

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 Mai 1945 Guelma



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

Département de Mathématiques

Mémoire de Master Académique en Mathématiques
Option : Equations aux Dérivées Partielles et Analyse Numérique

Présenté par :

ELADASSI Imen

Intitulé

**Sur la convergence de quelques méthodes
hybrides du gradient conjugué non linéaire**

Dirigé par : MELLAL Romaiassa (M.C.B, Univ. Guelma)

Soutenu le : 13/07/2021. Devant le Jury composé de :

Président Mme. SAKRANI Samia Prof Univ. Guelma

Encadreur Mme. MELLAL Romaiassa M.C .B Univ. Guelma

Examineur Mme. DJENAOUI Saliha M.C .B Univ. Guelma

Année Universitaire 2020/2021

Remerciements

J'exprime à travers ces quelques phrases modestes toute ma gratitude pour tous ceux qui ont aidés à aboutir ce travail.

A Dieu ALLAH

L'éternel tout puissant, le Clément, le Miséricordieux.

Je suis sur le point de finir avec ces lignes pour dire merci et je me rends compte qu'elles n'expriment pas tout à fait pleinement tous les mercis que je veux dire... En fin, voici quelques mots pour tous ceux qui ont contribué à la réussite de ce modeste travail.

Je tiens à exprimer ma profonde et sincère gratitude, à Mme "**MELLAL Romaisa**"

Docteur à l'Université, 8 mai 1945 Guelma.

Qui a accepté de m'encadrer avec disponibilité et bienveillance. Pour son enseignement, sa patience, et ses précieux conseils.

Mes vifs remerciements vont également aux membres de jury qui ont acceptés d'examiner ce travail.

Pour m'avoir fait l'honneur de juger mon mémoire et de rapporter mon travail.

Je remercie mon enseignant "**BOUSSETILA Nadjib**" Professeur à l'Université, 8 mai 1945 Guelma.

Pour son aide précieuse, son soutien, sa gentillesse et surtout pour son écoute attentive.

Je tiens aussi, à remercier toutes les personnes qui m'ont aidé à la réalisation de ce travail, en commençant par les enseignants de mon cursus universitaire, jusqu'à mes collègues.

Dédicace

Je dédie ce modeste travail :

À mes très chers parents, pour leur soutien et leur amour qui m'ont donné toute ma vie, ils ont toujours prié pour moi, et n'ont pas cessé de m'encourager et de me soutenir et qui ont fait de moi ce que je suis aujourd'hui.

À mes frères "**Chawki**" pour ces encouragements, aides et conseils précieux qui ont contribué à l'amélioration de ce travail et "**Ibrahim**" qui m'a donné toute la joie et l'amour.

À mes précieux amis "**Hamida**", "**Abderrezak**", "**Salah Eddine**", "**Maya**", "**Aymen**", "**Hala**" et "**Boutheina**" pour leur bonne compagnie, soutien, encouragement, ainsi que leur aide morale. Merci infiniment les amis.

À tous ceux qui me sont chers, et tous mes enseignants qui doivent voir dans ce travail la fierté d'un savoir bien acquis.

Eladassi Imen

RÉSUMÉ

Les méthodes du gradient conjugué non linéaire représentent une contribution majeure à la résolution des problèmes d'optimisation sans contrainte surtout de grande taille.

A travers ce mémoire, on se propose d'exposer une nouvelle classe de méthode du gradient conjugué non linéaire en déterminant le pas par la recherche linéaire inexacte de Wolfe forte, il s'agit de la classe des méthodes hybrides du gradient conjugué non linéaire.

L'idée derrière cette classe de méthodes hybrides est de combiner deux ou plusieurs variantes de la méthode du gradient conjugué non linéaire afin d'exploiter les caractéristiques intéressantes de chacune d'entre elles. Cette hybridation se fait par deux choix, soit en se basant sur le concept de projection, soit en basculant entre eux au cours de la méthode (des combinaisons convexes). Cette classe de méthodes possède des bonnes propriétés de convergence ainsi des résultats numériques meilleures que celles du gradient conjugué non linéaire classiques.

Nos simulations numériques (Scilab) illustrent l'efficacité des méthodes hybrides du gradient conjugué non linéaires pour la minimisation de quelques fonctions testées bien connues.

Mots clés : Optimisation sans contraintes, méthodes du gradient conjugué, hybridation, recherche linéaire inexacte de Wolfe, algorithme, convergence.

ABSTRACT

Nonlinear conjugate gradient methods represent a major contribution to the solution of unconstrained optimization problems especially of large size.

Through this thesis, we propose to expose a new class of nonlinear conjugate gradient methods by determining the step by the strong Wolfe inexact linear search, it is the class of hybrid nonlinear conjugate gradient methods.

The idea behind this class of hybrid methods is to combine two or more variants of the nonlinear conjugate gradient method in order to exploit the interesting features of each of them. This hybridization is done by two choices, either based on the projection concept, or by switching between them during the method (convex combinations). This class of methods have good convergence properties and numerical results better than the classical nonlinear conjugate gradient methods.

Our numerical simulations (Scilab) illustrate the efficiency of hybrid nonlinear conjugate gradient methods for the minimization of some well known test functions.

Key words : Unconstrained optimization, conjugate gradient methods, hybridization, Wolfe's inexact linear search, algorithm, convergence.

ملخص

تمثل طرق التدرج المترافق غير الخطي مساهمة كبيرة في حل مشاكل التحسين غير المقيدة خاصة ذات الحجم الكبير.

من خلال هذه الأطروحة ، نقترح كشف فئة جديدة من طرق التدرج المترافق غير الخطي من خلال تحديد الخطوة بواسطة البحث الخطي غير الدقيق القوي لـ *Wolfe*، وهي فئة طرق التدرج المترافق غير الخطي الهجين.

الفكرة وراء هذه الفئة من الطرق الهجينة هي الجمع بين متغيرين أو أكثر من طريقة التدرج اللاخطي المترافق من أجل استغلال الميزات المثيرة لكل منها. يتم هذا التهجين من خلال خيارين ، إما بناءً على مفهوم الإسقاط ، أو عن طريق التبديل بينهما أثناء الطريقة (مجموعات محدبة). هذه الفئة من الطرق لها خصائص تقارب جيدة ونتائج عددية أفضل من طرق التدرج اللاخطي المترافق التقليدي.

توضح عمليات المحاكاة العددية (*Scilab*) كفاءة طرق التدرج المترافق غير الخطي الهجين لتقليل بعض وظائف الاختبار المعروفة.

الكلمات المفتاحية : التحسين غير المقيد ، طرق التدرج المترافق ، التهجين ، البحث الخطي غير الدقيق لـ *Wolfe* ، الخوارزمية ، التقارب.

Table des matières

Introduction	1
1 Notions préliminaires	4
1.1 Définitions et notions de base	4
1.2 Optimisation sans contraintes	9
1.2.1 Problème d'optimisation sans contraintes	9
1.3 Direction de descente	10
1.3.1 Exemples de choix de directions de descente	12
1.4 Conditions d'optimalité des problèmes d'optimisation sans contraintes . . .	13
1.4.1 Conditions nécessaires d'optimalité	13
1.4.2 Conditions suffisantes d'optimalité	14
1.4.3 Cas convexe	15
2 Recherches linéaires inexactes	16
2.1 Recherche linéaire	16
2.1.1 Principe des recherches linéaires	16
2.1.2 Objectifs à atteindre	18
2.2 Recherche linéaire inexacte d'Armijo	19
2.2.1 Principe de la règle d'Armijo	19
2.2.2 Algorithme de la Règle d'Armijo	20
2.2.3 Existence du pas d'Armijo	20
2.3 Recherche linéaire inexacte de Wolfe	21
2.3.1 Conditions de Wolfe faibles	21
2.3.2 Conditions de Wolfe fortes	22

2.3.3	Algorithme de la Règle de Wolfe	23
2.3.4	Existence du pas de Wolfe	23
2.4	Contribution des recherches linéaires inexactes dans la convergence de la classe des méthodes à directions de descente	25
2.4.1	Condition de Zoutendijk	25
3	Méthode du gradient conjugué	28
3.1	Méthode du gradient	28
3.1.1	Algorithme de la méthode du gradient	29
3.1.2	Convergence de la méthode du gradient	29
3.2	La méthode du gradient conjugué linéaire	30
3.2.1	Principe général	30
3.2.2	Détermination du pas α_k	31
3.2.3	Détermination du coefficient β_k	32
3.2.4	Algorithme du gradient conjugué linéaire	33
3.2.5	Convergence de méthode du gradient conjugué linéaire	33
3.2.6	Vitesse de convergence	37
3.2.7	Les avantages de la méthode du gradient conjugué linéaire	37
3.3	Exemple	37
3.4	La méthode du gradient conjugué non-linéaire	40
3.4.1	Principe général	40
3.4.2	Quelques variantes de la méthode du gradient conjugué non linéaire	41
3.4.3	Algorithme du gradient conjugué non linéaire avec la recherche linéaire inexacte de Wolfe forte	43
3.4.4	Convergence des méthodes du gradient conjugué non linéaires . . .	43
4	Méthodes hybrides du gradient conjugué non linéaire	46
4.1	Principe général	46
4.2	Méthodes hybrides du gradient conjugué non linéaire de projection	48
4.2.1	Méthode hybride de Wei, Yao et Liu	49
4.2.2	Méthode hybride de Yao, Wei et Huang	49

4.2.3	Méthode hybride de Jian , Han et Jiang	49
4.2.4	Algorithme des méthodes hybrides du gradient conjugué non linéaire de projection	50
4.2.5	Descente de la direction	51
4.2.6	La convergence globale	51
4.3	Méthodes hybrides du gradient conjugué non linéaire basées sur des combinaisons convexes	53
4.3.1	Méthode hybride du gradient conjugué non linéaire LS-DY	54
4.3.2	Algorithme hybrides du gradient conjugué non linéaire sur des combinaisons convexes	54
4.3.3	Descente de la direction	55
4.3.4	La convergence globale	57
5	Simulation numérique	59
5.1	Performances des méthodes hybrides du gradient conjugué non linéaire basées sur le concept de projection	60
5.1.1	La fonction TRIDIA (cute)	60
5.1.2	La fonction DBVF (Discrete boundary value)	61
5.2	Performances des méthodes hybrides du gradient conjugué non linéaire basées sur sur des combinaisons convexes	63

Table des figures

1.1	Extrema d'une fonction à une variable	10
1.2	La direction de descente	12
2.1	Règle d'Armijo	19
2.2	Règle de Wolfe	22
5.1	Application des variantes de WYL, YWH et JHJ pour la fonction TRIDIA.	60
5.2	Comparaison des méthodes de FR, WYL, PRP, DY YWH et JHJ pour la fonction TRIDIA.	61
5.3	Comparaison des méthodes de FR, WYL, PRP, DY YWH et JHJ pour la fonction DBVF	62
5.4	Comparaison de la méthode LS-DY pour la fonction TRIDIA.	63

Introduction

L'optimisation est une discipline des mathématiques qui s'intéresse à la modélisation, à l'analyse et à la résolution de problèmes consistant à minimiser ou à maximiser une fonction avec ou sans contraintes, de manière analytique ou numérique.

Ces fonctions peuvent être linéaires ou non linéaires.

Les racines de l'optimisation remontent au 18^{ième} siècle, grâce aux travaux réalisés par : TAYLOR, NEWTON et LAGRANGE qui ont créé les bases des développements limités. CAUCHY[7] a également été le premier à avoir appliqué une méthode d'optimisation, la méthode du gradient, pour la résolution des problèmes sans contraintes. Peu de progrès ont été réalisés grâce à ces contributions importantes.

À la fin de la Seconde Guerre mondiale, une série d'avancées spectaculaires dans les techniques d'optimisation sont apparues. Il est intéressant de noter que ces progrès ont été réalisés principalement en Grande-Bretagne.

La méthode du gradient conjugué [Voir [3, 28]] est l'une des plus anciennes méthodes de la résolution de ce type de problème. Cette méthode est souvent utilisée pour résoudre des problèmes de grande taille. Elle a été proposée par HESTENES et STEIFEL [14] en 1952, afin de minimiser des fonctions quadratiques strictement convexes. Plusieurs mathématiciens ont étendu la méthode du gradient conjuguée pour le cas non linéaires. Cela a été réalisé en 1964 pour la première fois par FLETCHER et REEVES [11] (Méthode de Fletcher-Reeves) ensuite par POLAK [22], RIBIÈRE et POLYAK [23] (Méthode de Polak-Ribière-Polyak) en 1969. Une autre variante a été étudiée en 1987 par Fletcher (Méthode de la descente conjuguée), en 1999 par DAI et YUAN [9] (méthode de Dai-Yuan) et beaucoup d'autres méthodes.

Récemment, plusieurs mathématiciens ont développé des nouvelles méthodes du gradient conjugué non linéaire efficaces afin de résoudre des problèmes d'optimisation sans contraintes.

La question naturelle qui se pose maintenant est la suivante :
Peut-on combiner deux ou plusieurs variantes du gradient conjugué non linéaire afin d'obtenir des meilleurs résultats que les autres méthodes?

A travers ce mémoire, on répond à cette question et on rend compte des principaux résultats obtenus en s'appuyant sur cinq chapitres, comme suit :

PREMIER CHAPITRE :

Dans le premier chapitre 1, on présente les notions préliminaires de l'optimisation sans contraintes. On commence par introduire quelques outils fonctionnels de base nécessaires pour l'optimisation. Ensuite on rappelle quelques notions sur les problèmes de minimisation sans contrainte.

DEUXIEME CHAPITRE :

Le deuxième chapitre 2 est dédié à l'étude des recherches linéaires inexactes, on insiste ici particulièrement sur les recherches linéaires inexactes dites d'Armijo et Wolfe, de plus l'étude de la contribution de cette dernière à la convergence des algorithmes à directions de descente.

TROISIEME CHAPITRE :

Le troisième chapitre 3 aborde de manière générale les méthodes du gradient conjugué linéaire et non linéaire.

QUATRIEME CHAPITRE :

Le quatrième chapitre 4 qui est le cœur de ce mémoire, on expose ici la descente des deux classes de la méthode hybride du gradient conjuguée non linéaire et on démontre leurs convergence.

CINQUIEME CHAPITRE :

Le cinquième chapitre 5 met en évidence les résultats obtenus numériquement de notre

méthodes hybrides à l'aide du logiciel (SCILAB) concernant l'étude de convergence et de performances avec une discussion et des interprétations adéquates.

Ce mémoire se terminera par une conclusion générale qui récapitule le travail effectué.

Notions préliminaires

*La patience est un arbre dont la racine est amère,
et dont les fruits sont très doux.*

L'objectif de ce chapitre est de rappeler brièvement les définitions de quelques notions importantes sur lesquelles on va s'appuyer dans notre travail, ainsi que quelques propriétés utiles.

1.1 Définitions et notions de base

Définition 1.1.1. [19]

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

1. Le **GRADIENT** d'une fonction de plusieurs variables en un certain point $x_0 \in \mathbb{R}^n$ est un vecteur qui caractérise la variabilité de cette fonction au voisinage de ce point.

Le gradient est la généralisation à plusieurs variables de la dérivée d'une fonction d'une seule variable.

