

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
Laboratoire de domiciliation : Laboratoire des mathématiques appliquées et de
Modélisation

THÈSE
EN VUE DE L'OBTENTION DU DIPLOME DE
DOCTORAT EN 3^{ème} CYCLE

Domaine : Mathématiques. Filière : Mathématiques Appliquées
Spécialité : Mathématiques Appliquées

Présentée par
HAFIDIA Imane

Intitulée

Quelle est la meilleure condition de conjugaison ?

Soutenue le : 17/11/2022

Devant le Jury composé de :

Nom et Prénom	Grade		
Mr AISSAOUI Mohamed-Zine	Prof	Univ-Guelma	Président
Mr GUEBBAI Hamza	Prof	Univ-Guelma	Encadreur
Mr BENRABIA Nouredine	MCA	Univ-Souk-Ahras	Co-encadreur
Mr HADJI Mohamed Lakhdar	Prof	Univ-Annaba	Examineur
Mr ELLAGOUNE Fateh	Prof	Univ-Guelma	Examineur

Année Universitaire : 2022/2023

Résumé

L'objectif de cette thèse est de traiter la problématique "Quelle est la meilleure condition de conjugaison ? Afin d'y apporter une réponse, même partielle. Dans ce contexte, nous proposons une nouvelle méthode en développant un nouvel algorithme de gradient conjugué pour la résolution de problèmes d'optimisation sans contraintes qui satisfait la condition de descente suffisante et la condition de conjugaison, et nous étudions également sa convergence globale. Pour déterminer l'efficacité de cette méthode, nous devons effectuer une comparaison numérique avec d'autres méthodes efficaces.

Mots clés : Optimisation sans contraintes, méthodes du gradient conjugué, recherche linéaire, condition de conjugaison et de descente suffisante

Abstracts

The aim of this thesis is to treat the problem "What is the best conjugacy condition?" in order to give it an answer, even if it is partial. In this context, we propose a new method by developing a new conjugate gradient algorithm to solve unconstrained optimization problems that satisfy the sufficient descent condition and the conjugacy condition, and we also study their global convergence. To determine the effectiveness of this method, we need to make a numerical comparison with other effective methods.

Keywords: Unconstrained optimization, conjugate gradient method, linear search, conjugacy and sufficient descent condition.

ملخص

الهدف من هذه الرسالة هو معالجة مشكلة "ما هو أفضل شرط اقتران؟ لإعطائها إجابة ، حتى لو كانت جزئية. في هذا السياق ، نقتراح طريقة جديدة من خلال تطوير خوارزمية مشاكل التحسين غير المقيدة التي تفي بشرط النسب الكافي وشرط تدرج مترافق جديدة لحل الاقتران ، كما ندرس التقارب الإجمالي لتحديد فعالية هذه الطريقة ، نحتاج إلى إجراء مقارنة رقمية مع الطرق الفعالة الأخرى.

الكلمات المفتاحية : التحسين غير المقيد ، طريقة التدرج المترافق ، البحث الخطي ، الاقتران وظروف النسب الكافية

Table des matières

1 Optimisation sans contrainte	9
1.1 Rappel de quelques définitions	10
1.2 Conditions d'optimalité pour l'optimisation sans contrainte	11
1.2.1 Le minimum local et le minimum global	11
1.2.2 Une Direction de descente	12
1.2.3 Direction de descente suffisante	12
1.2.4 Conditions nécessaires	12
1.2.5 Conditions suffisantes	13
1.2.6 Les modes de convergence	13
1.3 Règles de recherche linéaire	14
1.3.1 La recherche linéaire exacte	15
1.3.2 La recherche linéaire inexacte	16
1.4 Méthode du gradient conjugué	18
1.4.1 Méthode du gradient conjugué linéaire	19
1.4.2 Méthode du gradient conjugué non linéaire	21
1.4.3 Méthodes classiques de gradient conjugué	25
1.4.4 Méthodes hybrides de gradient conjugué	28
1.5 Conditions de conjugaison	30

2 Une nouvelle hybridation de gradient conjugué pour une optimisation sans contrainte	33
2.1 Introduction	33
2.2 Algorithme de gradient conjugué hybride	36
2.3 La convergence de l'algorithme	43
2.4 Résultats numériques	47
3 Une nouvelle méthode d'optimisation basée sur l'idée de A. Perry [30].	53
3.1 Introduction	53
3.2 Amélioration de la méthode du gradient conjugué.	57
3.3 L'accélération du nouvel algorithme de gradient conjugué	62
3.4 Résultat de la convergence globale	65
3.5 Résultats numériques	68
4 Conclusion et Perspectives	73

Introduction

L'optimisation mathématique est l'une des disciplines les plus étudiées en raison de son utilisation fréquente en sciences de l'ingénieur, en économie et en industrie en général, où l'objectif principal est de trouver la meilleure solution parmi toutes les solutions obtenues.

Notre travail consiste, principalement, à étudier et tester, quelques problèmes d'optimisation sans contraintes, dans lesquels les variables sont des nombres réels et les fonctions objectives sont des fonctions réelles à plusieurs variables.

Par conséquent, étant donné une certaine fonction objective $f : \mathbb{R}^n \rightarrow \mathbb{R}$, notre problème est de trouver $x \in \mathbb{R}^n$ qui minimise $f(x)$.

Au cours de ces dernières années, la résolution des problèmes d'optimisation sans contraintes connaît un intérêt croissant, à cause de l'application des variantes de cette méthode à des problèmes modélisant les phénomènes sociaux-économiques industriels environnementaux.

Notre étude concerne sur les méthodes du gradient conjugué non linéaires, qui sont largement utilisées dans nombreux problèmes pratiques d'optimisation de grande taille, en raison de la simplicité de ces méthodes particulièrement leurs faibles besoins de mémoire.

Plusieurs mathématiciens ont étendu les méthodes du gradient conjugué pour

le cas non linéaire. Ceci a été réalisé pour la première fois, en 1964 par Fletcher et Reeves [19] (méthode de Fletcher-Reeves), suivi en 1969 par Polak, Ribière [31] et Polyak [32] (méthode de Polak-Ribière-Polyak). Une autre variante a été étudiée en (1999) par Dai-Yuan [18] (La méthode de Dai-Yuan).

Dans un sens large, les algorithmes de gradient conjugué peuvent être classés en algorithmes standard, hybrides, algorithmes de gradient standard modifier, algorithmes de gradient conjugué à trois termes, et autres.

Un facteur important dans les algorithmes de gradient conjugué est l'amélioration essentiellement de l'accélération. Il existe plusieurs techniques utiles pour accroître l'accélération, en particulier les recherches linéaires inexacts d'Armijo et de Wolfe forte, ainsi que les conditions de conjugaison.

Nous pouvons formulée la problématique dans la question suivante "Quelle est la meilleure condition de conjugaison ?" qui permet d'accroître l'accélération.

Dans le but de répondre, partiellement à la problématique nous allons développer en premier temps un algorithme, qui permet l'amélioration de l'accélération selon un procédé comparative basé sur les conditions respectivement de conjugaison suivante

$$y_k^\top d_{k+1} = 0, \quad (0.1)$$

et de conjugaison généralisé

$$y_k^\top d_{k+1} = -\sigma s_k^\top g_{k+1}. \quad (0.2)$$

Notre travail est principalement partagé en trois chapitres précédés d'une introduction et se termine par une conclusion avec des perspectives.

Le premier chapitre est consacré à un rappel des concepts mathématiques relative à l'optimisation sans contrainte, où nous essayons de nous rappeler Les méthodes de gradient conjugué linéaire et non linéaires et les propriétés de convergence, en présentant les grandes lignes de plusieurs méthodes fondamentales du gradient conjugué non linéaire ainsi leurs propriétés.

le deuxième chapitre qui représente l'essentiel de notre contribution, est consacré à présenter notre nouvelle méthode de gradient conjugué, obtenue par la combinaison linéaire des méthodes de gradient conjugué de Dai-Yuan (DY) et de Hager-Zhang (HZ) en exploitant les avantages des deux méthodes, sachant que les conditions de descente suffisantes et de conjugaison sont vérifiées. En suite, pour prouver l'efficacité de cette méthode nous avons établi une comparaison numérique de notre méthode avec les méthodes de gradient conjugué standards, de plus ainsi qui avec une deuxième méthode hybride qu'on a mis au point vérifient les conditions de descente suffisantes et de conjugaison généralisées, et qui sera développé dans nos perspectives.

Le troisième chapitre est consacré à présenter une nouvelle méthode de gradient conjugué sans l'utilisation de la condition de conjugaison cette méthode est basée sur la technique de A. Perry [30] modifiée où nous montrons il y a toujours une amélioration de l'accélération.

Chapitre 1

Optimisation sans contrainte

L'optimisation sans contrainte consiste à minimiser une fonction qui dépend d'un nombre de variables réelles, sans aucune restriction sur les valeurs de ces variables. Lorsque le nombre de variables est grand, la résolution de ce genre de problème devient assez difficile, pour cela, on utilise les méthodes de gradient conjugué considérées fiables et efficaces. Ce sont des méthodes itératives, elles commencent par une estimation initiale des variables, puis elles génèrent une suite d'estimations améliorées, jusqu'à ce qu'elles se terminent par un ensemble de valeurs pour les variables. Pour vérifier que cet ensemble des valeurs, pour ces variables, est bien la solution du problème, nous devons utiliser les conditions d'optimalités. Si les conditions d'optimalités, ne sont pas satisfaites, elles peuvent être utilisées pour améliorer l'estimation de la solution. Les algorithmes exploitent les valeurs de la fonction de minimisation, de la première et éventuellement la deuxième dérivée de cette fonction. On cite parmi les méthodes d'optimisation sans contraintes classiques jugées importantes : l'algorithme de la plus forte pente, la méthode de Newton, de quasi-Newton, de quasi-Newton à mémoire limitée et l'algorithme à région

de confiance.

1.1 Rappel de quelques définitions

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

Soit \mathbb{R}^n l'espace vectoriel réel de dimension $n \in \mathbb{N}$. Les vecteurs sont toujours des vecteurs colonnes. La transposée de x est notée x^T .

Nous utilisons $\|x\|$ pour représenter la norme euclidienne, c'est-à-dire, pour $x \in \mathbb{R}^n$

$$\|x\| = \|x\|_2 = \sqrt{x^T x}.$$

Définition 1.1.1 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction différentiable, on appelle le gradient de f la fonction notée ∇f définie par, $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$,

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

Définition 1.1.2 Si ∇f est différentiable, on appelle le hessien de f est la fonction $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ définie comme

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

où $\partial_{x_i x_j}^2 f$ est la dérivée partielle de f par rapport à x_j , puis par rapport à x_i

Position du problème

Le problème d'optimisation sans contrainte consiste à minimiser une fonction f de n variables à valeur réelle.

Nous cherchons à trouver, un point x^* tel que

$$f(x^*) \leq f(x),$$

où x^* est appelé minimum local.

Ce problème sera formulé comme suit

$$(P) : \min_{x \in \mathbb{R}^n} f(x). \quad (1.3)$$

Le problème (1.3) avec f est définie pour tout x et sans aucune condition sur les variables appelé problème sans contrainte.

Par conséquence, la méthode de résolution de ce type de problème est de nature itérative, et consiste à construire une suite x_k à partir d'un point initial x_0 , et de démontrer sa convergence vers une solution dite optimale.

1.2 Conditions d'optimalité pour l'optimisation sans contrainte

Pour résoudre un problème d'optimisation de type (1.3) nous avons besoin de conditions d'optimalité représentant l'outil principal pour trouver la solution plus efficace de notre problème.

1.2.1 Le minimum local et le minimum global

Définition 1.2.1

1. Un point $\hat{x} \in \mathbb{R}^n$ est dit minimum local du problème (1.3) s'il existe un voisinage $V_\varepsilon(\hat{x})$ tel que,

$$f(\hat{x}) \leq f(x); \forall x \in V_\varepsilon(\hat{x}) \quad (1.4)$$

Dans le cas où, $f(\hat{x}) < f(x)$, le point $\hat{x} \in \mathbb{R}^n$ est dit minimum local strict.

2. Un point $\hat{x} \in \mathbb{R}^n$ est dit minimum global du problème (1.3) si

$$f(\hat{x}) \leq f(x) \quad ; \forall x \in \mathbb{R}^n \quad (1.5)$$

1.2.2 Une Direction de descente

Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ une fonction différentiable, et considérons x, d deux valeurs de \mathbb{R}^n , d est dite direction de descente en x si

$$\nabla f(x) \cdot d < 0,$$

où $\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)$, et $d = (d_1, \dots, d_n)$.

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

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

1.2.3 Direction de descente suffisante

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

$$g_k^T d_k \leq -c \|g_k\|^2, \quad \forall k, \quad c > 0,$$

avec $g_k = \nabla f(x_k)$.

1.2.4 Conditions nécessaires

Le résultat, ci-dessous, nous permet la caractérisation d'une solution optimale moyennant des conditions de différentiabilité de la fonction f du problème (1.3), le théorème suivant illustre les conditions nécessaires d'optimalité.

Théorème 1.2.1 *Soit f deux fois continûment différentiable et soit x^* un minimum local de f , alors*

$$\nabla f(x^*) = 0,$$

de plus

$$x^T \nabla^2 f(x^*) x \geq 0 \text{ (semi-défini positif).}$$

Remarque 1.2.1 *La condition $\nabla f(x^*) = 0$ est appelée la condition nécessaire de premier ordre et le point x^* est dit point stationnaire ou point critique.*

1.2.5 Conditions suffisantes

Par conséquent la condition $\nabla f(x^*) = 0$ n'est pas suffisant, on doit alors lui associer la positivité de la dérivée seconde pour confirmer la nature minimale du point x^* , le théorème suivant qui illustre les conditions dites suffisantes d'optimalité.

Théorème 1.2.2 *Soit f deux fois continûment différentiable au voisinage de x^* .*

Supposons que $\nabla f(x^) = 0$ et que $\nabla^2 f(x^*)$ est définie positive, alors x^* est un minimum local de f .*

1.2.6 Les modes de convergence

Définition 1.2.2 *Soit $\{x_k\}_{k \in \mathbb{N}}$ une suite dans \mathbb{R}^n qui converge vers x^* .*

1. *On dit que $\{x_k\}_{k \in \mathbb{N}}$ converge vers x^* linéairement avec α le taux si*

$$\limsup_{k \rightarrow \infty} \frac{\|x_k - x^*\|}{\|x_{k-1} - x^*\|} = \alpha < 1,$$

2. On dit que $\{x_k\}_{k \in \mathbb{N}}$ converge vers x^* superlinéaire si

$$\lim_{k \rightarrow \infty} \frac{\|x_k - x^*\|}{\|x_{k-1} - x^*\|} \implies 0.$$

3. La convergence est dite superlinéaire d'ordre γ si

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

En particulier si

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

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

1.3 Règles de recherche linéaire

En optimisation mathématique, la recherche linéaire est l'une des méthodes classiques permettant de traiter la convergence des algorithmes de calcul d'un minimum \hat{x} , d'une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Considérons le problème d'optimisation sans contraintes (P)

$$(P) : \quad \min\{f(x); x \in \mathbb{R}^n\} \tag{1.6}$$

où f est une fonction réelle à une variable.

Pour résoudre le problème d'optimisation (1.6) on fait appel aux algorithmes et les schémas s'écrivent de la manière suivante pour $x \in \mathbb{R}^n$

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

où, par conséquent λ_k appelé le pas est la solution optimale du problème d'optimisation unidimensionnel

$$\min_{\lambda > 0} f(x_k + \lambda d_k), \tag{1.8}$$

λ_k vérifie l'inégalité

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k), \quad \forall \lambda > 0. \quad (1.9)$$

où, x_k et d_k sont fixés et la fonction à minimiser notée φ_k , est une fonction d'une variable réelle, définie comme suit $\varphi_k : \mathbb{R} \rightarrow \mathbb{R}$,

$$\lambda \rightarrow \varphi_k(\lambda) = f(x_k + \lambda d_k) \quad (1.10)$$

Remarque 1.3.1 *Il faut noter que dans les problèmes d'optimisation sans contraintes, nous avons besoin de résoudre à chaque itération x_k , un problème d'optimisation dans \mathbb{R} .*

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

1. les recherches linéaires exactes.
2. les recherches linéaires inexactes.

