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

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

Mémoire

Présenté en vue de l'obtention du diplôme de

Master en Mathématiques

Option : **EDP Et Analyse numérique**

Par : HANNACHI Marwa

Intitulé

Sur
L'hybridation des méthodes du gradient conjugué

Dirigé par : Pr. GUEBBAI Hamza

Devant le jury

PRESIDENT	**Dr. TABOUCHE Nora**	**M.C.B Univ-Guelma**
RAPPORTEUR	**Pr. GUEBBAI Hamza**	**Professeur Univ-Guelma**
EXAMINATEUR	**Dr. MEHRI ALLAOUA**	**M.C.A Univ-Guelma**

Session septembre 2020

Hybridation des méthodes du gradient conjugué

Hannachi Marwa
Mémoire de fin d'études
Universté 08 Mai 1945-Guelma

8 octobre 2020

Table des matières

Remerciement

En tout premier lieu, je remercie le bon **Dieu** , tout puissant, de m'avoir donné la force, le courage, la patience et la motivation pour donner de mon mieux et accomplir ce travail.

Aussi,j'adresse mes sincères remerciements et ma grande gratitude à mon directeur de mémoire **Monsieur Guebbai Hamza** qui m'a toujours aidée et guidée . Il a toujours été disponible pour m'orienter. Je n'aurai jamais pu réaliser ce mémoire ces précieux conseils.

Je voudrais exprimer ma reconnaissance aussi à **Hafaidia Imen**, pour son aide précieuse, sa patience avec moi et soutien inestimable durant toute la période de la préparation de ce travail.

Je remercie infiniment toute l'équipe du **Laboratoire de Mathématiques Appliquées et de Modélisation** au sein de notre université.

Dédicace

Je dédie ce mémoire avec fierté, amour et une immense joie,

À **mes parents**, la prunelle des mes yeux, ceux qui ont toujours été à mes côtés pour me rassurer, m'encourager, me soutenir, me gâter et surtout pour me remplir la vie de joie . Je vous dois tout dans ce monde j'espère que je vous ai rendu fiers de moi avec ce que j'ai réussi à accomplir avec ce modeste travail.

À **Nadir** mon frère, celui qui m'a toujours poussé à être forte et à ne jamais baisser les bras, qui m'a toujours comblé de sa tendresse et sa protection.

À l'âme de **ma grand-mère** chérie, que le bon Dieu l'accueille dans son vaste paradis, celle qui nous a grandis , nous a éduqués et nous a aimés de tout son cœur, je voudrai tellement que tu seras parmi nous jusqu'au jour où assister à ma soutenance mais malheureusement Dieu a voulu autrement .

À **Ilhem** ma moitié et ma meilleure amie, celle qui m'as accompagnée durant les pires et les meilleures moments de ma vie. À **Bibou** ma source de motivation qui est toujours là pour moi depuis notre enfance, ma soeur de coeur que j'aime tant. À **Soumia, Nada, Khawla** et **Lamis** qui m'ont tant encouragée à toujours donner de mon mieux pour y arriver.

À toute ma famille, mes amis, et toute personne qui m'a souhaité le succès.

Résumé

Dans ce mémoire, nous nous intéressons à résoudre un problème d'optimisation sans contrainte en construisant une nouvelle méthode du gradient conjugué en utilisant une recherche linéaire inexacte qui est : une pseudo-convexe hybridation de trois vecteurs de descente suivants : le vecteur de descente de la méthode de Lui-Storey (LS), le vecteur de descente de Polak-Ribière Polyak (PRP) et celui de Dai-Yuan (DY) afin de construire une nouvelle méthode du gradient conjugué pour l'optimisation sans contraintes qui satisfait la condition de conjugaison et la condition de descente suffisante, on démontre la descente et la convergence de cette nouvelle méthode .

Mots clés : méthodes du gradient conjugué, recherche linéaire, condition de Zoutendjik, condition de conjugaison.

Abstract

In this thesis, we are interested in solving optimization problems without constraints by building a new method of conjugate gradient method using a line search which is : the pseudo-convex hybridization of the following three descent vectors : the descent vector of ui-Storey (LS), the descent vector of Polak-Ribière Polyak (PRP), and the descent vector of Dai-Yuan (DY) in order to build a new gradient conjugate method for without constraints problems that satisfies condition of conjugation and the condition of sufficient descent. Also, we prove the descent and the convergence of that new method .

Key words : conjugate gradient method, line search, Zoutendijk's condition, condition of conjugation.

Introduction

L'optimisation est une branche des mathématiques cherchant à modéliser, analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à minimiser ou à maximiser une fonction avec ou sans contraintes. L'optimisation possède ses racines qui remonte 18ième siècle dans les travaux de :

-Taylor, Newton , Lagrange, qui ont élaboré les bases des développements limités. - Cauchy, qui était le premier à mettre en oeuvre une méthode d'optimisation, méthode du pas de descente, pour la résolution de problèmes sans contraintes.

Après la fin de la seconde guerre mondiale des avancées spectaculaires en termes de techniques d'optimisation sont apparues. A noter, ces avancées ont été essentiellement obtenues en Grande Bretagne. Parmi les plus anciennes méthodes utilisées pour résoudre cet problème du type , on peut citer la méthode du Gradient conjugué [Voir [1] [5] [20]]. Cette méthode est surtout utilisée pour les problèmes de grande taille. Cette méthode a été découverte en 1952 par Hestenes et Steifel , pour la minimisation de fonctions quadratiques strictement convexes. Plusieurs mathématiciens ont étendu cette méthode pour le cas non linéaire. Cela a été réalisé pour la première fois, en 1964 par Fletcher et Reeves (méthode de Fletcher-Reeves) puis en 1969 par

Polak, Ribière et Polyak (méthode de Polak- Ribière-Polyak). Une autre variante a été étudiée en 1987 par Fletcher (Méthode de la descente conjuguée).

Le travail dans ce mémoire est planifié comme suit :
Dans **le premier chapitre**, on rappelle les concepts mathématiques à appliquer au domaine d'optimisation.

Dans **le deuxième chapitre** on aborde les grandes lignes des méthodes d'optimisation sans contraintes basées sur les directions de descente et les recherches linéaires.

Dans **le troisième chapitre**, qui est le noyau de notre travail, on expose la descente de notre pseudo-convexe hybridation et on démontre sa convergence.

Dans **le quatrième chapitre**, on met les résultats des tests numérique effectuée pour la comparaison de notre nouvelle méthode avec la méthode de l'hybridation convexe des deux vecteurs de descente du PRP et DY où on observe clairement la rapidité et l'efficacité de notre méthode par rapport l'autre.

Chapitre 1

Définitions et rappels

1.1 Position du problème

L'optimisation est la minimisation ou la maximisation d'une fonction vectorielle à image dans $\mathbb{R}$ ($\mathbb{R}^n \to \mathbb{R}$) assez régulière. Un problème d'optimisation linéaire demande de minimiser une fonction linéaire sur un domaine connu. La fonction que l'on minimise ainsi que les contraintes sont décrites par des fonctions linéaires, d'où le nom donné à ces problèmes. Et Le présent travail ne se focalise que sur les problèmes d'optimisation sans contraintes et non linéaires. Ce problème prend la forme suivante :

$$(P): \quad \min\{f(x); x \in \mathbb{R}^n\} \tag{1.1}$$

où $f : \mathbb{R}^n \to \mathbb{R}$.

La méthode de résolution d'optimisation qu'on va utiliser est de nature itérative, c'est à dire qu'à partir d'un point initial x_0, on construit une suite $x_1, x_2, .., x_k..$ qu'on doit démontrer sa convergence vers la solution optimale.

1.2 Conditions d'optimalité des problèmes de minimisation sans contraintes

Pour résoudre un problème d'optimisation nous avons besoin de conditions d'optimalité qui sont l'outil principal pour trouver les solutions les plus efficaces à notre problème.