On le note par ∇f tel que :

$$\nabla f(x_0) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x_0) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x_0) \end{bmatrix} \quad (1.1.1)$$

▷ Il est courant aussi de noter le gradient par : $\overrightarrow{Grad}f$.

2. On appelle **LA MATRICE HESSIENNE** (ou simplement la hessienne) de la fonction f en un certain point $x_0 \in \mathbb{R}^n$ la matrice carrée, notée $H(f)$ de ses dérivées partielles secondes.

Plus précisément en supposant que toutes les dérivées partielles secondes de f existent, le coefficient d'indice i, j de la matrice hessienne de f vaut :

$$H_{ij}(f(x_0)) = \frac{\partial^2 f}{\partial x_i \partial x_j}(x_0) \quad \text{avec } i, j = (1, 2, \dots, n). \quad (1.1.2)$$

Ou, en d'autres termes :

$$H(f(x_0)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(x_0) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x_0) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x_0) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x_0) & \frac{\partial^2 f}{\partial x_2^2}(x_0) & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x_0) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x_0) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x_0) & \dots & \frac{\partial^2 f}{\partial x_n^2}(x_0) \end{bmatrix}. \quad (1.1.3)$$

Définition 1.1.2. [21]

Soit A une matrice carrée, $A \in \mathcal{M}_{n \times n}$

— la matrice A est dite **SYMÉTRIQUE** si :

$$A^T = A \quad (1.1.4)$$

- la matrice A est dite **DÉFINIT POSITIVE** si et seulement si elle est positive et inversible, autrement dit :

$$\forall x \in \mathbb{R}^n, x^T A x > 0 \quad (1.1.5)$$

Définition 1.1.3. [4] Soit $A \in \mathcal{M}_{n \times n}$ une matrice symétrique définie positive, soit x et y deux vecteurs de \mathbb{R}^n .

On dit que deux vecteurs non nuls x et y sont **CONJUGUÉS** par rapport à la matrice A ou (A -conjugués), s'ils vérifient :

$$x^T A y = \langle x, y \rangle_A = 0 \quad (1.1.6)$$

On dit que ces deux vecteurs sont orthogonaux pour ce produit scalaire.

▷ Une famille de vecteurs sont alors conjugués par rapport à A s'ils sont orthogonaux au sens de ce produit scalaire. En particulier, ils forment donc une famille libre.

Définition 1.1.4. [21]

Soit $A \in \mathcal{M}_{n \times n}$ une matrice symétrique, b un vecteur de \mathbb{R}^n et c une constante.

- On dit que la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une **FORME QUADRATIQUE** si on peut l'écrire sous la forme suivante :

$$f(x) = \frac{1}{2} x^T A x - b^T x + c \quad (1.1.7)$$

Remarque 1.1.1. La fonction $f(x)$ est dite une forme quadratique définie positive (resp. semi-définie positive) lorsque A est une matrice définie positive (resp. semi-définie positive).

Définition 1.1.5. [6] Soit Ω une partie de \mathbb{R}^n , $f : \Omega \rightarrow \mathbb{R}$ une application, On dit que f est **LOCALEMENT LIPSCHITZIENNE** si : $\forall M > 0, \forall (x, y) \in \Omega^2$

$$\|f(x) - f(y)\| \leq M \|x - y\| \quad (1.1.8)$$

Définition 1.1.6. [6] Une partie Ω de \mathbb{R}^n est dite **CONVEXE** si,

$$\forall t \in [0, 1], \forall (x, y) \in \Omega^2, (1 - t)x + ty \in \Omega \quad (1.1.9)$$

▷ Cela signifie que, $\forall (x, y) \in \Omega^2$, le segment $[x, y]$ est tout entier inclus dans Ω

Définition 1.1.7. [21] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une application, on dit que :

1. l'application f est **CONVEXE** sur \mathbb{R}^n si :

$$\forall t \in [0, 1], \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n : f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y) \quad (1.1.10)$$

2. l'application f est **STRICTEMENT CONVEXE** sur \mathbb{R}^n si :

$$\forall t \in]0, 1[, \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \text{ avec } x \neq y : f(tx + (1 - t)y) < tf(x) + (1 - t)f(y)$$

Définition 1.1.8. [19]

Soit Ω un ouvert de \mathbb{R}^n et soit f une application à valeurs dans \mathbb{R} et soit x un point de Ω , f est dite **DIFFÉRENTIABLE** au point x s'il existe une application linéaire continue L de \mathbb{R}^n dans \mathbb{R} telle que :

$$\forall h \in \mathbb{R}^n; f(x + h) = f(x) + L(h) + o(\|h\|), \text{ avec } x + h \in \Omega \quad (1.1.11)$$

Ou, d'une façon équivalente :

$$\lim_{h \rightarrow 0} \frac{f(x + h) - f(x) - L(h)}{\|h\|} = 0 \quad (1.1.12)$$

▷ Une telle application linéaire L si elle existe, est alors unique. Elle s'appelle **différentielle** de f au point x et notée par : $L = Df(x)$. De plus, sa continuité assure la continuité de f en x .

Définition 1.1.9. [25]

Soit f une fonction définie sur \mathbb{R}^n à valeurs dans \mathbb{R} et $d \in \mathbb{R}^n$.

1. On dit que f admet **UNE DÉRIVÉE DIRECTIONNELLE** au point x suivant le vecteur d , si l'application $t \in \mathbb{R} \mapsto f(x + td)$ est dérivable en 0 et se note par :

$$\frac{\partial f}{\partial d}(x) = \lim_{t \rightarrow 0} \frac{f(x + td) - f(x)}{t}$$

▷ Si d est le vecteur nul, cette dérivée directionnelle existe et a une valeur nulle, alors on va supposer que d n'est pas le vecteur nul dans ce qui suit.

2. Dans ce cas, si $d = e_k$ (l'un des vecteurs de base de \mathbb{R}^n), on appelle cette dérivée directionnelle la $k^{\text{ème}}$ **DÉRIVÉE PARTIELLE** de f au point x , que l'on note :

$$\frac{\partial f}{\partial e_k}(x) = \lim_{t \rightarrow 0} \frac{f(x_1, x_2, \dots, x_k + t, \dots, x_n) - f(x_1, x_2, \dots, x_n)}{t} \quad (1.1.13)$$

Définition 1.1.10. [4] Soit $\{x_k\}_{k \in \mathbb{N}}$ une suite dans \mathbb{R}^n convergeant vers \tilde{x} telle que $x_k \neq \tilde{x}$

◇ Si

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - \tilde{x}\|}{\|x_k - \tilde{x}\|} = \tau < 1,$$

on dit que $\{x_k\}_{k \in \mathbb{N}}$ converge vers \tilde{x} linéairement avec le taux τ .

◇ Si

$$\frac{\|x_{k+1} - \tilde{x}\|}{\|x_k - \tilde{x}\|} \longrightarrow 0 \text{ quand } k \longrightarrow \infty,$$

alors on dit que la convergence est superlinéaire.

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

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

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

En particulier si

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - \tilde{x}\|}{\|x_k - \tilde{x}\|^2} < +\infty,$$

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

1.2 Optimisation sans contraintes

D'un point de vue mathématique, Le but de l'optimisation est de trouver le minimum (ou maximum) d'une certaine quantité donnée, que l'on appelle le coût ou l'objectif.

Dans ce travail nous supposons que l'objectif est déterminé par n variables réelles, qui sont rassemblées en un vecteur $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, et donneront une valeur $f(x)$ où f est une fonction de \mathbb{R}^n à valeurs dans \mathbb{R} . Les variables du vecteur $x = (x_1, x_2, \dots, x_n)$ peuvent prendre n'importe quelle valeur dans \mathbb{R}^n , et ça ce que l'on appelle une optimisation sans contrainte.

1.2.1 Problème d'optimisation sans contraintes

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ on appelle problème d'optimisation (minimisation) sans contraintes le problème (P) suivant :

$$(P) : \min f(x) : x \in \mathbb{R}^n \quad (1.2.1)$$

▷ On considère le problème (1.2.1) :

Définition 1.2.1. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, une fonction continûment différentiable :

1. Minimum global :

$\tilde{x} \in \mathbb{R}^n$ est un **MINIMUM GLOBAL** de (1.2.1) si et seulement si :

$$f(\tilde{x}) \leq f(x) : \forall x \in \mathbb{R}^n \quad (1.2.2)$$

2. Minimum local :

$\tilde{x} \in \mathbb{R}^n$ est un **MINIMUM LOCAL** de (1.2.1) si et seulement si il existe un voisinage $\mathcal{V}_\varepsilon(\tilde{x})$ tel que :

$$f(\tilde{x}) \leq f(x) : \forall x \in \mathcal{V}_\varepsilon(\tilde{x}) \quad (1.2.3)$$

3. Minimum local strict :

$\tilde{x} \in \mathbb{R}^n$ est un **MINIMUM LOCAL STRICT** de (1.2.1) si et seulement si il existe un voisinage $\mathcal{V}_\varepsilon(\tilde{x})$ tel que :

$$f(\tilde{x}) < f(x) : \forall x \in \mathcal{V}_\varepsilon(\tilde{x}) \text{ avec } x \neq \tilde{x} \quad (1.2.4)$$

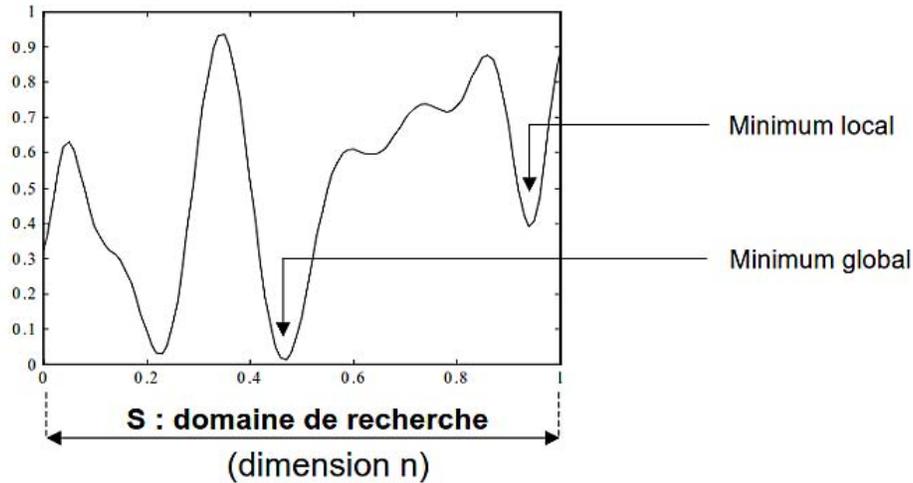


FIGURE 1.1 – Extrema d'une fonction à une variable

1.3 Direction de descente

Définition 1.3.1. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, d un vecteur non nul de \mathbb{R}^n est dite **DIRECTION DE DESCENTE** de f en $\tilde{x} \in \mathbb{R}^n$ si et seulement si il existe un réel $\delta > 0$; tel que pour tout α dans l'intervalle $]0, \delta[$ on a :

$$f(\tilde{x} + \alpha d) < f(\tilde{x}) \quad (1.3.1)$$

Théorème 1.3.1. [25] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable au point $\tilde{x} \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$ une direction vérifiant la condition suivante :

$$\nabla^T f(\tilde{x}) \cdot d < 0 \quad (1.3.2)$$

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

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

$$f(\tilde{x} + \alpha d) = f(\tilde{x}) + \alpha \nabla f(\tilde{x})^T \cdot d + \alpha \|d\| o(\tilde{x}, \alpha d), \quad \text{avec } o(\tilde{x}, \alpha d) \xrightarrow{\alpha \rightarrow 0} 0 \quad (1.3.3)$$

ceci implique que

$$\lim_{\alpha \rightarrow 0} \frac{f(\tilde{x} + \alpha d) - f(\tilde{x})}{\alpha} = \nabla^T f(\tilde{x}) \cdot d < 0 \quad (1.3.4)$$

La limite étant strictement négative, alors il existe un voisinage de zéro $\mathcal{V}(0) =]-\delta, +\delta[$ tel que :

$$\frac{f(\tilde{x} + \alpha d) - f(\tilde{x})}{\alpha} < 0, \quad \forall \alpha \in]-\delta, +\delta[\quad (1.3.5)$$

La relation (1.3.5) est particulièrement vraie pour tout $\alpha \in]0, +\delta[$. On obtient le résultat cherché en multipliant la relation (1.3.5) par $\alpha > 0$.

C'est à dire :

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

□

L'ensemble des directions de descente de f en x est donné par :

$$\{d \in \mathbb{R}^n : \nabla f(x)^T \cdot d < 0\} \quad (1.3.6)$$

Remarque 1.3.1. La direction de descente d fait un angle θ avec l'opposé du gradient $-\nabla f(x)$ qui est strictement inférieur à 90° et on a :

$$\theta = \arccos \frac{-\nabla^T f(x) \cdot d}{\|\nabla^T f(x)\| \|d\|} \in [0, \frac{\pi}{2}[\quad (1.3.7)$$

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

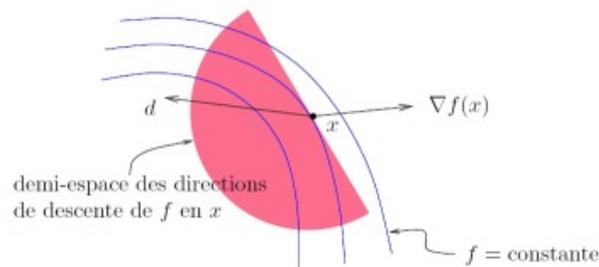


FIGURE 1.2 – La direction de descente

1.3.1 Exemples de choix de directions de descente

Direction du gradient

Si on choisit l'opposé du gradient comme directions de descente ; c'est à dire si on prend :

$$d = -\nabla f(x) \quad \text{avec } \nabla f(x) \neq 0 \quad (1.3.8)$$

On obtient alors la direction du **GRADIENT**

Bien sur $-\nabla f(x)$ est une direction de descente , puisque

$$\nabla f(x)^T . d = -\nabla f(x)^T \nabla f(x) = \langle \nabla f(x), \nabla f(x) \rangle = -\|\nabla f(x)\|^2 < 0 \quad (1.3.9)$$

Direction de Newton

On va supposer que la fonction f est deux fois différentiable en x

La direction de **NEWTON** en un point x en lequel le hessien de f est inversible est définie par :

$$d = -(\nabla^2 f(x))^{-1} \nabla f(x) \quad (1.3.10)$$

Il s'agit bien d'une direction de descente si :

- ▷ $\nabla f(x) \neq 0$.
- ▷ $\nabla^2 f(x)$ est définie positive.

En Effet

$$\nabla f(x).d = \langle \nabla f(x), d \rangle = -\langle \nabla f(x), \nabla^2 f(x)^{-1} \nabla f(x) \rangle \leq -\lambda_{max}^{-1} \|\nabla f(x)\|^2 < 0 \quad (1.3.11)$$

Où λ_{max} désigne la plus grande valeur propre de $\nabla^2 f(x)$.

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

En mathématique, les conditions d'optimalité sont un ensemble d'équations, d'inéquations ou de propriétés vérifiées par une solution de problème (1.2.1) (on parle alors de conditions nécessaires d'optimalité) ou qui permettent d'assurer qu'un point qui les vérifie est une solution du problème (1.2.1) (on parle alors des conditions suffisantes d'optimalité).