1.3.1 La recherche linéaire exacte

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

$$\min_{\lambda > 0} \varphi_k(\lambda).$$

Ce procédé est appelé la règle de Cauchy et le pas déterminé par cette règle est appelé "pas de Cauchy" ou "pas optimal". Dans certains cas, on préfère

le plus petit point stationnaire de φ_k qui la rend décroissante.

On aura alors, le pas λ_k telle que la fonction :

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

Remarque 1.3.2

1. *Dans la plupart des algorithmes d'optimisation modernes, on ne fait jamais de recherche linéaire exacte, car trouver λ_k signifie qu'il va falloir calculer un grand nombre de fois la fonction φ_k et cela peut être dissuasif du point de vue du temps de calcul. En pratique, on cherche plutôt une valeur de \hat{x} qui assure une décroissance suffisante de f .*
2. *Ces deux règles ne sont utilisées que dans des cas particuliers, par exemple lorsque φ est quadratique, la solution de la recherche linéaire s'obtient d'une façon exacte et par un nombre fini d'itérations.*

1.3.2 La recherche linéaire inexacte

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

Règle d'Armijo [10]

La règle d'Armijo se base sur le choix d'un paramètre ρ avec

$$0 < \rho < 1$$

consiste à déterminer une valeur approchée de λ_k par la condition,

$$\varphi_k(\lambda) \leq \varphi_k(0) + \rho\lambda\varphi'_k(0).$$

L'inconvénient de cette méthode est de favoriser les valeurs trop petites, par conséquent, elle est rarement utilisée seule.

Règle de Goldstein [21]

C'est une méthode proposée en 1967 et qui est basée sur le choix de deux paramètres ρ et δ vérifiant,

$$0 < \rho < \delta < 1$$

elle détermine les valeurs approchées de λ_k en utilisant deux conditions,

$$\begin{cases} \varphi_k(\lambda) \leq \varphi_k(0) + \rho\lambda\varphi'_k(0) \\ \varphi_k(\lambda) \geq \varphi_k(0) + \delta\lambda\varphi'_k(0) \end{cases}$$

Règle de Wolfe [44]

C'est une méthode proposée en 1969 et qui est basée sur le choix de deux paramètres ρ et δ tel que

$$0 < \rho < \delta < 1$$

elle détermine les valeurs approchées de λ_k selon les deux conditions suivantes :

$$\varphi_k(\lambda) \leq \varphi_k(0) + \rho\lambda\varphi'_k(0) \tag{1.11}$$

$$\varphi'_k(\lambda) \geq \delta \varphi'_k(0) \quad (1.12)$$

Deux valeurs sont usuelles des deux paramètres sont $\rho = 0, 1$ et $\delta = 0, 7$.

Remarque 1.3.3

1. La règle de Wolfe fait appel au calcul de φ'_k , elle est donc en théorie plus coûteuse que la règle de Goldstein. Cependant dans de nombreuses applications, le calcul du gradient $\nabla f(x)$ représente un faible coût additionnel en comparaison du coût d'évaluation de $f(x)$, c'est pourquoi cette règle est très utilisée.
2. Pour certains algorithmes (par exemple le gradient conjugué non linéaire) il est parfois nécessaire d'avoir une condition plus restrictive que (1.12). Pour cela la deuxième condition (1.12) est remplacée par :

$$|\varphi'_k(\lambda)| \leq -\delta \varphi'_k(0). \quad (1.13)$$

On obtient alors les conditions de Wolfe fortes (1.11) et (1.13).

1.4 Méthode du gradient conjugué

La méthode du gradient conjugué, est l'une des méthodes les plus utiles pour l'optimisation de systèmes linéaires et non linéaires. Elle est généralement utilisée comme un algorithme itératif, et peut être utilisée aussi comme une méthode directe, qui donne une solution numérique. Généralement, la méthode de gradient conjugué, est utilisée dans les très grands systèmes difficile à résoudre à l'aide des méthodes directes.

1.4.1 Méthode du gradient conjugué linéaire

Considérons le problème de la minimisation d'une fonction quadratique $f : \mathbb{R}^n \rightarrow \mathbb{R}$ avec

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x, \quad (1.14)$$

où $A \in \mathbb{R}^{n \times n}$ est une matrice définie positive, symétrique et $b \in \mathbb{R}^n$.

ainsi on obtient en se référant à la symétrie de A que

$$\nabla f(x) = Ax - b$$

et

$$\nabla^2 f(x) = A.$$

Dans le cas où A est définie positive, la fonction f est strictement convexe et possède un minimum global unique au point x^* ($\nabla f(x^*) = 0$). En conséquence, la solution du problème de la minimisation (1.14) est également une solution du problème suivant trouver $x \in \mathbb{R}^n$ satisfaisant,

$$Ax = b. \quad (1.15)$$

La méthode du gradient conjugué a été introduite par Hestenes et Stiefel dans les années 1950 [26], comme méthode itérative pour résoudre les systèmes d'équations linéaires $Ax = b$, symétrique définie positive où $A \in \mathbb{R}^{n \times n}$. Elle peut également être appliquée à des systèmes plus généraux avec inversible, mais pas symétrique ou définie positive, après la transformation de (1.15) en $A^\top Ax = A^\top b$, où la méthode du gradient conjugué résout ces problèmes en au plus n itérations. Cette méthode est souvent appelée "méthode du gradient conjugué linéaire".

Le nom de la méthode vient du fait qu'elle produit une séquence de directions

$\{d_0, \dots, d_{n-1}\}$, telle que les directions sont conjuguées par rapport à la matrice A , c'est-à-dire

$$d_i A d_j = 0, \quad \text{pour tous } i \neq j. \quad (1.16)$$

Si l'on note $g_k = \nabla f(x_k)$, l'algorithme de la méthode du gradient conjugué pour les fonctions quadratiques prend la forme suivante :

Cet algorithme consiste à générer une suite itérative $\{x_k\}$ sous la forme

$$x_{k+1} = x_k + \alpha_k d_k, \quad (1.17)$$

où $d_k \in \mathbb{R}^n$ est une direction de recherche et $\alpha_k \in \mathbb{R}^+$ est le pas de recherche obtenu par une recherche linéaire exacte ou inexacte.

A chaque étape k , la direction d_k est obtenue de la combinaison linéaire du gradient en x_k et de la direction précédente d_{k-1}

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad (1.18)$$

où β_k est une suite des scalaires non nulle, $g_{k+1} = \nabla f(x_k)$.

Le choix des scalaires β_k assure la propriété de conjugaison (1.16), on en déduit alors

$$d_{k+1} A d_k = 0 \Rightarrow (-g_{k+1} + \beta_k d_k) A d_k = 0 \quad (1.19)$$

$$\Rightarrow -g_{k+1} A d_k + \beta_k d_k A d_k = 0 \quad (1.20)$$

$$\Rightarrow \beta_k = \frac{g_{k+1} A d_k}{d_k A d_k}, \quad k = 0, 1, \dots, n-1. \quad (1.21)$$

Dans le cas de la fonction quadratique f , il est facile de déterminer le pas α_k , comme solution analytique exacte pour le problème de minimisation unidimensionnel

$$f(x_{k+1}) = \min_{\alpha \in \mathbb{R}} f(x_k + \alpha d_k). \quad (1.22)$$

Ainsi α_k est une solution optimale de (1.22) si et seulement si α_k vérifie $f'(\alpha_k) = 0$. Or

$$f'(\alpha_k) = d_k^\top \nabla f(x_{k+1}) = 0$$

$$d_k^\top \nabla f(x_{k+1}) = d_k^\top (Ax_{k+1} - b) = 0.$$

où, $d_k = (d_1, \dots, d_n)$ le vecteur direction, $\nabla f = (\frac{\partial f}{\partial x_i})_{i=1, \dots, n}$.

Soit

$$d_k^\top (A(x_k + \alpha_k d_k) - b) = 0.$$

c'est à dire

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

avec $g_k = \nabla f(x_k) = Ax_k - b$.

1.4.2 Méthode du gradient conjugué non linéaire

Les algorithmes non linéaires de gradient conjugué présentent l'avantage significatif d'un faible stockage. Ils ont été généralisés par Fletcher, Reeves [19], Polak, Ribière [31], au problème non linéaire de la minimisation sans contrainte.

Pour résoudre le problème d'optimisation non linéaire sans contrainte

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1.23)$$

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

Tout algorithme non linéaire de gradient conjugué génère une suite $\{x_k\}$ de la forme

$$x_{k+1} = x_k + \alpha_k d_k, \quad (1.24)$$

où α_k est la longueur de pas obtenue par la recherche linéaire et d_k est la direction de recherche, calculée par

$$\begin{cases} d_1 = -g_1 \\ d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \geq 1, \end{cases} \quad (1.25)$$

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

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

Résultats de convergence généraux pour la méthode de gradient conjugué non linéaire

Un algorithme de gradient conjugué donné par (1.24) et (1.25), génère une suite $\{x_k\}$, notre souci est de déterminer les conditions sous lesquelles cette suite converge vers la solution x^* du problème (1.23). Puisque l'algorithme donné par (1.24) et (1.25) ne dépend que des paramètres β_k , il s'en suit que l'objectif est de chercher les valeurs de ces paramètres assurant la convergence de cet algorithme.

Une condition importante pour les méthodes d'optimisation basées sur la recherche linéaire, c'est que la direction de la recherche doit être une direction de descente, par conséquent elle doit vérifier les propriétés suivante :

$$g_k^\top d_k < 0, \quad (\text{propriété de descente}) \quad (1.26)$$

$$g_k^\top d_k \leq -c \|g_k\|^2, \quad ((\text{propriété de descente suffisante}).) \quad (1.27)$$

pour tous les $k = 0, 2, \dots, n - 1$ où $c > 0$ est une constante.

Les propriétés de convergence du gradient conjugué non linéaire, peuvent être étudiées en mesurant l'efficacité de la direction de recherche et de la

longueur du pas. La qualité d'une direction de recherche d_k peut être déterminée, en étudiant l'angle entre la direction de descente la plus pente $-g_k$ et la direction de la recherche d_k . Cet angle est donné par

$$\cos \theta_k = \frac{-g_k^\top d_k}{\|g_k\| \|d_k\|}, \quad (1.28)$$

Pour établir les résultats de convergence générale de toute méthode donnée par (1.24) et (1.25), on introduit les hypothèses suivantes concernant la fonction f .

1. L'ensemble $S = \{f(x) \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ est borné, c'est-à-dire qu'il existe une constante $B > 0$ pour que $\|x_k\| \leq B$ pour tout x dans l'ensemble S .
2. Dans un certain voisinage V de l'ensemble S , f est continument différentiable et son gradient est Lipschitz continu, il existe une constante $L > 0$ de sorte que

$$\|g(x) - g(y)\| \leq L\|x - y\|, \text{ pour tous } x, y \in N. \quad (1.29)$$

Selon les hypothèses 1 et 2 ci-dessus, le théorème suivant, dû à Zoutendijk (1970) et Wolfe (1969, 1971), est essentiel pour prouver les résultats de convergence globale des algorithmes d'optimisation sans contrainte dans le cas non linéaire, y compris le gradient conjugué.

Théorème 1.4.1 *Supposons que f est bornée dans \mathbb{R}^n et que f est continument différentiable dans un voisinage N de l'ensemble S . Supposons aussi que le gradient est Lipschitz continu, c'est-à-dire qu'il existe une constante $L > 0$ de sorte que (1.29) est satisfaite pour tout $x, y \in N$. Considérons toute itération de la forme (1.24), où d_k est une direction de descente et α_k*

satisfait aux conditions de recherche linéaire de Wolfe (1.11) et (1.12). Alors,

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|g_k\|^2 < \infty. \quad (1.30)$$

Démonstration 1 D'après (3.5), il s'en suit que

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

D'autre part, la continuité de Lipschitz (1.14) donne le résultat

$$(g_{k+1} - g_k)^\top d_k \leq \alpha_k L \|d_k\|^2.$$

Par conséquent, la combinaison de ces deux relations donne

$$\alpha_k \geq \frac{(\sigma - 1) g_k^\top d_k}{L \|d_k\|^2}. \quad (1.31)$$

Maintenant, en utilisant la première condition de Wolfe (3.4) et (1.31), il en résulte que

$$f_{k+1} \leq f_k + \rho \frac{(\sigma - 1) (g_k^\top d_k)^2}{L \|d_k\|^2}. \quad (1.32)$$

D'après la définition (1.28) de $\cos \theta_k$, il s'en suit que (1.32) peut s'écrire comme suite

$$f_{k+1} \leq f_k + c \cos^2 \theta_k \|g_k\|^2. \quad (1.33)$$

où $c = \rho(\sigma - 1)/L$. En sommant (1.33) pour $k \geq 1$ et en considérant que f est bornée en dessous, on obtient (1.30).

La relation (1.30) est appelée condition de Zoutendijk. cette relation peut être écrite d'après (1.28), sous la forme

$$\sum_{k=1}^{\infty} \frac{(g_k^\top d_k)^2}{\|d_k\|^2} < \infty. \quad (1.34)$$

1.4.3 Méthodes classiques de gradient conjugué

1. La méthode de Fletcher-Reeves

C'est une méthode du gradient conjugué notée méthode FR introduite par Fletcher-Reeves et développée dans la référence où le coefficient β_k , noté β_k^{FR} est donné comme quotient de la norme par,

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad (1.35)$$

En 1970, Zoutendijk [47] a montré que la méthode FR associée à la recherche linéaire exacte est globalement convergente. Powell [33] a démontré que la méthode FR était faible par rapport aux résultats de Zoutendijk, à cause de sa sensibilité envers le comportement de "brouillage". Cela veut dire que si la méthode génère une mauvaise direction de recherche, qui est presque orthogonale au gradient, il est probable que le pas soit minuscule et que la direction suivante soit également mauvaise. Al-Baali [2] a donné une extension de ce résultat à la recherche linéaire inexactes de Wolfe fortes (1.11), (1.12) avec $\sigma < 0.5$.

2. La méthode de Polak-Ribière

Un autre choix du paramètre β_k a été proposé de manière indépendante par Polak et Ribière [31] et Polyak [32] en 1969, où β_k , noté β_k^{PR} , donné par :

$$\beta_k^{PR} = \frac{g_k^\top y_{k-1}}{\|g_{k-1}\|^2}, \quad (1.36)$$

Si f est fortement convexe, et la recherche linéaire est exacte, de plus β_k est donnée par (1.36) alors, Polak et Ribière et Polyak ont établi la convergence globale de la méthode Polak-Ribière-Polyak note PRP.

Powell (1977) a prouvé que pour une fonction générale non linéaire f , si la taille du pas $s_k = x_{k+1} - x_k$ tend vers zéro, (où la recherche linéaire est exacte, et la continuité de Lipschitzien), alors la méthode PRP est globalement convergente. D'autre part, Powell (1984a), dans un article laborieux, a construit un contre-exemple à trois variables et a montré que la méthode PRP peut se dérouler à l'infini, sans approcher aucune solution. Par conséquent, l'hypothèse selon laquelle la taille du pas tend vers le zéro est nécessaire pour la convergence. Plus tard, en supposant que la direction de recherche est une direction de descente, Yuan (1993) a établi la convergence globale de la méthode PRP pour les fonctions objectives fortement convexes associées à la recherche linéaire de Wolfe. Cependant, Dai (1997), dans sa thèse de doctorat, a présenté un exemple qui montre que même lorsque la fonction objective est fortement convexe et que $\sigma \in]0, 1[$ est suffisamment petit, la méthode PRP peut encore échouer en générant une direction de recherche ascendante. Dai, Han, Liu, Sun, Yin et Yuan (1999) ont présenté un exemple montrant que la délimitation de l'ensemble de niveaux est nécessaire pour la convergence de la méthode PRP même si la recherche linéaire est exacte. Par conséquent, la convergence de la méthode PRP n'est pas certaine. Cependant, cette méthode s'est avérée être l'une des méthodes les plus efficaces pour résoudre les problèmes d'optimisation sans contrainte à grande échelle.