1.2.1 Le minimum local et le minimum global

Définition 1.2.1

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

$$f(\hat{x}) \leq f(x) \quad ; \forall x \in V_\varepsilon(\hat{x}) \tag{1.2}$$

2. *$\hat{x} \in \mathbb{R}^n$ est un minimum local strict de (P) si et seulement si il existe un voisinage $V_\varepsilon(\hat{x})$ tel que*

$$f(\hat{x}) < f(x) \quad ; \forall x \in V_\varepsilon(\hat{x}), \quad x \neq \hat{x} \tag{1.3}$$

3. *$\hat{x} \in \mathbb{R}^n$ est un minimum global de (P) si et seulement si*

$$f(\hat{x}) \leq f(x) \quad ; \forall x \in \mathbb{R}^n \tag{1.4}$$

Théorème 1.2.1 *Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction convexe définie sur l'ensemble convexe X. Alors, tout minimum local de f est également un minimum global. Si f est strictement convexe, alors il existe au plus un minimum global de f.*

Preuve 1.2.1 *Voir [19]*

Remarque 1.2.1

1. *La qualité du minimum local dépend de la "taille" de* $V_\varepsilon(\hat{x})$. *Plus* $V_\varepsilon(\hat{x})$ *est "grand", mieux c'est*
2. *Si* $\hat{x}$ *est un minimum global, alors c'est un minimum local.*
3. *Si on connait tous les minimums locaux alors le plus petit est le minimum global.*

1.2.2 Une direction de descente

Considérons le problème d'optimisation sans contraintes :

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

où $f : \mathbb{R}^n \to \mathbb{R}$ est supposée régulière.
Notons $\nabla f(x)$ et $\nabla^2 f(x)$ respectivement le gradient et la matrice hessienne de f en x.
On dit que d est une direction de descente de f en $x \in \mathbb{R}^n$ si

$$\nabla f(x)^T.d < 0. \tag{1.5}$$

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

$$\{d \in \mathbb{R}^n \backslash \nabla f(x)^T.d < 0\} \tag{1.6}$$

Exemple de choix du direction de descente

Direction du gradient

La direction du gradient d est, en réalité, l'opposé du gradient :

$$d = -\nabla f(x)$$

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

$$\nabla f(x) \neq 0$$

Puisque :

$$\nabla f(x)^T \cdot (-\nabla f(x)) = \langle \nabla f(x), -\nabla f(x) \rangle = -\|\nabla f(x)\|^2 < 0$$

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

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

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

1.2.4 Schémas général des algorithmes d'optimisation sans contraintes

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

$$x_{k+1} = x_k + \lambda_k d_k, \qquad \forall \lambda_k \in]0, +\infty[.$$

Selon la définition de la direction de descente, on est assuré que :

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

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

Exemple de choix de pas λ_k :

Notre choix du λ_k est généralement d'une façon optimale c'est à dire : λ_k doit vérifier :

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

Autrement dit, on est ramenés à étudié à chaque itération un problème de minimisation d'une variable réelle. C'est ce qu'on appelle la recherche linéaire.

1.2.5 Conditions nécessaires d'optimalité

Prenons un vecteur $\hat{x}$, nous voulons déterminer si ce vecteur est un minimum local ou global de (P). La propriété de différentiabilité de f fournit une première manière de caractériser une solution optimale. Énonçons tout d'abord un théorème pour établir une première condition nécessaire d'optimalité.

Condition nécessaire d'optimalité du premier ordre

Théorème 1.2.2 *Soit $f : \mathbb{R}^n \to \mathbb{R}$ différentiable au point $\hat{x} \in \mathbb{R}^n$. Si $\hat{x}$ est un minimum local de (P) alors $\nabla f(\hat{x}) = 0$.*

Preuve 1.2.2 *Supposons que $\nabla f(\hat{x}) \neq 0$. Puisque la direction $d = \nabla f(\hat{x})$ est une direction de descente, alors il existe $\delta > 0$ tel que*

$$f(\hat{x} + \lambda_k d_k) < f(\hat{x}) \qquad \forall \lambda_k \in]0, \delta[$$

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

Remarque 1.2.2 *Si f est convexe, la condition nécessaire du premier ordre est également suffisante pour que $\hat{x}$ soit un minimum global. Dans le cas où f est deux fois différentiable, une autre condition nécessaire est donnée par le théorème (1.2.3). Elle est appelée condition nécessaire du second ordre car elle fait intervenir la matrice hessienne de f (que nous noterons $\nabla^2 f(x)$, dont les éléments sont ses secondes dérivées partielles).*

Condition nécessaire d'optimalité du second ordre

Théorème 1.2.3 *Soit $f : \mathbb{R}^n \to \mathbb{R}$ deux fois différentiable au point $\hat{x} \in \mathbb{R}^n$. Si $\hat{x}$ est un minimum local de (P) alors $\nabla f(\hat{x}) = 0$ et la matrice hessiene de f au point $\hat{x}$ est semi définie positive.*

1.2.6 Conditions suffisantes d'optimalité

Les conditions données précédemment sont nécessaires (si f n'est pas convexe), c'est à dire qu'elles doivent être satisfaites pour tout minimum local, cependant, tout vecteur vérifiant ces conditions n'est pas nécessairement un minimum local. Le théorème (1.2.4) établit une condition suffisante pour qu'un vecteur soit un minimum local, si f est deux fois différentiable.

Condition suffsante d'optimalité du second ordre

Théorème 1.2.4 *Soit $f : \mathbb{R}^n \to \mathbb{R}$ deux fois différentiable au point $\hat{x} \in \mathbb{R}^n$. Si $\nabla f(\hat{x}) = 0$ et la matrice hessiene de f au point $\hat{x}$ est définie positive,alors $\hat{x}$ est un minimum local strict de (P).*

Remarque 1.2.3 *Si f est convexe, alors tout minimum local est aussi global. De plus si f est strictement convexe, alors tout minimum local devient non seulement global mais aussi unique.*

1.3 Règles de recherche linéaire

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

Considérons le problème d'optimisation sans contraintes (P) :

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

où $f : \mathbb{R}^n \to \mathbb{R}$.

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

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

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

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

c'est à dire, λ_k vérifie

$$f(x_k + \lambda_k d_k) \leq f(x_k + \lambda d_k), \qquad \forall \lambda > 0. \tag{1.9}$$

x_k et d_k sont fixés et la fonction à minimiser est une fonction d'une variable réelle défnie comme suit :

$$\lambda \to \varphi(\lambda) = f\,(x_k + \lambda d_k) \tag{1.10}$$

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 :

1. les recherches linéaires exactes.
2. les recherches linéaires inexacte.

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

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

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

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

Remarque 1.3.1

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 et cela peut être dissuatif du point de vue du temps de calcul. En pratique, on chercher 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 dans un nombre fini d'itérations.*

La recherche linéaire exacte dans le cas des fonctions quadratiques

Soit Q est une matrice (n, n) symétrique et définie positive et $b \in \mathbb{R}^n$ et $f(x) = \frac{1}{2}x^T Qx - b^T x$. Considérons le problème $(PQSC)$

$$\min_{x\in\mathbb{R}^n} f(x) = \min_{x\in\mathbb{R}^n} \left\{ \tfrac{1}{2}x^T Qx - b^T x \right\} \qquad (PQSC).$$

Les méthodes à directions de recherche linéaire génèrent des suites $\{x_k\}_{k=1,2,3...}$ de la manière suivante. On démarre par $x_1 \in \mathbb{R}^n$ à l'itération k si on a $x_k \in \mathbb{R}^n$. Le successeur x_{k+1} de x_k est donné par la relation suivante :

$$x_{k+1} = x_k + \lambda_k d_k \tag{1.11}$$

où, $d_k \in \mathbb{R}^n$ est une direction de recherche et $\lambda_k \in \mathbb{R}^+$ est le pas de recherche obtenu par une recherche linéaire exacte ou inexacte. Dans le cas d'une recherche linéaire exacte λ_k vérifie

$$f(x_k + \lambda_k d_k) = \min_{\lambda>0} \{ f(x_k + \lambda d_k)\} \tag{1.12}$$

Notons

$$g_k = \nabla f(x_k) = Qx_k - b \tag{1.13}$$

Théorème 1.3.1 *Soit Q est une matrice (n, n) symétrique et définie positive et $b \in \mathbb{R}^n$ et $f(x) = \frac{1}{2}x^T Qx - b^T x$. Considérons le problème $(PQSC)$*

$$\min_{x\in\mathbb{R}^n} f(x) = \min_{x\in\mathbb{R}^n} \left\{ \tfrac{1}{2}x^T Qx - b^T x \right\} \qquad (PQSC).$$

Supposons qu'à l'itération k on a une direction d_k de descente, c'est à dire que d_k vérifie

$$g_k^T d_k = (Qx_k - b)^T d_k < 0 \tag{1.14}$$

soit $\lambda_k > 0$ obtenu par une recherche linéaire exacte c'est à dire que λ_k vérifie :

$$f(x_k + \lambda_k d_k) = \min_{\lambda > 0} \{ f(x_k + \lambda d_k)\} \tag{1.15}$$