1.4.1 Conditions nécessaires d'optimalité

Condition nécessaire d'optimalité du premier ordre :

Théorème 1.4.1. [19] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable au point $\tilde{x} \in \mathbb{R}^n$. Si \tilde{x} est un minimum local de (1.2.1) alors $\nabla f(\tilde{x}) = 0$.

Preuve. En effet, supposons que $\nabla f(\tilde{x}) \neq 0$.

Puisque la direction $d = -\nabla f(\tilde{x})$, est une direction de descente, il existe $\delta > 0$ tel que :

$$f(\tilde{x} + \alpha d) < f(\tilde{x}), \quad \forall \alpha \in]0, \delta[\quad (1.4.1)$$

Ceci est contradiction avec le fait que \tilde{x} est une solution optimal local de (1.2.1) d'où :

$$\nabla f(\tilde{x}) = 0 \quad (1.4.2)$$

□

Condition nécessaire d'optimalité du second ordre :

Théorème 1.4.2. [21] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable au point $\tilde{x} \in \mathbb{R}^n$. Si \tilde{x} est un minimum local de (1.2.1) alors $\nabla f(\tilde{x}) = 0$ et la matrice hessienne de f au point \tilde{x} , qu'on note $H(\tilde{x})$, est semi définie positive.

Preuve. Soit $x \in \mathbb{R}^n$ quelconque, f étant deux fois différentiable au point \tilde{x} , on aura pour tout $\alpha \neq 0$

$$f(\tilde{x} + \alpha d) = f(\tilde{x}) + \frac{1}{2}\alpha^2 d^t H(\tilde{x})d + \alpha^2 \|d\|^2 o(\tilde{x}, \alpha d), \quad \text{avec } o(\tilde{x}, \alpha d) \xrightarrow{\alpha \rightarrow 0} 0 \quad (1.4.3)$$

Ceci implique :

$$\frac{f(\tilde{x} + \alpha d) - f(\tilde{x})}{\alpha^2} = \frac{1}{2}d^t H(\tilde{x})d + \|d\|^2 o(\tilde{x}; \alpha d) \quad (1.4.4)$$

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

$$d^t H(\tilde{x})d + \|d\|^2 o(\tilde{x}; \alpha d) \geq 0 \text{ pour } \alpha \text{ petit.} \quad (1.4.5)$$

On passe à la limite quand $\alpha \rightarrow 0$, on obtient $d^t H(\tilde{x})d \geq 0$, d'où $H(\tilde{x})$ est semi définie positive.

□

1.4.2 Conditions suffisantes d'optimalité

Théorème 1.4.3. [19] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable au point $\tilde{x} \in \mathbb{R}^n$. Si $\nabla f(\tilde{x}) = 0$ et $H(\tilde{x})$ est définie positive, alors \tilde{x} est un minimum local strict de (1.2.1).

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

$$f(x) = f(\tilde{x}) + \frac{1}{2}(x - \tilde{x})^t H(\tilde{x})(x - \tilde{x}) + \|x - \tilde{x}\|^2 o(\tilde{x}; x - \tilde{x}), \quad \text{avec } o(\tilde{x}; x - \tilde{x}) \xrightarrow{x \rightarrow \tilde{x}} 0 \quad (1.4.6)$$

Supposons que \tilde{x} n'est pas un minimum local strict. Donc il existe une suite $\{x_k\}$ conver-

gente vers \tilde{x} telle que

$$f(x_k) \leq f(\tilde{x}) \text{ avec } x_k \neq \tilde{x}, \quad \forall k. \quad (1.4.7)$$

Posons $d_k = \frac{x_k - \tilde{x}}{\|x_k - \tilde{x}\|}$. Donc $\|d_k\| = 1$ et on obtient :

$$\frac{f(x_k) - f(\tilde{x})}{\|x_k - \tilde{x}\|^2} = \frac{1}{2} d_k^t H(\tilde{x}) d_k + o(\tilde{x}; x_k - \tilde{x}) \leq 0, \forall k \quad (1.4.8)$$

et comme $\|d_k\| = 1, \forall k$ alors il existe $\{d_k\}_{k \in \mathbb{N}_1 \subset \mathbb{N}}$ telle que $d_k \rightarrow d$ pour $k \rightarrow \infty$ et $k \in \mathbb{N}_1$.

On a bien sûr $\|d\| = 1$.

Considérons donc $\{d_k\}_{k \in \mathbb{N}_1}$ et le fait que $o(x; x - x) \rightarrow 0$ pour $k \rightarrow \infty$ et $k \in \mathbb{N}_1$. Alors :

$$d^t H(\tilde{x}) d \leq 0 \quad (1.4.9)$$

Ce qui contredit le fait que $H(\tilde{x})$ est définie positive car $\|d\| = 1$ (donc $d \neq 0$). Donc \tilde{x} est un minimum local strict. \square

1.4.3 Cas convexe

Théorème 1.4.4. [25] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ telle que f est convexe et différentiable. Alors \tilde{x} est un minimum globale de f si et seulement si

$$\nabla f(\tilde{x}) = 0 \quad (1.4.10)$$

Remarque 1.4.1. Dans le cas où la fonction f est convexe, il n'y a pas de différence entre un minimum local et un minimum global, c'est-à-dire que tout minimum local est également global. De plus, si f est strictement convexe, tous les minima locaux deviennent non seulement globaux, mais également uniques.

Recherches linéaires inexactes

Le but de ce chapitre consiste à décrire les principales règles de recherche linéaire inexacte (Armijo, Wolfe), ainsi que l'étude de la contribution de cette dernière à la convergence des algorithmes à directions de descente.

2.1 Recherche linéaire

2.1.1 Principe des recherches linéaires

Si la recherche linéaire nous intéresse, ce n'est pas seulement parce que dans les applications, on rencontre naturellement des problèmes unidimensionnels, mais plutôt parce que la recherche linéaire est une pièce fondamentale de toutes les méthodes traditionnelles d'optimisation multidimensionnelle.

Effectuer une recherche linéaire consiste à déterminer un pas α_k le long d'une direction de descente d_k , en d'autres termes résoudre le problème unidimensionnel suivant :

$$\min f(x_k + \alpha d_k), \quad \alpha \in \mathbb{R}_+ \quad (2.1.1)$$

Habituellement, nous avons le schéma suivant d'une méthode de minimisation multi-

dimensionnelle sans contrainte :

En observant le comportement local de la fonction-coût f sur l'itération courante x_k , la méthode sélectionne "la direction du mouvement" d_k (qui est souvent une direction de descente de l'objectif $f : \nabla^T f(x) \cdot d < 0$) et exécute un pas dans cette direction de la forme suivante :

$$x_k \rightarrow x_{k+1} = x_k + \alpha_k \cdot d_k \quad (2.1.2)$$

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

$$\min_{\alpha > 0} f(x_k + \alpha d_k) \quad (2.1.3)$$

en vue de réaliser une certaine progression de la valeur de l'objectif f , c'est-à-dire de faire en sorte que

$$f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k), \quad \forall \alpha > 0 \quad (2.1.4)$$

La plupart des méthodes considèrent que x_k, d_k sont des valeurs fixes et que le pas α_k est déterminé par la minimisation unidimensionnelle de la fonction $h_k : \mathbb{R}_+ \rightarrow \mathbb{R}$ tel que :

$$\alpha \mapsto h_k(\alpha) = f(x_k + \alpha d_k) \quad (2.1.5)$$

Il convient de noter que dans les problèmes d'optimisation sans contraintes, nous devons résoudre un problème d'optimisation dans \mathbb{R} pour chaque itération x_k .

La méthode de recherche linéaire est ainsi une brique de base de toute méthode multidimensionnelle.

Il existe deux grandes catégories principales de méthodes qui concernent l'optimisation unidimensionnelle :

- ◆ Les recherches linéaires exactes.
- ◆ Les recherches linéaires inexactes.

2.1.2 Objectifs à atteindre

Il s'agit d'atteindre deux objectifs [18] :

Le premier objectif

Consiste à diminuer f suffisamment. Cela se fait le plus souvent en réalisant une égalité sous la forme :

$$f(x_k + \alpha_k d_k) = f(x_k) + \text{"un terme négatif"} \quad (2.1.6)$$

Le terme négatif, dit k , joue un rôle essentiel pour la convergence de l'algorithme qui utilise cette recherche linéaire.

L'argument va comme suit.

Dans le cas où $f(x_k)$ est minorée il existe une constante C telle que $f(x_k) \geq C, \forall k$ alors ce terme négatif doit nécessairement tendre vers zéro. C'est bien souvent à partir de la convergence vers zéro de cette suite que nous arrivons à montrer que le gradient lui-même doit tendre vers zéro. Le terme négatif doit donc prendre une forme tout à fait particulière si l'on veut parvenir à en tirer des informations.

Notamment, il ne suffit pas d'imposer $f(x_k + \alpha_k d_k) < f(x_k)$.

Le second objectif

Cela consiste à éviter que le pas $\alpha_k > 0$ soit trop petit, trop proche de zéro. En fait, le premier objectif est insuffisant puisque l'inégalité (2.1.6) est généralement satisfaite par des pas $\alpha_k > 0$ arbitrairement petit.

Mais cela peut conduire à une "fausse convergence", c'est-à-dire à la convergence des itérés vers un point non stationnaire.

2.2 Recherche linéaire inexacte d'Armijo

2.2.1 Principe de la règle d'Armijo

On vise à réduire " d'une manière importante " la valeur de l'objectif par un pas $x_k \rightarrow x_{k+1} = x_k + \alpha_k d_k$ de x_k dans la direction d_k , tel que $f(x_k + \alpha_k d_k) < f(x_k)$.

La règle d'Armijo [5] impose une contrainte sur le choix de α_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 :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k \nabla^T f(x_k) d_k \quad (2.2.1)$$

Elle est de la forme (2.2.1) car ρ devra être choisi dans $]0, 1[$.

Il faut qu'en α_k , la fonction h_k prenne une valeur plus petite que celle prise par la fonction $\psi_\rho(\alpha) : \alpha \rightarrow h_k(0) + \rho h'_k(0)\alpha$ autrement dit

$$\alpha \rightarrow f(x_k) + \rho \alpha_k \nabla^T f(x_k) d_k$$

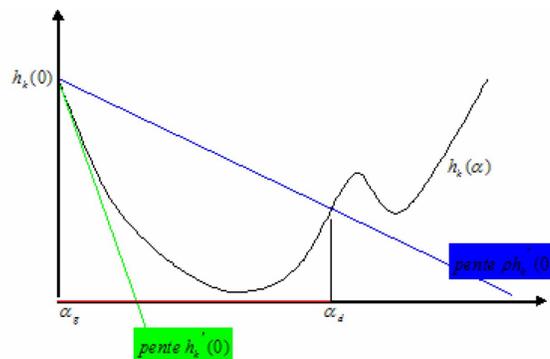


FIGURE 2.1 – Règle d'Armijo

On voit bien à la figure (2.1) ce que signifie cette condition.

2.2.2 Algorithme de la Règle d'Armijo

Algorithm 1: Règle d'Armijo

Étape 0 : (initialisation)

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

Étape 1 :

- si $h_k(\alpha_k) = h_k(0) + \rho h'_k(0)\alpha_k$: STOP ($\tilde{\alpha} = \alpha_k$).
- si $h_k(\alpha_k) > h_k(0) + \rho h'_k(0)\alpha_k$, alors $\alpha_{d,k+1} = \alpha_k, \alpha_{g,k+1} = \alpha_g$ et aller à l'étape 2.

Étape 2 :

- si $\alpha_{d,k+1} = 0$ déterminer $\alpha_{k+1} \in]\alpha_{g,k+1}, +\infty[$
 - si $\alpha_{d,k+1} \neq 0$ déterminer $\alpha_{k+1} \in]\alpha_{g,k+1}, \alpha_{d,k+1}[$ remplacer k par $k + 1$ et aller à l'étape 1.
-

2.2.3 Existence du pas d'Armijo

Pour assurer l'existence du pas d'Armijo en posant quelques conditions sur la fonction h_k .

Théorème 2.2.1. [12] Si $h_k : \mathbb{R}_+ \rightarrow \mathbb{R}$, définie par (2.1.5) $h_k(\alpha) = f(x_k + \alpha d_k)$ 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.

Preuve. On a

$$h_k(\alpha) = f(x_k + \alpha d_k)$$

$$\psi_\rho(\alpha) = f(x_k) + \rho \alpha \nabla^T f(x_k) d_k$$

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

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

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

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

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

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

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

$$\psi_\rho(\alpha) \rightarrow -\infty; \alpha \rightarrow +\infty,$$

on déduit que la fonction $\psi_\rho(\alpha) - h_k(\alpha)$ possède la propriété :

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

Donc s'annule au moins une fois pour $\alpha > 0$.

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

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

Ce qui achève la démonstration. □

2.3 Recherche linéaire inexacte de Wolfe

En 1969, **WOLFE** [27] a proposé une méthode basée sur le choix de deux paramètres.

2.3.1 Conditions de Wolfe faibles

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

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha \nabla^T f(x_k) d_k \quad (2.3.1)$$

$$\nabla^T f(x_k + \alpha d_k) d_k \geq \sigma \nabla^T f(x_k) d_k \quad (2.3.2)$$

Autrement dit :

$$h_k(\alpha) \leq h_k(0) + \rho h'_k(0)\alpha \quad (2.3.3)$$

$$h'_k(\alpha) \geq \sigma h'_k(0) \quad (2.3.4)$$

On voit bien à la figure (2.2) ce que signifie cette condition.

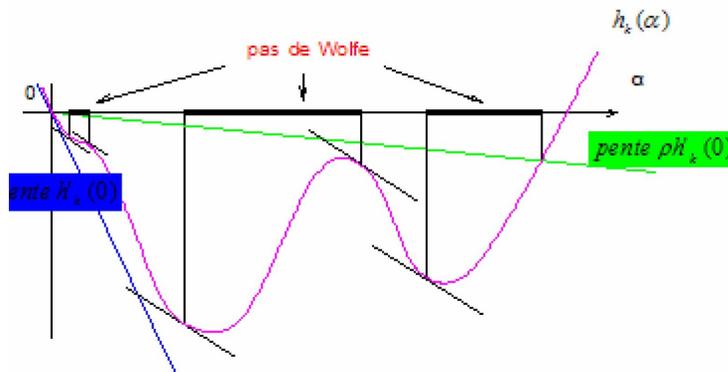


FIGURE 2.2 – Règle de Wolfe

2.3.2 Conditions de Wolfe fortes

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha \nabla^T f(x_k) d_k \quad (2.3.5)$$

$$|\nabla^T f(x_k + \alpha d_k) d_k| \leq -\sigma \nabla^T f(x_k) d_k \quad (2.3.6)$$

autrement dit :

$$h_k(\alpha) \leq h_k(0) + \rho h'_k(0)\alpha \quad (2.3.7)$$

$$|h'_k(\alpha)| \leq -\sigma h'_k(0) \quad (2.3.8)$$

où $0 < \rho < \sigma < 1$.

