) (4.8) et (4.6). Alors il existe  $c_1 > 0$ , telle que

$$\|s_k\| \geq \frac{c_1(-g_k^T d_k)}{\|d_k\|^2}, \quad (4.13)$$

où  $s_k = x_{k+1} - x_k$ .

**Preuve.** D'après la supposition 4.1 (A<sub>1</sub>)

$$\|g(x) - g(y)\| \leq L \|x - y\|,$$

c'est-à-dire

$$\|y_k\| \leq L \|s_k\|, \quad (4.14)$$

donc on a

$$y_k^T d_k \leq L \|s_k\| \|d_k\|, \quad (4.15)$$

et d'après (1.10) on peut avoir

$$y_k^T d_k = d_k^T (g_{k+1} - g_k) \geq (1 - \sigma_1)(-g_k^T d_k) \quad (4.16)$$

Ainsi, d'après (2.7) et (2.8) on obtient

$$\|s_k\| \geq \frac{(1 - \sigma_1)}{L} \times \frac{(-g_k^T d_k)}{\|d_k\|^2} \quad (4.17)$$

soit  $c_1 = \frac{(1 - \sigma_1)}{L}$  et d'après (4.17), on obtient alors (4.13).

**Lemme 4.2.** Supposons que la supposition 4.1 soit satisfaite,  $\lambda_k$  est donné par (4.5) et (4.6), et  $\beta_k = \beta_k^{FR}$ . Alors il existe une constante positive  $M = \frac{L^2}{\eta}$  telle que

$$\frac{\|y_k\|^2}{y_k^T s_k} \leq M \quad (4.18)$$

**Preuve :** D'après la supposition de la convexité, on a



$$y_k^T d_k = d_k^T (g_{k+1} - g_k) \geq \eta \lambda_k \|d_k\|^2 \quad (4.19)$$

et d'après la continuité de lipschitz (4.9), on peut obtenir que

$$\|y_k\| = \|g_{k+1} - g_k\| \leq L \|x_{k+1} - x_k\| = L \lambda_k \|d_k\| \quad (4.20)$$

En utilisant (4.3) et (4.20), on peut obtenir que

$$\frac{\|y_k\|^2}{y_k^T s_k} \leq \frac{L^2 \lambda_k^2 \|d_k\|^2}{\eta \lambda_k^2 \|d_k\|^2} = \frac{L^2}{\eta} = M \quad (4.21)$$

ce qui achève la preuve. ■

**Lemme 4.3.** Supposons que l'inégalité suivante soit satisfaite pour tout  $k$

$$0 < m_1 \leq \|g_k\| \leq m_2 \quad (4.22)$$

et  $\lambda_k$  est donné par (4.8) et (4.6), alors

(1) il existe une constante positive  $b > 1$  telle que

$$|\beta_k| \leq b \quad (4.23)$$

(2) il existe une constante positive  $\mu$ , telle que  $\|y_{k-1}\| \leq \mu$ , alors  $|\beta_k| \leq \varepsilon$  pour tout  $\varepsilon > 0$ .

**Preuve.** Lorsque  $\beta_k = \beta_k^{FR}$ , d'après (4.12) et (4.22) on peut obtenir que :

$$\begin{aligned}
 |\beta_k| &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} = \frac{\|g_k - g_{k-1} + g_{k-1}\|^2}{\|g_{k-1}\|^2} \\
 &= \frac{\|g_k - g_{k-1} + g_{k-1}\| \|g_k - g_{k-1} + g_{k-1}\|}{\|g_{k-1}\|^2} \\
 &\leq \frac{(\|g_k - g_{k-1}\| + \|g_{k-1}\|)(\|g_k - g_{k-1}\| + \|g_{k-1}\|)}{\|g_{k-1}\|^2} \\
 &= \frac{\|g_k - g_{k-1}\|^2 + 2\|g_k - g_{k-1}\| \|g_{k-1}\| + \|g_{k-1}\|^2}{\|g_{k-1}\|^2} \\
 &\leq \frac{\|g_k\|^2 + \|g_{k-1}\|^2 + 2\|g_k\| \|g_{k-1}\| + 2\|g_k - g_{k-1}\| \|g_{k-1}\| + \|g_{k-1}\|^2}{\|g_{k-1}\|^2} \\
 &\leq \frac{\|g_k\|^2 + \|g_{k-1}\|^2 + 2\|g_k\| \|g_{k-1}\| + 2\{\|g_k\| + \|g_{k-1}\|\} \|g_{k-1}\| + \|g_{k-1}\|^2}{\|g_{k-1}\|^2} \\
 &= \frac{\|g_k\|^2 + 2\|g_{k-1}\|^2 + 2\|g_k\| \|g_{k-1}\| + 2\|g_k\| \|g_{k-1}\| + 2\|g_{k-1}\|^2}{\|g_{k-1}\|^2} \\
 &= \frac{\|g_k\|^2 + 4\|g_{k-1}\|^2 + 4\|g_k\| \|g_{k-1}\|}{\|g_{k-1}\|^2} \\
 &\leq \frac{9m_2^2}{m_1^2} = \left(\frac{3m_2}{m_1}\right)^2
 \end{aligned}$$