Alors

$$\lambda_k = -\frac{g_k^T d_k}{d_k^T Q d_k} \tag{1.16}$$

Preuve 1.3.1 *Considérons*

$$\varphi(\lambda) = f(x_k + \lambda d_k) \tag{1.17}$$

Donc λ_k est la solution optimale du problème unidimensionnel suivant

$$\varphi(\lambda_k) = \min_{\lambda > 0} \varphi(\lambda). \tag{1.18}$$

Q est définie positive, alors $f(x) = \frac{1}{2}x^T Q x - b^T x$ est strictement convexe sur $\mathbb{N}^n$. Par conséquent $\varphi(\lambda) = f(x_k + \lambda d_k)$ est strictement convexe sur $]0, +\infty[$ qui est un ouvert convexe. Donc λ_k est une solution optimale de (1.18) si et seulement si λ_k vérifie $\varphi'(\lambda_k) = 0$. Or

$$\begin{aligned}\varphi'(\lambda) &= \nabla f(x_k + \lambda d_k)^T d_k = [Q(x_k + \lambda d_k) - b]^T d_k \\ &= [Qx_k - b + \lambda Q d_k]^T d_k = [g_k + \lambda Q d_k]^T d_k \\ &= g_k^T d_k + \lambda d_k^T Q d_k.\end{aligned}$$

Donc

$$\varphi'(\lambda) = 0 \Longleftrightarrow \lambda d_k^T Q d_k = -g_k^T d_k,$$

c'est à dire

$$\lambda_k = -\frac{g_k^T d_k}{d_k^T Q d_k}.$$

Remarque 1.3.2 *Si Q n'est pas définie positive mais seulement semi définie positive, alors f n'est pas strictement convexe, et $d_k^T Q d_k$ peut être égale à zéro. Par conséquent λ_k ne sera plus définie par* (1.16).

1.3.2 La recherche linéaire inexacte

Au lieu de demander que λ_k minimise φ, 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 unique pas (ou quelques pas) vérifiant ces conditions mais tout un intervalle de pas (ou plusieurs intervalles), ce qui rendra d'ailleurs leur recherche plus aisée. C'est ce qu'on fait avec les règles d'Armijo, de Goldstein et de Wolfe.

Règle d'Armijo

La règle d'Armijo [Voir [17]] se base sur le choix d'un paramètre :

$$0 < \rho < 1$$

et détermine une valeur approchée de λ_k par la condition :

$$\varphi_k(\lambda) \leqslant \varphi_k(0) + \rho\lambda\varphi_k'(0)$$

Le risque de cette méthode est de favoriser les valeurs trop petites, aussi, elle est rarement utilisée seule.

Règle de Goldstein

C'est une méthode qui a été proposée en 1967 et qui est basée cette fois ci sur le choix de deux paramètres

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

et d(étermine les valeurs approchées de λ_k par deux conditions

$$\begin{cases} \varphi_k(\lambda) \leqslant \varphi_k(0) + \rho\lambda\varphi_k'(0) \\ \varphi_k(\lambda) \geqslant \varphi_k(0) + \delta\lambda\varphi_k'(0) \end{cases}$$

Règle de Wolfe

C'est une méthode qui a été proposée en 1969 et qui est basée cette fois ci sur le choix de deux paramètres

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

et détermine les valeurs approchées de λ_k par deux conditions :

$$\varphi_k(\lambda) \leqslant \varphi_k(0) + \rho\lambda\varphi_k'(0) \tag{1.19}$$

$$\varphi_k'(\lambda) \geqslant \delta\varphi_k'(0) \tag{1.20}$$

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 couteuse que la règle de Goldstein. Cependant dans de nombreuses applications, le calcul du gradient $\nabla f(x)$ représente un faible cout additionnel en comparaison du cout 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 (1.20).Pour cela la deuxième condition* (1.20) *est remplacée par :*

$$|\varphi_k'(\lambda)| \leq -\delta\varphi_k'(0). \tag{1.21}$$

On obtient alors les conditions de Wolfe fortes (1.19) *et* (1.21).

1.4 La méthode du gradient conjugué

En analyse numérique, la méthode du gradient conjugué est un algorithme pour résoudre des systèmes d'équations. Cette méthode, imaginée en 1950 simultanément par Cornelius Lanczos et Magnus Hestenes, est une méthode itérative qui converge en un nombre fini d'itérations.
Cette méthode génère une suite $\{x_n\}_{n\in\mathbb{N}}$ telle que :

$$x_{k+1} = x_k + \lambda_k d_k \tag{1.22}$$

Le pas $\lambda_k \in \mathbb{R}$ est déterminé par une optimisation unidimensionnelle ou une recherche linéaire exacte ou inexacte.
Les directions d_k sont calculées de façon récurrente par les formules suivantes :

$$d_k = \begin{cases} -g_0 \quad si \quad k = 0 \\ -g_k + \beta_k d_{k-1} \quad si \quad k \geq 1 \end{cases}$$

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.

1.4.1 La méthode du gradient conjugué non linéaire

Pour réussir à minimiser une fonction non linéaire f en utilisant l amé-thode du gradient conjugué nous devons suivre les étapes suivantes :

1. Faire une recherche linéaire afin de calculer le pas $\lambda_k$qui minimise notre fonction f le long de d_k.
2. Le résidu r doit être remplacé par le gradient de f.

Nous obtenons ainsi un algorithme très efficace pour l'optimisation non linéaire. Le paramètre λ_k doit vérifier certaines conditions pour assurer que la direction d_{k+1} soit une direction de descente. Si la recherche linéaire est exacte, nous voyons que :

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

Mais si la recherche n'est pas exacte (parce que la recherche exacte est trop coûteuse), il faut imposer d'autres conditions.
L'idée de la méthode est de construire itérativement des directions $d_0, ..., d_k$ mutuellement conjuguées. A chaque étape k la direction d_k est obtenue comme combinaison linéaire du gradient en x_k et de la direction précédente $dk - 1$, les coefficients étant choisis de sorte que d_k soit conjuguée avec toutes les directions précédentes.
On s'intéresse ici à la minimisation d'une fonction

$$f : \mathbb{R}^n \to \mathbb{R}$$

non nécessairement quadratique.
Comme nous l'avons déjà mentionné, la direction dk est définie par la formule de récurrence suivante :

$$d_k = \begin{cases} -g_0 \quad si \quad k = 0 \\ -g_k + \beta_k d_{k-1} \quad si \quad k \geq 1 \end{cases}$$

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

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

et le pas λ_k est obtenu par une recherche linéaire.

Remarque 1.4.1 *Ces méthodes sont des extensions de la méthode du gradient conjugué si β_k prend l'une des valeurs données comme suit :*

$\beta_k^{HS} = \dfrac{g_{k+1}^T y_k}{d_k^T y_k}$	*Proposé par Hestenes et Stiefel[1952].*
$\beta_k^{FR} = \dfrac{\lVert g_k \rVert^2}{\lVert g_{k-1} \rVert^2}$	*(NCG), proposé par Fletcher et reeves[1964].*
$\beta_k^{D} = \dfrac{g_{k+1}^T \nabla^2 f(x_k) d_k}{d_k^T \nabla^2 f(x_k) d_k}$	*(NCG), proposé par Daniel[1967].*
$\beta_k^{PRP} = \dfrac{g_{k+1}^T y_k}{g_k^T g_k}$	*(NCG), proposé par Polak, Ribiere et Polyak[1969].*
$\beta_k^{PRP+} = \max\{0, \dfrac{g_{k+1}^T y_k}{g_k^T g_k}\}$	*(NCG), proposé par Powell[1984].*
$\beta_k^{CD} = \dfrac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}$	*(NCG), proposé par Flétcher[1987].*
$\beta_k^{LS} = \dfrac{g_{k+1}^T y_k}{-d_k^T g_k}$	*(NCG), proposé par Lieu et Story[1991].*
$\beta_k^{DY} = \dfrac{g_{k+1}^T g_{k+1}}{d_k^T g_k}$	*(NCG), proposé par Dai et Yuan[1999]*
$\beta_k^{DL} = \dfrac{g_{k+1}^T (y_k - t s_k)}{d_k^T g_k}, \qquad t > 0$	*(NCG), proposé par Dai et Lieu[2001].*
$\beta_k^{YT} = \dfrac{g_{k+1}^T (z_k - t s_k)}{d_k^T z_k}$	*(NCG), proposé par Yabe et Takano[2004].*

là où

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

et

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

1.5 Convergence des méthodes à directions de descente

1.5.1 Convergence

Étudier la convergence d'un algorithme, c'est étudier la convergence de la suite des itérés générés par l'algorithme. Un algorithme de descente selon le modèle précédent, est dit convergent si la suite de ses itérés $\{x_n\}_{n\in\mathbb{N}}$ converge vers un point limite x_* solution du problème :