3. La méthode de Hestenes-Steifel

Dans la méthode originale proposée par Hestenes et Steifel [26], pour résoudre des systèmes d'équations linéaires, le coefficient β_k , noté β_k^{HS} ,

était donné par la formule suivante :

$$\beta_k^{HS} = \frac{g_k^\top y_{k-1}}{d_{k-1}^\top y_{k-1}}, \quad y_{k-1} = g_k - g_{k-1}. \quad (1.37)$$

La méthode HS possède la propriété de conjugaison suivante

$$d_{k+1}^\top y_k = 0$$

indépendamment de la recherche linéaire.

Pour la recherche linéaire exacte, $\beta_k^{HS} = \beta_k^{PR}$. La méthode HS possède donc des propriétés similaires à celles de la méthode PR. En particulier, le contre-exemple de convergence de Powell [34] est également valable pour la méthode HS. et la modification suivante est utilisée pour assurer la convergence globale.

4. La méthode de Liu-Storey

Cette méthode a été proposée par Liu-Storey [29], dont ils ont présenté la formule suivante où le scalaire β_k , noté β_k^{LS} , s'écrit

$$\beta_k^{LS} = \frac{g_{k+1}^\top y_k}{-d_k^\top g_k}, \quad (1.38)$$

Pour la recherche linéaire exactes, la méthode LS est identique à la méthode PRP. Liu et Storey ont étudié cette méthode, en prouvant sa convergence globale.

5. La méthode de Dai-Yuan

Cette méthode, notée DY, est élaborée par Dai et Yuan [18], où le paramètre de gradient conjugué β_k , noté β_k^{DY} , est calculé comme suit

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^\top y_k}, \quad (1.39)$$

Avec une recherche linéaire de Wolfe standard (1.11), (1.12), la méthode DY vérifie toujours la propriété de descente. De plus, lorsque l'hypothèse de Lipschitz est vérifiée, alors la méthode DY est globalement convergente.

6. La méthode de Hager-Zhang

Récemment, Hager et Zhang ont proposé [25] une nouvelle méthode avec $\beta_k = \max \{ \beta_k^N, \eta_k \}$, où,

$$\eta_k = \frac{1}{\|d_k\| \min\{\eta, \|g_k\|\}},$$

et

$$\beta_k^N = \left(y_k - 2 \frac{\|y_k\|^2 d_k}{d_k^\top y_k} \right)^\top \frac{g_k}{d_k^\top y_k},$$

avec, $\eta > 0$ une constante, est globalement convergent avec la recherche linéaire de Wolfe (1.11), (1.12). il ont montré une amélioration importante des performances numériques.

1.4.4 Méthodes hybrides de gradient conjugué

Des méthodes hybrides ont été proposées en combinant des méthodes qui existaient déjà afin d'atteindre les meilleures caractéristiques et minimiser l'influence des inconvénients.

Les méthodes du gradient conjugué standards peuvent être combinés de deux manières distinctes, en obtenant ainsi deux classes des méthodes hybrides à gradient conjugué.

La première classe, se sont des méthodes de gradient conjugué hybrides basées sur le concept de projection qui se caractérisent généralement par leur expressions algébrique simples. Dès qu'une méthode de gradient conjugué

entre dans une phase de brouillage, alors le schéma hybride déclenche une autre méthode de gradient conjugué, en essayant ainsi d'améliorer les performances numériques de l'hybridation. Par exemple

Touati-Ahmed et Storey [41] ont proposé la méthode hybride suivante :

$$\beta_k = \begin{cases} \beta_k^{PR} & \text{si } 0 \leq \beta_k^{PR} \leq \beta_k^{FR}, \\ \beta_k^{FR} & \text{sinon.} \end{cases}$$

Et Nocedal et Gilbert [20] ont proposé

$$\beta_k = \{-\beta_k^{FR}, \min\{\beta_k^{FR}, \beta_k^{PR}\}\}$$

par conséquent, prouvé, que toute méthode contenant $\|\beta_k\| < \beta_k^{FR}$ est globalement convergente avec la recherche linéaire forte de Wolfe (1.11), (1.12). Les résultats numériques par Dai annoncent que ces combinaisons ne sont pas mieux que ceux de la méthode PR.

La deuxième classe, se sont des méthodes hybrides de gradient conjugué basées sur la combinaison convexe des méthodes standards. Cela veut dire que les méthodes de gradient conjugué standards sont combinées de manière convexe, ce qui permet d'obtenir une méthode de gradient conjugué hybride. En général, les méthodes hybrides sont plus efficaces et plus robustes que les méthodes standards. Exemple N. Andrei a proposé une nouvelle méthode notée CCOMB, qui représente une combinaison convexe des deux méthodes du gradient conjugué (PRP) et (DY). Afin de profiter les propriétés de calcul de la méthode de PRP et les propriétés de convergence de la méthode de DY. Dont, le scalaire β_k prend la valeur suivante :

$$\beta_k^N = (1 - \theta_k) \beta_k^{PRP} + \theta_k \beta_k^{DY} = (1 - \theta_k) \frac{g_{k+1}^T y_k}{g_k^T g_k} + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} \quad (1.40)$$

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

$$\theta_k = \frac{(y_k^T g_{k+1})(y_k^T s_k) - (y_k^T g_{k+1})(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T s_k) - (g_{k+1}^T g_{k+1})(g_k^T g_k)} \quad (1.41)$$

obtenu en utilisant la condition de conjugaison : $y_k^T d_{k+1} = 0$.

N. Andrei a prouvé que les directions générées par cette méthode sont des directions de descente, satisfont certaines conditions, sous la condition de descente suffisante et il a démontré que cette méthode est convergente.

1.5 Conditions de conjugaison

Dans les fonctions quadratiques, il est bien connu que les méthodes de gradient conjugué linéaire génèrent une suite de directions de recherche d_k . alors la condition de conjugaison est

$$d_i^T A d_j = 0 \quad (1.42)$$

cette condition est satisfaite pour tous $i \neq j$. où A est le hessien de la fonction objectif

Dans les fonctions non linéaires générales, par le théorème de la valeur moyenne, il existe un paramètre $\nu \in [0, 1]$

$$d_{k+1}^T g_{k+1} = d_{k+1}^T g_k + \lambda_k d_{k+1}^T \nabla^2 f(x_k + \nu \lambda_k d_k) d_k \quad (1.43)$$

nous avons $y_k = g_{k+1} - g_k$, nous obtenons

$$d_{k+1}^T y_k = \lambda_k d_{k+1}^T \nabla^2 f(x_k + \nu \lambda_k d_k) d_k \quad (1.44)$$

Par conséquent, pour l'optimisation non linéaire, il est raisonnable de remplacer la condition de conjugaison du cas linéaire par la condition suivante

$$d_{k+1}^T y_k = 0 \quad (1.45)$$

Pour accélérer l'algorithme du gradient conjugué, Perry (voire également Shanno) a prolongé la condition de conjugaison en introduisant l'information du second degré, en effet, il a employé :

$$H_{k+1}y_k = s_k, \quad s_k = x_{k+1} - x_k$$

où H_k est une approximation de l'inverse de la matrice hessienne, symétrique et définie positive, .

Puisque pour la méthode quasi-newtonienne, la direction d_{k+1} est calculée par

$$d_{k+1} = H_{k+1}g_{k+1},$$

donc,

$$d_{k+1}^T y_k = (-H_{k+1}g_{k+1}) y_k = -g_{k+1}^T (H_{k+1}y_k) = -g_{k+1}^T s_k, \quad (1.46)$$

Ainsi on obtient (1.46) la nouvelle condition de conjugaison, Récemment, Dai et Liao ont prolongé cette condition, ils ont donné une nouvelle condition de conjugaison d'après

$$d_{k+1}^T y_k = (-H_{k+1}g_{k+1}) y_k = -g_{k+1}^T (H_{k+1}y_k) = -g_{k+1}^T s_k,$$

on obtient

$$d_{k+1}^T y_k = -\sigma g_{k+1}^T s_k \quad (1.47)$$

où, σ est un scalaire positif.

Chapitre 2

Une nouvelle hybridation de gradient conjugué pour une optimisation sans contrainte

2.1 Introduction

Les méthodes de gradient conjugué non linéaire sont des méthodes bien connues et pratiques, pour la minimisation des problèmes d'optimisation sans contraintes, de la forme.

$$\min_{x \in \mathbb{R}^n} f(x), \quad (2.1)$$

avec f une fonction continue différentiable.

la méthode du gradient conjugué génère, une suite itérative $\{x_k\}$ où

$$x_{k+1} = x_k + \alpha_k d_k, \quad (2.2)$$

sachant que α_k sont obtenus par la recherche linéaire de wolfe,

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^\top d_k, \quad (2.3)$$

et

$$\sigma g_k^\top d_k \leq g(x_k + \alpha_k d_k)^\top d_k \leq -\sigma g_k^\top d_k, \quad (2.4)$$

avec $0 < \delta < \frac{1}{2}$ et $\delta < \sigma < 1$.

les directions d_k sont données par,

$$\begin{cases} d_1 = -g_1 \\ d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \geq 1, \end{cases} \quad (2.5)$$

où $g_k = g(x_k) = \nabla f(x_k)$, et β_k est un paramètre de gradient conjugué.

Nous rappelons que les Différents algorithmes du gradient conjugué sont associé aux différentes valeurs du paramètre β_k , en particulier pour $y_k = g_{k+1} - g_k$,

Gradient conjugué variante Hestenes-Stiefel (HS)

$$\beta_k^{HS} = \frac{g_{k+1}^\top y_k}{d_k^\top y_k},$$

Gradient conjugué variante Fletcher Reeves (FR)

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

Gradient conjugué variante Polak-Ribière-Polyak (PRP)

$$\beta_k^{PRP} = \frac{g_{k+1}^\top y_k}{\|g_k\|^2},$$

Gradient conjugué variante de Dai-Yuan (DY)

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^\top y_k},$$

Gradient conjugué variante Hager-Zhang (HZ)

$$\beta_k^{HZ} = \frac{g_{k+1}^\top y_k}{d_k^\top y_k} - 2\|y_k\|^2 \frac{d_k^\top g_{k+1}}{(d_k^\top y_k)^2},$$

qui sont proposés respectivement en [26] [19] [31] [18] et [25].

Afin d'assurer la convergence globale d'une méthode de gradient conjugué,

la propriété de descente suffisante ci-dessous est importante

$$d_{k+1}^\top g_{k+1} \leq -c \|g_{k+1}\|, \quad c > 0, \quad (2.6)$$

Pour une fonction générale, le résultat de convergence dépend du choix du paramètre β_k et de la technique de la recherche linéaire. Le premier résultat de l'essai pratique de convergence globale concernant la méthode FR, obtenue par [2], en se basant sur la vérification de la propriété de descente suffisante, pour une certaine valeur de c , en vérifiant les conditions de Wolfe fortes ci-dessus avec $\sigma < 1/2$.

Cependant, le résultat de convergence de la méthode de Dai-Yuan (DY) nécessite les conditions de Wolfe données par (3.4) et la partie gauche de l'inégalité (3.5), alors que pour la méthode de Hager-Zhang (HZ), la convergence est assurée pour toute technique de recherche linéaire.

Il s'avens en pratique, que les méthodes DY et HZ, permettent d'obtenir de résultat meilleurs que de nombreuses méthodes classique de gradient conjugué (pour plus de détails, voir par exemple [25]). Par conséquent, nous envisageons la possibilité de combiner les meilleures caractéristiques des deux méthodes DY et HZ en définissant la direction de recherche comme une combinaison linéaire de la forme

$$d_{k+1} = \theta_k d_{k+1}^{DY} + \vartheta_k d_{k+1}^{HZ}, \quad (2.7)$$

avec un choix judicieux des paramètres θ_k et ϑ_k .

Ces paramètres θ_k et ϑ_k sont calculé dans la section 2 ou la condition (2.6) est vérifiée sous le signe de l'égalité, en plus de la condition de conjugaison $y_k^\top d_{k+1} = 0$. Nous nous concentrons ici, sur la combinaison linéaire (2.7),

qui peut être étendue à d'autres méthodes de gradient conjugué, nos résultats principaux dans la section 3 est que notre nouvelle méthode converge globalement si les conditions de Wolfe sont vérifiées. Dans la section 4, nous présentons les résultats numériques qui montrent que notre nouvelle méthode est plus performante que les méthodes DY et HZ.

2.2 Algorithme de gradient conjugué hybride

Nous proposons, dans cette partie, une méthode du gradient conjugué hybride, en définissant la direction de recherche par la combinaison linéaire (2.7), qui peut s'écrire comme suit

$$d_{k+1} = -(\theta_k + \vartheta_k)g_{k+1} + (\theta_k\beta_k^{DY} + \vartheta_k\beta_k^{HZ})d_k, \quad (2.8)$$

où les paramètres ϑ_k et θ_k sont des scalaires, que nous choisissons de telle sorte que la direction de la recherche d_k , satisfait la condition de descente suffisante et la condition de conjugaison, donnés respectivement par

$$d_{k+1}^\top g_{k+1} = -c\|g_{k+1}\|^2, \quad c > 0 \quad (2.9)$$

et

$$y_k^\top d_{k+1} = 0. \quad (2.10)$$

En multipliant (2.8) par g_{k+1}^\top et y_k^\top et en utilisant (2.9) et (2.10), nous obtenons respectivement

$$(\|g_{k+1}\|^2 - \beta_k^{DY} d_k^\top g_{k+1})\theta_k + (\|g_{k+1}\|^2 - \beta_k^{HZ} d_k^\top g_{k+1})\vartheta_k = c\|g_{k+1}\|^2 \quad (2.11)$$

et

$$-\vartheta_k g_{k+1}^\top y_k - \theta_k g_{k+1}^\top y_k + \vartheta_k \beta_k^{HZ} (d_k^\top y_k) + \theta_k \beta_k^{DY} (d_k^\top y_k) = 0. \quad (2.12)$$

Pour voir toutes les solutions possibles de ce système de deux équations linéaires, nous traitons d'abord l'équation (2.12), ainsi, en utilisant les définitions de β_k^{DY} et β_k^{HZ} , nous obtenons

$$d_k^\top y_k g_k^\top g_{k+1} \theta_k - 2 \|y_k\|^2 d_k^\top g_{k+1} \vartheta_k = 0. \quad (2.13)$$

Ensuite nous allons résoudre le système (2.53) et (2.54) en se basant sur les quatre cas possibles suivants :

i) Si les deux valeurs de $d_k^\top g_{k+1} = 0$ et $g_k^\top g_{k+1} = 0$, sont vérifiées et si la recherche linéaire est exacte en plus si la fonction f est quadratique, alors l'équation (2.54) est vérifiée pour toute valeur de θ_k et de ϑ_k qui satisfont (2.53) avec $c = \theta_k + \vartheta_k$ (par exemple, $\theta_k = 0$ et $\vartheta_k = c = 1$).

ii) Si $d_k^\top g_{k+1} = 0$ et $g_k^\top g_{k+1} \neq 0$, (2.54) implique que $\theta_k = 0$ et donc de (2.53) on vérifie que $\vartheta_k = c$ (si $c = 1$, on revient à l'étape (i)).

iii) Si $d_k^\top g_{k+1} \neq 0$ et $g_k^\top g_{k+1} = 0$, (2.54) implique que $\vartheta_k = 0$ et donc de (2.53), il s'ensuit que $\theta_k = \tilde{\theta}_k$, où

$$\tilde{\theta}_k = -c \frac{d_k^\top y_k}{d_k^\top g_k} \quad (2.14)$$

iv) Pour le cas qui reste, $d_k^\top g_{k+1} \neq 0$ et $g_k^\top g_{k+1} \neq 0$, nous résolvons le système pour obtenir $\theta_k = \hat{\theta}_k$ et $\vartheta_k = \hat{\vartheta}_k$, où

$$\hat{\theta}_k = \frac{c \|g_{k+1}\|^2 (y_k^\top g_{k+1} - \beta_k^{HZ} d_k^\top y_k)}{\Delta_k (\beta_k^{DY} - \beta_k^{HZ})}, \quad (2.15)$$

$$\hat{\vartheta}_k = \frac{-c \|g_{k+1}\|^2 (y_k^\top g_{k+1} - \beta_k^{DY} d_k^\top y_k)}{\Delta_k (\beta_k^{DY} - \beta_k^{HZ})}, \quad (2.16)$$

avec

$$\Delta_k = \|g_{k+1}\|^2 (d_k^\top y_k) - (d_k^\top g_{k+1})(y_k^\top g_{k+1}). \quad (2.17)$$

La solution précédente existe si

$$\Delta_k(\beta_k^{DY} - \beta_k^{HZ}) \neq 0 \quad (2.18)$$

Cependant, si le déterminant (2.18) est nul, le système (2.53) et (2.54) n'a pas de solution, dans ce cas, nous appliquons uniquement la condition de descente suffisante (2.53), en choisissant, comme dans le cas (iii), $\vartheta_k = 0$ et $\theta_k = \tilde{\theta}_k = -c \frac{d_k^T y_k}{d_k^T g_k}$.