Soit  $b = \left(\frac{3m_2}{m_1}\right)^2$ , évidemment on a  $b > 1$  car  $m_2 \geq m_1 > 0$ .

soit  $\mu = \frac{2m_1^3}{9m_2^2}\varepsilon$ ,

$$\begin{aligned}
 |\beta_k| &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \leq \frac{\|g_k - g_{k-1}\|^2 + 2\|g_k - g_{k-1}\| \|g_{k-1}\| + \|g_{k-1}\|^2}{\|g_{k-1}\|^2} \\
 &\leq \frac{\|g_k - g_{k-1}\|^2 + 2\|g_k - g_{k-1}\| \|g_{k-1}\| + g_{k-1}^T d_{k-1}}{\|g_{k-1}\|^2} \\
 &\leq \frac{\|g_k - g_{k-1}\|^2 + 2\|g_k - g_{k-1}\| \|g_{k-1}\| + g_{k-1}^T d_{k-1}}{\|g_{k-1}\|^2} \\
 &\leq \frac{\|g_k - g_{k-1}\|^2 + 2\|g_k - g_{k-1}\| \|g_{k-1}\|}{\|g_{k-1}\|^2} + \frac{g_{k-1}^T d_{k-1}}{\|g_{k-1}\|^2}
 \end{aligned}$$

on a

$$g_{k-1}^T d_{k-1} < 0$$

donc

$$\begin{aligned} |\beta_k| &\leq \frac{\|g_k - g_{k-1}\|^2 + 2\|g_k - g_{k-1}\| \|g_{k-1}\|}{\|g_{k-1}\|^2} \\ &= \frac{\|g_k - g_{k-1}\| (\|g_k - g_{k-1}\| + 2\|g_{k-1}\|)}{\|g_{k-1}\|^2} \\ &\leq \frac{(\|g_k\| + 3\|g_{k-1}\|) \|y_{k-1}\|}{\|g_{k-1}\|^2} \\ &= \frac{4m_2}{m_1^2} \|y_{k-1}\| = \frac{8m_2^2}{2m_1^2 m_2} \|y_{k-1}\| \\ &\leq \frac{8m_2^2}{2m_1^2 m_1} \|y_{k-1}\| \leq \frac{9m_2^2}{2m_1^3} \|y_{k-1}\| \\ &\leq \frac{9m_2^2}{2m_1^3} \mu \leq \varepsilon \blacksquare \end{aligned}$$

**Lemme 4.4.** Supposons que la supposition 4.1 (A<sub>1</sub>) soit satisfaite et  $\lambda_k$  est obtenu par la nouvelle recherche linéaire non monotone (4.8) et (4.6), notons  $\zeta_k = \delta \lambda_k g_k^T d_k$ , alors  $\{f(x_k)\}$  est décroissante et

$$\lim_{k \rightarrow \infty} \zeta_{l(k+1)-1} = 0 \quad (4.24)$$

où

$$l(k) = \max \left\{ i \mid 0 \leq k - i \leq m(k), f(x_i) = \max_{0 \leq j \leq m(k)} f(x_{k-j}) \right\} \quad (4.25)$$

et  $k - m(k) \leq l(k) \leq (k)$

**Preuve.** D'après (4.8) et (4.25) on peut obtenir que :

$$f(x_k + \lambda_k d_k) \leq \alpha \max_{0 \leq j \leq m(k)} f(x_{k-j}) + (1 - \alpha) \min_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta \lambda_k g_k^T \quad (4.26)$$

$$\leq f(x_{l(k)}) + \zeta_k \quad (4.27)$$

Puisque  $\zeta_k < 0$ , alors :

$$f(x_{(k+1)}) \leq f(x_{l(k)})$$

d'après (4.25) et

$$m(k) \leq m(k-1) + 1$$

pour tout  $k$ , on peut obtenir que

$$f(x_{l(k)}) \leq \max_{0 \leq j \leq m(k-1)+1} f(x_{k-j}) = \max \left\{ \max_{0 \leq j \leq m(k-1)} f(x_{k-1-j}), f(x_k) \right\} \quad (4.28)$$