2.3.3 Algorithme de la Règle de Wolfe

Algorithm 2: Règle de Wolfe

Etape 0 : (initialisation)

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

Etape 1 :

- si $h_k(\alpha_k) \leq h_k(0) + \rho h'_k(0)\alpha_k$ et $h'_k(\alpha) \geq \sigma h'_k(0)$: STOP ($\tilde{\alpha} = \alpha_k$).
- si $h_k(\alpha_k) > h_k(0) + \rho h'_k(0)\alpha_k$, alors $\alpha_{d,k+1} = \alpha_k, \alpha_{g,k+1} = \alpha_{g,k}$ et aller à l'étape 2.
- si $h'_k(\alpha) < \sigma h'_k(0)$, alors $\alpha_{d,k+1} = \alpha_{d,k}, \alpha_{g,k+1} = \alpha_k$ et aller à l'étape 2.

Etape 2 :

- si $\alpha_{d,k+1} = 0$ déterminer $\alpha_{k+1} \in]\alpha_{g,k+1}, +\infty[$
 - si $\alpha_{d,k+1} \neq 0$ déterminer $\alpha_{k+1} \in]\alpha_{g,k+1}, \alpha_{d,k+1}[$ remplacer k par $k + 1$ et aller à l'étape 1.
-

2.3.4 Existence du pas de Wolfe

Théorème 2.3.1. [13] Si $h_k : \mathbb{R}_+ \rightarrow \mathbb{R}$; définie par (2.1.5) $h_k(\alpha) = f(x_k + \alpha d_k)$ 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) (2.3.1)-(2.3.2) est non vide.

Preuve. On a

$$h_k(\alpha) = f(x_k + \alpha d_k)$$

$$\psi_\rho(\alpha) = f(x_k) + \rho \alpha \nabla^T f(x_k) d_k$$

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

$$h_k(\alpha) = f(x_k + \alpha d_k) = f(x_k) + \rho \alpha \nabla^T f(x_k) d_k + \alpha \zeta(\alpha) o_\zeta(\alpha) \rightarrow 0, \alpha \rightarrow 0.$$

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

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

On voit que pour $\alpha > 0$ assez petit on a : $h_k(\alpha) < \psi_\rho(\alpha)$

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

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

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

donc s'annule au moins une fois pour $\alpha > 0$. En choisissant le plus petit de ces zéros on voit qu'il existe $\alpha > 0$ tel que :

$$h_k(\bar{\alpha}) = \psi_\rho(\bar{\alpha}) \text{ et } h_k(\alpha) < \psi_\rho(\alpha) \text{ pour } 0 < \alpha < \bar{\alpha} \quad (2.3.9)$$

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

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

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

Donc $\tilde{\alpha}$ satisfait (2.3.2) D'autre part, $\alpha = \hat{\alpha}$ satisfait (2.3.1) en effet, $\hat{\alpha}$ satisfait (2.3.9) $h_k(\hat{\alpha}) < \psi_\rho(\hat{\alpha})$ n'est autre que :

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

Ce qu'il fallait démontrer . □

2.4 Contribution des recherches linéaires inexactes dans la convergence de la classe des méthodes à directions de descente

2.4.1 Condition de Zoutendijk

Dans cette partie, nous allons étudier la contribution de la recherche linéaire inexacte à la convergence des algorithmes de classe des méthodes à directions de descente.

Tout d'abord on va définir une condition importante sur les méthodes à direction de descente afin d'assurer leurs convergence

Définition 2.4.1. On dit qu'une règle de recherche linéaire vérifie la condition de Zoutendijk s'il existe une constante $C > 0$ de telle sorte que pour chaque indice $k \geq 1$ on ait :

$$f(x_{k+1}) \leq f(x_k) - C \|\nabla f(x_k)\|^2 \cos^2 \theta_k \quad (2.4.1)$$

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

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

Voici comment on se sert de la condition de Zoutendijk [Voir [28]].

Proposition 2.4.1. Si la suite $\{x_k\}$ générée par un algorithme d'optimisation vérifie la condition de Zoutendijk (2.4.1) et si la suite $\{f(x_k)\}$ est minorée, alors

$$\sum_{k \geq 1} \|\nabla f(x_k)\|^2 \cos^2 \theta_k < \infty \quad (2.4.2)$$

Preuve. En sommant les inégalités (2.4.1), on a

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

La série est donc convergente puisqu'il existe une constant C' telle que pour tout k ,

$f(x_k) \geq C'$ La proposition suivante précise les circonstances dans les quelles la condition de Zoutendijk (2.4.1) est vérifiée avec la règle de Wolfe. \square

Proposition 2.4.2. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continûment différentiable dans un voisinage de $\Lambda = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ On considère un algorithme à directions de descente d_k , qui génère une suite $\{x_k\}_{k \in \mathbb{N}}$ en utilisant la recherche linéaire de Wolfe (2.3.1) – (2.3.2).

Alors il existe une constante $C > 0$ telle que, pour tout $k \geq 1$ la condition de Zoutendijk (2.4.1) est vérifiée.

Preuve. D'après (2.3.2)

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

Et du fait que f est continument différentiable :

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

En utilisant (2.3.1) , on aura :

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \rho \alpha_k \nabla f(x_k) d_k \\ \Rightarrow f(x_k + \alpha_k d_k) &\leq f(x_k) + \rho \alpha \nabla f(x_k) d_k \\ \Rightarrow f(x_k + \alpha_k d_k) &\leq f(x_k) + |\rho \alpha \nabla f(x_k) d_k| \\ \Rightarrow f(x_k + \alpha_k d_k) &\leq f(x_k) - \rho \alpha \nabla f(x_k) d_k \end{aligned}$$

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

On en déduit (2.4.1)

□

Méthode du gradient conjugué

En analyse numérique, la méthode du gradient conjugué est un algorithme développé dans le but de résoudre des systèmes d'équations linéaires dont la matrice est symétrique et définie positive.

Il s'agit d'une méthode itérative à direction de descente qui converge en un nombre fini d'itérations, elle a été proposée par **HESTENES** et **STEIFEL**[14] en 1950.

Dans ce chapitre, on va décrire les méthodes du gradient conjugué mais avant d'y accéder on va tout d'abord étudier de façon brève la méthode du gradient.

3.1 Méthode du gradient

Il s'agit d'une méthode fondamentale proposée par **CAUCHY** [7] en 1847 afin de minimiser des problèmes d'optimisation sans contrainte. En effet, cette méthode prend comme direction : $d_k = -g_k$. Dans ce cas donc on peut dire que la méthode du gradient est une méthode à directions de descente.

Au point courant x_k , la direction du gradient négatif représente la meilleure direction pour la recherche d'un minimum de l'objectif f , c'est pour ce là que l'algorithme du gradient est également connu sous le nom d'algorithme de la plus profonde descente ou la plus forte pente (steepest descent).

3.1.1 Algorithme de la méthode du gradient

Algorithm 3: Méthode du gradient

Etape 0 : (initialisation)

Soit x_0 le point de départ.

Poser $d_0 = -\nabla f(x_0)$,

Poser $k = 0$ et aller à l'étape 1.

Etape 1 :

"Test d'arrêt" : si $\|\nabla f(x_k)\| \simeq 0$: STOP ($\tilde{x} = x_k$).

Si non aller à l'étape 2.

Etape 2 :

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

$$d_k = -\nabla f(x_k) \quad (3.1.1)$$

$$\alpha_k \text{ est déterminé par une recherche linéaire} \quad (3.1.2)$$

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

3.1.2 Convergence de la méthode du gradient

La convergence de l'algorithme du gradient est assurée par le théorème suivant.

Théorème 3.1.1. [4] *Supposons que f soit deux fois continûment différentiable. Si la Hessienne $\nabla^2 f(\tilde{x})$ de la fonction f est définie positive et possède la plus petite valeur propre $\lambda_1 > 0$ et la plus grande valeur propre $\lambda_n > 0$, alors la suite de valeurs objectives $\{f(x_k)\}$ générée par l'algorithme du gradient converge vers $f(\tilde{x})$ linéairement avec un taux de convergence non supérieur à*

$$\left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 = \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 \quad (3.1.3)$$

i.e.

$$f(x_{k+1}) \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 f(x_k) \quad (3.1.4)$$

où $\kappa = \frac{\lambda_n}{\lambda_1}$ est le nombre de conditions de la Hessienne.

3.2 La méthode du gradient conjugué linéaire

3.2.1 Principe général

La méthode du gradient conjugué est une méthode très puissante et très utilisée pour la résolution des problèmes d'optimisation sans contraintes surtout de grande taille.

Elle se base sur le concept des directions conjuguées, car les gradients successifs sont à la fois orthogonaux entre eux et aux directions précédentes.

L'idée principale de cette méthode est de construire une suite de directions de descente que sont mutuellement conjuguées par rapport à la fonction quadratique afin de résoudre le problème d'optimisation sans contraintes (P) suivant :

$$(P) : \{\min f(x) : x \in \mathbb{R}^n\} \quad (3.2.1)$$

Où f est une fonction quadratique strictement convexe de n variables.

Ça veut dire que la minimisation de $f(x)$ revient à la minimisation de $q(x)$ où :

$$q(x) = \frac{1}{2}x^T Ax - b^T x + c, \quad \text{avec } x \in \mathbb{R}^n \quad (3.2.2)$$

- $A \in \mathcal{M}_{n \times n}$ est une matrice symétrique et définie positive.

- $b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$ un vecteur de \mathbb{R}^n .

- $c \in \mathbb{R}$ est une constante.

La méthode du gradient conjugué linéaire est une méthode itérative, elle consiste à générer une suite $\{x_k\}$ de la manière suivante :

✓ On commence par un vecteur x_0 donné comme point de départ.

✓ A chaque étape k si on a $x_k \in \mathbb{R}^n$ alors le successeur x_{k+1} de x_k est déterminé par la relation suivante :

$$x_{k+1} = x_k + \alpha_k d_k \quad (3.2.3)$$

Où $\{d_0, d_1, \dots, d_n\}$ constitue un ensemble de directions de descente ayant la propriété d'être mutuellement conjuguées par rapport à la fonction quadratique $q(x)$.

Les directions de descente d_k cherchées sont définies avec les formules de récurrence qui suit :

$$d_k = \begin{cases} -g_0 = \nabla q(x_0) & \text{si } k = 0 \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1 \end{cases} \quad (3.2.4)$$

Le gradient de f au point x_k est notée par g_k . On détermine la valeur du pas $\alpha_k \in \mathbb{R}_+$ par une recherche linéaire c'est ce qu'on appelle une optimisation unidimensionnelle et le coefficient $\beta_k \in \mathbb{R}$ est déterminé d'une manière où les directions d_k soient mutuellement conjuguées par rapport à la forme quadratique $q(x)$.

3.2.2 Détermination du pas α_k

En effet, une méthode pour déterminer le pas α_k est de résoudre le problème d'optimisation unidimensionnel suivant :

$$\alpha_k = \min f(x_k + \alpha d_k), \alpha > 0 \quad (3.2.5)$$

On note

$$g_k = \nabla f(x_k) = \nabla q(x_k) = Ax_k - b \quad (3.2.6)$$

Comme α_k minimise q dans la direction d_k alors le pas α_k vérifie la condition nécessaire d'optimalité et on a $\forall k$:

$$\frac{dq}{d\alpha}(\alpha_k) = q'(\alpha_k) = d_k^T \nabla q(x_{k+1}) = 0 \quad (3.2.7)$$

$$d_k^T \nabla q(x_{k+1}) = d_k^T (Ax_{k+1} - b) = 0 \quad (3.2.8)$$

D'où :

$$d_k^T A(x_k + \alpha_k d_k) - d_k^T b = 0 \quad (3.2.9)$$

Alors

$$d_k^T A \alpha_k d_k = -d_k^T A x_k + d_k^T b \quad (3.2.10)$$

$$\Rightarrow d_k^T A \alpha_k d_k = -d_k^T (A x_k + b) \quad (3.2.11)$$

D'où l'on tire

$$\alpha_k = \frac{-d_k^T (A x_k + b)}{d_k^T A d_k} \quad (3.2.12)$$

Donc

$$\alpha_k = \frac{-d_k^T g_k}{d_k^T A d_k} \quad (3.2.13)$$

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

3.2.3 Détermination du coefficient β_k

Les coefficients β_k étant choisis de telle manière que d_k soit conjuguée avec toutes les directions précédentes. Autrement dit il faut avoir :

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

on en déduit :

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

3.2.4 Algorithme du gradient conjugué linéaire

Nous allons utiliser l'algorithme suivant pour résumer tout ce qui est dit ci-dessus :

Algorithm 4: Méthode du gradient conjugué linéaire

Etape 0 : (initialisation)

Soit x_0 le point de départ, $g_0 = \nabla q(x_0) = Ax_0 - b$.

Poser $d_0 = -g_0$.

Poser $k = 0$ et aller à l'étape 1.

Etape 1 :

Si $\|g_k\| = 0$: STOP ($\tilde{x} = x_k$). "Test d'arrêt"

Si non aller à l'étape 2.

Etape 2 :

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

$$\alpha_k = \frac{-d_k^T g_k}{d_k^T A d_k} \quad (3.2.14)$$

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k \quad (3.2.15)$$

$$\beta_{k+1} = \frac{g_{k+1}^T A d_k}{d_k^T A d_k} \quad (3.2.16)$$

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

3.2.5 Convergence de méthode du gradient conjugué linéaire

On a vu que la méthode du gradient conjugué linéaire converge vers la solution optimale en au plus n itérations (en supposant que les calculs numériques sont exacts). Pour montrer sa validité, nous devons tout d'abord énoncer les lemmes suivants :

Lemme 3.2.1. *On note :*

$$g_k = \nabla q(x_k) = Ax_k - b \quad (3.2.17)$$

où $x_k, k = 0, \dots, n - 1$ sont obtenus par l'algorithme du gradient conjugué linéaire. Alors on a :

$$g_{k+1}^T g_i = 0; \quad k = 0, \dots, n - 1; \quad j = 0, \dots, kn \quad (3.2.18)$$

Lemme 3.2.2. Les directions $\{d_0, d_1, \dots, d_{n-1}\}$ engendrées par l'algorithme du gradient conjugué linéaire sont A conjuguées.

Théorème 3.2.1. [19] A une itération k quelconque de l'algorithme ou l'optimum de $q(x)$ n'est pas encore atteint (c-à-d $g_i \neq 0, i = 0, 1, \dots, k$) on a :

a)

$$\alpha_k = \frac{g_k^T g_k}{d_k^T A d_k} \quad (3.2.19)$$

b)

$$\beta_{k+1}^T = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k} \quad (3.2.20)$$

$$= \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \quad (3.2.21)$$

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

Preuve. La Démonstration de ce théorème se fait par utilisant la récurrence sur k . Supposons que d_0, d_1, \dots, d_k soient tous mutuellement conjugués.

a) On va d'abord montrer l'équivalence de (3.2.19) et de (3.2.13) :

On a : $d_k = -g_k + \beta_k d_{k-1}$, et (3.2.13) peut s'écrire de la façon suivante :

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

Comme $(d_0, d_1, \dots, d_{k-1})$ sont mutuellement conjuguées, x_k est l'optimum de $q(x)$ sur la variété ϑ_k passant par x_0 et engendrée par $(d_0, d_1, \dots, d_{k-1})$. Donc $d_{k-1}^T g_k = 0$ d'où l'on déduit (3.2.19).

b) Pour démontrer (3.2.20) remarquons que :

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

On a alors :

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

En utilisant (3.2.19)

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

il vient

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

Or

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

Ce qui démontre (3.2.20).