Nous résumons maintenant l'analyse précédente dans le théorème suivant.

Théorème 2.2.1 *Pour θ_k et ϑ_k vérifiant*

$$(\theta_k, \vartheta_k) = \begin{cases} (\hat{\theta}_k, \hat{\vartheta}_k) & \text{si ; } (\beta_k^{DY} - \beta_k^{HZ})\Delta_k \neq 0, \\ (0, c) & \text{si ; } d_k^T g_{k+1} = 0, \\ (\tilde{\theta}_k, 0) & \text{autrement} \end{cases} \quad (2.19)$$

et supposant que les conditions de recherche linéaire de Wolfe sont vérifiées. Alors il existe $c > 0$ telle que la condition de descente suffisante (2.9) est satisfaite. De plus, la condition de conjugaison (2.10) est assurée, sauf quand $\Delta_k(\beta_k^{DY} - \beta_k^{HZ}) = 0$.

Dans la suite, nous allons procéder à l'amélioration de nos résultats en essayant d'affaiblir les conditions (2.19) en se procédant de la manière suivante, nous gardons dans (2.8)

$$\eta_k = \theta_k + \vartheta_k, \quad \beta_k = \theta_k \beta_k^{DY} + \vartheta_k \beta_k^{HZ}, \quad (2.20)$$

de sorte que la direction de recherche (2.8) peut être écrite comme suit

$$d_{k+1} = -\eta_k g_{k+1} + \beta_k d_k, \quad (2.21)$$

avec

$$(\eta_k, \beta_k) = \begin{cases} (\hat{\eta}_k, \hat{\beta}_k) & \text{si } \Delta_k \neq 0, \\ (c, \beta_k^{PRP}) & \text{si } d_k^\top g_{k+1} = 0, \\ (\tilde{\theta}_k, \beta_k^{FR}) & \text{autrement,} \end{cases} \quad (2.22)$$

où

$$\hat{\eta}_k = \frac{c \|g_{k+1}\|^2 (d_k^\top y_k)}{\Delta_k}, \quad (2.23)$$

$$\hat{\beta}_k = \frac{c \|g_{k+1}\|^2 (y_k^\top g_{k+1})}{\Delta_k}. \quad (2.24)$$

Nous notons que la condition $\beta_k^{DY} \neq \beta_k^{HZ}$, qui apparaît dans le premier cas de (2.19) et disparaît dans le premier cas de (2.22) en raison de la résolution du système (2.53) et (2.54) par rapport à η_k et θ_k ; en effet, en substituant $\vartheta_k = \eta_k - \theta_k$ dans (2.53) et (2.54), nous obtenons le système équivalent suivant :

$$-(\|g_{k+1}\|^2 - \beta_k^{HZ} d_k^\top g_{k+1})\eta_k + (\beta_k^{DY} - \beta_k^{HZ})d_k^\top g_{k+1}\theta_k = -c\|g_{k+1}\|^2, \quad (2.25)$$

et

$$(y_k^\top g_{k+1} - \beta_k^{HZ} d_k^\top y_k)\eta_k - (\beta_k^{DY} - \beta_k^{HZ})d_k^\top y_k\theta_k = 0. \quad (2.26)$$

D'après (2.26), nous obtenons

$$(\beta_k^{DY} - \beta_k^{HZ})d_k^\top y_k\theta_k = (y_k^\top g_{k+1} - \beta_k^{HZ} d_k^\top y_k)\eta_k. \quad (2.27)$$

en multipliant (2.25) par $d_k^\top y_k$, il s'ensuit que

$$-(\|g_{k+1}\|^2 - \beta_k^{HZ} d_k^\top g_{k+1})d_k^\top y_k\eta_k + (\beta_k^{DY} - \beta_k^{HZ})d_k^\top g_{k+1}d_k^\top y_k\theta_k = -c\|g_{k+1}\|^2 d_k^\top y_k, \quad (2.28)$$

en substituant (2.27), nous obtenons simplement

$$(-\|g_{k+1}\|^2 d_k^\top y_k + y_k^\top g_{k+1} d_k^\top g_{k+1})\eta_k = -c\|g_{k+1}\|^2 d_k^\top y_k,$$

donc

$$\eta_k = \frac{-c\|g_{k+1}\|^2 d_k^\top y_k}{(-\|g_{k+1}\|^2 d_k^\top y_k + y_k^\top g_{k+1} d_k^\top g_{k+1})}$$

qui sera noté $\hat{\eta}_k$, ce qui implique (2.23), avec $\Delta_k \neq 0$, nous remarquons que, $\Delta_k \neq 0$ est indépendante de la condition $\beta_k^{DY} = \beta_k^{HZ}$. De même, en substituant $\vartheta_k = \eta_k - \theta_k$ dans (2.20) nous avons

$$\beta_k = \beta_k^{HZ} \eta_k + (\beta_k^{DY} - \beta_k^{HZ}) \theta_k \quad (2.29)$$

en utilisant (2.27), il s'ensuit que

$$\beta_k = \frac{y_k^\top g_{k+1}}{d_k^\top y_k} \eta_k \quad (2.30)$$

d'après la définition (2.23) de $\hat{\eta}_k$, nous obtenons (2.24) la définition de $\hat{\beta}_k$.

Nous allons, maintenant, établir le lemme suivant, qui montre la possibilité de jumeler, les conditions (2.19) et (2.22) en une seule et nouvelle condition (Ainsi tout sera réduit à un seul choix).

Lemme 2.2.1

Si l'une des deux conditions $d_k^\top g_{k+1} = 0$ ou $g_k^\top g_{k+1} = 0$ est vérifiée, alors le premier choix dans (2.19) et (2.22) respectivement se réduit aux deuxième et troisième choix.

Preuve 2.2.1 En utilisant les définitions de β_k^{HZ} et β_k^{DY} , les formules (2.15), (2.16) et (2.17), peuvent être réarrangées respectivement comme suit,

$$\hat{\theta}_k = \frac{2c\|g_{k+1}\|^2 \|y_k\|^2 d_k^\top g_{k+1}}{\Delta_k (\beta_k^{DY} - \beta_k^{HZ}) d_k^\top y_k}, \quad (2.31)$$

$$\hat{\vartheta}_k = \frac{c\|g_{k+1}\|^2 g_k^\top g_{k+1}}{\Delta_k (\beta_k^{DY} - \beta_k^{HZ})}, \quad (2.32)$$

$$\Delta_k = (d_k^\top g_{k+1})(g_k^\top g_{k+1}) - d_k^\top g_k \|g_{k+1}\|^2. \quad (2.33)$$

De plus,

$$\beta_k^{DY} - \beta_k^{HZ} = \frac{g_k^\top g_{k+1}}{d_k^\top y_k} + 2\|y_k\|^2 \frac{d_k^\top g_{k+1}}{(d_k^\top y_k)^2}. \quad (2.34)$$

Par conséquent, la condition $d_k^\top g_{k+1} = 0$ réduit le couple $(\hat{\theta}_k, \hat{\vartheta}_k)$ à $(0, c)$, tandis que l'équation $g_k^\top g_{k+1} = 0$ réduit $(\hat{\theta}_k, \hat{\vartheta}_k)$ à $(\tilde{\theta}_k, 0)$.

De même, pour le choix (2.22), en utilisant les réarrangements ci-dessus, nous observons à partir de (2.23) et (2.24) que les conditions $d_k^\top g_{k+1} = 0$ et $g_k^\top g_{k+1} = 0$, réduisent respectivement la paire $(\hat{\eta}_k, \hat{\beta}_k)$ à (c, β_k^{PRP}) et $(\hat{\eta}_k, \hat{\beta}_k)$ à $(\tilde{\theta}_k, \beta_k^{FR})$.

Ce résultat suggère l'écriture des deux premiers cas et une partie du troisième cas dans (2.19) et (2.22) comme un seul cas, cependant cette réécriture n'est plus valable quand on est dans les situation $(\beta_k^{DY} - \beta_k^{HZ})\Delta_k = 0$ et $\Delta_k = 0$, respectivement.

Si les conditions $(\beta_k^{DY} - \beta_k^{HZ})\Delta_k = 0$ et $\Delta_k = 0$ sont satisfaites, notre méthode n'est pas applicable. Cependant nous proposons d'utiliser toute méthode de gradient conjugué globalement convergente.

plus précisément, nous fixons le couple (θ_k, ϑ_k) qui peut être égal à $(0, 1)$ ce qui correspond au couple $(\eta_k, \beta_k) = (1, \beta_k^{HZ})$, ainsi la direction de recherche correspondante au cas précédent, satisfait la condition de descente suffisante (2.6) pour toute technique de recherche linéaire.

En conclusion, nous allons illustrer nos nouvelles hypothèses jumelé, en effet, nous remplaçons les choix (2.19) et (2.22) par les deux expressions suivantes

$$(\theta_k, \vartheta_k) = \begin{cases} (\hat{\theta}_k, \hat{\vartheta}_k) & \text{si } \beta_k^{DY} \neq \beta_k^{HZ}, \Delta_k \neq 0, \\ (0, 1) & \text{autrement} \end{cases} \quad (2.35)$$

et

$$(\eta_k, \beta_k) = \begin{cases} (\hat{\eta}_k, \hat{\beta}_k) & \text{si } \Delta_k \neq 0, \\ (1, \beta_k^{HZ}) & \text{autrement.} \end{cases} \quad (2.36)$$

Cependant, nous allons montrer que la condition $\Delta_k \neq 0$ est souvent satisfaite sous certains condition. Tout d'abord, nous allons montrer que $\Delta_k > 0$, ce qui est prouvé dans le lemme suivant,

Lemme 2.2.2 *L'inégalité $\Delta_k > 0$ est vérifiée si soit*

(i) $(d_k^\top g_{k+1})(g_k^\top g_{k+1}) \geq 0$ ou (ii) la condition de Wolfe forte (3.4) et (3.5) sont appliquées avec des valeurs suffisamment petites du paramètre σ .

Preuve 2.2.2 *Considérons (2.33), à savoir*

$$\Delta_k = (d_k^\top g_{k+1})(g_k^\top g_{k+1}) - d_k^\top g_k \|g_{k+1}\|^2, \quad (2.37)$$

nous observons que $\Delta_k > 0$ si la condition (i) est vérifiée, puisque la propriété de descente $d_k^\top g_k < 0$ est vérifiée.

Maintenant, en notant que

$$\Delta_k \geq -|d_k^\top g_{k+1}| |g_k^\top g_{k+1}| - d_k^\top g_k \|g_{k+1}\|^2 \quad (2.38)$$

et en utilisant la condition de wolfe $|d_k^\top g_{k+1}| \leq -\sigma d_k^\top g_k$, il s'ensuit que

$$\Delta_k \geq -d_k^\top g_k (\|g_{k+1}\|^2 - \sigma |g_k^\top g_{k+1}|) \quad (2.39)$$

qui est positif pour des valeurs suffisamment petites de σ .

Ce résultat montre que $\Delta_k > 0$ si

$$\|g_{k+1}\|^2 - \sigma |g_k^\top g_{k+1}| \geq \hat{\sigma} \quad (2.40)$$

qui est valable pour des valeurs suffisamment petites des paramètres σ et $\hat{\sigma} > 0$.

Le lemme précédent nous permet de modifier les choix de (θ_k, ϑ_k) (2.35) et de (η_k, β_k) (2.36) respectivement comme suit.

$$(\theta_k, \vartheta_k) = \begin{cases} (\hat{\theta}_k, \hat{\vartheta}_k) & \text{si ; } \beta_k^{DY} \neq \beta_k^{HZ}, ; \|g_{k+1}\|^2 \geq \sigma |g_k^\top g_{k+1}| + \hat{\sigma}, \\ (0, 1) & \text{autrement} \end{cases} \quad (2.41)$$

et

$$(\eta_k, \beta_k) = \begin{cases} (\hat{\eta}_k, \hat{\beta}_k) & \text{si ; } \|g_{k+1}\|^2 \geq \sigma |g_k^\top g_{k+1}| + \hat{\sigma}, \\ (1, \beta_k^{HZ}) & \text{autrement.} \end{cases} \quad (2.42)$$

Ainsi, nous obtenons un algorithme hybride, noté DYHZ/HZ, qui est donné par :

Algorithm 1 Algorithme de gradient conjugué hybride.

Étape 1. Soit un point initial x_0 et $\varepsilon \geq 0$. Poser $k = 0$.

Étape 2. Calculer $g_0 = g(x_0)$. Si $\|g_k\| \leq \varepsilon$: stop, sinon non définir $d_0 = -g_0$ et aller a **Étape 3**.

Étape 3. Calculer la longueur de pas α_k avec les conditions de recherche de la ligne de Wolfe forte.

Étape 4. Poser $x_{k+1} = x_k + \alpha_k d_k$.

Étape 5. Calculer $g_{k+1} = g(x_{k+1})$.

Étape 6. Si $\|g_{k+1}\| \leq \varepsilon$, stop.

Étape 7. Calculer θ_k et ϑ_k par (2.17) et donc d_{k+1} par (2.7). Poser $k = k + 1$, et aller à **Étape 3**.

2.3 La convergence de l'algorithme

Dans cette section, nous analysons la convergence globale de l'algorithme. Avant cela, nous introduisons les hypothèses suivantes pour la fonction ob-

jective $f(x)$.

H1. f est minorée dans \mathbb{R}^n et f est continument différentiable dans un voisinage V de l'ensemble $S = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$, où x_0 est le point de départ de l'itération.

H2 Le gradient de f est Lipschitzien dans V , autrement dit, il existe une constante $L > 0$ tel que :

$$\|\nabla f(\tilde{x}) - \nabla f(x)\| \leq L\|\tilde{x} - x\| \quad (2.43)$$

Nous avons le lemme technique suivant

Lemme 2.3.1 *Supposons que les hypothèses **H1** et **H2** soient satisfaites et que la suite $\{x_k\}$ soit générée par (3.2), de sorte que la propriété de descente est satisfaites. et que α_k soit déterminé telle que les conditions de Wolfe (3.4) et l'inégalité de gauche de (3.5) sont vérifiées. Alors la condition de Zoutendijk est assurée, plus précisément*

$$\sum_{k=0}^{\infty} \frac{(g_k^\top d_k)^2}{\|d_k\|^2} < \infty \quad (2.44)$$

Le théorème ci-dessous montre la convergence de notre nouvelle algorithme.