$$= \max f(x_{l(k-1)}), f(x_k) = f(x_{l(k-1)}), \quad k = 1, 2, \dots$$

donc  $\{f(x_k)\}$  est décroissante et puisque

$l(k+1) - 1 \geq k + 1 - m(k+1) - 1 \geq k - M_0$ , on a

$$f(x_{l(l(k+1)-1)}) \leq f(x_{l(k-M_0)}) \quad (4.29)$$

d'après l'inégalité ci-dessus et (4.27), on peut obtenir

$$f(x_{l(k+1)}) \leq f(x_{l(l(k+1)-1)}) + \zeta_{l(k+1)-1}$$

$$\leq f(x_{l(k-M_0)}) + \zeta_{l(k+1)-1}$$

donc, on a

$$0 \leq -\zeta_{l(k+1)-1} \leq f(x_{l(k-M_0)}) - f(x_{l(k+1)}) \quad (4.30)$$

d'après (4.30) et lorsque la supposition 4.1 (A<sub>1</sub>) est satisfaite, on a

$$\lim_{k \rightarrow \infty} \zeta_{l(k+1)-1} = 0$$

**Lemme 4.5.** Supposons que  $\lambda_k$  est donné par (4.8) et (4.6) et  $\beta_k = \beta_k^{FR}$ . Alors  $\{l(k)\}$  est croissante. ■

**Preuve.** Supposons que

$$l(k+1) < l(k) \tag{4.31}$$

alors on a

$$k+1 \geq k \geq l(k) > l(k+1) \geq k+1 - m(k+1) \tag{4.32}$$

d'après la définition de  $l(k+1)$  et (4.28), on peut obtenir

$$f(x_{l(k+1)}) \geq f(x_{l(k)}) \tag{4.33}$$

mais d'après le lemme 4.4, on a

$$f(x_{l(k+1)}) \leq f(x_{l(k)}) \tag{4.34}$$

donc, par la définition de  $l(k+1)$  et (4.33), on a

$$l(k+1) \geq l(k) \tag{4.35}$$

ceci est contredit (4.33) . Donc  $l(k) \leq l(k+1)$ , *i.e*  $\{l(k)\}$  est croissante. ■

Maintenant on peut présenter la nouvelle méthode du gradient conjugué de la descente comme suit

**Algorithme de la nouvelle recherche linéaire non monotone de Wolfe forte (RLN2).**

**étape 1 :** Etant donné  $x_1 \in \mathbb{R}^n$ ,  $\varepsilon \geq 0$ ,  $0 < \delta \leq \sigma_1 \leq 1$ ,  $0 \leq \sigma_2 \leq 1$ ,  $M_0 \in \mathbb{N}$ ,  $0 \leq \alpha \leq 1$ ,

posons  $d_1 = -g_1$ ,  $k = 1$ , if  $\|g_k\| \leq \varepsilon$ , alors stop

**étape 2 :** Trouver  $\lambda_k \geq 0$  par (4.8) et (4.6).

**étape 3 :** Soit  $x_{k+1} = x_k + \lambda_k d_k$  et  $g_{k+1} = g(x_{k+1})$ , si  $\|g_{k+1}\| \leq \varepsilon$ , alors stop.

**étape 4 :** Calculer  $\beta_k = \beta_k^{FR}$  par la formule (1.5) et générer  $d_{k+1}$  par (4.3).

étape 5 : Poser  $k = k + 1$ , aller à l'étape 2.

## 4.2 convergence Globale

**Théorème 4.2.** Supposons que la supposition 4.1 est satisfaite et  $\lambda_k$  est obtenu par la nouvelle recherche linéaire non monotone de Wolfe (RLN2), (4.8) et (4.6), considérons la méthode d'itération de la forme (4.2) et (4.3), où  $\beta_k = \beta_k^{FR}$ . Alors

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (4.36)$$

**Preuve.** Si la supposition 4.1, alors il existe une constante  $\gamma_1$  telle que

$$0 < \gamma_1 \leq \|g(x)\| \leq \gamma_2$$

pour tout  $k$ . D'après (4.3) et lemme 4.3, on a

$$\begin{aligned} \|d_{l(k+1)-1}\| &\leq \|g_{l(k+1)-1}\| + |\beta_{l(k+1)-1}| \|d_{l(k+1)-2}\| \\ &\leq \gamma_2 + b \|d_{l(k+1)-2}\| \\ &\leq \dots \\ &\leq \gamma_2 \sum_{j=0}^{l(k+1)-l(k)-3} (b)^j + (b)^{l(k+1)-l(k)-2} \|d_{l(k)+1}\| \\ &\leq \gamma_2 \sum_{j=0}^{l(k+1)-l(k)-2} (b)^j + (b)^{l(k+1)-l(k)-2} |\beta_{l(k)+1}| \|d_{l(k)}\| \end{aligned} \quad (4.37)$$