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

De plus, la convergence est dite locale si elle n'a lieu que pour des points initiaux x_0 dans un voisinage de x_* . Sinon elle est dite globale. En pratique, le but d'un algorithme d'optimisation ne sera que de trouver un point critique (i.e. un point vérifiant la condition d'optimalité du premier ordre : $\nabla f(x_*) = 0$)).
On introduit alors la notion de convergence globale d'un algorithme d'optimisation :

Définition 1.5.1 *Soit un algorithme itératif qui génère une suite* $\{x_n\}_{n\in\mathbb{N}}$ *dans* $\mathbb{R}^n$ *afin de résoudre le problème* $\min\{f(x); x \in \mathbb{R}^n\}$ *où* $f : \mathbb{R}^n \to \mathbb{R}$. *est une application de classe* C^1 *: L'algorithme est dit globalement convergent si : quel que soit le point initial* $x_0 \in \mathbb{R}^n$ *,*

$$\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0$$

1.5.2 Condition de Zoutendijk

Dans cette section on va étudier la contribution de la recherche linéaire inexacte à la convergence des algorithmes à directions de descente.
On dit qu'une règle de recherche linéaire satisfait la condition de Zoutendijk s''il existe une constante $C > 0$ telle que pour tout indice $k \geq 1$ on a :

$$f(x_{k+1}) \leq f(x_k) - C \left\|\nabla f(x_k)\right\|^2 \cos^2 \theta_k \tag{1.23}$$

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

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

Voici comment on se sert de la condition de Zoutendijk [Voir [20]].

Proposition 1.5.1 *Si la suite* $\{x_k\}$ *générée par un algorithme d'optimisation vérifie la condition de Zoutendijk* (1.23) *et si la suite* $\{f(x_k)\}$ *est minorée, alors*

$$\sum_{k \geq 1} \left\|\nabla f(x_k)\right\|^2 \cos^2 \theta_k < \infty \tag{1.24}$$

Preuve 1.5.1 *En sommant les inégalités* (1.23) *on obtient*

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

La série est donc convergente puisqu'il existe une constant C' telle que pour tout $k, f(x_k) \geq C'$ La proposition suivante précise les circonstances dans les quelles la condition de Zoutendijk (1.23) est vérifie avec la règle de Wolfe.

Proposition 1.5.2 *Soit*

$$f : \mathbb{R}^n \to \mathbb{R}$$

une fonction continument différentiable dans un voisinage de $\Gamma = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$. *On considère un algorithme à directions de descente* d_k*, qui génère une suite* $\{x_k\}_{k\in\mathbb{N}}$*, en utilisant la recherche linéaire de Wolfe* $(1.19)-(1.20)$*. Alors il existe une constante* $C > 0$ *telle que, pour tout* $k \geq 1$ *, la condition de Zoutendijk* (1.23) *est vérifiée.*

Preuve 1.5.2

D'après (1.20)

$$\nabla^T f(x_k + \lambda_k d_k)\, d_k \geq \delta \nabla^T f(x_k)\, d_k \Rightarrow$$

$$(\nabla f(x_k + \lambda_k d_k) - \nabla f(x_k))^T d_k \geq (\delta - 1)\nabla^T f(x_k)\, d_k \Rightarrow$$

$$-(1-\delta)\nabla^T f(x_k)\, d_k = (1-\delta)\left|\nabla^T f(x_k)\, d_k\right| \Leftrightarrow$$

$$(1-\delta)\left|\nabla^T f(x_k)\, d_k\right| \leq (\nabla f(x_k + \lambda_k d_k) - \nabla f(x_k))^T d_k.$$

Et du fait que f *est continument différentiable :*

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

$$(1-\delta)\left\|\nabla^T f(x_k)\right\| \cos\theta_k \leq \mathrm{Lr}_k \|d_k\| \Rightarrow$$

$$\lambda_k \|d_k\| \leq ((1-\delta)/L)\left\|\nabla^T f(x_k)\right\| \cos\theta_k.$$

En utilisant (1.19)*, on aura :*

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \rho\lambda_k \nabla^T f(x_k)\, d_k \Rightarrow$$

$$f(x_k + \lambda d_k) \leq f(x_k) + \left|\rho\lambda\nabla^T f(x_k)\, d_k\right| \Rightarrow$$

$$f(x_k + \lambda d_k) \leq f(x_k) - \rho\lambda\nabla^T f(x_k)\, d_k \Rightarrow$$

$$f(x_k + \lambda d_k) \leq f(x_k) - \rho\lambda\left\|\nabla^T f(x_k)\right\| \|d_k\| \cos\theta_k \Rightarrow$$

$$f(x_k + \lambda d_k) \leq f(x_k) - ((\rho(1-\delta))/L)\left\|\nabla^T f(x_k)\right\|^2 \cos^2\theta_k$$

On en déduit (1.23).

Notre objectif d'après ce travail est d'utiliser les notions qu'on vient de citer et définir pour réaliser une pseudo-convexe hybridation de trois vecteurs de descente afin de construire une nouvelle méthode du gradient conjugué pour l'optimisation sans contraintes où on va étudier la descente et la convergence de cette nouvelle méthode.
Cette nouvelle méthode a été acceptée pour être exposée dans deux conférences dont l'une des deux s'est déroulée en ligne le mois de mai passé ([Voir [11]]) [l'attestation de participation dans cette conférence est remise avec ce mémoire], et la deuxième [Voir [12]] aura lieu entre le 12 et le 16 octobre et elle va se dérouler de la même manière que la première (son email d'acceptation est aussi remis avec ce mémoire).

Chapitre 2

Les méthodes du gradient conjugué

Dans ce chapitre, on va énoncer les méthodes qui ont été utilisées dans la réalisation de notre hybridation et qui sont des extensions de la méthode du gradient conjugué où chaque méthode possède un paramètre λ_k spécifique. Mais avant de présenter chacune de ces méthodes nous allons définir d'abord la notion de "la condition de conjugaison".

2.1 Condition de conjugaison

Si la direction initiale d_0 est choisie de la manière suivante :

$$d_0 = -g_0$$

et la fonction objective f à minimiser est une fonction quadratique convexe c'est à dire :

$$f(x) = \frac{1}{2} x^T Q x - b^T x + c, \tag{2.1}$$

et la recherche linéaire exacte est employée, veut dire :

$$\lambda_k = \min_{\lambda > 0} f\left(x_k + \lambda d_k\right), \tag{2.2}$$

alors la condition de conjugaison est :

$$d_i^T Q d_j = 0 \tag{2.3}$$

et elle est satisfaite pour tous $i \neq j$.

Cette relation (2.3) est la condition originale de conjugaison employée par Hestenes et Stiefel pour dériver les algorithmes du gradient conjugué, principalement pour résoudre les systèmes des équations linéaires définie positives.

En employant (2.1),(2.2) et (2.3) on montre que x_{k+1} est le minimum de la fonction quadratique (2.1) et il est dans le sous-espace vectoriel engendré par $\{g_1, g_2, \ldots g_k\}$ et les gradients $g_1, g_2, \ldots g_k$ sont mutuellement orthogonaux sauf si g_k.

Ce qui donne que pour des fonctions quadratiques et convexes la solution sera trouvée dans n itérations au plus [Voir [15]].

Powell a prouvé que si la direction initiale d_0 n'est pas égale à $-g_0$ alors les algorithmes de gradient conjugué ne se terminent pas dans un nombre fini d'itérations pour la fonction (2.1). Il est bien connu que l'algorithme du gradient conjugué converge au moins linéairement [Voir [14]]. Une limite supérieure pour le taux de convergence des algorithmes de gradient conjugué a été donnée par Yuan . Dénotons :

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

Pour une fonction f nonlinéaire deux fois différentiable,on a par le théorème de valeur moyenne, il existe certains :

$$\nu \in [0, 1]$$

tel que

$$d_{k+1}^T y_k = \lambda_k d_{k+1}^T \nabla^2 f\left(x_k + \nu \lambda_k d_k\right) d_k \tag{2.4}$$

Par conséquent, il semble raisonnable de remplacer (2.3) par la condition suivante :

$$d_{k+1}^T y_k = 0 \tag{2.5}$$

Pour accélérer l'algorithme de gradient conjugué Perry (voire également Shanno) a prolongé la condition de conjugaison en incorporant l'information du second degré.