Théorème 2.3.1 *Soit x_1 tel que les hypothèses H1 et H2 sont vérifiées et le premier cas de (2.42), la condition (2.40) sont vérifiées et la suite $\{x_k\}$ est générée par l'algorithme 2 précédent. Alors*

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (2.45)$$

Preuve 2.3.1 *Nous allons démontrer le théorème en utilisant par contradiction.*

Pour cela nous supposons qu'il existe une constante $\gamma_1 > 0$ telle que

$$\|g_k\| \geq \gamma_1, \quad \forall k \geq 1, \quad (2.46)$$

en utilisant cette hypothèse et la condition de descente suffisante (2.9), il s'ensuit que

$$(g_k^\top d_k)^2 \geq c^2 \|g_k\|^4 \geq c^2 \gamma_1^4$$

donc d'après (2.44) on obtient

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} < \infty,$$

ce qui contredit la condition $\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} = \infty$, que ne peut réaliser que si d_k est borné.

Par conséquent, nous allons montrer que $\|d_k\|$ est borné en procédant de la manière suivante :

D'après les relations (2.21) et (2.30), nous avons

$$d_{k+1} = -\eta_k g_{k+1} + \frac{y_k^\top g_{k+1}}{d_k^\top y_k} \eta_k d_k,$$

examinons le terme $d_k^\top y_k$ nous obtenons, alors en utilisant les conditions de Wolfe fortes (3.5) et la condition de descente suffisante (2.9), la relation suivante

$$c(1 - \sigma) \|g_k\|^2 \leq -(1 - \sigma) d_k^\top g_k \leq d_k^\top y_k \leq -(1 + \sigma) d_k^\top g_k. \quad (2.47)$$

Ainsi, à partir des hypothèses H1 et H2, on a $\|g_{k+1}\| < \gamma_2$ associé à la condition de Lipschitz (2.43), nous obtenons

$$\begin{aligned} \|d_{k+1}\| &\leq |\eta_k| \|g_{k+1}\| + \frac{\|y_k\| \|g_{k+1}\|}{d_k^\top y_k} |\eta_k| \|d_k\| \\ &\leq |\eta_k| \left(\gamma_2 + \frac{L \alpha_k \|d_k\| \gamma_2}{c(1 - \sigma) \gamma_1^2} \|d_k\| \right). \end{aligned} \quad (2.48)$$

Ainsi pour que d_k soit bornée il suffit de montrer que η_k dans (2.48) est bornée. Supposons que les conditions de Wolfe fortes sont satisfaites et prenons σ assez petite de , d'après (2.40), la définition (2.42) de η_k , c et et les relations (2.23), (2.39), ainsi que la condition de courbure (2.47), puis la borne γ_2 , nous obtenons

$$\begin{aligned} |\eta_k| &= \frac{c\|g_{k+1}\|^2(d_k^\top y_k)}{\Delta_k} \\ &\leq \frac{c\|g_{k+1}\|^2(d_k^\top y_k)}{-d_k^\top g_k(\|g_{k+1}\|^2 - \sigma|g_k^\top g_{k+1}|)} \\ &\leq \frac{c\|g_{k+1}\|^2(d_k^\top y_k)}{-d_k^\top g_k \nu} \\ &\leq \frac{c\|g_{k+1}\|^2(1 + \sigma)}{\nu} \\ &\leq M, \end{aligned}$$

où $M = \frac{c\gamma_2^2(1 + \sigma)}{\nu}$.

En substituant le résultat précédent dans (2.48), il s'ensuit que

$$\|d_{k+1}\| \leq M\gamma_2 + M \frac{L\gamma_2\alpha_k\|d_k\|}{c(1 - \sigma)\gamma_1^2} \|d_k\|, \quad (2.49)$$

comme $\alpha_k\|d_k\| \rightarrow 0$, qui découle de la première condition de Wolfe (3.4), et d'après (2.49), il existe constante $m \in (0, 1)$ et un entier k_1 , tels que

$$M \frac{L\gamma_2\alpha_k\|d_k\|}{c(1 - \sigma)\gamma_1} \leq m < 1, \quad \forall ; k \geq k_1, \quad (2.50)$$

par conséquent, nous obtenons

$$\|d_{k+1}\| \leq M\gamma_2 + m\|d_k\|, \quad \forall ; k \geq k_1 \quad (2.51)$$

ce qui implique

$$\|d_{k+1}\| \leq \frac{M\gamma_2}{1 - m} + \|d_{k_1}\| m^{k+1-k_1}, \quad \forall ; k \geq k_1. \quad (2.52)$$

Ainsi, $\|d_k\|$ est borné, donc la contradiction est démontré. Par conséquent le résultat de la convergence globale (2.45) est satisfaite.

Enfin nous terminons notre analyse par l'étude de deuxième cas de (2.42), en supposant que $(\eta_k, \beta_k) = (1, \beta_k^{HZ})$.

Nous remarquons bien que dans le cas (2.42) qui définit une méthode globalement convergente, donc la condition de convergence (2.45) est satisfaite, ainsi s'achève la preuve du théorème.

2.4 Résultats numériques

nous présentons ici, une comparaison de l'efficacité de notre nouvelle méthode hybride du gradient conjugué (2.42) avec les méthodes HZ et DY, afin de déterminer les performances de tous les algorithmes sur un ensemble de problèmes tests d'optimisation sans contrainte [8]. Chaque problème est testé selon les paramètres suivant n nombre de variables, fixé à $n = 2, 10, 50, 100, 1000, 1500, 2000, 5000, \text{ et } 10000$. Avec le paramètre $c = 7/8$, en utilisant les conditions de recherche linéaire de Wolfe forte (3.4) et (3.5) avec $\delta = 0,0001, \sigma = 0,1, \sigma = 0,9$ (les méthodes sont notées DYHZ/HZ.1 et DYHZ/HZ.9, respectivement). Le critère d'arrêt pour tous les algorithmes est que $\|g_k\|^2 \leq 10^{-6}$.

Nous adoptons les profils de performance proposés par Dolan et Moré [15].

Afin d'obtenir les figures 2.1, 2.2 et 2.3 montrant la vitesse CPU, le nombre d'itérations et le nombre de fonctions et d'évaluations du gradient respectivement.

Les figures montrent clairement que la méthode hybride proposée performe mieux que les méthodes efficaces DY et HZ, nous observons également que

les performances de la nouvelle méthode avec $\sigma = 0.1$ sont meilleures que $\sigma = 0.9$, nous avons utilisé la première valeur pour les expériences suivantes. Nous avons répété l'exécution pour la méthode (2.42), mais avec β_k^{HZ} dans le deuxième cas être remplacé par β_k^{DY} , β_k^{PR+} et zéro (les méthodes sont appelées DYHZ/DY.1, DYHZ/PR+.1 et DYHZ/SD.1, respectivement). se sont représentées par les trois figures 2.4, 2.5 et 2.6. Nous avons observé que les performances de la méthode DYHZ/HZ.1 sont un peu meilleures que celles des autres méthodes.

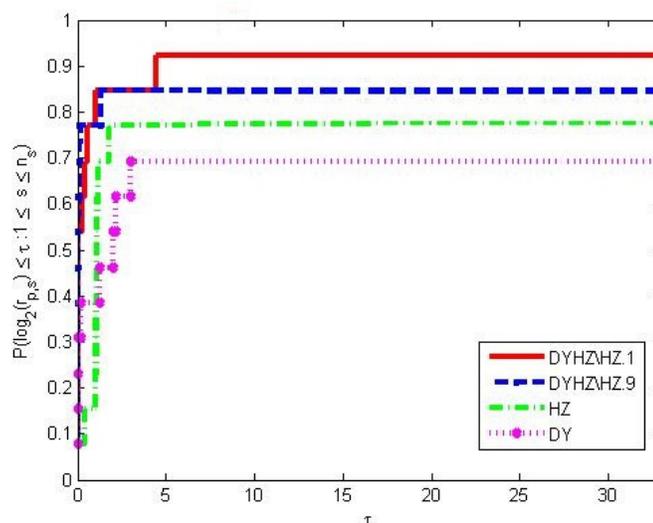


FIGURE 2.1 – Profil de performance pour le temps CPU

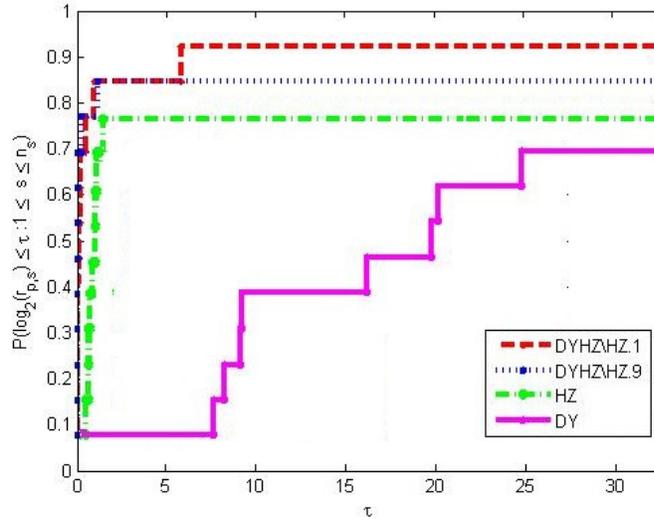


FIGURE 2.2 – Profil de performance pour le nombre d'itérations

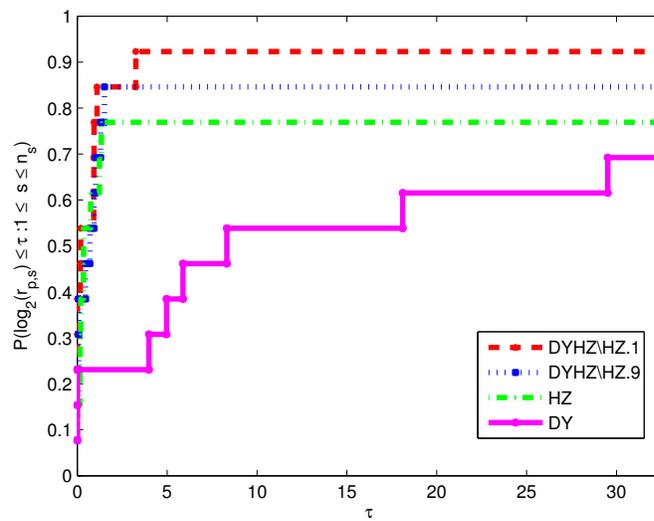


FIGURE 2.3 – Profil de performance pour le nombre d'évaluations de fonctions et de gradients.

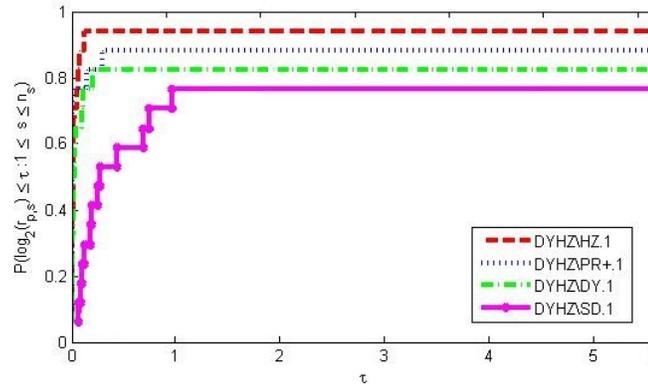


FIGURE 2.4 – Profil de performance pour le temps CPU

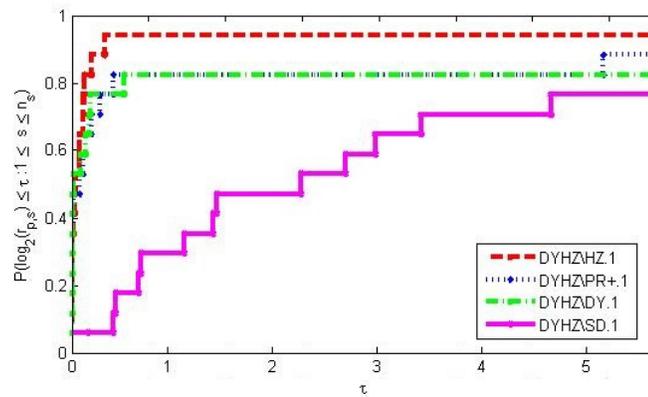


FIGURE 2.5 – Profil de performance pour le nombre d'itérations

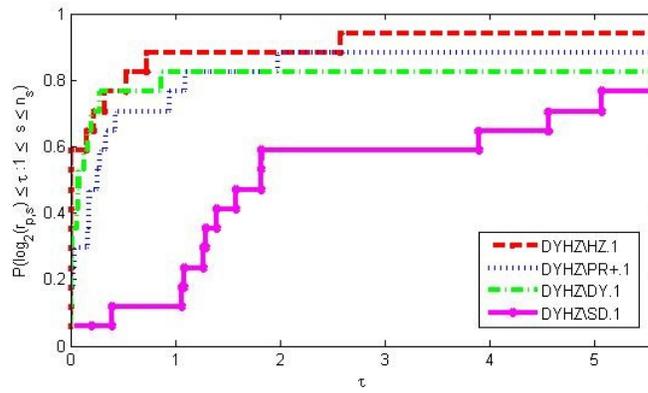


FIGURE 2.6 – Profil de performance pour le nombre d'évaluations de fonctions et de gradients.

— **En utilisant la condition de conjugaison généralisée**

Le système déjà obtenu lorsqu'on travaille avec la condition de conjugaison devient le suivant lorsqu'on utilise la condition de conjugaison généralisée.

$$(\|g_{k+1}\|^2 - \beta_k^{DY} d_k^\top g_{k+1})\theta_k + (\|g_{k+1}\|^2 - \beta_k^{HZ} d_k^\top g_{k+1})\vartheta_k = -c\|g_{k+1}\|^2 \quad (2.53)$$

et

$$d_k^\top y_k g_k^\top g_{k+1} \theta_k - 2\|y_k\|^2 d_k^\top g_{k+1} \vartheta_k = -\nu s_k^\top g_{k+1}. \quad (2.54)$$

Nous calculons les valeurs des inconnus θ_k et ϑ_k en utilisant les mêmes techniques qui sont déjà utilisées, et de la même façon aussi nous obtenons sa convergence.

Ce graphe représente une comparaison entre les deux méthodes proposées.

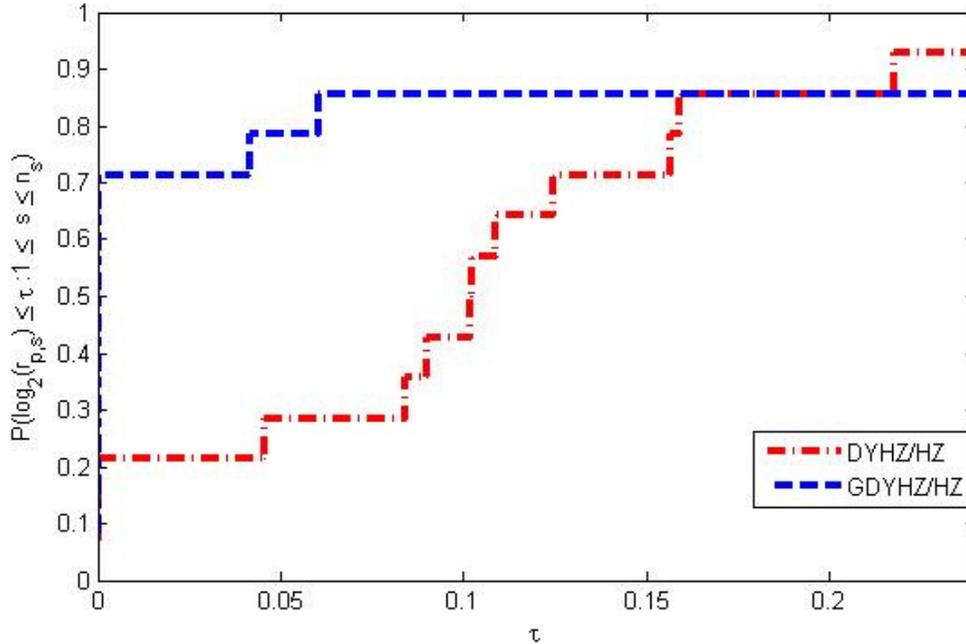


FIGURE 2.7 – Profil de performance pour le temps CPU

Au départ, on voit bien que la méthode qui vérifie la condition de conjugaison généralisée est mieux que la méthode qui vérifie la condition de conjugaison . Mais, à un certain moment la méthode qui vérifie la condition de conjugaison devient la plus performante.