puisque

$$l(k+1) - l(k) \leq k + 1 - [k - m(k)] \leq M_0 + 2$$

on peut obtenir

$$\sum_{j=0}^{l(k+1)-l(k)-2} (b)^j \leq \sum_{j=0}^{M_0} (b)^j, (b')^{l(k+1)-l(k)-2} \leq (b)^{M_0}$$

où

$$\gamma_2 \sum_{j=0}^{M_0} (b)^j = h_1, (b)^{M_0} = h_2$$

Alors

$$\|d_{l(k+1)-1}\| \leq h_1 + h_2 |\beta_{l(k)}| \|d_{l(k)}\| \quad (4.38)$$

d'après le lemme 4.2 et (4.6), on a

$$0 \leq \|y_k\| \leq \sqrt{M y_k^T s_k} \leq \sqrt{M(1 + \sigma_2)(-\lambda_k g_k^T d_k)}$$

d'après le lemme 4.3, on note que  $\varepsilon = \frac{1}{h_2 b^2}$ , il existe un entier non négative  $k_0$ , tel que pour  $k \geq k_0$  on a  $|\beta_{l(k)}| < \varepsilon$ .

Donc on peut obtenir

$$\begin{aligned} \|d_{l(k+1)-1}\| &\leq h_1 + \frac{\|d_{l(k)}\|}{b^2} \\ &\leq h_1 + \frac{\|g_{l(k)}\| + |\beta_{l(k)}| \|d_{l(k)-1}\|}{b^2} \\ &\leq h_1 + \frac{\gamma_2 + b \|d_{l(k)-1}\|}{b^2} \\ &\leq h_3 + \frac{\|d_{l(k)-1}\|}{b} \end{aligned}$$

où  $h_3 = h_1 + \frac{\gamma_2}{b^2}$ , on a une équation réursive qui conduit à

$$\begin{aligned}
 \|d_{l(k+1)-1}\| &\leq h_3 + \frac{\|d_{l(k)-1}\|}{b} \\
 &\leq h_3 + \frac{h_3 + \frac{\|d_{l(k-1)-1}\|}{b}}{b} \leq \dots \\
 &\leq h_3 \sum_{j=0}^{k-k_0} \left(\frac{1}{b}\right)^j + \frac{1^{k-k_0+1}}{b} \|d_{l(k_0)-1}\| \\
 &\leq h_3 \sum_{j=0}^{\infty} \left(\frac{1}{b}\right)^j + \|d_{l(k_0)-1}\| \\
 &\leq h_3 \frac{b}{b-1} + \|d_{l(k_0)-1}\|
 \end{aligned} \tag{4.39}$$

En appliquant  $l(k) \geq k - m(k) \geq k - M_0$  et le lemme 4.5, on peut supposer

$$l(i) - 1 \leq j < l(i+1) - 1, i \geq k_0 + 2$$

pour tout  $j \geq l(k_0 + 2) - 1$ , donc on a

$$\begin{aligned}
 \|d_j\| &\leq \|g_j\| + |\beta_j| \|d_{j-1}\| \leq \gamma_2 + b(\|g_{j-1}\| + |\beta_{j-1}| \|d_{j-1}\|) \leq \dots \\
 &\leq \gamma_2 \sum_{t=0}^{j-l(i)} (b)^t + (b)^{j-l(i)+1} \|d_{l(i)-1}\|
 \end{aligned} \tag{4.40}$$

Donc, d'après

$$\begin{aligned}
 j - l(i) + 1 &\leq [l(i+1) - 1] - l(i) + 1 \\
 &\leq i + 1 - [i - m(i)] \\
 &\leq M_0 + 1
 \end{aligned}$$

et (4.40), on a



$$\|d_j\| \leq \gamma_2 \sum_{t=0}^{j-l(i)} (b)^t + (b)^{M_0+1} \|d_{l(i)-1}\| \quad (4.41)$$