Alors du fait que :

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

Car

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

Appartient au sous-espace engendré par (d_0, d_1, \dots, d_k) et que g_{k+1} est orthogonal à ce sous-espace, de cela nous obtenons la formule (3.2.21).

c) Enfin on va montrer que d_{k+1} est conjuguée par rapport à (d_0, d_1, \dots, d_k) , alors on a :

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

Vérifions maintenant que : $d_{k+1}^T Ad_i = 0$ pour $i = 0, 1, \dots, k-1$.

On a :

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

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

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

$$Ad_i = \frac{1}{\alpha_i} (Ax_{i+1} - Ax_i) = \frac{1}{\alpha_i} (g_{i+1} - g_i).$$

En écrivant :

$$g_{i+1} = d_{i+1} - \beta_i d_i,$$

$$g_i = d_i - \beta_{i-1} d_{i-1},$$

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

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

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

3.2.6 Vitesse de convergence

Le résultat suivant est de Luenenberger [17], fournit une estimation de la décroissance de la norme d'erreur $x_k - \tilde{x}$, en terme κ du conditionnement de la matrice A :

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \quad (3.2.22)$$

où : $\lambda_{\max}(A)$, $\lambda_{\min}(A)$ sont respectivement, les plus grandes, petites des valeurs propres de la matrice A . La norme utilisée est donc la suivante :

$$\|x\|_A = \left(x^T A x\right)^{\frac{1}{2}} \quad (3.2.23)$$

Théorème 3.2.2. [17] Soit κ le conditionnement de A . La suite $\{x_k\}$ générée par l'algorithme du gradient conjugué linéaire vérifie l'estimation

$$\|x_{k+1} - \tilde{x}\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|x_1 - \tilde{x}\|_A \quad (3.2.24)$$

3.2.7 Les avantages de la méthode du gradient conjugué linéaire

1– L'algorithme du gradient conjugué linéaire est très utile dans le but de résoudre des grands systèmes creux, en effet il suffit de savoir comment appliquer la matrice A à un vecteur .

2– La convergence peut être très rapide si A admet seulement r ($r < n$) valeurs propres distincts la convergence a lieu en au plus r itérations.

3.3 Exemple

Exemple 3.3.1. Dans cet exemple on va minimiser la fonction quadratique suivante en utilisant la méthode du gradient conjuguée linéaire :

Soit $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, $(x_1, x_2) \mapsto f(x_1, x_2) = \frac{1}{2}x_1^2 + x_2^2 - x_1 - 2x_2$

On peut l'écrire sous la forme : $f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} x - \begin{pmatrix} 1 \\ 2 \end{pmatrix} x$, avec $x_0 = (0, 0)^T$

Remarquons tout d'abord que $A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ est définie positive car $\Delta_1 = 1 > 0$ et

$\Delta_2 = \begin{vmatrix} 1 & 0 \\ 0 & 2 \end{vmatrix} = 2 > 0 \Rightarrow f$ est une fonction quadratique strictement convexe.

De plus on a : $g_k = \nabla f(x_k) = Ax_k - b$,

et on prend comme direction initial $d_0 = -g_0$, $\varepsilon = 10^{-6}$

$$g_0 = \nabla f(x_0) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix} = - \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$d_0 = -g_0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix},$$

on calcule $\|g_0\|_2 = \sqrt{5} > 10^{-6} = \varepsilon$

Avant toute chose, rappelons que les itérations de la méthode du gradient conjugué sont définies comme suit : $x_{k+1} = x_k + \alpha_k d_k$, avec

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

Donc, on obtient $\alpha_0 = \frac{-d_0^T g_0}{d_0^T A d_0} = \frac{5}{9}$

k = 1 :

$$x_1 = x_0 + \alpha_0 d_0$$

$$\begin{aligned} \Rightarrow x_1 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{5}{9} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} \frac{5}{9} \\ \frac{10}{9} \end{pmatrix} \\ g_1 &= \nabla f(x_1) = Ax_1 - b \\ \Rightarrow g_1 &= \begin{pmatrix} -4 \\ \frac{9}{2} \\ \frac{9}{9} \end{pmatrix}, \\ \|g_1\|_2 &= \sqrt{0.24691358} = 0.496903995 > 10^{-6} = \varepsilon \\ \text{Comme } d_{k+1} &= -g_{k+1} + \beta_{k+1}d_k, \end{aligned}$$

$$\begin{aligned} \beta_{k+1} &= \frac{g_{k+1}^T Ad_k}{d_k^T Ad_k} \Rightarrow \beta_1 = \frac{g_1^T Ad_0}{d_0^T Ad_0} = \frac{0.44}{9} = 0.04938, \\ d_1 &= -g_1 + \beta_1 d_0 = \begin{pmatrix} 0.4938 \\ -0.1234 \end{pmatrix} \\ \alpha_1 &= \frac{-d_1^T g_1}{d_1^T Ad_1} = 0.85 \end{aligned}$$

k = 2 :

$$\begin{aligned} x_2 &= x_1 + \alpha_1 d_1 \\ \Rightarrow x_2 &= \begin{pmatrix} \frac{5}{9} \\ \frac{10}{9} \end{pmatrix} + 0.914 \begin{pmatrix} 0.4938 \\ -0.1234 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ g_2 &= \nabla f(x_2) = Ax_2 - b \\ \Rightarrow g_2 &= \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 0, \\ \|g_2\|_2 &= 0 < 10^{-6} = \varepsilon \end{aligned}$$

En conclusion, la méthode converge en 2 itérations, c'est à dire que le min de f est atteint en $\tilde{x} = x_2$ et on a :

$$\tilde{x} = x_2^T = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$f(\tilde{x}) = \frac{1}{2} \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

En effet, $f(\tilde{x}) = -1.5$ est le minimum global de la fonction f avec la précision donnée.

3.4 La méthode du gradient conjugué non-linéaire

3.4.1 Principe général

La méthode du gradient conjugué linéaire a été proposée afin de minimiser des fonctions quadratiques strictement convexes. La question qui se pose maintenant est peut-on utiliser cette méthode pour minimiser des fonctions non convexes ou même des fonctions non-linéaires ?

Afin de minimiser une fonction non-linéaire f , la méthode du gradient conjugué doit être adaptée selon les étapes ci-dessous :

- ✓ D'abord, on effectue une recherche linéaire pour déterminer le pas α_k qui minimise la fonction non-linéaire f le long de d_k .
- ✓ Ensuite, le résidu ∇q devra être remplacé par le gradient de la fonction f .

On obtient alors un algorithme très efficace pour l'optimisation non-linéaire.

Pour garantir que la direction d_{k+1} est une direction descente, le paramètre α_k devra vérifier certaines conditions. En effet, si la recherche linéaire est exacte, on voit que :

$$\nabla^T f(x_k) d_k < 0$$

Par contre, si la recherche linéaire n'est pas exacte (puisque la recherche exacte est trop coûteuse), d'autres conditions doivent être imposées.

L'idée de cette méthode est de construire des directions d_0, \dots, d_k mutuellement conjuguées d'une manière itérative.

La direction d_k étant obtenue à chaque étape k comme une combinaison linéaire du gradient au point x_k et de la direction précédente d_{k-1} , les coefficients seront choisis de telle manière que d_k soit conjugué avec toutes les directions précédentes.

Nous nous intéressons ici à la minimisation d'une fonction f de n variables à valeurs réelles

$$\min \{f(x) : x \in \mathbb{R}^n\} \quad (3.4.1)$$

pas nécessairement quadratique.

Comme indiqué précédemment, la direction d_k est définie par la formule de récurrence suivante :

$$d_k = \begin{cases} -g_0 & \text{si } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1 \end{cases} \quad (3.4.2)$$

avec $g_k = \nabla f(x_k)$, $\beta_k \in \mathbb{R}$ et $\{x_k\}$ est généré par la formule suivante :

$$x_{k+1} = x_k + \alpha_k d_k \quad (3.4.3)$$

et le pas α_k étant calculé par une recherche linéaire.

3.4.2 Quelques variantes de la méthode du gradient conjugué non linéaire

Les différentes variantes de la méthode du gradient conjugué non linéaire sont des extensions de la méthode du gradient conjugué linéaire pour le cas quadratique. Plusieurs mathématiciens ont étendu la méthode du gradient conjugué pour le cas non linéaire. Ceci a été réalisé pour la première fois, en 1964 par Fletcher et Reeves [11] (méthode de Fletcher-Reeves), en 1969 par Polak-Ribière [22] et Ployak [23] (méthode de Polak-Ribière-Ployak), en 1999 par Dai et Yuan [9] (méthode de Dai-Yuan).

Dans le tableau ci-dessous on va citer quelques variantes de la méthode du gradient conjugué non linéaire :

TABLE 3.1 – Les différentes variantes du gradient conjugué

$\beta_{k+1}^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$	Proposé par <i>Hestenes</i> et <i>Stiefel</i> [1952]
$\beta_{k+1}^{FR} = \frac{\ g_{k+1}\ ^2}{\ g_k\ ^2}$	Proposé par <i>Fletcher</i> et <i>Reeves</i> [1964]
$\beta_{k+1}^D = \frac{g_{k+1}^T \nabla^2 f(x_k) d_k}{d_k^T \nabla^2 f(x_k) d_k}$	Proposé par <i>Daniel</i> [1967]
$\beta_{k+1}^{PRP} = \frac{g_{k+1}^T y_k}{\ g_k\ ^2}$	Proposé par <i>Polak, Ribiere</i> et <i>Polyak</i> [1969]
$\beta_{k+1}^{CD} = \frac{\ g_{k+1}\ ^2}{-d_k^T g_k}$	Proposé par <i>Fletcher</i> [1987]
$\beta_{k+1}^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}$	Proposé par <i>Lieu</i> et <i>Story</i> [1991]
$\beta_{k+1}^{DY} = \frac{\ g_{k+1}\ ^2}{d_k^T g_k}$	Proposé par <i>Dai</i> et <i>Yuan</i> [1999]
$\beta_{k+1}^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T g_k}, \quad t > 0$	Proposé par <i>Dai</i> et <i>Lieu</i> [2001]
$\beta_{k+1}^{YT} = \frac{g_{k+1}^T (z_k - t s_k)}{d_k^T z_k}$	Proposé par <i>Yabe</i> et <i>Takano</i> [2004]

la où :

$$y_k = g_{k+1} - g_k$$

$$s_k = x_{k+1} - x_k$$

$$z_k = y_k + \frac{\rho v_k}{s_k^T u_k} u_k, \rho \geq 0 \text{ et } u_k \in [0, 1]$$

et

$$v_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k$$

3.4.3 Algorithme du gradient conjugué non linéaire avec la recherche linéaire inexacte de Wolfe forte

Nous allons présenter ci-dessous le schéma général de l'algorithme du gradient conjugué non linéaire pour des fonctions quelconques.

Algorithm 5: La méthode du gradient conjugué non-linéaire

Etape 0 : (initialisation)

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

Etape 1 :

Si $\|g_k\| = 0$: STOP ($\tilde{x} = x_k$). "Test d'arrêt" Si non aller à l'étape 2.

Etape 2 :

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

α_k : déterminé par la recherche linéaire inexacte de Wolfe forte .

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k \quad (3.4.4)$$

où

β_k : défini selon la méthode.

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

3.4.4 Convergence des méthodes du gradient conjugué non linéaires

Le premier résultat de convergence de la méthode du gradient conjugué non linéaire avec des recherches linéaires inexacte était démontré par Al-Baali [1].

Le résultat qui suit garantit la convergence de toutes les variantes de la méthode du gradient conjugué non linéaire avec la recherche linéaire inexacte de Wolfe pour des fonctions

quelconques.

Théorème 3.4.1. [4] Supposons que l'ensemble $\Lambda = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ est borné où $x_1 \in \mathbb{R}^n$ est le point initial et que sur un voisinage \mathcal{V} de Λ , la fonction objectif f est continûment différentiable et son gradient est lipschitzien. Considérons une méthode du gradient conjugué non linéaire du type (3.4.2) et (3.4.3) où α_k est déterminée avec la recherche linéaire inexacte de Wolfe forte (2.3.5) et (2.3.6). Alors

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0, \quad (3.4.5)$$

où

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \quad (3.4.6)$$

Preuve. De (3.4.4) pour tout $k \geq 1$,

$$d_{k+1} + g_{k+1} = \beta_{k+1} d_k \quad (3.4.7)$$

en appliquant le carré sur les deux côtés de (3.4.7) résulte :

$$\|d_{k+1}\|^2 = -\|g_{k+1}\|^2 - 2g_{k+1}^T d_{k+1} + \beta_{k+1}^2 \|d_k\|^2 \quad (3.4.8)$$

Puisque d_k satisfait la condition de descente $g_k d_k < 0$ d'après (3.4.8), il en déduit que

$$\|d_{k+1}\|^2 \geq -\|g_{k+1}\|^2 + \beta_{k+1}^2 \|d_k\|^2 \quad (3.4.9)$$

De (3.4.4)

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

lequel, avec la condition de recherche linéaire de Wolfe forte (2.3.5) montre que

$$\left| g_{k+1}^T d_{k+1} \right| + \sigma |\beta_{k+1}| \left| g_k^T d_k \right| \geq \|g_{k+1}\|^2 \quad (3.4.10)$$

En appliquant l'inégalité $(a + \sigma b)^2 \leq (1 + \sigma^2)(a^2 + b^2)$ pour tout $a, b, \sigma \geq 0$ avec,

$a = |g_{k+1}^T d_{k+1}|$ et $b = |\beta_{k+1}| |g_k^T d_k|$, de (3.4.10) on peut écrire

$$\left(g_{k+1}^T d_{k+1}\right)^2 + \beta_{k+1}^2 \left(g_k^T d_k\right)^2 \geq c \|g_{k+1}\|^4 \quad (3.4.11)$$

où $c = (1 + \sigma^2)^{-1}$. Par conséquent, de (3.4.9) et (3.4.11), il résulte que

$$\begin{aligned} \frac{\left(g_{k+1}^T d_{k+1}\right)^2}{\|d_{k+1}\|^2} + \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} &= \frac{1}{\|d_{k+1}\|^2} \left[\left(g_{k+1}^T d_{k+1}\right)^2 + \frac{\|d_{k+1}\|^2}{\|d_k\|^2} \left(g_k^T d_k\right)^2 \right] \\ &\geq \frac{1}{\|d_{k+1}\|^2} \left[\left(g_{k+1}^T d_{k+1}\right)^2 + \beta_{k+1}^2 \left(g_k^T d_k\right)^2 - \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} \|g_{k+1}\|^2 \right] \\ &\geq \frac{1}{\|d_{k+1}\|^2} \left[c \|g_{k+1}\|^4 - \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} \|g_{k+1}\|^2 \right] \end{aligned} \quad (3.4.12)$$

Or, si (3.4.5) n'est pas vrai, alors (3.4.12) et la condition de Zoutendijk (2.4.2) impliquent que l'inégalité suivante

$$\frac{\left(g_{k+1}^T d_{k+1}\right)^2}{\|d_{k+1}\|^2} + \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} \geq \frac{c}{2} \frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2} \quad (3.4.13)$$

est vérifiée pour k suffisamment grand. Avec cela, l'inégalité (3.4.6) suit (3.4.13) et (3.4.12).