Il a employé :

$$H_{k+1} y_k = s_k,$$

où, h_k est une approximation symétrique de l'inverse de la matrice hessienne. Puisque pour la méthode quasi-newtonienne la direction d_{k+1} est calculée par

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

donc :

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

de ce fait on obtient une nouvelle condition de conjugaison. Récemment, Dai et Liao ont prolongé cette condition ils ont donné la nouvelle condition de conjugaison comme suit :

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

$$d_{k+1}^T y_k = -\sigma g_{k+1}^T s_k \tag{2.6}$$

où, σ est un scalaire positif.

2.2 La méthode de Polak-Ribière Polyak

Cette méthode a été proposée en 1969 par Polak [Voir[8]] et Ribière et Polyak [Voir [4]]. Pour cette méthode, le paramètre β_k est donné par :

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

d'où :

$$\beta_{k+1}^{PRP} = \frac{\nabla f_{k+1}^T\left(\nabla f_{k+1}-\nabla f_k\right)}{\left\|\nabla f_k\right\|^2} = \frac{g_{k+1}^T y_k}{\left\|g_k\right\|^2}, \tag{2.7}$$

et son algorithme est donné comme suit :

2.2.1 Algorithme de la méthode de Polak-Ribière-Polyak

Étape 0 : (Initialisation)

Soit x_0 le point de départ,$g_0=\nabla f\left(x_0\right)$, poser $d_0=-g_0$.

Poser $k=0$ et aller à l'étape 1.

Étape 1

Si $g_k=0$: STOP ($x^*=x_k$). "Test d'arrêt"

Si non aller à l'étape 2.

Étape 2

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

$$\begin{gathered}
\lambda_k=\min_{\lambda>0} f\left(x_k+\lambda d_k\right) \\
d_{k+1}=-g_{k+1}+\beta_{k+1}^{PRP} d_k
\end{gathered}$$

où

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

Poser $k = k+1$ et aller à l'étape 1.

Le résultat suivant est du à Polak et Ribière (Voir [8]). Il établit la convergence de cette méthode pour une fonction fortement convexe avec une recherche linéaire exacte.

Proposition 2.2.1 *Si f est fortement convexe, de classe C^1 avec un gradient lipschitzien, alors la méthode de Polak-Ribiére avec une recherche linéaire exacte génère une suite $\{x_n\}$ convergente vers l'unique point x^* réalisant le minimum de f. [Voir [8]]*

Remarque 2.2.1 *Si f n'est pas convexe, la méthode de Polak-Ribière-Polyak peut ne pas converger.*

2.3 La méthode de Dai-Yuan

Cette méthode a été proposée en 1999 par Dai et Yuan [Voir [18]], et son paramètre β_k est donné par :

$$\begin{aligned}
\beta_{k+1}^{DY} &= \frac{g_{k+1}^T H d_k}{d_k^T H d_k} = \frac{g_{k+1}^T \left(g_{k+1} - g_k\right)}{d_k^T \left(g_{k+1} - g_k\right)} \\
&= \frac{g_{k+1}^T g_{k+1} - g_{k+1}^T g_k}{d_k^T \left(g_{k+1} - g_k\right)} = \frac{\|g_{k+1}\|^2 - g_{k+1}^T g_k}{d_k^T \left(g_{k+1} - g_k\right)} \\
&= \frac{\|g_{k+1}\|^2}{d_k^T \left(g_{k+1} - g_k\right)}
\end{aligned}$$

d'où :

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}. \tag{2.8}$$

Cette méthode possède plusieurs propriétés comme la propriété de descente à chaque itération et la convergence si le pas est déterminé par les règles de (Wolfe faible, Armijo et Goldstein) et si f est strictement convexe ou régulière.

2.3.1 Descente de la méthode de DY

Proposition 2.3.1

1. *L'ensemble* $\Omega = \{x \in \mathbb{R}^n ; f(x) \leq f(x_1)\}$ *est borné ; où* $x_1 \in \mathbb{R}^n$ *est le point initial.*
2. *Sur un voisinage* $\mathcal{N}$ *de* Ω, *la fonction objectif* f *est continument différentiable et son gradient est lipschitzien i.e*

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

Ces suppositions impliquent quil existe $\gamma > 0$ *tel que*

$$g(x)\| \leq \gamma \quad \forall x \in \Omega$$

Théorème 2.3.1 *Soit* x_1 *un point de départ pour lequel la proposition* (2.2.1) *est satisfaite. Considérons une méthode du gradient conjugué avec* β_k *satisfait* (2.8).
Si f *est strictement convexe sur l'ensemble convexe* ω *c'est à dire :*

$$(g(x) - g(y))^T (x - y) > 0 \quad \forall x, y \in \omega \tag{2.9}$$

alors $\forall k \geq 1$ *:*

$$g_k^T d_k < 0 \tag{2.10}$$

En 1999 ils ont généralisé ce résultat pour toute fonction régulière avec la recherche de Wolfe faible. [Voir [17]]

Remarque 2.3.1 *Rappelons que les conditions de Wolf faibles sont donnée par :*

$$\begin{aligned} f\left(x_k+\lambda_k d_k\right) &\leq f\left(x_k\right)+\delta\lambda\nabla f\left(x_k\right)^T\cdot d_k \\ \sigma d_k^T g_k &\leq d_k^T g_{k+1} \end{aligned} \tag{2.11}$$

où, $0<\delta<\sigma<1$.

Théorème 2.3.2 (Voir [18]) *Supposons sue la proposition* (2.2.1) *est satisfaite. Pour toute méthode du gradient conjugué dont* β_k *satisfait* (2.8) *et le pas* λ_k *satisfait les conditions de Wolfe faibles, alors toutes les directions générées sont des directions de descente, autrement dit :*

$$d_k^T g_k<0, \quad \forall k\geq 1. \tag{2.12}$$

Preuve 2.3.1 *La démonstration se fait par récurrence.*
Pour k=1 :

$$d_1^T g_1=-\|g_1\|^2<0.$$

Supposons que (2.12) *est satisfaite pour* $k>1$ *et démontrons qu'elle le sera pour* $k+1$.
C'est à dire, supposons que :

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

en utilisant (2.11), *on obtient*

$$d_k^T y_k=d_k^T(g_{k+1}-gk)=d_k^T g_{k+1}-dk^T gk\geq\sigma d_k^T g_k-d_k^T g_k=-(1-\sigma)d_k^T g_k>0.$$

D'autre part,

$$\begin{aligned}
d_k^T g_{k+1} &= (-g_{k+1} + \beta_{k+1}^{DY} d_k)^T g_{k+1} \\
&= -\left\|g_{k+1}\right\|^2 + \beta_{k+1}^{DY} d_k^T g_{k+1} \\
&= -\left\|g_{k+1}\right\|^2 + \frac{\left\|g_{k+1}\right\|^2}{d_k^T y_k} d_k^T (y_k + g_k) \\
&= -\left\|g_{k+1}\right\|^2 + \left\|g_{k+1}\right\|^2 + \frac{\left\|g_{k+1}\right\|^2}{d_k^T y_k} d_k^T g_k \\
&= \frac{\left\|g_{k+1}\right\|^2}{d_k^T y_k} d_k^T g_k
\end{aligned}$$

puisque : $d_k^T g_k < 0$ *et* $d_k^T y_k > 0$*, il en résulte :*

$$d_k^T g_{k+1} < 0.$$

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

2.3.2 Convergence de la méthode de Dai-Yuan

Dai et Yuan ont assuré la convergence de leur méthode pour une fonction strictement convexe avec une recherche linéaire inexacte d'Armijo et Goldstein [Voir[17]].