D'après (4.40) et (4.41), on a

$$\|d_j\| \leq \gamma_2 \sum_{t=0}^{M_0} (b)^t + (b)^{M_0+1} \left[ h_3 \frac{b}{b-1} + \|d_{l(k_0)-1}\| \right] \quad (4.42)$$

pour tout  $i - 1 > k_0$ . En utilisant le lemme 4.1 et le lemme 4.2, on a

$$\begin{aligned} -\zeta_{l(k+1)-1} &= \frac{\delta \|s_{l(k+1)-1}\| (-g_{l(k+1)-1}, d_{l(k+1)-1})}{\|d_{l(k+1)-1}\|} \\ &\geq \delta c_1 \frac{(-g_{l(k+1)-1}, d_{l(k+1)-1})^2}{\|d_{l(k+1)-1}\|} \\ &\geq \rho c_1 \frac{\gamma_1^4}{\|d_{l(k+1)-1}\|} \geq 0 \end{aligned} \quad (4.43)$$

mais d'après (4.24) et (3.8), on peut obtenir que :

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

ceci contredit (4.44).

Donc

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

### 4.3 Expériences numériques

On procède a quelques expériences numériques sur les méthodes du gradient conjugué avec les nouvelles recherches linéaires non monotones du type Wolfe forte . On teste la méthode de (FR), la méthode de (DY) , la méthode de (HS) et la méthode de (PRP) . Les méthodes sont testées d'après [41] , pour  $\varepsilon = 10^{-6}$ ,  $\delta = 0.01$ ,  $\sigma_1 = \sigma_2 = 0.1$ ,  $\alpha = 0$  ,  $M_0 = 150$ , pour  $\varepsilon = 10^{-6}$ ,

## Chapitre 4. Convergence de la méthode de FR avec une nouvelle classe de recherche linéaire non monotone

---

$\delta = 0.01, \sigma_1 = \sigma_2 = 0.1, \alpha = \frac{1}{2}, M_0 = 150$ . Les problèmes testés sont classés dans le tableau 4.1, et les résultats numériques détaillés de nos tests sont rapportés dans le tableau 4.2, les résultats numériques détaillés sont classés dans la forme NI/NF/NG, où NI, NF, NG dénote le nombre d'itération, évaluations de fonction et gradient respectivement.

Problèmes	Noms
1	fonction de Roth
2	fonction de Beale
3	fonction de Helicavalley
4	fonction de recherche de Gulf et developement
5	fonction singulière de Powell
6	fonction de Wood
7	fonction de Kowalikand Osborne
8	fonction de Brownand Dennis
9	fonction de Watson
10	fonction de Rosenbrock étendre
11	fonction Trigonometrique
12	fonction singulière de Powell étendre
13	fonction de Pénalité I
14	fonction de Pénalité II

Tableau 4.1. Liste des problèmes testés

## Chapitre 4. Convergence de la méthode de FR avec une nouvelle classe de recherche linéaire non monotone

---

Problèmes	dimension	FR	DY	HS	PRP
1	2	11/70/56 *	42/168/139	55/187/155	11/76/57
2	2	14/57/45	75/186/164	74/177/155	13/58/45 *
3	3	40/148/123	37/118/98 *	56/157/132	66/183/156
4	3	1/2/2 *	1/2/2 *	1/2/2 *	1/2/2 *
5	4	103/333/383	2286/4555/4547	425/1036/947	113/380/328
6	4	78/278/230 *	100/291/240	184/438/399	118/357/304
7	4	68/249/220	536/1449/1271	254/723/633	93/269/240
8	5	41/178/136	39/158/121	44/171/133	37/156/123 *
9	5	403/1210/1062	126/348/298	87/276/238 *	133/374/338
	20	2287/7972/7045	1945/5658/4924	4375/12695/11223	3293/10458/9246
10	100	33/195/152	31/157/121	62/223/182	29/168/128 *
	200	24/167/125 *	26/160/121	25/159/117	25/175/132
11	100	56/137/120 *	306/401/401	FAIL	59/120/114
	200	64/168/140	314/399/398	FAIL	64/135/127
12	500	219/490/465 *	4796/6823/6822	5089/7049/7058	1645/2889/2889
	1000	38/69/64 *	414/449/449	2406/3114/3115	147/251/252
13	500	6/13/7 *	7/16/8	7/16/8	7/16/8
	1000	7/14/8	7/15/8	7/15/8	6/13/7 *
14	1000	33/75/60 *	52/107/102	52/114/109	36/78/75
	5000	32/75/60 *	65/137/131	72/149/145	35/76/73

Tableau 4. 2. Les résultats numérique des méthodes

(\*) signifie que ce résultat est le meilleur parmi ces méthodes.

**Remarque 4.2 :** Dans le tableau 4.3, les CPU des méthodes sont données. Les tableaux 4.2 et 4.3 montrent que la méthode de (*FR*) a la meilleure performance par rapport aux nombres d'itérations et temps CPU. Tous les résultats numériques montrent l'efficacité de la méthode de (*FR*).