□

Méthodes hybrides du gradient conjugué non linéaire

Dans ce chapitre on va présenter le noyau de notre travail qu'on a eu l'honneur d'exposer dans la conférence nationale " National Conference on Applied Mathematics and Didactics (NCAMD2021)" organisée par l'université de Constantine, le 26 Juin 2021 [Voir [10]].

Dans cette partie on va aborder la classe des méthodes hybrides du gradient conjugué non linéaire.

Cette dernière se divise en deux parties : méthodes hybrides de projection et méthodes hybrides basées sur des combinaisons convexes .

Avant d'étudier ces deux catégories, on va tout d'abord présenter le principe d'hybridation.

4.1 Principe général

Les méthodes hybrides du gradient conjugué non linéaires sont des méthodes qui combine deux ou plusieurs variantes du gradient conjugué non linéaire afin de minimiser des problèmes d'optimisation sans contraintes, soit en se basant sur le concept de projection (méthode hybride de projection), soit en basculant entre eux au cours de la méthode

(méthode hybride convexe).

Ceci est généralement fait pour combiner les caractéristiques souhaitées de chaque variante, de sorte que la méthode globale (hybride) soit meilleur que les composantes individuelles.

On considère le problème d'optimisation sans contrainte suivant :

$$(p) : \min \{f(x); x \in \mathbb{R}^n\} \quad (4.1.1)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction continûment différentiable.

Les méthodes du gradient conjugué sont très efficaces pour résoudre des problèmes d'optimisation sans contrainte de ce type, surtout lorsque la dimension n est grande. Généralement, les itérations générées par la méthode du gradient conjugué afin de résoudre le problème (4.1.1) sont obtenues par :

$$x_{k+1} = x_k + \alpha_k d_k \quad (4.1.2)$$

Où, le pas $\alpha_k \in \mathbb{R}_+$ est calculée en effectuant une recherche linéaire exacte ou inexacte, et la direction d_k est générée par la formule de récurrence suivante :

$$d_k = \begin{cases} -g_0 & \text{si } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{si } k \geq 1 \end{cases} \quad (4.1.3)$$

avec un scalaire approprié $\beta_k \in \mathbb{R}$ et $g_k = \nabla f(x_k)$. Le pas α_k est généralement calculée par la recherche linéaire de Wolfe.

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k, \quad g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k \quad (4.1.4)$$

ou par la recherche linéaire de Wolfe forte

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k, \quad |g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k \quad (4.1.5)$$

où les paramètres et vérifient $0 < \rho < \sigma < 1$.

En ce qui concerne les différentes valeurs de β_k , il existe des méthodes distinctes de gradient conjugué; on cite par exemple quelques méthodes bien connues : méthode de Fletcher-Reeves, méthode Polak-Ribière-Polyak, méthode Hestenes-Stiefel, méthode Dai-Yuan et la méthode de Liu-Storey Leurs paramètres β_k sont spécifiés comme suit :

$$\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (4.1.6)$$

$$\beta_{k+1}^{PRP} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} \quad (4.1.7)$$

$$\beta_{k+1}^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \quad (4.1.8)$$

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} \quad (4.1.9)$$

$$\beta_{k+1}^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k} \quad (4.1.10)$$

où $\|\cdot\|$ représente la norme Euclidienne et $y_k = g_{k+1} - g_k$

Les propriétés de convergence de ces méthodes ont été mentionnées dans plusieurs références [Voir [23, 8, 28, 9, 22]]

4.2 Méthodes hybrides du gradient conjugué non linéaire de projection

Dans cette section on va présenter trois différentes variantes de la classe des méthodes hybrides du gradient conjugué non linéaire de projection. Ces variantes sont égaux théoriquement mais pratiquement elles sont différentes. Après avoir présenter ces variantes on va citer leurs algorithmes, prouver le théorème qui assure leurs convergence.

4.2.1 Méthode hybride de Wei, Yao et Liu

La méthode hybride WYL [26] a été proposée en 2006 par Wei, Yao et Liu. Cette variante est une modification de la méthode PRP. Vu que la méthode PRP est plus performante que les autres variantes du gradient conjugué non linéaire, mais aucun résultat de convergence n'a été établi. Cette méthode hybride de projection possède les avantages de la méthode PRP en plus Wei, Yao et Liu ont prouvé que la direction de cette méthode est de descente suffisante et converge globalement si le pas est déterminé par la recherche linéaire inexacte de Wolfe forte si $\sigma < \frac{1}{4}$. Le paramètre β_k^{WYL} est donné par :

$$\beta_k^{WYL} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}}{\|g_{k-1}\|^2} \quad (4.2.1)$$

4.2.2 Méthode hybride de Yao, Wei et Huang

La méthode hybride YWH [24] a été proposée en 2007 par Yao, Wei et Huang. Cette variante est une modification de la méthode HS. Cette méthode hybride de projection possède les avantages de la méthode HS en plus Yao, Wei et Huang ont prouvé que la direction de cette méthode est de descente suffisante et converge globalement si le pas est déterminé par la recherche linéaire inexacte de Wolfe. Le paramètre β_k^{YWH} est donné par :

$$\beta_k^{YWH} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} \quad (4.2.2)$$

4.2.3 Méthode hybride de Jian , Han et Jiang

La méthode hybride JHJ [15] a été proposée en 2015 par Jinbao Jian , Lin Han et Xianzhen Jiang. Cette variante est une hybridation des deux variantes présentées dessus (WYL et YWH).

Le paramètre β_k^{JHJ} est donné par :

$$\beta_k^{JHJ} = \frac{\|g_k\|^2 - \max \left\{ 0, \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1} \right\}}{\max \left\{ \|g_{k-1}\|^2, d_{k-1}^T (g_k - g_{k-1}) \right\}} \quad (4.2.3)$$

Il est facile de voir que $\beta_k^{JHJ} = \beta_k^{DY}$ ou β_k^{FR} ou β_k^{WYL} ou β_k^{YWH} , donc β_k^{JHJ} est l'une des hybrides de β_k^{DY} , β_k^{FR} , β_k^{WYL} et β_k^{YWH} . Ainsi, une nouvelle méthode du gradient conjugué hybride est induite.

Cette nouvelle méthode hybride possède de bonnes propriétés de convergence et de descente. En effet ses directions sont de descente et ceci quelle que soit la technique de recherche linéaire inexacte utilisée, autrement dit qu'elle est indépendante de la règle de Wolfe et d'Armijo, ce qui permet d'avoir une large gamme de choix du pas.

En particulier, si le pas est donnée par la recherche linéaire de Wolfe (4.1.4), alors la méthode proposée est globalement convergente.

4.2.4 Algorithme des méthodes hybrides du gradient conjugué non linéaire de projection

Algorithm 6: Algorithme des méthodes hybrides du gradient conjugué non linéaire de projection

Étape 0 : (initialisation) Soit x_1 un point initial et $\varepsilon \geq 0$. Définir $d_1 = -g_1$, poser $k = 1$.

Étape 1 : Si $\|g_k\| \leq \varepsilon$ STOP. Si non aller à l'étape 2.

Étape 2 : Déterminer le pas α_k par une recherche linéaire appropriée.

Étape 3 : Générer l'itération suivante par $x_{k+1} = x_k + \alpha_k d_k$ calculer $g_{k+1} = g(x_{k+1})$ et β_{k+1}^{JHJ} par (4.2.3)

Étape 4 : Calculer la direction en utilisant $d_{k+1} = -g_{k+1} + \beta_{k+1}^{JHJ} d_k$.

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

4.2.5 Descente de la direction

Le lemme suivant indique que les directions de recherche de l'algorithme des méthodes hybrides du gradient conjugué non linéaire de projection sont toutes de descentes, quelle que soit la méthode de recherche linéaire utilisée.

Lemme 4.2.1. [15] *Soit f une fonction objectif continûment différentiable. Supposons que les directions d_k sont générées par l'algorithme des méthodes hybrides du gradient conjugué non linéaire de projection sont de descente.*

Autrement dit $g_k^T d_k < 0$, est vérifiée pour tout $k \geq 1$.

Lemme 4.2.2. [15] *Pour tout $k \geq 1$ la relation $0 \leq \beta_k^{IHJ} \leq \frac{g_k^T d_k}{g_{k-1}^T d_{k-1}}$, est toujours vérifiée.*

4.2.6 La convergence globale

Dans cette section, la convergence globale de l'algorithme des méthodes hybrides du gradient conjugué non linéaire de projection est prouvée si le pas est déterminé par la recherche linéaire inexacte de Wolfe. C'est-à-dire le pas α_k est donnée par la condition (4.1.4).

Le lemme suivant est une brick essentiel dans la démonstration de la convergence des méthodes hybrides de projection, en effet il assure la satisfaction de la condition de Zoutendijk [28]

Lemme 4.2.3. [16] *Supposons que l'ensemble $\Lambda = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ est borné où $x_1 \in \mathbb{R}^n$ est le point initial et que sur un voisinage \mathcal{V} de Λ , la fonction objectif f est continûment différentiable et son gradient est lipschitzien.*

Considérons toutes méthodes de type (4.1.2), si la direction d_k satisfait $g_k^T d_k < 0$ et le pas α_k satisfait la condition de la recherche linéaire de Wolfe (4.1.4), alors $\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty$.

Le théorème suivant présente le résultat de convergence globale de l'algorithme 6.

Théorème 4.2.1. [15] Supposons que l'ensemble $\Lambda = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ est borné; où $x_1 \in \mathbb{R}^n$ est le point initial et que sur un voisinage \mathcal{V} de Λ , la fonction objectif f est continûment différentiable et son gradient est lipschitzien. Soit $\{x_k\}$ généré par l'algorithme 6.

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

Preuve. Supposons, par contradiction, que la conclusion énoncée ne soit pas vraie. Alors, étant donné $\|g_k\| > 0$ il existe une constante $\gamma > 0$ tel que $\|g_k\|^2 \geq \gamma, \forall k$.

De (4.1.3) il s'ensuit que $d_k + g_k = \beta_k^{JHJ} d_{k-1}$ ceci avec le lemme (4.2.2) il implique que :

$$\|d_k\|^2 = \left(\beta_k^{JHJ}\right)^2 \|d_{k-1}\|^2 - 2g_k^T d_k - \|g_k\|^2 \leq \left(\frac{g_k^T d_k}{g_{k-1}^T d_{k-1}}\right)^2 \|d_{k-1}\|^2 - 2g_k^T d_k - \|g_k\|^2 \quad (4.2.4)$$

Divisons les deux côtés de (4.2.4) par $(g_k^T d_k)^2$ on obtient :

$$\begin{aligned} \frac{\|d_k\|^2}{(g_k^T d_k)^2} &\leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} - \frac{2}{g_k^T d_k} - \frac{\|g_k\|^2}{(g_k^T d_k)^2} \\ &= \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} - \left(\frac{1}{\|g_k\|} + \frac{\|g_k\|}{g_k^T d_k}\right)^2 + \frac{1}{\|g_k\|^2} \\ &\leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\|g_k\|^2} \end{aligned} \quad (4.2.5)$$

En combinant avec $\frac{\|d_1\|^2}{(g_1^T d_1)^2} = \frac{1}{\|g_1\|^2}$, par une récurrence de la relation (4.2.5) et $\|g_k\|^2 \geq \gamma$, on a :

$$\begin{aligned} \frac{\|d_k\|^2}{(g_k^T d_k)^2} &\leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\|g_k\|^2} \leq \frac{\|d_{k-2}\|^2}{(g_{k-2}^T d_{k-2})^2} + \frac{1}{\|g_{k-1}\|^2} + \frac{1}{\|g_k\|^2} \\ &\leq \dots \leq \sum_{i=1}^k \frac{1}{\|g_i\|^2} \leq \frac{k}{\gamma} \end{aligned} \quad (4.2.6)$$

Ainsi, $\frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \frac{\gamma}{k}$, cela montre encore que $\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = \infty$, ce qui contredit le lemme 4.2.3 par conséquent, le résultat souhaité est vérifié. \square

4.3 Méthodes hybrides du gradient conjugué non linéaire basées sur des combinaisons convexes

Les méthodes hybrides du gradient conjugué non linéaire basées sur des combinaisons convexes sont des méthodes itératifs de type (4.1.2)-(4.1.3), où le paramètre du gradient conjugué β_k est calculé de la manière suivante :

$$\beta_{k+1} = (1 - \theta_k) \beta_{k+1}^{M1} + \theta_k \beta_{k+1}^{M2} \quad (4.3.1)$$

où β_{k+1}^{M1} et β_{k+1}^{M2} sont les paramètres du gradient conjugué non linéaire que nous voulons hybrider et $0 \leq \theta_k \leq 1$ est un paramètre.

L'idée est de combiner de manière convexe les algorithmes classiques afin d'obtenir des algorithmes ayant de meilleures performances.

Le paramètre β_{k+1} dans la combinaison convexe est déterminé par la condition de conjugaison $y_k^T d_{k+1} = 0$ où notre direction de la recherche est choisie de la manière suivante :

$$d_{k+1} = -g_{k+1} + \left((1 - \theta_k) \beta_{k+1}^{M1} + \theta_k \beta_{k+1}^{M2} \right) d_k \quad (4.3.2)$$

Evidemment, au lieu de la condition de conjugaison classique $y_k^T d_{k+1} = 0$, il est tout à fait possible d'utiliser la condition de conjugaison de Dai et Liao.

$$y_k^T d_{k+1} = -ts_k^T g_{k+1} \quad (4.3.3)$$

où $t \geq 0$ est un paramètre scalaire. L'utilisation de la condition de conjugaison (4.3.3) dans la formule de d_{k+1} donné par (4.3.2), nous permet d'aboutir à

$$\theta_k = \frac{y_k^T g_{k+1} - ts_k^T g_{k+1} - \beta_{k+1}^{M1} (y_k^T d_k)}{\left(\beta_{k+1}^{M2} - \beta_{k+1}^{M1} \right) (y_k^T d_k)} \quad (4.3.4)$$

4.3.1 Méthode hybride du gradient conjugué non linéaire LS-DY

Cette méthode hybride du gradient conjugué non linéaire est basée sur une combinaison convexe de LS et DY. En effet on va prendre comme cas particulier dans la formule (4.3.1)

$$\beta_{k+1}^{M1} = \beta_{k+1}^{LS} = -\frac{y_k^T g_{k+1}}{d_k^T g_k} \quad \text{et} \quad \beta_{k+1}^{M2} = \beta_{k+1}^{DY} = \frac{g_{k+1}^T g_{k+1}}{d_k^T y_k} \quad (4.3.5)$$

En utilisant la condition de conjugaison de Dai et Liao (4.3.3) dans la formule de θ_k donnée par (4.3.4) où $\beta_{k+1}^{M1} = \beta_{k+1}^{LS}$ et $\beta_{k+1}^{M2} = \beta_{k+1}^{DY}$, on obtient

$$\theta_k^{LD} = \frac{(y_k^T g_{k+1})(d_k^T g_{k+1}) - t(s_k^T g_{k+1})(d_k^T g_k)}{\|g_{k+1}\|^2 (d_k^T g_k) + (y_k^T g_{k+1})(d_k^T y_k)} \quad (4.3.6)$$

4.3.2 Algorithme hybrides du gradient conjugué non linéaire sur des combinaisons convexes

Algorithm 7: Algorithme des méthodes du GCH basées sur combinaisons convexes.