Ainsi, les expériences numériques préliminaires montrent que DYHZ/HZ et DYHZG/HZ sont des algorithmes très efficaces pour les problèmes d'optimisation sans contrainte à grande taille.

Chapitre 3

Une nouvelle méthode d'optimisation basée sur l'idée de A. Perry [30].

3.1 Introduction

Il est bien connu que la méthode du gradient conjugué non linéaire, est caractérisée par de fortes propriétés de convergence locale et globale, plus pratique aussi que d'autres méthodes par ce qu'elle minimise les problèmes d'optimisation sans contraintes de grandes tailles.

Dans ce chapitre, nous présentons une nouvelle méthode du gradient conjugué obtenue après avoir modifié la méthode de HS, qui vérifie toujours la condition de conjugaison $y_k^T d_{k+1} = 0$, cependant nous pouvons quand-même l'améliorer en effectuant une accélération d'algorithme et sans l'utilisation de la condition de conjugaison.

Pour résoudre le problème d'optimisation sans contrainte [11, 36, 37, 38, 42, 39],

$$\min_{x \in \mathbb{R}^n} f(x), \quad (3.1)$$

où f est une fonction continue différentiable.

La méthode du gradient conjugué génère une suite $\{x_k\}_{k \in \mathbb{N}}$, en fixant un point initial $x_0 \in \mathbb{R}^n$ et en utilisant la relation de récurrence suivante

$$x_{k+1} = x_k + \alpha_k d_k, \quad (3.2)$$

où,

α_k est la longueur du pas de la recherche linéaire.

et les directions d_k sont données par

$$\begin{cases} d_1 = -g_1 \\ d_{k+1} = -g_{k+1} + \beta_k d_k, \quad \forall k \geq 1, \end{cases} \quad (3.3)$$

avec, $g_k = g(x_k) = \nabla f(x_k)$, et β_k est un paramètre de gradient conjugué.

Il est bien connu que les formules pour β_k incluent le Hestenes-Stiefel (HS) [26], le Fletcher-Reeves (FR) [19], le Polak-Ribiere-Polyak (PRP) [31], le Liu-Storey (LS) [29] et le Hager-Zhang (Hz) [25].

Nous savons que pour obtenir les résultats de convergence globale de la méthode de gradient conjugué, il est généralement nécessaire que la longueur de pas α_k doit satisfaire certaines conditions de recherche linéaire, telles que la recherche linéaire forte de Wolfe donnée par

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^\top d_k, \quad (3.4)$$

$$\sigma g_k^\top d_k \leq g(x_k + \alpha_k d_k)^\top d_k \leq -\sigma g_k^\top d_k, \quad (3.5)$$

avec, $0 < \delta < \frac{1}{2}$ et $\delta < \sigma < 1$.

La direction de recherche d_{k+1} est nécessaire pour satisfaire la condition de descente suffisante,

$$d_{k+1}^\top g_{k+1} \leq -c \|g_{k+1}\|^2, \quad c > 0. \quad (3.6)$$

En plus de ce que nous avons mentionné, les méthodes Quasi-Newton [18, 46] sont des méthodes parfois efficaces pour résoudre (3.1).

La direction de recherche des méthodes Quasi-Newton est donnée par,

$$d_{k+1} = -H_{k+1}g_{k+1},$$

où H_{k+1} est une approximation de l'inverse de la matrice hessienne $\nabla^2 f(x_k)^{-1}$. Certains auteurs utilisent cette technique dans la méthode du gradient conjugué, par exemple, pour calculer le paramètre β_k , Perry [30] a proposé la formule suivante

$$\beta_k^p = \frac{y_k^\top g_{k+1} - s_k^\top g_{k+1}}{d_k^\top y_k}, \quad (3.7)$$

où, $s_k = x_{k+1} - x_k$ et $y_k = g_{k+1} - g_k$.

En remplaçant (3.7) dans (3.3) et en appliquant quelques techniques algébriques simples, nous obtenons la direction de recherche de Perry correspondante comme suit,

$$d_{k+1}^p = - \left(I - \frac{s_k y_k^\top}{y_k^\top s_k} + \frac{s_k s_k^\top}{y_k^\top s_k} \right) g_{k+1} = -P_{k+1}g_{k+1}. \quad (3.8)$$

Dans la méthode de Perry, la matrice P_{k+1} sert à estimer l'approximation de l'inverse de la matrice hessienne. Si la recherche linéaire est exacte, ($d_k^\top g_{k+1} = 0$), alors (3.7) est identique à l'algorithme du gradient conjugué de Hestenes et Stiefel [26].

Comme P_{k+1} n'est pas symétrique, certains auteurs ont modifié cette matrice

pour répondre aux exigences précédentes en utilisant d'autres techniques différentes (voir par exemple [27, 4, 43]), en particulier Andrei [9] a présenté une matrice symétrique pour estimer l'inverse de l'approximation de la matrice hessienne comme suit,

$$P_{k+1}^N = I - \frac{s_k y_k^\top + y_k s_k^\top}{y_k^\top s_k} + \eta_k \frac{s_k s_k^\top}{y_k^\top s_k}, \quad (3.9)$$

où, le paramètre η_k est calculé d'une manière différente [45].

Notre travail est basé nécessairement sur l'observation de Perry [30], en utilisant la méthode HS, dans laquelle la direction d_{k+1} dans (3.3) peut être réécrite comme suit,

$$d_{k+1} = -D_{k+1}g_{k+1} = -\left(I - \frac{s_k y_k^\top}{s_k^\top y_k}\right)g_{k+1}. \quad (3.10)$$

La matrice D_{k+1} est une matrice d'itération du gradient conjugué, qui représente l'inverse d'une matrice d'approximation hessienne, mais elle n'est pas symétrique. Plusieurs procédures de symétrie ont été proposées, telle que la technique symétrique de Powell [14], dans laquelle la matrice d'itération du gradient conjugué D_{k+1} peut être symétrisée par \tilde{D}_{k+1} comme suit,

$$\tilde{D}_{k+1} = \left(I - \frac{s_k y_k^\top}{s_k^\top y_k}\right) \left(I - \frac{y_k s_k^\top}{s_k^\top y_k}\right). \quad (3.11)$$

Ainsi, en se basant sur la méthode HS, de nombreuses variantes largement utilisées en pratique ont été développées, car elles présentent de meilleures performances de calcul.

De plus, il est très important de choisir une matrice d'itération pour une méthode du gradient conjugué non-linéaire. A partir de D_{k+1} , nous proposons une nouvelle matrice définie positive, qui satisfait toujours la condition

de descente suffisante pour toute recherche linéaire. Nous utilisons également une technique d'adaptation accélérée dans nos algorithmes de gradient conjugué, qui montre que la méthode proposée converge globalement, en utilisant l'analyse spectrale. Finalement, nous décrivons les résultats numériques.

3.2 Amélioration de la méthode du gradient conjugué.

Dans cette section, nous proposons une modification de l'algorithme de gradient conjugué pour l'adapter à la résolution des problèmes à grandes tailles, à n'importe quelle itération. La matrice D_{k+1} dans (3.10) n'est pas symétrique ; donc, nous proposons sa symétrisation donnée par

$$D_{k+1}^{sym} = \frac{D_{k+1}^\top + D_{k+1}}{2} = I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}. \quad (3.12)$$

Nous observons que notre famille est symétrique, le résultat suivant va nous éclaircir le caractère définie positive de la nouvelle famille des matrices D_{k+1}^{sym} afin d'assurer la condition de descente suffisante.

Théorème 3.2.1

Soit D_{k+1}^{sym} défini par (3.12), et $s_k^\top y_k \neq 0$. Alors, D_{k+1}^{sym} une valeur propre $\mu = 1$ de multiplicité $(n - 2)$; et les deux valeurs propres restantes (respectivement maximale et minimale) note μ_{\max}^{k+1} et μ_{\min}^{k+1} donnée par

$$\mu_{\max}^{k+1} = \frac{1}{2} + \frac{1}{2} \sqrt{\frac{(s_k^\top s_k)(y_k^\top y_k)}{(s_k^\top y_k)^2}}, \quad (3.13)$$

$$\mu_{\min}^{k+1} = \frac{1}{2} - \frac{1}{2} \sqrt{\frac{(s_k^\top s_k)(y_k^\top y_k)}{(s_k^\top y_k)^2}}. \quad (3.14)$$

Preuve 3.2.1

En utilisant la formule algébrique suivante

$$\det(I + xy^\top + uv^\top) = (1 + y^\top x)(1 + v^\top u) - (x^\top v)(y^\top u),$$

nous obtenons

$$\begin{aligned} \det(D_{k+1}^{sym}) &= \det\left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right) \\ &= \left(1 - \frac{1}{2} \frac{s_k^\top y_k}{s_k^\top y_k}\right) \left(1 - \frac{1}{2} \frac{y_k^\top s_k}{s_k^\top y_k}\right) - \frac{1}{4} \frac{(s_k^\top s_k)(y_k^\top y_k)}{(s_k^\top y_k)^2} \\ &= \frac{1}{4} - \frac{(s_k^\top s_k)(y_k^\top y_k)}{4(s_k^\top y_k)^2}. \end{aligned} \quad (3.15)$$

Par conséquent, la matrice D_{k+1}^{sym} est non singulière lorsque $\frac{(s_k^\top s_k)(y_k^\top y_k)}{(s_k^\top y_k)^2} > 1$.

Pour prouver que la matrice D_{k+1}^{sym} aux possède comme suit, a la valeur propre $\mu = 1$ (avec la multiplicité $(n - 2)$).

Soit $V = \text{span}\{y_k, s_k\}$, est l'ensemble de tous les vecteurs qui peuvent être représentés par une combinaison linéaire de $\{y_k, s_k\}$. on a $\dim(V) \leq 2$ et $\dim(V^\perp) \geq n - 2$. Ainsi, $\forall \zeta \in V^\perp$, pour que $s_k^\top \zeta = y_k^\top \zeta = 0$, nous avons

$$D_{k+1}^{sym} \zeta = \left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right) \zeta = \zeta - \frac{y_k^\top \zeta}{2s_k^\top y_k} s_k - \frac{s_k^\top \zeta}{2s_k^\top y_k} y_k = \zeta,$$

ainsi,

$$D_{k+1}^{sym} \zeta = \zeta,$$

donc, $\mu = 1$ est une valeurs propres de multiplicité $(n - 2)$.

Reste a montrer les relations (3.13) et (3.14).

En effet a partir de la formule (3.12), nous pouvons obtenir la trace de D_{k+1}^{sym} comme suit

$$\begin{aligned} \text{tr}(D_{k+1}^{sym}) &= \text{tr}\left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right) \\ &= n - \frac{1}{2} \frac{y_k^\top s_k + s_k^\top y_k}{s_k^\top y_k} \\ &= n - 1, \end{aligned} \quad (3.16)$$

et

$$\text{tr}(D_{k+1}^{sym}) = \underbrace{1 + \dots + 1}_{(n-2)\text{ fois}} + \mu_{\max}^{k+1} + \mu_{\min}^{k+1}. \quad (3.17)$$

Par conséquent, nous obtenons que

$$\mu_{\max}^{k+1} + \mu_{\min}^{k+1} = 1. \quad (3.18)$$

Aussi, pour les relations concernant les déterminants on a

$$\det(D_{k+1}^{sym}) = \mu_{\max}^{k+1} \mu_{\min}^{k+1},$$

ainsi, en utilisant (3.15) on obtient

$$\det(D_{k+1}^{sym}) = \mu_{\max}^{k+1} \mu_{\min}^{k+1} = \frac{1}{4} - \frac{(s_k^\top s_k)(y_k^\top y_k)}{4(s_k^\top y_k)^2}. \quad (3.19)$$

A partir de (3.18) et (3.19), μ_{\max}^{k+1} et μ_{\min}^{k+1} sont deux solutions de l'équation suivante

$$\mu^2 - \mu + \frac{1}{4} - \frac{(s_k^\top s_k)(y_k^\top y_k)}{4(s_k^\top y_k)^2} = 0.$$

Ainsi, les deux autres valeurs propres sont déterminées par (3.13) et (3.14), respectivement.

Corollaire 3.2.1 *Que la matrice D_{k+1}^{sym} non définie positive.*

Car d'après le théorème précédent, $\frac{(s_k^\top s_k)(y_k^\top y_k)}{(s_k^\top y_k)^2} > 1$, par conséquent $\mu_{\min}^{k+1} \leq 0$.

Donc la matrice D_{k+1}^{sym} n'est pas définie positive.

Pour rendre la matrice D_{k+1}^{sym} définie positive, nous avons besoin d'augmenter sa puissance à $2p$, $p \in \mathbb{N}^*$. Pour cela, nous avons deux cas :

Si s_k et y_k sont des vecteurs linéairement indépendants, nous mettons

$$M_{k+1} = (D_{k+1}^{sym})^{2p} = \left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k} \right)^{2p}. \quad (3.20)$$

Alors, en appliquant le théorème précédent, les valeurs propres de M_{k+1} sont similaires à celles de D_{k+1}^{sym} .

Seulement, les max et les min de ces valeur propre notés λ_{k+1}^+ et λ_{k+1}^- , sont donnés par

$$\lambda_{k+1}^+ = (\mu_{\max}^{k+1})^{2p}, \quad \lambda_{k+1}^- = (\mu_{\min}^{k+1})^{2p}.$$

Cependant, si s_k et y_k étaient des vecteurs linéairement dépendants, c'est-à-dire $s_k = \sigma y_k$, alors M_{k+1} serait réécrite sous la forme

$$M_{k+1} = (D_{k+1}^{sym} + (1 - \mu_{\min}^{k+1}) I)^{2p},$$

Ce qui donne

$$M_{k+1} = \left(2I - \frac{y_k y_k^\top}{y_k^\top y_k} \right)^{2p}. \quad (3.21)$$

Par conséquent, les valeurs propres de la matrice M_{k+1} sont constituées de 2^{2p} valeur propre de multiplicité $(n-1)$, et de la valeur propre minimal $\lambda_{k+1}^- = 1$, car D_{k+1}^{sym} est diagonalisable pour toute fonction H holomorphe définie sur le spectre de D_{k+1}^{sym} ($Sp(D_{k+1}^{sym})$) inclut dans le domaine de la fonction H notée $D(H)$ où ($Sp(D_{k+1}^{sym}) \subset D(H)$), alors

$$sp(H(D_{k+1}^{sym})) = H(Sp(D_{k+1}^{sym})).$$

Pour obtenir les valeurs propres ci-dessus, nous avons pris

$$H(\mu) = (\mu + (1 - \mu_{\min}^{k+1}))^{2p}.$$

Remarque 3.2.1 Nous observons que la matrice donnée par (3.21) et (3.20) est une matrice symétrique définie positive.

A partir de la modification simple sur la matrice D_{k+1} , nous avons introduit direction de la recherche suivante

$$d_{k+1} = -M_{k+1}g_{k+1}, \quad (3.22)$$

où

$$M_{k+1} = \begin{cases} (3.21) & \text{si } s_k = \sigma y_k \\ (3.20) & \text{autrement.} \end{cases} \quad (3.23)$$

Le théorème suivant montre que notre méthode satisfait la condition de descente suffisante.