Problème	dimension	FR	DY	HS	PRP
1	2	0.0451*	0.1775	0.1964	0.0516
2	2	0.0721	0.3100	0.2717	0.0713*
3	3	0.2530	0.1264*	0.1995	0.5374
4	3	0.0008*	0.0041	0.0044	0.0036
5	4	0.3892	8.9351	2.5667	0.7034
6	4	0.3788*	0.4389	0.7155	0.5266
7	4	0.3570	2.1098	0.6122	0.4177
8	5	0.2364	0.3521	0.1614*	0.2172
9	5	2.2057	0.5362	0.3141*	0.6104
	20	7.5890	6.100*	16.9904	10.9497
10	100	0.2136*	0.2332	0.5122	0.2718
	200	0.3657*	0.4020	0.3961	0.4309
11	100	0.3899*	1.9000	<i>FAIL</i>	0.4573
	200	1.9860	6.5000	<i>FAIL</i>	1.8896*
12	500	8.7861*	56.4843	57.1386	28.8092
	1000	1.8963*	15.0000	40.6372	6.1500
13	500	1.9425	2.1589	2.1428	1.8554*
	1000	8.3452	9.7060	8.5487	7.3760*
14	1000	0.5001*	0.6897	0.7284	0.5066
	5000	1.3960*	3.1576	3.6011	1.7479

Tableau 4.3. Les temps CPU correspondant aux méthodes.

Afin de classer les méthodes numériques itératives on peut calculer le nombre totale d'évaluations de la fonction et du gradient par la formule suivante :

$$N_{total} = NF + NG \quad (4.44)$$

Similairement, on compare la méthode (*PRP*) , (*HS*) , (*DY*) et la méthode (*FR*) comme suit : pour chaque problème  $i$ , on calcule le nombre totale d'évaluation de la fonction et d'éva-

luation du gradient obtenu de ces méthodes et celle de (*FR*) à l aide de la formule (4.44) et on les note par  $N_{total,i}$  (*EM*) et  $N_{total,i}$  (*FR*); alors le taux est calculé comme suit :

$$r_i [EM (J)] = \frac{N_{total, i} [EM (J)]}{N_{total, i} (FR)} \quad (4.45)$$

Si la méthode [*EM* ( $J_0$ )] ne fonctionne pas par exemple pour  $i_0$ , mais la méthode de (*FR*) peut fonctionner, on remplace  $r_{i_0} [EM (J_0)]$  par une constante positive  $\tau_1$  qui est définie comme suit :

$$\tau_1 = \max \{r_i [EM (J_0)] : (i, j_0) \in S_1\} \quad (4.46)$$

soit

$$S_1 = \{(i, j_0) : \text{la méthode } j_0 \text{ ne fonctionne pas par exemple pour } i\} \quad (4.47)$$

Si la méthode de (*FR*) ne fonctionne pas par exemple pour  $i_0$ , mais la méthode de [*EM* ( $J_0$ )] peut fonctionner, on remplace  $r_{i_0} [EM (J_0)]$  par une constante positive  $\tau_2$  qui est définie comme suit :

$$\tau_2 = \min \{r_i [EM (J_0)] : (i, j_0) \in S_1\} \quad (4.48)$$

Soit la méthode de (*FR*) et la méthode [*EM* ( $J_0$ )] fonctionne, on définit  $r_{i_0} [EM (J_0)] = 1$ . Le sens géométrique de ces taux pour la méthode [*EM* ( $J$ )] sur tous les problèmes testés est défini par

$$r [EM (J)] = \{\prod_{i \in S} r_i [EM (J)]\}^{\frac{1}{|S|}} \quad (4.49)$$

où  $S$  désigne l'ensemble des problèmes testés et  $|S|$  le nombre des éléments dans  $S$ .

1. Les valeurs de  $r(FR)$ ,  $r(PRP)$ ,  $r(HS)$ , et  $r(DY)$  sont classés dans le tableau 4.4 suivant :

FR	PRP	HS	DY
1.212	0.923	0.896	0.775

D'après le tableau 4.4 on peut voir que la nouvelle méthode est plus efficace que la méthode de  $(HS)$  et la méthode de  $(PRP)$  .

# Conclusion

A travers ce travail nous avons voulu montrer comment les recherches linéaires non monotone pouvaient contribuer dans la convergence globale des algorithmes des méthodes du gradient conjugué. Nous avons explicité les recherches linéaires inexactes d'Armijo, Goldstein et Wolfe et nous les avons étendu aux recherches linéaires non monotone en particulier d'Armijo et Wolfe.