Théorème 2.3.3 (Voir [17]) *Soit* x_1 *un point de départ pour lequel la supposition* (2.2.1)*. Considérons une méthode du gradient conjugué avec* β_k *satisfait* (2.8)*.*
Si f *est strictement convexe sur l'ensemble convexe* Ω *et le pas* λ_k *satisfait les conditions de Goldstein*

$$\omega_2 \lambda_k \nabla^T f(x_k) d_k \leq f(x_k + \lambda_k d_k) - f(x_k) \leq \omega_1 \lambda_k V^T f(x_k) d_k \tag{2.13}$$

où, $g_k = \nabla f(x)$ *et* ω_1 *et* ω_2 *sont deux constantes vérifiant :*

$$0<\omega_1<\frac{1}{2}<\omega_2<1.$$

Alors,

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

Théorème 2.3.4 *Soit x_1 un point de départ pour lequel la supposition (2.2.1). Considérons une méthode du gradient conjugué avec β_k satisfait (2.8). Si f est uniformément convexe sur l'ensemble convexe Ω, c'est à dire : s'il existe une constante $\eta > 0$ tel que*

$$(g(x)-g(y))^T(x-y)\geq\eta\|x-y\|^2,\quad \forall x,y\in\Omega, \tag{2.14}$$

et le pas λ_k satisfait les conditions d'Armijo

$$f(x_k+\lambda_k d_k)-f(x_k)\leq\omega\lambda_k\nabla^T f(x_k)\,d_k, \tag{2.15}$$

où $g_k=\nabla f(x)$ et $0<\omega<1$ pour lequel la condition suffisante de descente est satisfaite. Alors

$$\lim_{k\to\infty}\|g_k\|=0$$

2.4 La méthode de Liu-Storey

Cette méthode a été proposée par Liu et Storey [voir [13]] où ils ont présenté la formule suivante pour le scalaire β_k :

$$\beta_{k+1}^{LS}=\frac{g_{k+1}^T y_k}{-d_k^T g_k}. \tag{2.16}$$

Chapitre 3

Hybridation pseudo-convexe

Dans ce chapitre on expose le fruit de notre travail,qu'on a eu l'honneur de l'exposer dans la conférence internationale MÉTHODES MODERNES THÉORIE DES PROBLÈMES AUX LIMITES Matériaux [Voir [11]] et après être accepté à la Conférence Scientifique Internationale KOLMOGOROV READINGS - IX [Voir [12]], on l'y présentera avec fierté entre le 12 et le 16 Octobre.

3.1 Introduction

Considérons le problème d'optimisation suivant :

$$(P): \quad \min\{f(x); x \in \mathbb{R}^n\} \tag{3.1}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction différentiable.
Les méthodes du gradient conjugué sont aujourd'hui les méthodes les plus fiables et efficaces pour résoudre ce type d'optimisation sans contraintes. Ces méthodes suivent le schéma itératif suivant :

$$x_{k+1} = x_k + \lambda_k d_k, \tag{3.2}$$

la direction d_k est définie par la formule de récurrence suivante :

$$d_k = \begin{cases} -g_0 \quad si \quad k = 0, \\ -g_k + \beta_k d_{k-1} \quad si \quad k \geq 1. \end{cases} \tag{3.3}$$

Où, le pas $\lambda_k \in \mathbb{R}$ est déterminé par une optimisation unidimensionnelle ou une recherche linéaire inexacte, $g_k = \nabla f(x_k)$ et $\beta_k \in \mathbb{R}$.

Les λ_k vérifient les conditions de Wolfe fortes suivantes :

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \omega_1 \lambda_k d_k^T g_k \tag{3.4}$$

$$\left| d_k^T g_{k+1} \right| \leq -\omega_2 d_k^T g_k \tag{3.5}$$

avec $0 < \omega_1 < \omega_2 < 1$. Le scalaire β_k est choisi de telle sorte que la méthode (3.2)(3.3) est réduite à la méthode du gradient conjugué linéaire dans le cas ou f est quadratique convexe et la recherche linéaire est exacte. Pour des fonctions générales, de différentes formules pour le scalaire β_k résultent des méthodes du gradient conjugué.

Parmi ces méthodes on trouve : la méthode de Polak-Ribière Polyak, la méthode de Dai-Yuan, la méthode de Liu-Storey... et leurs paramètres sont donnés comme suit :

$$\beta_{k+1}^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \tag{3.6}$$

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}. \tag{3.7}$$

$$\beta_{k+1}^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}, \tag{3.8}$$

respectivement, où $\|.\|$ signifie la norme Euclidienne et $y_k = g_{k+1} - g_k$.

Les propriétés de convergence de ces méthodes ont été mentionnées dans plusieurs de références, [voir [19] [17] [20] [8] [4] [18]...]

Afin de garantir la convergence de la méthode (3.2) et (3.1), les propriétés suivantes sont indispensables :

$$d_{k+1}^{T} g_{k+1} < 0 \quad \text{(propriété de descente)} \tag{3.9}$$

$$d_{k+1}^{T} g_{k+1} \leq -c_0 \left\| g_{k+1} \right\|^2, c_0 > 0 \quad \text{(propriété de descente suffisante)} \tag{3.10}$$

Cependant, si la condition imposée à ω_2 dans (3.5) est uniquement $\omega_2 < 1$, ni l'une ni l'autre des trois méthodes du gradient conjugué nonlinéaires citées ci-dessus ne peut assurer la descente avec la recherche linéaire de Wolfe forte (3.5), même si f est quadratique [voir [16]].

Dans [voir [3]] N. Andrei a construit une nouvelle méthode qu'il a nommé CCOMB qui représente une combinaison convexe de deux méthodes du gradient conjugué qui sont (PRP) et (DY). Il a choisi l'hybridation de ses deux méthodes pour bénéficier des propriétés de calcul qu'elle possède la méthode de PRP et des propriétés de convergence de la méthode de DY.
Pour cette méthode, le scalaire β_k prend la valeur suivante :

$$\beta_k^N = (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{DY} = (1-\theta_k)\,\frac{g_{k+1}^T y_k}{g_k^T g_k} + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k} \tag{3.11}$$

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

$$\theta_k = \frac{\left(y_k^T g_{k+1}\right)\left(y_k^T s_k\right) - \left(y_k^T g_{k+1}\right)\left(g_k^T g_k\right)}{\left(y_k^T g_{k+1}\right)\left(y_k^T s_k\right) - \left(g_{k+1}^T g_{k+1}\right)\left(g_k^T g_k\right)} \tag{3.12}$$

qui a été trouvé 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 qui, sous certaines conditions, satisfont la condition de descente suffisante et il a démontré que cette méthode est convergente.

3.2 Pseudo-convexe hybridation

Dans ce mémoire, on expose une nouvelle méthode qui est une combinaison peuso-convexe de trois méthodes de gradient conjugué qui sont (PRP),(DY) et (LS). Le choix de ses trois méthode était fait pour leurs résultats et leur performance chacune de sa part, donc leur hybridation sera surement plus efficace performante.

Notre direction de la recherche est sélectionnée comme suit :

$$d_{k+1} = -(1+\gamma_k)g_{k+1} + (\gamma_k\beta_k^{LS} + (1-\theta_k)\beta_k^{PRP} + \theta_k\beta_k^{DY})d_k, \tag{3.13}$$

Où, les paramètres γ_k et θ_k sont des scalaires que nous choisirons de telle sorte que la direction de la recherche d_{k+1} qui satisfait la condition de conjugaison et de descente suffisante respectivement à chaque itération.

Nous allons écrire le paramètre γ_k en termes de θ_k telle que direction de recherche d_{k+1}, définie par (3.13), satisfait la condition de conjugaison

$$y_k^\top d_{k+1} = 0. \tag{3.14}$$

En multipliant (3.13) par y_k, nous obtenons

$$y_k^\top d_{k+1} = -\gamma_k g_{k+1}^\top y_k - g_{k+1}^\top y_k + \gamma_k\beta_k^{LS}(d_k^\top y_k) + (1-\theta_k)\beta_k^{PRP}(d_k^\top y_k) + \theta_k\beta_k^{DY}(d_k^\top y_k),$$

$$\gamma_k = \frac{g_{k+1}^\top y_k - (1-\theta_k)\beta_k^{PRP}(d_k^\top y_k) - \theta_k\beta_k^{DY}(d_k^\top y_k)}{-g_{k+1}^\top y_k + \beta_k^{LS}(d_k^\top y_k)} \tag{3.15}$$

Nous allons choisir le paramètre θ_k telle que direction de recherche d_{k+1} garantit la condition de descente suffisante que nous avons modifié en une égalité, comme suit

$$d_{k+1}^\top g_{k+1} = -c\|g_{k+1}\|^2, \quad c > 0, \tag{3.16}$$

En multipliant (3.13) par g_{k+1}, nous obtenons

$$-\gamma_k\|g_{k+1}\|^2-\|g_{k+1}\|^2+\gamma_k\beta_k^{LS}(d_k^\top g_{k+1})+(1-\theta_k)\beta_k^{PRP}(d_k^\top g_{k+1})+\theta_k\beta_k^{DY}(d_k^\top g_{k+1}) = -c\|g_{k+1}\|,$$

et en utilisant (3.15), nous obtenons

$$\theta_k = \frac{-c\|g_{k+1}\|^2(-g_{k+1}^\top y_k + \beta_k^{LS}(d_k^\top y_k)) + (\beta_k^{LS} - \beta_k^{PRP})(\|g_{k+1}\|^2(d_k^\top y_k) - (d_k^\top g_{k+1})(g_{k+1}^\top y_k))}{(\beta_k^{DY}(d_k^\top g_{k+1}) - \beta_k^{PRP}(d_k^\top g_{k+1}))(\|g_{k+1}\|^2 - (g_{k+1}^\top y_k))}$$

Ainsi, nous obtenons un algorithme hybride, noté PCH, qui est donné par :

Algorithm 3.2.1

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

Etape 2. *Calculer* $g_0 = g(x_0)$. *Si* $\|g_k\| \leq \varepsilon$ *: stop, si non définir* $d_0 = -g_0$ *et aller à* **l'étape 3**.