Théorème 3.2.2 Soit la suite $\{d_{k+1}\}_{k \in \mathbb{N}}$ générée par (3.22), alors la direction de recherche d_{k+1} satisfait la condition de descente suffisante

$$d_{k+1}^\top g_{k+1} \leq -c \|g_{k+1}\|^2, \quad c > 0,$$

où, $g_{k+1} = g(x_{k+1}) = \nabla f(x_{k+1})$, et c est une constante positive.

Preuve 3.2.2 Pour tout $k \geq 1$, nous avons les directions de recherche définies par

$$d_{k+1} = -M_{k+1}g_{k+1}, \quad (3.24)$$

En multipliant (3.24) par g_{k+1}^\top , nous obtenons

$$d_{k+1}^\top g_{k+1} = -g_{k+1}^\top M_{k+1}g_{k+1}, \quad (3.25)$$

et du fait que M_{k+1} est une matrice symétrique définie positive, nous obtenons

$$d_{k+1}^\top g_{k+1} = -g_{k+1}^\top M_{k+1}g_{k+1} \leq -\lambda_{k+1}^- \|g_{k+1}\|^2. \quad (3.26)$$

Cela montre que la condition de descente suffisante est satisfaite.

3.3 L'accélération du nouvel algorithme de gradient conjugué

Nous savons que les meilleures caractéristiques des méthodes de gradient conjugué sont ses itérations simples du fait qu'elles ne nécessitent pas un grand espace de stockage; cependant, la matrice proposée précédemment, nécessite un grand espace de stockage, d'où la difficulté de son application aux problèmes d'optimisation sans contraintes de grande taille.

Afin d'éviter cette difficulté, nous proposons une formule accélérée pour calculer la matrice (3.20) plus efficacement.

Théorème 3.3.1 Soit M_{k+1} définie par (3.20), alors,

$$\left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k} \right)^{2p} = I + \eta_{2p} (s_k y_k^\top + y_k s_k^\top) + \delta_{2p} s_k s_k^\top + \gamma_{2p} y_k y_k^\top, \quad (3.27)$$

où

$$\begin{aligned} \eta_{2p} &= \frac{1}{2} \left(\left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p-1} + \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p-1} \right) \left(\frac{-1}{2(s_k^\top y_k)} - \frac{1}{(4a_k b_k - 1)(s_k^\top y_k)} \right) \\ &\quad - \frac{2b_k}{(4a_k b_k - 1)(s_k^\top y_k)} \left(\frac{-a_k}{2\sqrt{a_k b_k}} \left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p-1} + \frac{a_k}{2\sqrt{a_k b_k}} \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p-1} \right) \\ &\quad + \frac{1}{(4a_k b_k - 1)(s_k^\top y_k)}, \end{aligned}$$

$$\begin{aligned} \delta_{2p} &= \frac{-\sqrt{a_k b_k}}{2a_k} \left(\left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p-1} - \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p-1} \right) \left(\frac{-4a_k b_k - 1}{2(4a_k b_k - 1)(s_k^\top y_k)} \right) \\ &\quad - \frac{2b_k}{(4a_k b_k - 1)(s_k^\top y_k)} \left(\frac{1}{2} \left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p-1} + \frac{1}{2} \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p-1} - 1 \right), \end{aligned}$$

$$\begin{aligned} \gamma_{2p} &= \frac{-\sqrt{a_k b_k}}{2b_k} \left(\left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p-1} - \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p-1} \right) \left(\frac{-4a_k b_k - 1}{2(4a_k b_k - 1)(s_k^\top y_k)} \right) \\ &\quad - \frac{2a_k}{(4a_k b_k - 1)(s_k^\top y_k)} \left(\frac{1}{2} \left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p-1} + \frac{1}{2} \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p-1} - 1 \right), \end{aligned}$$

$$\text{avec } a_k = -\frac{s_k^\top s_k}{2(s_k^\top y_k)} \text{ et } b_k = -\frac{y_k^\top y_k}{2(s_k^\top y_k)}.$$

Preuve 3.3.1 Pour démontrer le théorème nous procédons par récurrence.

Pour $p = 1$, on a

$$\left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right)^2 = I - \frac{3}{4(s_k^\top y_k)} s_k y_k^\top - \frac{3}{4(s_k^\top y_k)} y_k s_k^\top + \frac{y_k^\top y_k}{4(s_k^\top y_k)^2} s_k s_k^\top + \frac{s_k^\top s_k}{4(s_k^\top y_k)^2} y_k y_k^\top.$$

Nous supposons alors que pour toute $p \geq 1$,

$$\left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right)^{2p} \text{ vérifie (3.27),}$$

et nous montrons que $\left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right)^{2p+1}$ est vérifié.

En effet

$$\begin{aligned} \left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right)^{2p+1} &= \left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right)^{2p} \left(I - \frac{1}{2} \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k}\right) \\ &= \left(I + \eta_{2p} s_k y_k^\top + \eta_{2p} y_k s_k^\top + \delta_{2p} s_k s_k^\top + \gamma_{2p} y_k y_k^\top\right) \left(I - \frac{s_k y_k^\top + y_k s_k^\top}{2(s_k^\top y_k)}\right) = \\ &= I + \left(\frac{-1}{(s_k^\top y_k)} + \frac{\eta_{2p}}{2} - \frac{\delta_{2p} s_k^\top s_k}{2(s_k^\top y_k)}\right) s_k y_k^\top + \left(\frac{-1}{(s_k^\top y_k)} + \frac{\eta_{2p}}{2} - \frac{\gamma_{2p} y_k^\top y_k}{2(s_k^\top y_k)}\right) y_k s_k^\top \\ &\quad + \left(\frac{\delta_{2p}}{2} - \frac{\eta_{2p} y_k^\top y_k}{2(s_k^\top y_k)}\right) s_k s_k^\top + \left(\frac{\gamma_{2p}}{2} - \frac{\eta_{2p} s_k^\top s_k}{2(s_k^\top y_k)}\right) y_k y_k^\top \\ &= I + \eta_{2p+1} \left(s_k y_k^\top + y_k s_k^\top\right) + \delta_{2p+1} s_k s_k^\top + \gamma_{2p+1} y_k y_k^\top, \end{aligned}$$

où, de la même façon, nous obtenons

$$\begin{aligned} \eta_{2p+1} &= \left(\frac{-1}{(s_k^\top y_k)} + \frac{\eta_{2p}}{2} - \frac{\delta_{2p} s_k^\top s_k}{2(s_k^\top y_k)}\right) = \left(\frac{-1}{(s_k^\top y_k)} + \frac{\eta_{2p}}{2} - \frac{\gamma_{2p} y_k^\top y_k}{2(s_k^\top y_k)}\right) \\ &= \frac{1}{2} \left(\left(\frac{1}{2} - \sqrt{a_k b_k}\right)^{2p} + \left(\frac{1}{2} + \sqrt{a_k b_k}\right)^{2p} \right) \left(\frac{-1}{2(s_k^\top y_k)} - \frac{1}{(4a_k b_k - 1)(s_k^\top y_k)} \right) \\ &\quad - \frac{2b_k}{(4a_k b_k - 1)(s_k^\top y_k)} \left(\frac{-a_k}{2\sqrt{a_k b_k}} \left(\frac{1}{2} - \sqrt{a_k b_k}\right)^{2p} + \frac{a_k}{2\sqrt{a_k b_k}} \left(\frac{1}{2} + \sqrt{a_k b_k}\right)^{2p} \right) \\ &\quad + \frac{1}{(4a_k b_k - 1)(s_k^\top y_k)}, \end{aligned}$$

$$\begin{aligned}
\delta_{2p+1} &= \left(\frac{\delta_{2p}}{2} - \frac{\eta_{2p} y_k^\top y_k}{2(s_k^\top y_k)} \right) \\
&= \frac{-\sqrt{a_k b_k}}{2a_k} \left(\left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p} - \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p} \right) \left(\frac{-4a_k b_k - 1}{2(4a_k b_k - 1)(s_k^\top y_k)} \right) \\
&\quad - \frac{2b_k}{(4a_k b_k - 1)(s_k^\top y_k)} \left(\frac{1}{2} \left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p} + \frac{1}{2} \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p} - 1 \right), \\
\gamma_{2p+1} &= \left(\frac{\gamma_{2p}}{2} - \frac{\eta_{2p} s_k^\top s_k}{2(s_k^\top y_k)} \right) \\
&= \frac{-\sqrt{a_k b_k}}{2b_k} \left(\left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p} - \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p} \right) \left(\frac{-4a_k b_k - 1}{2(4a_k b_k - 1)(s_k^\top y_k)} \right) \\
&\quad - \frac{2a_k}{(4a_k b_k - 1)(s_k^\top y_k)} \left(\frac{1}{2} \left(\frac{1}{2} - \sqrt{a_k b_k} \right)^{2p} + \frac{1}{2} \left(\frac{1}{2} + \sqrt{a_k b_k} \right)^{2p} - 1 \right),
\end{aligned}$$

ce qui achève la preuve.

Cette nouvelle formulation [\(3.27\)](#) est la mieux adapté pour la programmation numérique. Ainsi, d'après le théorème 3, nous obtenons une nouvelle définition de M_{k+1} peut être modifié comme

$$M_{k+1} = \begin{cases} \left(2I - \frac{y_k y_k^\top}{y_k^\top y_k} \right)^{2p} & \text{if } s_k = \sigma y_k \\ I + \eta_{2p} (s_k y_k^\top + y_k s_k^\top) + \delta_{2p} s_k s_k^\top + \gamma_{2p} y_k y_k^\top & \text{otherwise.} \end{cases} \quad (3.28)$$

Ainsi, nous obtenons l'algorithme de descente du gradient conjugué sous la forme suivante

Algorithm 1 Généralisation des puissances de Perry (GPP)

Étape 1. Soit un point initial x_0 et $\varepsilon \geq 0$. Poser $k = 0$.

Étape 2. Calculer $g_0 = g(x_0)$. Si $\|g_k\| \leq \varepsilon$: stop, sinon non définir $d_0 = -g_0$ et aller a **Étape 3**.

Étape 3. Calculer la longueur de pas α_k avec les conditions de recherche de la ligne de Wolfe forte (3.4) et (3.5).

Étape 4. Poser $x_{k+1} = x_k + \alpha_k d_k$.

Étape 5. Calculer $g_{k+1} = g(x_{k+1})$.

Étape 6. Si $\|g_{k+1}\| \leq \varepsilon$, stop.

Étape 7. Calculer la direction d_{k+1} par (3.22) où M_{k+1} est calculé par (3.28).

Poser $k = k + 1$, et aller à **Étape 3**.

3.4 Résultat de la convergence globale

Nous analysons la convergence globale de notre algorithme, en utilisant l'outil de la théorie spectrale. Précédemment, on a introduit les hypothèses suivantes sur la fonction objective $f(x)$.

H1 : f est minorée dans \mathbb{R}^n et continûment différentiable dans un voisinage V de l'ensemble $S = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$, où x_0 est le point de départ de l'itération.

H2 : Le gradient de f est une fonction Lipschitzien continue sur V , par conséquent, il existe une constante $L > 0$ telle que

$$\|\nabla f(\tilde{x}) - \nabla f(x)\| \leq L\|\tilde{x} - x\|.$$

Lemme 3.4.1

*En supposant que les hypothèses **H1** et **H2** sont satisfaites, que la suite $\{x_k\}_k$ est générée par (3.2) et que les α_k sont déterminés de telle sorte que les conditions de Wolfe (3.4) et (3.5) soient vérifiées, alors la condition de Zoutendijk est vérifiée. Plus précisément on a*

$$\sum_{k=0}^{\infty} \frac{(g_k^\top d_k)^2}{\|d_k\|^2} < +\infty.$$

Preuve 3.4.1 Voir [6].

Le théorème ci-dessous permet d'illustrer la condition spectrale de la convergence globale de notre algorithme en se basant sur le lemme précédent.

Théorème 3.4.1 *Soit la fonction objective $f(x)$ satisfaisant les hypothèses H1 et H2. Pour la méthode du gradient conjugué non linéaire, sa suite itérative générée par (3.2) et sa direction de recherche linéaire sont calculées par*

$$\begin{cases} d_1 = -g_1 \\ d_{k+1} = -M_{k+1}g_{k+1}, \quad \forall k \geq 1, \end{cases} \quad (3.29)$$

telles que

i)- la condition de descente suffisante (3.6) soit vérifiée.

ii)- Les α_k soient déterminés de telle sorte que les conditions de Wolfe forte (3.4) et (3.5) soient vérifiées.

iii)- la condition spectrale soit vérifiée. Plus précisément

$$\sum_{k=1}^{\infty} (\lambda_{k+1}^+)^{-2} = +\infty, \quad (3.30)$$

où, λ_{k+1}^+ est la valeur propre maximale de la matrice M_{k+1} .

Alors

$$\liminf_{k \rightarrow \infty} \|g_{k+1}\| = 0. \quad (3.31)$$

de plus, si $\lambda_{k+1}^+ \leq \Lambda$, où Λ est une constante positive, alors

$$\lim_{k \rightarrow \infty} \|g_{k+1}\| = 0. \quad (3.32)$$

Remarque 3.4.1 Si la matrice M_{k+1} est une matrice symétrique définie positive, alors la condition spectrale (3.30) peut se réécrire sous la forme

$$\sum_{k=1}^{\infty} (\kappa_2(M_{k+1}))^{-2} = +\infty, \quad (3.33)$$

où κ_2 est la valeur propre maximale divisée par la valeur propre minimale notée le nombre de conditions spectrales de la matrice M_{k+1} .

Preuve 3.4.2

Par contradiction, Supposons que g_{k+1} n'est pas bornée, il existe $\gamma > 0$, par conséquent

$$\|g_{k+1}\| \geq \gamma, \quad \forall k \geq 1. \quad (3.34)$$

Alors,

D'après (3.29) et du fait que M_{k+1} est une matrice symétrique définie positive, il s'ensuit que

$$\|d_{k+1}\|^2 = g_{k+1}^\top M_{k+1}^\top M_{k+1} g_{k+1} \leq \lambda_{k+1}^+ \|g_{k+1}\|^2, \quad (3.35)$$

d'autre part nous avons,

$$\cos^2 \theta_k = \frac{(g_{k+1}^\top d_{k+1})^2}{\|d_{k+1}\|^2 \|g_{k+1}\|^2},$$

où, θ_k est l'angle entre d_{k+1} et $(-g_{k+1})$.

il résulte de (3.6) et (3.35) que

$$\cos^2 \theta_k = \frac{(g_{k+1}^\top d_{k+1})^2}{\|d_{k+1}\|^2 \|g_{k+1}\|^2} \geq \frac{(\lambda_{k+1}^-)^2 \|g_{k+1}\|^4}{(\lambda_{k+1}^+)^2 \|g_{k+1}\|^4} = (\kappa_2(M_{k+1}))^{-2}.$$

Par conséquent, à partir de (3.34), il s'en suit que

$$\sum_{k \geq 0} \|g_{k+1}\|^2 \cos^2 \theta_k \geq \gamma^2 \sum_{k=0}^{\infty} (\kappa_2(M_{k+1}))^{-2} = +\infty, \quad (3.36)$$

ce qui contredit la condition du Zoutendijk

$$\sum_{k \geq 0} \|g_{k+1}\|^2 \cos^2 \theta_k \leq \sum_{k=0}^{\infty} \frac{(g_{k+1}^\top d_{k+1})^2}{\|d_{k+1}\|^2} < +\infty \quad (3.37)$$

Cette dernière contradiction implique que les résultats du Lemme 2.4.1 sont vrais.

Donc, $\liminf_{k \rightarrow \infty} \|g_{k+1}\| = 0$.