Nous avons aussi présenté une synthèse sur les importants résultats de convergence des différentes variantes de la méthode du gradient conjugué avec les recherches linéaires inexactes d'Armijo, Goldstein et Wolfe puis on s'est posé la question de pouvoir faire la même chose avec les recherches linéaires non monotone d'Armijo et Wolfe.

Nous avons fait des tests numériques et testé plusieurs méthodes du gradient conjugué pour les comparer avec la méthode (FR) et la nouvelle recherche linéaire non monotone de Wolfe. Les résultats numériques ont montré l'efficacité de cette dernière.

# 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] Bonnans, J. F., Panier, E., Titas, A., and Zhou, J. L., Avoiding the Maratos Effect by Means of a Nonmonotone Line Search, II : Inequality Constrained Problems Feasible Iterates, *SIAM Journal on Numerical Analysis*, Vol. 29, pp. 1187–1202, 1992.
- [8] C. W. Carroll (1961), The created response surface technique for optimizing nonlinear restrained systems, *Operations Research*, Vol. 9, pp. 169-84.



- [9] 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.
- [10] Y. Dai , A nonmonotone conjugate gradient algorithm for unconstrained optimization. J. Syst. Sci. Complex. 15 (2002), no. 2, 139–145. MR1900944 (2003b :90097).
- [11] Y. H. Dai, On the Nonmonotone Line Search, Journal of Optimization theory and Applications : Vol. 112, No. 2, pp. 315–330, February 2002.
- [12] 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.
- [13] Y. H. Dai and Y. Yuan (1996), Convergence properties of the Fletcher-Reeves method, *IMA J Numer. Anal.*, Vol.16(2), pp. 155-164.
- [14] Y. H. Dai and Y. Yuan (1996), Convergence properties of the conjugate descent method, *Advances in Mathematics*, 6, pp.552-562.Y.
- [15] H. Dai and Y. Yuan (2001), New properties of a new conjugate gradient method, *Numer. Math.*, Vol. 89(2), pp. 83-98.
- [16] 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.
- [17] Y. H. Dai and Y. Yuan (2002), A note on the nonlinear conjugate gradient methode Engineering Computing, *Chinese Academy of Sciences, P. O. Box 2719, Beijing 100080, China*).
- [18] 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*.
- [19] 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.
- [20] R. Fletcher (1987), Practical methods of optimization, *John Wiley&Sons, Chichster*.
- [21] R. Fletcher and M. J. D. Powell (1963), A rapidly convergent descent method for minimization, *Computer. J.*, 6, pp. 163-168.

- [22] R. Fletcher and C. Reeves (1964), Function minimization by conjugate gradients. *Comput. J.*, 7, pp.149-154.
- [23] A. V. Fiacco and G. P. McCormick (1968), Nonlinear Programming, *John Wiley, New York*.
- [24] J. C. Gilbert (2007), Eléments d'Optimisation Différentiable : Théorie et Algorithmes, *Notes de cours, École Nationale Supérieure de Techniques Avancées, Paris*.
- [25] 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*.
- [26] A.A. Goldstein (1967), Constructive Real Analysis, *A Harper International Edition*.
- [27] A. A. Goldstein (1965), On Steepest Descent, *SIAM J. on Control A, Vol. 3, No. 1, pp. 147-151*.
- [28] A.A. Goldstein and J.F. Price (1969), An Effective Algorithm For Minimization, *Num.Math.*, 10, pp. 184-189.
- [29] L. Grippo and S. Lucidi (1997), A Globally Convergent Version Of The Polak-Ribière Conjugate Gradient Method, *Mathematical Programming, Vol. 78, pp. 375-391*.
- [30] Grippo, L. Lampariello, F, and Lucidi, S., A Nonmonotone Line Search Technique for Newton's Method, *SIAM Journal on Numerical Analysis, Vol. 23, pp. 707-716, 1986*.
- [31] W.W. Hager e and H. Zhang (2005), A New Conjugate Gradient Method With Guaranteed Descent And an Efficient Line Search, *SIAM J. Optim.*, 16, pp. 170-192.
- [32] 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.
- [33] Y.F. Hu and C. Story (1991), Global Convergence Result For Conjugate Gradient Methods, *JOTA*, 71(2), pp. 399-405.
- [34] 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*.
- [35] 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.