Etape 1 : (initialisation) Soit x_1 un point initial et $\varepsilon \geq 0$ et $\varepsilon_b \geq 0$. calculer $f(x_0)$ et g_0

Définir $d_0 = -g_0$ et $\alpha_0 = \frac{1}{\|g_0\|}$, poser $k = 0$.

Etape 2 : Si $\|g_k\| \leq \varepsilon$ STOP. Si non aller à l'étape 3.

Etape 3 : Déterminer le pas α_k par une recherche linéaire de Wolfe.

Etape 4 : Générer l'itération suivante par $x_{k+1} = x_k + \alpha_k d_k$ calculer $f(x_{k+1})$ et g_{k+1}

calculer $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$

Etape 5 : Si $\left| \|g_{k+1}\|^2 (d_k^T g_k) + (y_k^T g_{k+1})(d_k^T y_k) \right| \leq \varepsilon_b$ alors poser $\theta_k = 0$ si non calculer θ_k par (4.3.6).

Etape 6 : Poser $\beta_{k+1}^{M1} = \beta_{k+1}^{LS}$ et $\beta_{k+1}^{M2} = \beta_{k+1}^{DY}$.

Etape 7 : Calculer $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$.

Etape 8 : Si $|g_{k+1}^T g_k| \geq 0.2 \|g_{k+1}\|^2$ est satisfait, alors $d_{k+1} = -g_{k+1}$.

Etape 9 : Calculer $\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|}$.

Etape 10 : Poser $k = k + 1$ et aller à l'étape 2.

4.3.3 Descente de la direction

Le théorème suivant montre que la direction de recherche d_k^{LD} générée par l'algorithme 7 satisfait à la condition de descente suffisante.

Théorème 4.3.1. [4] Soit $\{g_k\}$ et $\{d_k\}$ les suites générées par l'algorithme 7 avec la recherche linéaire de Wolfe forte. Alors, la direction d_k satisfait à la condition de descente suffisante

$$d_k^T g_k \leq -c \|g_k\|^2 \quad (4.3.7)$$

Pour tout $k \geq 0$ où $c = \frac{(1 - 1.2\sigma)}{(1 - \sigma)}$, $\sigma < 0.5$

Preuve. Supposons que le critère de redémarrage de Powell n'est pas vérifié, c'est-à-dire,

$$\left| g_{k+1}^T g_k \right| < 0.2 \|g_{k+1}\|^2 \quad (4.3.8)$$

La preuve est donnée par récurrence. Pour $k = 0$, $g_0^T d_0 = \|g_0\|^2$ Puisque $c < 1$ il suit que (4.3.7) est vérifiée. Maintenant, supposons que (4.3.7) est vérifiée pour un certain $k \geq 1$.

De la deuxième condition forte de Wolfe $\left| \nabla f(x_k + \alpha_k d_k)^T d_k \right| \leq -\sigma d_k^T g_k$

$$d_k^T y_k = d_k^T g_{k+1} - d_k^T g_k \geq -(1 - \sigma) d_k^T g_k \geq 0 \quad (4.3.9)$$

De (4.1.3)

$$d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2 + \beta_{k+1} d_k^T g_{k+1} \quad (4.3.10)$$

Alors, lorsque $\theta_k \geq 1$ comme vu à l'étape 6 de l'algorithme 7, il s'ensuit que $\beta_k = \beta_k^{DY}$. Par conséquent, à partir des relations (4.3.9) et (4.3.10) ci-dessus, il résulte que

$$d_{k+1}^T g_{k+1} \leq -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2}{d_k^T y_k} \left| d_k^T g_{k+1} \right| \leq -\frac{1 - 2\sigma}{1 - \sigma} \|g_{k+1}\|^2 \quad (4.3.11)$$

D'après l'étape 6 de l'algorithme 7, lorsque $\theta_k \leq 0$, $\beta_{k+1} = \beta_{k+1}^{LS}$. Par conséquent, à partir

la deuxième condition de la recherche linéaire de Wolfe forte et (4.3.9), il en résulte que

$$d_{k+1}^T g_{k+1} \leq -\|g_{k+1}\|^2 + \frac{|g_{k+1}^T y_k|}{|d_k^T g_k|} |d_k^T g_{k+1}| \leq -(1 - 1.2\sigma) \|g_{k+1}\|^2 \quad (4.3.12)$$

Enfin, lorsque $\theta_k \in (0, 1)$ comme indiqué à l'étape 6 de l'algorithme 7, β_k est donnée par (4.3.1). Remarquons que β_{k+1}^{DY} peut s'écrire comme $\beta_{k+1}^{DY} = \frac{d_{k+1}^T g_{k+1}}{d_k^T g_k}$.

En utilisant (4.3.9), les relations ci-dessus et la définition de β_{k+1}^{LS} il s'ensuit que

$$\begin{aligned} d_{k+1}^T g_{k+1} &\leq -\|g_{k+1}\|^2 + \left| \beta_{k+1}^{LS} \right| |d_k^T g_{k+1}| + \left| \beta_{k+1}^{DY} \right| |d_k^T g_{k+1}| \\ &\leq -\|g_{k+1}\|^2 + \sigma \left| \beta_{k+1}^{LS} \right| |d_k^T g_k| + \sigma \left| \beta_{k+1}^{DY} \right| |d_k^T g_k| \\ &= -\|g_{k+1}\|^2 + \sigma \left| \frac{g_{k+1}^T y_k}{d_k^T g_k} \right| |d_k^T g_k| + \sigma \left| \frac{d_{k+1}^T g_{k+1}}{d_k^T g_k} \right| |d_k^T g_k| \\ &= -\|g_{k+1}\|^2 + \sigma \frac{|g_{k+1}^T y_k|}{|d_k^T g_k|} |d_k^T g_k| + \sigma \frac{|d_{k+1}^T g_{k+1}|}{|d_k^T g_k|} |d_k^T g_k| \\ &= -\|g_{k+1}\|^2 + \sigma |g_{k+1}^T y_k| + \sigma |d_{k+1}^T g_{k+1}| \\ &\leq -\|g_{k+1}\|^2 + \sigma \|g_{k+1}\|^2 + \sigma |g_{k+1}^T g_k| + \sigma |d_{k+1}^T g_{k+1}| \\ &\leq -\|g_{k+1}\|^2 + 1.2\sigma \|g_{k+1}\|^2 + \sigma |d_{k+1}^T g_{k+1}|. \end{aligned} \quad (4.3.13)$$

Mais, d'après (4.3.13)

$$d_{k+1}^T g_{k+1} - \sigma |d_{k+1}^T g_{k+1}| \leq -(1 - 1.2\sigma) \|g_{k+1}\|^2$$

Puisque $\sigma < 0.5$; il s'ensuit qu'il y a toujours une constante v tel que

$$d_{k+1}^T g_{k+1} - \sigma |d_{k+1}^T g_{k+1}| = v \left(d_{k+1}^T g_{k+1} \right)$$

Donc,

$$d_{k+1}^T g_{k+1} \leq -q \|g_{k+1}\|^2 \quad (4.3.14)$$

Où $q = (1 - 1.2\sigma)/v$, $v = 1 + \sigma$ ou $1 - \sigma$.

En conclusion (4.3.11), (4.3.12) et (4.3.14) montrer que (4.3.7) est vérifié pour $k + 1$. \square

4.3.4 La convergence globale

Le théorème suivant prouve la convergence globale de l'algorithme 7.

Théorème 4.3.2. [4] supposons que l'ensemble $\Lambda = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ est borné où $x_1 \in \mathbb{R}^n$ est le point initial et que sur un voisinage \mathcal{V} de Λ , la fonction objectif f est continûment différentiable et son gradient est lipschitzien. Soit $\{g_k\}$ et $\{d_k\}$ les suites générées par l'algorithme 7 avec la recherche linéaire de Wolfe forte. Alors, $\liminf_{k \rightarrow \infty} \|g_k\| = 0$.

Preuve. Comme f est continûment différentiable et son gradient est lipschitzien, alors il existe une constante positive $\Gamma > 0$ tel que $\|g(x)\| \leq \Gamma$ pour tout $x \in \Lambda$. Remarquons que, d'après la proposition (1.2.3) $\alpha_k \geq \lambda$ où λ est une constante positive. supposons que $\liminf_{k \rightarrow \infty} \|g_k\| \neq 0$. Alors, il existe une constante positive $r > 0$ tel que pour tout k assez grand

$$\|g_k\| \geq r. \quad (4.3.15)$$

D'après la deuxième condition de Wolfe forte et de (4.3.7),

$$d_k^T y_k = d_k^T g_{k+1} - d_k^T g_k \geq -(1 - \sigma) d_k^T g_k \geq c(1 - \sigma) \|g_k\|^2 \quad (4.3.16)$$

De la continuité Lipschitzienne du gradient, il résulte que

$$\|y_k\| = \|g_{k+1} - g_k\| \leq L \|x_{k+1} - x_k\| \leq LD \quad (4.3.17)$$

Où $D = \max\{\|x - y\| : x, y \in S\}$ est le diamètre de l'ensemble de niveau Λ . Or, étant donné les inégalités ci-dessus, il en résulte que

$$\begin{aligned} |\beta_{k+1}| &= \left| (1 - \theta_k) \beta_{k+1}^{LS} + \theta_k \beta_{k+1}^{DY} \right| \\ &\leq \left| \beta_{k+1}^{LS} \right| + \left| \beta_{k+1}^{DY} \right| = \frac{|g_{k+1}^T y_k|}{|d_k^T g_k|} + \frac{\|g_{k+1}\|^2}{|d_k^T y_k|} \\ &\leq \frac{\|g_{k+1}\| \|y_k\|}{c \|g_k\|^2} + \frac{\|g_{k+1}\|^2}{c(1 - \sigma) \|g_k\|^2} \leq \frac{\Gamma LD}{cr^2} + \frac{\Gamma^2}{c(1 - \sigma)r^2} \equiv M \end{aligned}$$

Selon la sélection du paramètre β_{k+1} , à l'étape 6 de l'algorithme 7, lorsque $\theta_k \notin (0, 1)$ il est facile de voir que l'inégalité ci-dessus est aussi vérifiée.

Par conséquent,

$$\|d_{k+1}\| \leq \|g_{k+1}\| + |\beta_{k+1}| \|d_k\| = \|g_{k+1}\| + \frac{|\beta_{k+1}| \|s_k\|}{\alpha_k} \leq \Gamma + \frac{\text{MD}}{\lambda} \equiv E$$

Ce qui implique que

$$\sum_{k \geq 0} \frac{1}{\|d_k\|^2} = \infty \quad (4.3.18)$$

D'autre part, de (4.3.7) et (4.3.15) et de la condition de Zoutendijk

$$c^2 r^4 \sum_{k \geq 0} \frac{1}{\|d_k\|^2} \leq \sum_{k \geq 0} \frac{c^2 \|g_k\|^4}{\|d_k\|^2} \leq \sum_{k \geq 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \quad (4.3.19)$$

Ce qui contredit avec (4.3.18). Donc, (4.3.15) n'est pas vérifiée et donc $\liminf_{k \rightarrow \infty} \|g_k\| \neq 0$

□

Simulation numérique

Dans ce chapitre, on va étudier les performances des méthodes FR, DY et PRP et les méthodes hybrides du gradient conjugué non linéaire de projection (WYL, YWH et JHJ) et les méthodes hybrides du gradient conjugué non linéaire basées sur des combinaisons convexes (LS-DY) avec la recherche linéaire inexacte de Wolfe forte. Dans ce but, on va les comparer pour les fonctions test TRIDIA et DBVF.

L'algorithme de la recherche linéaire inexacte de Wolfe forte assure que la direction est de descente.

Les paramètres sont donnés : $\rho = 0.1$, et $\delta = 0.01$. Dans le cas où le dénominateur de la variante du gradient conjugué β_k est nul on redémarre en prenant la direction de la plus profonde pente.

Dans nos simulations numériques on a limité le nombre des itérations nécessaires dans l'algorithme de la recherche linéaire inexacte de Wolfe forte, et le pas initial est égal à un.

Les programmes sont écrits sous Scilab 6.1.0 et exécutés dans un ordinateur Lenovo i7 de vitesse 1.75 GHz et de RAM 8 Go et un système Windows dix.

5.1 Performances des méthodes hybrides du gradient conjugué non linéaire basées sur le concept de projection

5.1.1 La fonction TRIDIA (cute)

Cette fonction est numérotée 44 dans l'ensemble des fonctions testes proposée par (Andrei [[2], 2008])

$$f(x) = \gamma(\delta x_1 - 1)^2 + \sum_{i=2}^n (\alpha x_i - \beta x_{i-1})^2, \quad (5.1.1)$$

$$\alpha = 2, \beta = 1, \gamma = 1, \delta = 1 \text{ and } x_0 = [1, \dots, 1].$$

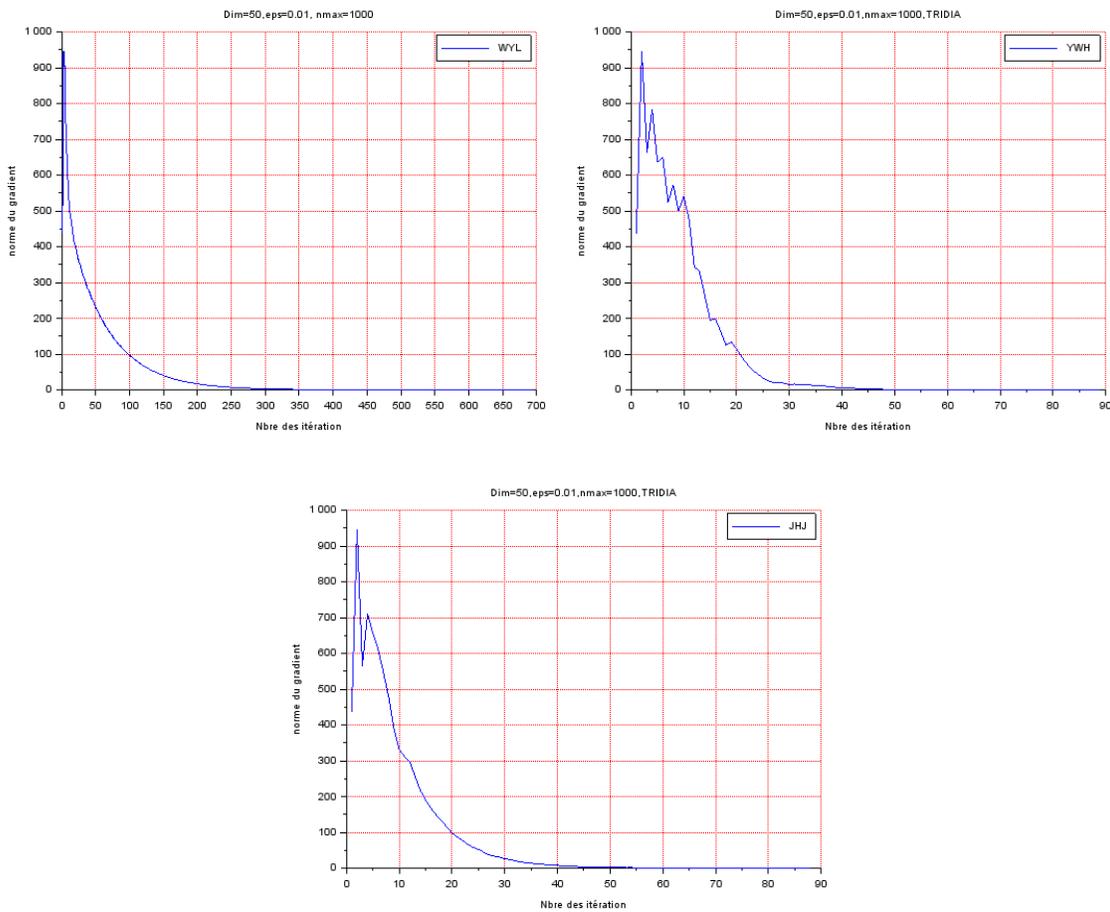


FIGURE 5.1 – Application des variantes de WYL, YWH et JHJ pour la fonction TRIDIA.