3.5 Résultats numériques

Pour montrer l'efficacité de notre nouvelle version de l'algorithme GPP (Généralisation des puissances de Perry) nous allons mener une comparaison de ce dernier avec les algorithmes classique : CG-DESCENT de Hager et Zhang [24], mBFGS [40] et l'algorithme SPDOC [28].

Nous allons présenter la direction des trois algorithmes :

L'algorithme CG-DESCENT de Hager et Zhang [24] et les directions de recherche sont formulées par

$$\begin{cases} d_1 = -g_1 \\ d_{k+1} = -g_{k+1} + \beta_k d_k, \quad \forall k \geq 1 \end{cases}$$

$$\text{où, } \beta_k = \max \{ \beta_k^N, \eta_k \}, \quad \eta_k = \frac{1}{\|d_k\| \min \{ \eta, \|g_k\| \}},$$

$$\beta_k^N = \left(y_k - 2 \frac{\|y_k\|^2 d_k}{d_k^\top y_k} \right)^\top \frac{g_k}{d_k^\top y_k},$$

avec $\eta = .01$, aussi avec l'algorithme mBFGS [40] et la direction de recherche est définie par

$$\begin{cases} d_1 = -g_1 \\ d_{k+1} = -Q_{k+1}^{mBFGS} g_{k+1}, \quad \forall k \geq 1, \end{cases}$$

sachant que

$$Q_{k+1}^{mBFGS} = I - \frac{s_k y_k^\top + y_k s_k^\top}{s_k^\top y_k} + \left(1 + \frac{y_k^\top y_k}{s_k^\top y_k}\right) \frac{s_k s_k^\top}{s_k^\top y_k},$$

et l'algorithme SPDOC [28] et la direction de recherche est définie par

$$\begin{cases} d_1 = -g_1 \\ d_{k+1} = -Q_{k+1}^{SPDOC} g_{k+1}, \quad \forall k \geq 1, \end{cases}$$

sachant que

$$Q_{k+1}^{SPDOC} = \left(I - \frac{s_k y_k^\top}{s_k^\top y_k}\right) \left(I - \frac{y_k s_k^\top}{s_k^\top y_k}\right) + \frac{y_k^\top y_k}{s_k^\top y_k} \frac{s_k s_k^\top}{s_k^\top y_k}.$$

Pour déterminer les performances de tous les algorithmes sur un ensemble de problèmes de test d'optimisation sans contraintes [8], chaque problème est testé en faisant varier le nombre de variable dont la dimension n à savoir : $n = 2, n = 10, n = 50, n = 100, n = 1000, n = 1500, n = 2000, n = 5000$ et $n = 10000$ pour un nombre total de problèmes de test égale à 80.

Nous les exécutons sur un PC ayant les caractéristiques suivantes : Intel(R) core (TM)i5 Processeur 650 @ 3,20 GHz, 3,00 Go de RAM.

En utilisant les conditions de recherche linéaire de Wolfe forte avec $\delta = 0.0001, \sigma = 0.1$ et le critère d'arrêt pour tous les algorithmes $\|g_k\|^2 \leq 10^{-6}$, nous adoptons les profils de performance donnés par Dolan et More [15] afin de comparer les performances.

Avant tout cela, en se référant à la figure 3.1 nous constatons que notre méthode GPP pour une puissance $p = 3$ plus performante que pour les puissances $p = 1, p = 4$.

D'autre par les figure 3.2, 3.3 et 3.4 respectivement représentent le profil de performance mesuré par le temps CPU, le nombre d'itérations et l'évaluations des fonctions test et de leur gradients. Généralement tous les graphes

montrent que notre algorithme GPP proposé est plus performant que les algorithmes CG-DESCENT, mBFGS et SPDOC par rapport au temps CPU, au nombre d'itérations et à l'évaluation des fonctions test et de leurs gradients.

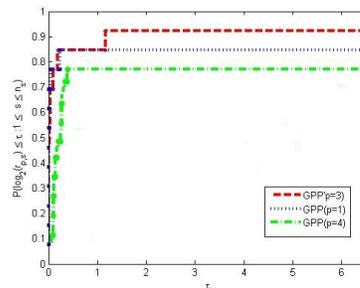


FIGURE 3.1 – Profil de performance pour le temps CPU

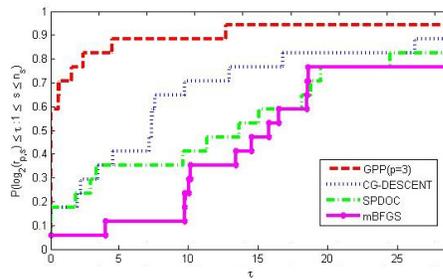


FIGURE 3.2 – Profil de performance pour le temps CPU

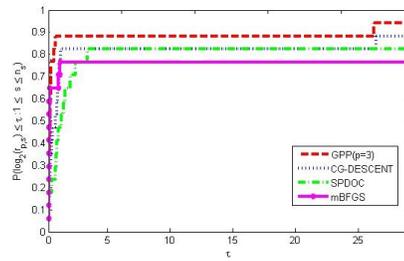


FIGURE 3.3 – Profil de performance pour le nombre d'itérations

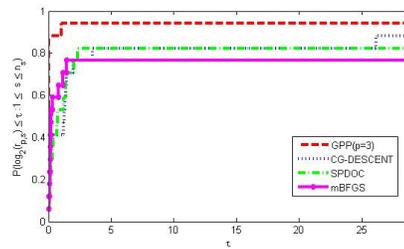


FIGURE 3.4 – Profil de performance pour le nombre d'évaluations de fonctions et de gradients.

Chapitre 4

Conclusion et Perspectives

L'objectif de notre travail est d'étudier les méthodes d'optimisation sans contrainte et d'améliorer leur accélération selon un procédé comparative. En particulier, nous sommes consacrés à l'étude des méthodes du gradient conjugué non linéaires en essayant de trouver la meilleure méthode à convergence rapide.

Pour répondre à ce souci nous avons développé trois algorithmes qui permettent la résolution de ce type de problème d'optimisation non linéaire.

En premier temps, nous avons élaboré l'hybridation des deux fameuse méthodes DY et HZ en se basant sur la combinaison linéaire des celles-ci, de telle sorte que les conditions de descente suffisantes et de conjugaison soient vérifiées.

Ensuite, notre hybridation nous a permis de construire une autre méthode plus performantes qui vérifie les conditions de descente suffisantes et de conjugaison généralisée.

Nous avons construit, enfin, une autre méthode, efficace numériquement, en utilisant l'idée de la méthode de Perry moyennant certains modifications,

ce qui a mené finalement à une combinaison linéaire, facteur essentielle pour l'amélioration de l'accélération de la convergence. Notre méthode construite a l'avantage d'assurer la condition de descente suffisante pour toute recherche linéaire

Nous avons démontré que nos trois nouveaux algorithmes construits, accélèrent la convergence, de façon significative et qu'ils sont numériquement plus performants comparativement par rapport aux algorithmes standard à partir desquels ils ont été conçus.

Pour montrer les avantages de notre combinaison, nous avons effectué des tests numériques qui montrent que les deux conditions sont équivalentes avec une petite différence pour les grandes tailles, puisqu'elle permet d'aboutir à une combinaison linéaire facteur essentielle pour l'amélioration de l'accélération de la convergence.

Ainsi, les perspectives futures dans un premier temps La finalisation de la publication des travaux qui concernent la nouvelle hybridation avec la condition de conjugaison et la condition de conjugaison généralisée.

L'utilisation d'autres méthodes et conditions d'amélioration de performance de la direction de descente.

L'application de ces techniques dans des problèmes et des modèles physiques concrets.

Bibliographie

- [1] Abubakar, A. B., Sabi'u, J., Kumam, P., & Shah, A. (2021). Solving nonlinear monotone operator equations via modified sr1 update. *Journal of Applied Mathematics and Computing*, 1-31.
- [2] Al-Baali, Mehiddin. Descent property and global convergence of the Fletcher-Reeves method with inexact line search. *IMA Journal of Numerical Analysis* 5.1 (1985) : 121-124.
- [3] Andrei, Neculai, *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization*. 2020, Springer *Optimization and Its Applications* 2020 : 158, Springer.
- [4] Andrei, N. (2017). Accelerated adaptive Perry conjugate gradient algorithms based on the self-scaling memoryless BFGS update. *Journal of Computational and Applied Mathematics*, 325,149-164.
- [5] Andrei, Neculai. "On three-term conjugate gradient algorithms for unconstrained optimization." *Applied Mathematics and Computation* 219.11 (2013) : 6316-6327.
- [6] Andrei, N.(2010) Another accelerated conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization. No. 7. ICI Technical Report, .

- [7] Andrei, Neculai. Hybrid conjugate gradient algorithm for unconstrained optimization. *Journal of Optimization Theory and Applications* 141.2 (2009) : 249-264.
- [8] Andrei, N.(2008). An unconstrained optimization test functions collection. *Adv. Model. Optim* 10.1 :147-161.
- [9] Andrei, N.(2007). Scaled conjugate gradient algorithms for unconstrained optimization. *Computational Optimization and Applications* 38.3 : 401-416.
- [10] Armijo, Larry. "Minimization of functions having Lipschitz continuous first partial derivatives." *Pacific Journal of mathematics* 16.1 (1966) : 1-3.
- [11] Baluch, B., Salleh, Z., & Alhawarat, A. (2018). A new modified three-term Hestenes-Stiefel conjugate gradient method with sufficient descent property and its global convergence. *Journal of Optimization*.
- [12] Benrabia, N., Laskri, Y., Guebbai, H., and Al-Baali, M. Applying the Powell's symmetrical technique to conjugate gradient methods with the generalized conjugacy condition. *Numerical Functional Analysis and Optimization* 37.7 (2016) : 839-849.
- [13] Cardenas, S. (1991). Efficient generalized conjugate gradient algorithms. I. Theory. *J. Optim. Theory Appl*, 69(1), 129-137.
- [14] Dongyi, L., & Genqi, X. (2011). Applying Powell's symmetrical technique to conjugate gradient methods. *Computational Optimization and Applications*, 49(2), 319-334.
- [15] Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2), 201-213.

- [16] Dai, Yu-hong and Yuan, Yaxiang. An efficient hybrid conjugate gradient method for unconstrained optimization. *Annals of Operations Research* 103.1-4 (2001) : 33-47.
- [17] Dai, Yu-Hong, Han, Jiye, Liu, Guanghui. Convergence properties of nonlinear conjugate gradient methods. *SIAM Journal on Optimization* 10.2 (2000) : 345-358.
- [18] Dai, Yu-Hong, and Yuan, Yaxiang. A nonlinear conjugate gradient method with a strong global convergence property, (1999) : 177-182.
- [19] Fletcher, R., & Powell, M. J. (1963). A rapidly convergent descent method for minimization. *The computer journal*, 6(2), 163-168.
- [20] Gilbert, Jean Charles, and Jorge Nocedal. "Global convergence properties of conjugate gradient methods for optimization." *SIAM Journal on optimization* 2.1 (1992) : 21-42.
- [21] Goldstein, A. A., and J. F. Price. "An effective algorithm for minimization." *Numerische Mathematik* 10.3 (1967) : 184-189.
- [22] Gould, Nicholas IM, Dominique Orban, and Philippe L. Toint. "CUTEr and SifDec : A constrained and unconstrained testing environment, revisited." *ACM Transactions on Mathematical Software (TOMS)* 29.4 (2003) : 373-394.
- [23] Hafaidia, I., Benrabia, N., Ghiat, M., et Guebbai, H. (2021). A new optimization method based on Perry's idea through the use of the matrix power. *Journal of Applied Mathematics and Computational Mechanics*, 20(4), 29-41.

- [24] Hager, W. W., & Zhang, H. (2006). Algorithm 851 : CG-DESCENT, a conjugate gradient method with guaranteed descent. *ACM Transactions on Mathematical Software (TOMS)*, 32(1), 113-137.
- [25] Hager, W. W., & Zhang, H. (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, 16(1), 170-192.
- [26] Hestenes, M. R., & Stiefel, E. (1952). *Methods of conjugate gradients for solving linear systems (Vol. 49, No. 1)*. Washington, DC : NBS.
- [27] Livieris, I. E., & Pintelas, P. (2015). A modified Perry conjugate gradient method and its global convergence. *Optimization Letters*, 9(5), 999-1015.
- [28] Liu, D., & Xu, G. (2013). Symmetric Perry conjugate gradient method. *Computational Optimization and Applications*, 56(2), 317-341.
- [29] Liu, Y., Storey, C. Efficient generalized conjugate gradient algorithms, Part1. *Theory. J. Optim. Theory. Appl.* 69, 129-137 (1991)
- [30] Perry, A. (1978). A modified conjugate gradient algorithm. *Operations Research*, 26(6), 1073-1078.
- [31] Polak, E., & Ribiere, G. (1969). Note sur la convergence de méthodes de directions conjuguées. *ESAIM : Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 3(R1), 35-43.
- [32] Polyak, B. (1969). The conjugate gradient method in extreme problems, *USSR Computational Mathematics and Mathematical Physics.*, 9, 94-112.

- [33] Powell, M. J. D. (1977). Restart procedures for the conjugate gradient method. *Mathematical programming*, 12(1), 241-254.
- [34] Powell, M. J. (1984). Nonconvex minimization calculations and the conjugate gradient method. In *Numerical analysis* (pp. 122-141). Springer, Berlin, Heidelberg.
- [35] Pytlak, Radosław, *Conjugate Gradient Algorithms in Nonconvex Optimization*. 2009 : 89, *Nonconvex Optimization and Its Applications*, Springer.
- [36] Sabi'u, J., Muangchoo, K., Shah, A., Abubakar, A. B., & Aremu, K. O. (2021). An Inexact Optimal Hybrid Conjugate Gradient Method for Solving Symmetric Nonlinear Equations. *Symmetry*, 13(10), 1829.
- [37] Salleh, Z., Alhamzi, G., Masmali, I., & Alhawarat, A. (2021). A Modified Liu and Storey Conjugate Gradient Method for Large Scale Unconstrained Optimization Problems. *Algorithms*, 14(8), 227.
- [38] Sabi'u, J., Shah, A., & Waziri, M. Y. (2020). Two optimal Hager-Zhang conjugate gradient methods for solving monotone nonlinear equations. *Applied Numerical Mathematics*, 153, 217-233.
- [39] Sabi'u, J., Shah, A., & Waziri, M. Y. (2021). A modified Hager-Zhang conjugate gradient method with optimal choices for solving monotone nonlinear equations. *International Journal of Computer Mathematics*, 1-23.
- [40] Shanno, D. F. (1978). Conjugate gradient methods with inexact searches. *Mathematics of operations research*, 3(3), 244-256.

- [41] Touati-Ahmed, D., and C. Storey. "Efficient hybrid conjugate gradient techniques." *Journal of optimization theory and applications* 64.2 (1990) : 379-397.
- [42] Waziri, M. Y., Ahmed, K., & Sabi'u, J. (2019). A family of Hager-Zhang conjugate gradient methods for system of monotone nonlinear equations. *Applied mathematics and Computation*, 361, 645-660.
- [43] Waziri, M. Y., Hungu, K. A., & Sabi'u, J. (2020). Descent Perry conjugate gradient methods for systems of monotone nonlinear equations. *Numerical Algorithms*, 85(3), 763-785.
- [44] Wolfe, P. (1969). Convergence conditions for ascent methods. *SIAM review*, 11(2), 226-235.
- [45] Yao, S., He, D., & Shi, L. (2018). An improved Perry conjugate gradient method with adaptive parameter choice. *Numerical Algorithms*, 78(4), 1255-1269.
- [46] Zhou, W. (2020). A modified BFGS type quasi-Newton method with line search for symmetric nonlinear equations problems. *Journal of Computational and Applied Mathematics*, 367, 112454.
- [47] Zoutendijk, G.(1970). Nonlinear programming, computational methods. *Integer and nonlinear programming*.37-86.