Etape 3. *Calculer la longueur du pas* λ_k *avec la recherche linéaire de Wolf :*

Etape 4. *Poser* $x_{k+1} = x_k + \lambda_k d_k$.

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

Etape 6. *Si* $\|g_{k+1}\| \leq \varepsilon$ *: stop.*

Etape 7. *Calculer la direction* d_{k+1} *via (3.13). Poser* $k = k + 1$ *et aller à* **l'étape 3**.

3.3 La convergence de l'algorithme

Dans cette section, nous analysons la convergence globale de l'algorithme. Avant cela, nous introduisons les hypothèses suivantes dans la fonction objective $f(x)$.

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

H2 Le gradient de f est Lipschitz continu dans N, autrement dit, il existe une constante L > 0 tel que :

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

Lemme 3.3.1 *Supposons que les hypothèses* **H1** *et* **H2** *sont vérifiées et que les suites* $\{g_k\}$ *et* $\{d_k\}$ *sont générées par l'algorithme déjà donné et que le pas* λ_k *soit déterminé par la recherche linéaire. alors la condition de Zoutendijk est :*

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

Preuve 3.3.1 *Voir [1].*

Lemme 3.3.2 *Considèrons toute mèthode itèrative du gradient conjugué, où* d_k *et* λ_k *satisfont la condition de descente suffisante et les conditions de Wolfe (3.4) et (3.5), respectivement. Supposons que les hypothèses* **H1** *et* **H2** *sont satisfaites et si*

$$\sum_{k \geq 1} \frac{1}{\|d_k\|^2} = \infty, \tag{3.17}$$

alors

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

Démonstration 1 *Supposons par contradiction qu'il existe une constante positive* γ *telle que* $\|g_k\| \geq \gamma$ *pour tout* $k \geq 1$*, il s'en suit que*

$$\frac{(d_{k+1}^\top g_{k+1})^2}{\|d_{k+1}\|^2} = c^2 \frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2}$$

Par conséquent, à partir du lemme (3.3.1), il s'en suit que

$$\sum_{k\geq 1}\frac{1}{\|d_k\|^2}\leq\frac{1}{\gamma^4}\sum_{k\geq 1}\frac{\|g_k\|^4}{\|d_k\|^2}<\infty$$

ce qui est en contradiction avec (3.17). Alors

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

Chapitre 4

Tests numériques

Dans ce chapitre, on présente une comparaison entre la performance de notre méthode **PCH** et celle la méthode de **CCOMB** avec le profil de performance.

4.1 Profil de performance

Le "Performance profil" est une méthode qui a montré son efficacité dans la comparaison entre les méthodes d'approximations numériques, surtout lorsque la comparaison théorique (ordre de convergence, complexité, ...) est impossible ou incertaine. Cette méthode présente un grand intérêt notamment dans le domaine de l'optimisation [voir [9]].

4.2 Résultats numériques

On expose ici une comparaison entre la performance de notre méthode **PCH** et celle la méthode de **CCOMB** avec le profil de performance en utilisant les conditions de recherche linéaire de Wolf avec $\delta = 0 : 0001$, $\sigma = 0,9$ et $c = \frac{7}{8}$ sur un nombre de d'optimisation problèmes sans contraintes du type (3.1) ,chaque fonction de test est effectué a une expérience avec le nombre de variables $n = 50, 100, 200, ..., 1000$ respectivement. Le critère d'arrêt pour tous les algorithmes est $\|g_k\| \leq 10^{-6}$.

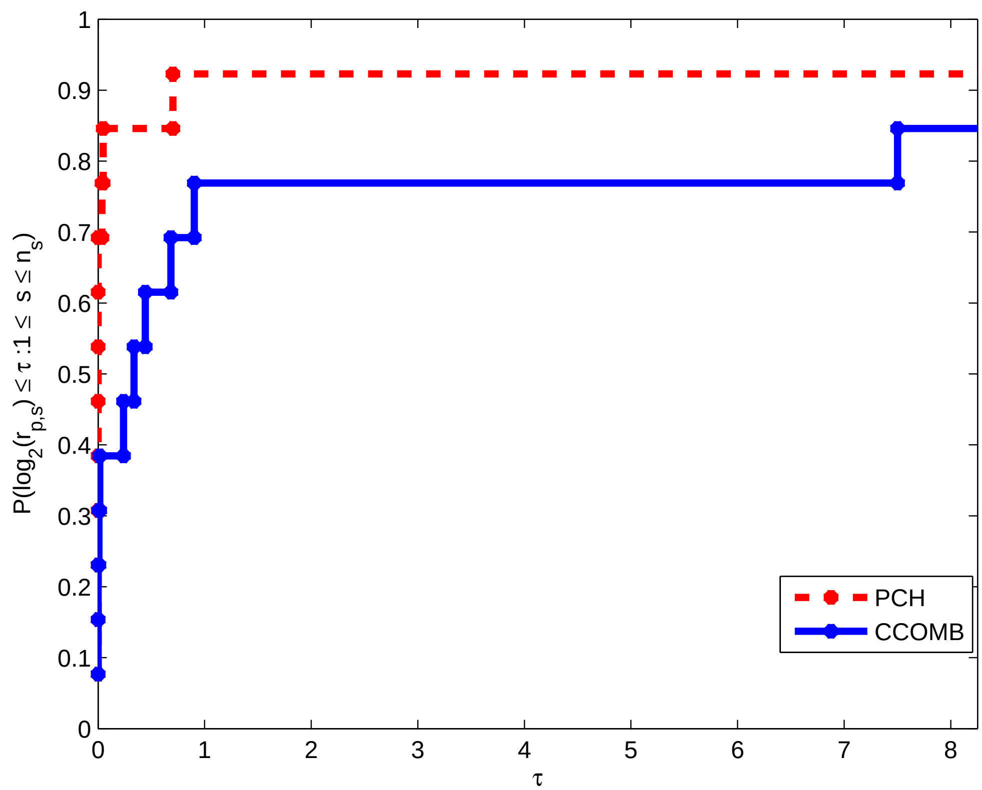

Figure 4.1 – Profil de performance par le temps (CPU).

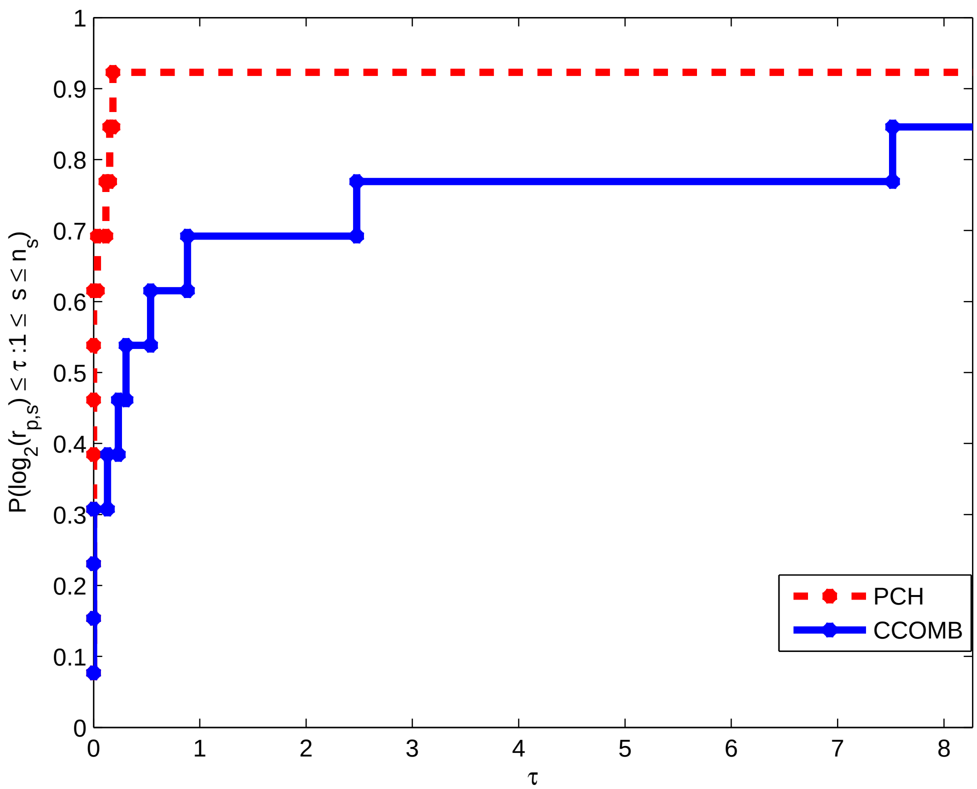

FIGURE 4.2 – Profil de performance par nombre d'itérations.