La figure (5.1) illustre la convergence des méthodes de WYL, YWH et JHJ respectivement appliquée à la fonction teste TRIDIA pour le cas où $n = 50$ et $\varepsilon = 0.01$. La convergence se fait avant 700 itérations pour la méthode de WYL et avant 90 itérations pour les méthodes de YWL et JHJ.

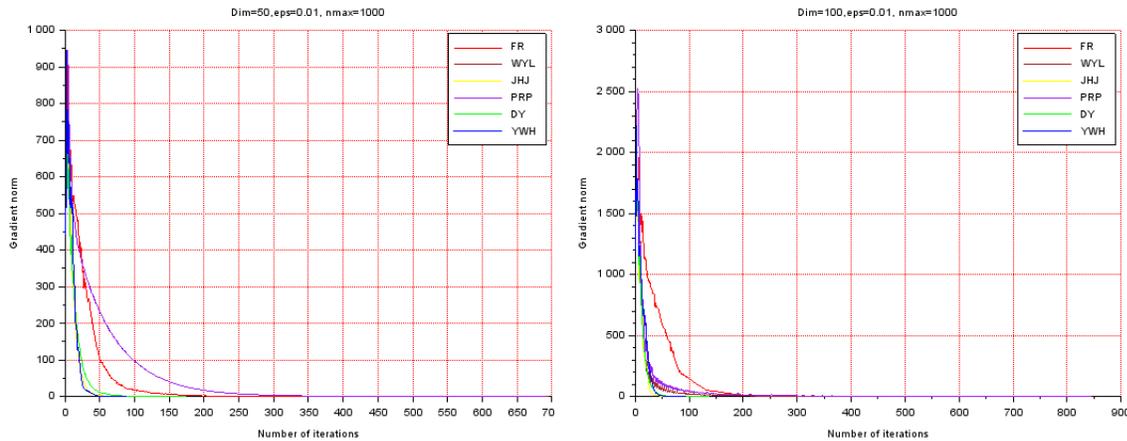


FIGURE 5.2 – Comparaison des méthodes de FR, WYL, PRP, DY YWH et JHJ pour la fonction TRIDIA.

La figure (5.2) représente une comparaison des méthodes de FR, WYL, PRP, DY, YWH et JHJ. On a exécuté ces méthodes pour la fonction TRIDIA, en prenant comme dimension $n = 50$ dans le graphe à gauche et $n = 100$ dans le graphe à droite, précision $\varepsilon = 0.01$ et un nombre maximale d’itérations $nmax = 1000$. On voit bien que dans ces cas la méthodes de JHJ converge en un nombre d’itérations inférieure à celles de FR, WYL, PRP, DY, YWH; c’est-à-dire qu’elle donne des meilleurs résultats de convergence que les autres méthodes.

5.1.2 La fonction DBVF (Discrete boundary value)

Cette fonction est numérotée 28 dans l’ensemble des fonctions testes proposées par (Moré and Hillstrom [[20], 1981])

$$f_i(x) = 2x_i - x_{i-1} - x_{i+1} + h^2(x_i + t_i + 1)^3 / 2 \tag{5.1.2}$$

$$h = 1/(n + 1), t_i = ih, x_0 = x_{n+1} = 0$$

$$\text{et } x_0 = (\xi_i) \text{ where } \xi_i = t_i(t_i - 1).$$

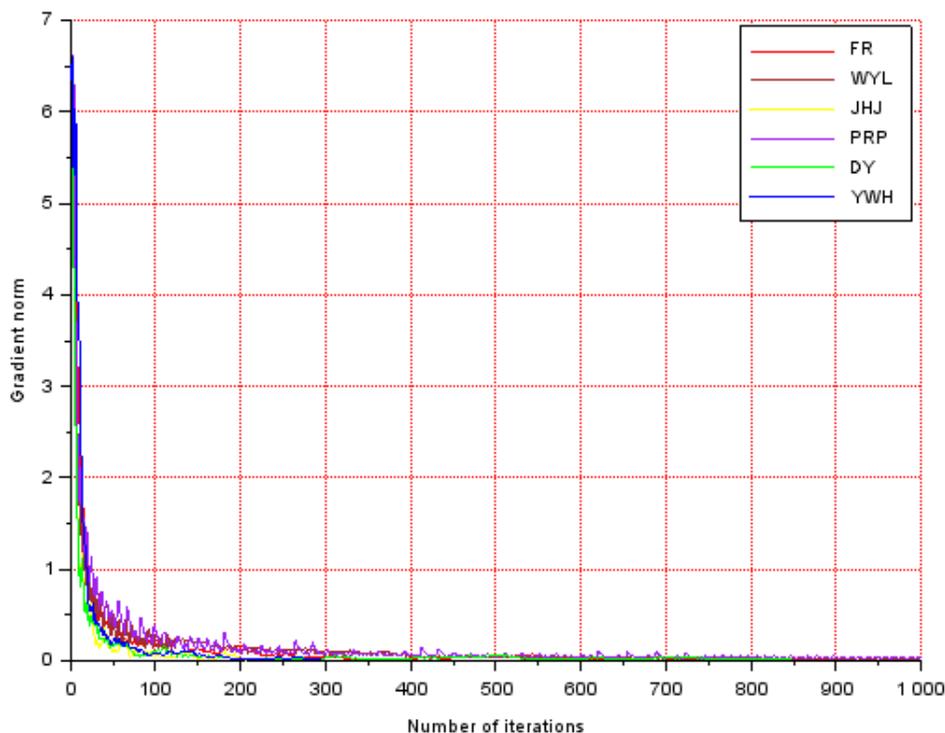


FIGURE 5.3 – Comparaison des méthodes de FR, WYL, PRP, DY, YWH et JHJ pour la fonction DBVF

La figure (5.3) représente une comparaison des méthodes de FR, WYL, PRP, DY, YWH et JHJ. On a exécuté ces méthodes pour la fonction DBVF, en prenant comme dimension $n = 50$, la précision $\varepsilon = 0.01$ et un nombre maximale d'itérations $n_{max} = 1000$. On voit bien que dans ce cas toutes les variantes divergent.

5.2 Performances des méthodes hybrides du gradient conjugué non linéaire basées sur sur des combinaisons convexes

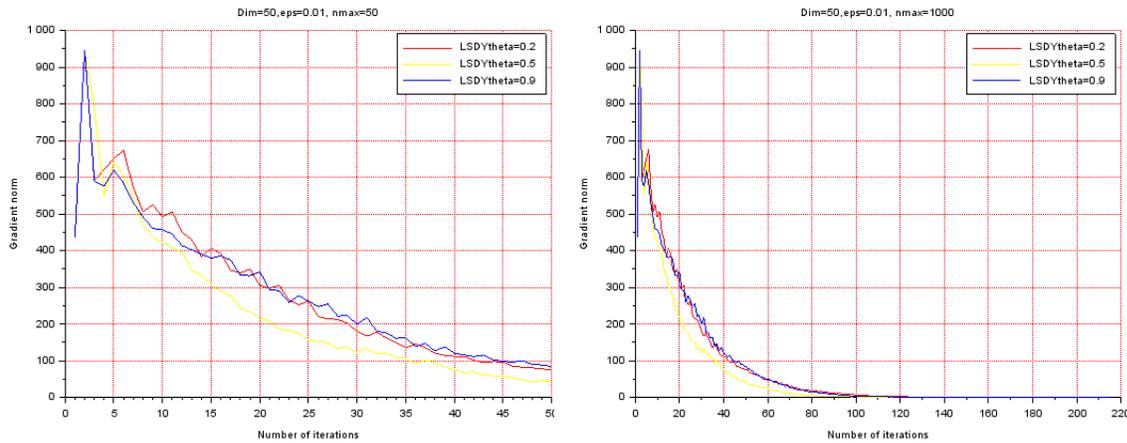


FIGURE 5.4 – Comparaison de la méthode LS-DY pour la fonction TRIDIA.

La figure (5.4) représente une comparaison de la méthode LS-DY. On a exécuté cette méthode pour la fonction TRIDIA, en prenant comme dimension $n = 50$, précision $\varepsilon = 0.01$ et des valeurs pour $\theta_k = 0.2, 0.5, 0.9$.

Dans le graphe à gauche on a limité le nombre maximale des itérations à $nmax = 50$ tandis que dans le graphe à droite on a limité le nombre maximale des itérations à $nmax = 1000$.

On voit bien que la méthode LS-DY donne des bons résultats de convergence. En effet la convergence se fait avant 110 itérations si $\theta_k = 0.5$ par contre la convergence se fait avant 130 et 220 itérations si $\theta_k = 0.2, 0.9$ respectivement.

Conclusion

Je pense et je réfléchis pendant des mois et des années. Quatre-vingt-dix-neuf fois, la conclusion est fausse. La centième fois, j'ai raison.
- Albert Einstein

Le travail effectué dans ce mémoire nous a permis d'étudier deux classes de méthodes hybrides du gradient conjugué non linéaire pour la résolution des problèmes d'optimisation sans contrainte. Il s'agit de deux méthodes hybrides, la première est basée sur le concept de projection et la deuxième est basée sur des combinaisons convexes.

Dans la première classe on a étudié trois variantes (WYL, YWH et JHJ) et comme cas particulier de la deuxième classe on a choisi la variante LS-DY. On a présenté leurs algorithmes ainsi que des théorèmes qui assurent leur convergence si le pas est déterminé par la recherche linéaire inexacte de Wolfe.

Afin de tester la performance des méthodes étudiées, nous avons mis en œuvre nos propres simulations numériques (SCILAB) qui montrent l'efficacité de celles-ci parmi les méthodes du gradient conjugué qui existent déjà.

Bibliographie

- [1] Mehiddin AL-BAALI. « Descent property and global convergence of the Fletcher—Reeves method with inexact line search ». In : *IMA Journal of Numerical Analysis* 5.1 (1985), p. 121-124.
 - [2] Neculai ANDREI. « An unconstrained optimization test functions collection ». In : *Adv. Model. Optim* 10.1 (2008), p. 147-161.
 - [3] Neculai ANDREI. *Another accelerated conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization*. Rapp. tech. ICI Technical Report, 2010.
 - [4] Neculai ANDREI. *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization*. Springer, 2020.
 - [5] Larry ARMIJO. « Minimization of functions having Lipschitz continuous first partial derivatives ». In : *Pacific Journal of mathematics* 16.1 (1966), p. 1-3.
 - [6] Stephen BOYD, Stephen P BOYD et Lieven VANDENBERGHE. *Convex optimization*. Cambridge university press, 2004.
 - [7] Augustin CAUCHY et al. « Méthode générale pour la résolution des systemes d'équations simultanées ». In : *Comp. Rend. Sci. Paris* 25.1847 (1847), p. 536-538.
 - [8] YH DAI et Y YUAN. « Some properties of a new conjugate gradient method ». In : *Advances in Nonlinear Programming*. Springer, 1998, p. 251-262.
 - [9] Yu-Hong DAI et Yaxiang YUAN. « A nonlinear conjugate gradient method with a strong global convergence property ». In : *SIAM Journal on optimization* 10.1 (1999), p. 177-182.
-

- [10] Imen ELADASSI. « Numerical study of some hybrid non linear conjugate gradient methods ». In : National Conference on Applied Mathematics et Didactics NCAMD2021, Constantine-Algeria. Ecole Normale Supérieure El Katiba Assia Djebbar Constantine Applied Mathematics et Didactics Laboratory, 2021.
- [11] Reeves FLETCHER et Colin M REEVES. « Function minimization by conjugate gradients ». In : *The computer journal* 7.2 (1964), p. 149-154.
- [12] JC GILBERT. « Éléments d'Optimisation Différentiable: Théorie et Algorithmes, notes de cours ». In : *École Nationale Supérieure de Techniques Avancées, Paris* (2007).
- [13] Robert L GRAVES et Philip WOLFE. « Recent advances in mathematical programming ». In : (1963).
- [14] Magnus R HESTENES et Eduard STIEFEL. « Methods of conjugate gradients for solving linear systems ». In : *Journal of Research of the National Bureau of Standards* 49.6 (1952), p. 409-436.
- [15] Jinbao JIAN, Lin HAN et Xianzhen JIANG. « A hybrid conjugate gradient method with descent property for unconstrained optimization ». In : *Applied Mathematical Modelling* 39.3-4 (2015), p. 1281-1290.
- [16] XZ JIANG, GD MA et JB JIAN. « A new global convergent conjugate gradient method with Wolfe line search ». In : *Chinese Journal of Engineering Mathematics* 28.6 (2011), p. 779-786.
- [17] David LUENBERGER. « G.(1969)“Optimization by Vector Space Methods,” » in : *New York: Wiley Luenberger Optimization by vector space methods 1969* (1969).
- [18] Romaïssa MELLAL. « Méthode du gradient conjugué et convergence uniforme des méthodes multigrilles ». Thèse de doct. Université Badji Mokhtar, 2015.
- [19] Michel MINOUX. *Programmation mathématique. Théorie et algorithmes*. Dunod, 1983.
- [20] Katia MOCELLIN. « Contribution à la simulation numérique tridimensionnelle du forgeage à chaud: étude du contact et calcul multigrille ». Thèse de doct. École Nationale Supérieure des Mines de Paris, 1999.

-
- [21] Jorge NOCEDAL et Stephen WRIGHT. *Numerical optimization*. Springer Science & Business Media, 2006.
- [22] E POLAK et G RIBIERE. « Note sur la convergence de directions conjuguées. Rev. Francaise Informat ». In : *Recherche Opertionelle, 3e année* 16 (1969), p. 35-43.
- [23] Boris Teodorovich POLYAK. « The conjugate gradient method in extremal problems ». In : *USSR Computational Mathematics and Mathematical Physics* 9.4 (1969), p. 94-112.
- [24] Yao SHENGWEI, Zengxin WEI et Hai HUANG. « A note about WYL's conjugate gradient method and its applications ». In : *Applied Mathematics and computation* 191.2 (2007), p. 381-388.
- [25] M Bazaraa HD Sherali CM SHETTY. *Nonlinear programming*. Wiley, 1993.
- [26] Zengxin WEI, Shengwei YAO et Liying LIU. « The convergence properties of some new conjugate gradient methods ». In : *Applied Mathematics and computation* 183.2 (2006), p. 1341-1350.
- [27] Philip WOLFE. « Convergence conditions for ascent methods ». In : *SIAM review* 11.2 (1969), p. 226-235.
- [28] G ZOUTENDIJK. « Nonlinear programming, computational methods ». In : *Integer and nonlinear programming* (1970), p. 37-86.