- [36] G. H. Liu, L. L. Jing, L. X. Han, And D. Han, A Class of Nonmonotone Conjugate Gradient Methods for Unconstrained Optimization *Journal Of Optimization Theory And Applications* : Vol. 101, No. 1, pp. 127-140, APRIL 1999.
- [37] 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.
- [38] Lucidi, S., and Roma, M, Nonmonotone Conjugate Gradient Methods for Optimization, System Modeling and Optimization, Edited by J. Henry and J. D.
- [39] D. G. Luenberger (1969), Optimization by Vector Space Methods, *John Wiley and Sons, Inc. New York*.
- [40] M. Minoux (1983), Programmation Mathématique, Théorie et Algorithmes, tome1, *Dunod*.
- [41] J.J.Moré, B.S.Garbow and K.E. Hillstrom, Testing Unconstrained Optimization Software, *ACMTansactions on Mathematical Software* 7 (1981) 17–41.
- [42] A.S. Nemirovsky et D.B. Yudin (1983), Problem Complexity and Method Efficiency, *New York, Wiley*.
- [43] J. Nocedal (1991), Theory of Algorithm For Unconstrained Optimization, *Acta Numerica*, pp. 199-242.
- [44] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, NewYork, 1970.
- [45] E. Polak and G. Ribière (1969), Note Sur La Convergence De Directions Conjuguées, *Rev. Française Informat. Recherche Operationelle*, 3e année, 16, pp. 35-43.
- [46] B.T. Polyak (1969), The Conjugate Gradient Method In Extremem Problems, *Comput. Math. Math. Phys.*, 9, pp. 94-112.
- [47] M.J.D. Powell (1986), Convergence Properties Of Algorithms For Nonlinear Optimization, *SIAM rev.*, 28, pp. 487-500.
- [48] 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.
- [49] M.J.D. Powell (1971), On the convergence of the variable metric algorithm; *J. Inst. Math. Appl*, 7, pp. 21-36.

- [50] M.J.D. Powell (1977), Restart Procedures For The Conjugate Gradient Method, *Math Programming*, 2 , pp 241-254.
- [51] Powell, M. J. D, Some Properties Of The Variable Metric Method, Numerical Methods For Nonlinear Optimization, Edited by F. A. Lootsma, Academic Press, London, England, pp. 1-17, 1972.
- [52] T. Bouali and Y. Laskri, New Convergence Method With Nonmonotone Line Search, International Journal Of Pure And Applied Mathematics, Vol 92 No. 2, pp. 225-242,(2014). [dx.doi.org/10.12732/ijpam.v92i2.7](https://doi.org/10.12732/ijpam.v92i2.7).
- [53] T. Bouali , Y. Laskri, and R. Guefaifia, New Global Convergence Of Nonmonotone Line Search Algorithm, International Journal Of Pure And Applied Mathematics, Vol 93 No. 5, pp. 225-242,(2014).[dx.doi.org/10.12732/ijpam.v93i5.8](https://doi.org/10.12732/ijpam.v93i5.8).
- [54] Toint, P. L., A Nonmonotone Trust-Region Algorithm for Nonlinear Optimization Subject to Convex Constraints, *Mathematical Programming*, Vol. 77, pp. 69–94, 1997.
- [55] D. Touati-Ahmed and C. Storey (1990), Efficient Hybrid Conjugate Gradient Techniques, *JOTA*, 64, pp. 379-397.
- [56] P. Wolfe (1961), A Duality Theorem For Nonlinear Programming, *Quart. Appl. Math.*, 19, pp. 239-244.
- [57] P. Wolfe (1969), Convergence Conditions For Ascent Methods, *SIAM Review*, 11, pp. 226-235.
- [58] P.Wolfe (1971), Conditions For ascent Methods Some Corrections 2, *SIAM Review*, 13, pp.185-188.
- [59] P. Wolfe (1963), Methods Of Nonlinear Programming, in *Recent Advances in Mathematical Programming* (Eds R. L. Graves and P. Wolfe), McGraw-Hill, New York.
- [60] Y.X.Yuan (1993), Numerical Methods For Non linear Programming, *Shanghai scientific & Technical Publishers (in Chinese)*.
- [61] Zhang, Li. Zhou, Weijun, Li, Donghui, Global Convergence Of a Modified Fletcher-Reeves Conjugate Gradient Method With Armijo-type Line Search. *Numer. Math.* 104 (2006), no. 4, 561–572. Doi 10.1007/s00211-006-0028.

## Bibliographie

---

- [62] Zhou, J. L, and Titas, A, Nonmonotone Line Search for Minimax Problems, *Journal of Optimization Theory and Applications*, Vol. 76, pp. 455–476, 1993.
- [63] G. Zoutendijk (1970), *Nonlinear Programming Computational Methods*, *Integer and Non-linear Programming*, North Holland, Amsterdam.