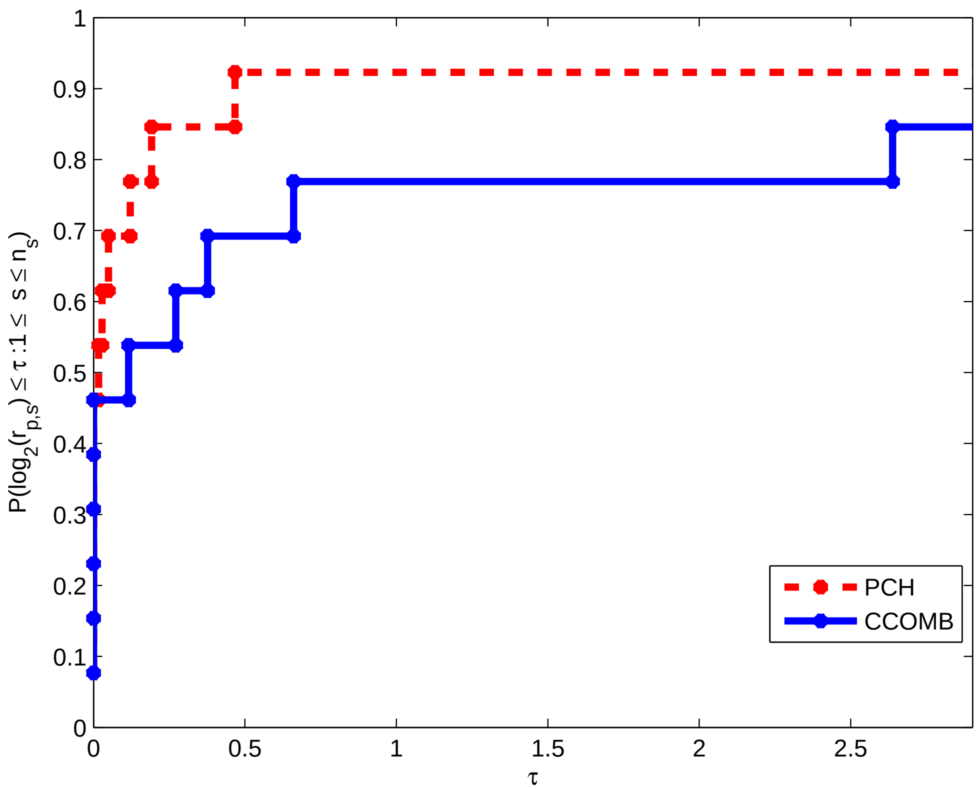

Figure 4.3 – Profil de performance pour le nombre de fonctions et l'évaluation du gradient.

Comme le montre les résultats numériques, qui sont présentés par les figures **4.1**, **4.2** et **4.1** la méthode **PCH** est plus rapide que la méthode **CCOMB**, plus compétitive au sens de nombre d'itération et beaucoup plus supérieure quand il s'agit de nombre de fonctions et d'évaluation du gradient.

Conclusion

Dans ce mémoire, nous avons réussi à construire une hybridation pseudo-convexe de trois méthodes su graduant conjugué pour arriver à résoudre les problèmes d'optimisation sans contraintes. Cette nouvelle méthode satisfait en même temps : la condition de conjugaison et la condition de descente suffisante. On a démontrer la convergence de notre méthode et aussi, on a donné quelques tests numériques qui montrent son efficacité parmi les méthodes du gradient conjugué qui existent déjà.

Bibliographie

[1] Andrei, Neculai. Another accelerated conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization. No. 7. ICI Technical Report, 2010.

[2] Andrei, Neculai. An unconstrained optimization test functions collection. Adv. Model. Optim 10.1 (2008) : 147-161.

[3] Andrei N. Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization / N. Andrei // J Optim Theory Appl. — 2009. 141. 249–264. https ://doi.org/10.1007/s10957-008-9505-0.

[4] B. T. Polyak. The conjugate gradient method in extreme problems. USSR Comp. Math. and Math. Phys. 9 (1969), pp. 94-112.

[5] 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.

[6] 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.

[7] Dolan, Elizabeth D. and More,Jorge J. Benchmarking optimization software with performance profiles. Mathematical programming 91.2 (2002) : 201-213.

[8] E. Polak and G. Ribière. Note sur la convergence de directions conjugées. Rev. Francaise Informat Recherche Operationelle, 3e Ann ee 16 (1969), pp. 35-43.

[9] Ghiat, M. Guebbai, H. Khellaf, A. (2018) Numerical Methods and their Performance Profile for a Weakly Singular Integro-Differential Nonlinear Volterra Equation.

[10] G, Zoutendijk. Nonlinear programming, computational methods. Integer and nonlinear programming. (1970) : 37-86.

[11] HANNACHI. M, HAFAIDIA. I, GHIAT. M, GUEBBAI. H. (2020). Pseudoconvex hybridization of three descent vectors to build a new method of the conjugate gradient. MÉTHODES MODERNES THÉORIE DES PROBLÈMES AUX LIMITES Matériaux Conférence internationale . LECTURES DE PONTRYAGIN XXXI, Dédié à la mémoire de Yuli Vitalievich Pokorny (80e anniversaire), (3-9 mai 2020) , École de mathématiques de printemps de Voronej, Voronej.

[12] HANNACHI. M, HAFAIDIA. I, GHIAT. M, GUEBBAI. H. (2020). New hybrid conjugate gradient method as a pseudoconvex combination of PRP,DY and LS methods. Lectures de Kolmogorov IX. Problèmes généraux de contrôle et leurs applications (OCP-2020). Conférence internationale, dédié au 70e anniversaire de la naissance d'Alexandre Ivanovitch Boulgakov et au 90e anniversaire de l'Institut de mathématiques, de phy-

sique et des technologies de l'information. Université d'État de Tambov nommée d'après G.R. Derzhavin Tambov, 12-16 octobre 2020.

[13] LIU, Y. , STOREY, C., Efficient generalized conjugate gradient algorithms, part 1, theory. J. Optim. Theory Appl. 69, 129-137 (1991).

[14] R., GLOWINSKI, Numerical Methods for Nonlinear Variational Problems SpringerVerlag, Berlin, 1984.

[15] TOUATI-AHMED, D., STOREY, C., Efficient hybrid conjugate gradient techniques, J. Optim. Theory Appl.64(2), 379–397 (1990) .

[16] Y. H. Dai, Analyses of nonlinear conjugale gradient method, Ph. D. thesis, Institute of Computational Mathematics and Scientfic/ E[3] ngineering Computing, Chinese Academy of Sciences, 1997.

[17] Y. H. Dai and Y. Yuan, Some properties of a new conjugale gradient method, in : Y. Yuan ed. , Advances in Nonlinear Programming (Kluwer,Boston, 1998), pp. 251-262.

[18] Y.H. Dai and Y. Yuan (1999). A nonlinear conjugate gradient with a strong global convergence property. SIAM J. Optimization, Vol. 10(1) pp.177-182.

[19] Y. H. Dai, Some new properties of a nonlinear conjugale gradient method, Research report ICM-98-010, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1998.

[20] ZOUTENDIJK, G., Nonlinear Programming, Computational Methods, In J. Abadie (Eds.), Integer and Nonlinear Programming (pp. 37-86). Amsterdam : North-Holland. (1970).