وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

BADJI MOKHTAR-ANNABA

UNIVERSITY

UNIVERSITE BADJI MOKHTAR
ANNABA

جامعة باجي مختار

ـعنابةـ

Faculté des Sciences **Année : 2024/2025**

Département de Mathématiques

THÈSE

Présentée en vue de l'obtention du diplôme de Doctorat

Sur quelques méthodes Itératives appliquées à l'optimisation sans contraintes

Filière
Mathématiques

Spécialité
Analyse Numérique et Optimisation

Par
BARROUK Bachir

DIRECTEUR DE THÈSE:	BELLOUFI Mohammed	**Prof**	**U.M.C.M. Souk Ahras**
CO- DIRECTEUR DE THÈSE :	ZEGHDOUDI Halim	**Prof**	**U.B.M. Annaba**

Devant le jury

PRESIDENT :	SAHARI Mohamed Lamine	**M.C.A**	**U.B.M. Annaba**
EXAMINATEUR :	BENRABAH Abderafik	**Prof**	**U.8 mai 1945.Guelma**
EXAMINATEUR :	EZZEBSA Abdelali	**M.C.A**	**U. 8 mai 1945.Guelma**
EXAMINATEUR :	KHEMIS Rabah	**M.C.A**	**U.20 août 1955. Skikda**

Dédicaces

À la mémoire de mon père,
À la mémoire de ma mère,

À ma femme,
À mes enfants Jaouad et Wissal,
À tous mes frères et sœurs,
À tous mes amis,
À tous ceux qui sont proches de mon cœur,
et dont je n'ai pas cité les noms,
Je dédie ce modeste travail.
Merci à tous ... !

Remerciements

Avant tout, je remercie "Allah" le tout puissant pour nous avoir donné l'opportunité et de me donner la force et la patience pour terminer ce travail.

Je tiens tout d'abord à remercier profondément mon encadreur Mr. BELLOUFI Mohammed, professeur à l'université de Souk Ahras, pour la gentillesse et la patience qu'il a manifestées à mon égard. Je le remercie, aussi, pour m'avoir guidé, encouragé, conseillé, tout au long de la réalisation de cette thèse.

Je tiens à témoigner toute ma gratitude à mon co-encadreur Mr. ZEGHDOUDI Halim professeur à l'université Badji Mokhtar de Annaba pour sa disponibilité et son aide.

Je tiens à remercier également Mr. SAHARI Mohamed Lamine, maître de conférences à l'université Badji Mokhtar de Annaba, pour l'honneur qu'il m'a fait en acceptant de présider le jury et aussi pour tout ce qu'il m'a appris.

Je remercie aussi tous les membres du jury pour leurs contributions. Je cite, Mr. BENRABAH Abderafik maître de conférences à l'université de Guelma et Mr. EZZEBSA Abdelali maître de conférences à l'université de Guelma et Mr. KHEMIS Rabah maître de conférences à l'université de Skikda, qui ont pris de leurs temps pour lire et juger ce travail, ainsi que pour leur déplacement le jour de la soutenance.

Mes remerciements ne seraient pas complets sans mentionner Pr. BENZINE Rachid mon Ex-encadreur pour sa disponibilité et son aide, dont je lui souhaite la bonne santé. Sans oublier Mr. BACHAOUETTE Taher pour sa grande et distinguée aide et aussi l'ensemble de mes enseignants qui ont participé à notre formation. Qu'ils trouvent ici, l'expression de mon profond respect et de ma haute considération.

Je n'oublie pas de remercier toutes les personnes qui m'ont facilité la tâche et tous ceux que j'ai connu au département de mathématiques de Annaba et de Souk Ahras, qui ont rendu mes séjours au département agréables.

Comme je remercie aussi ma grande famille pour son soutien et pour son aide dans la préparation de cette thèse et pour leurs encouragements... Merci !.

الملخـــص

الأمثَلة هي مجال من مجالات البحث الرياضي الذي يهدف إلى البحث عن الحد الأدنى أو الأقصى للدالة. هذا المجال واسع ويغطي مجموعة واسعة من المسائل، من حل المعادلات التفاضلية إلى المسائل الهندسية. الهدف من هذا العمل هو اقتراح تقريب جديد لتسريع تقارب طريقة التدرج المترافق PRP. يعد تحسين تقارب طريقة التدرج المترافق أمرًا مهمًا لأنه يسمح بحل المسائل غير الخطية بسرعة أكبر، وهذا يمكن أن يكون له تأثير كبير في العديد من المجالات، مثل الهندسة والعلوم والاقتصاد.

الكلمات المفتاحية: الأمثَلة غير المقيدة، طريقة التدرج المترافق، البحث الخطي القوي لوولف ، التقارب، ملفات تعريف الأداء، المقارنات العددية.

Abstract

Optimization is a field of mathematical research aimed at finding the minimum or maximum of a function. This field is vast and covers a wide range of problems, from solving differential equations to engineering challenges. The objective of this work is to propose a new approach to accelerate the convergence of the PRP conjugate gradient method, such that this approach combines existing techniques to improve the method's performance. Improving the convergence of the conjugate gradient method is important because it allows nonlinear problems to be solved more quickly. This can have a significant impact in many fields, such as engineering, science, and economics.

Keywords: Unconstrained optimization, conjugate gradient method, strong Wolfe linear search, convergence, performance profiles, numerical comparisons.

Résumé

L'optimisation est un domaine de recherche mathématique qui a pour objectif de trouver le minimum ou le maximum d'une fonction. Ce domaine est vaste et couvre un large éventail de problèmes, allant de la résolution d'équations différentielles aux problèmes d'ingénierie. L'objectif de ce travail est de proposer une nouvelle approche pour accélérer la convergence de la méthode du gradient conjugué PRP, telle que cette approche combine des techniques déjà existantes pour améliorer les performances de la méthode. L'amélioration de la convergence de la méthode du gradient conjugué est importante, car elle permet de résoudre des problèmes non linéaires plus rapidement. Cela peut avoir un impact significatif dans de nombreux domaines, tels que l'ingénierie, les sciences et l'économie.

Mots clés : Optimisation sans contrainte, méthode du gradient conjugué, recherche linéaire de Wolfe forte, convergence, profil de performance, comparaisons numériques.

Table des matières

Introduction

L'optimisation est un domaine fondamental des mathématiques appliquées qui trouve des applications dans une multitude de domaines, allant de l'ingénierie à l'économie en passant par la science des données. L'optimisation sans contraintes, en particulier, revêt une importance particulière, car elle se concentre sur la recherche de solutions optimales sans imposer de limitations strictes sur les variables du problème. Les méthodes itératives jouent un rôle central dans la résolution de ces problèmes, en visant à améliorer itérativement une solution initiale jusqu'à atteindre un optimum.

Cette thèse de doctorat vise à explorer en profondeur les méthodes itératives appliquées à l'optimisation sans contraintes. L'objective principal est de contribuer à l'avancement de ces méthodes, en identifiant leurs avantages, leurs limites et en proposant des améliorations. Dans ce contexte, cette introduction établira le cadre de notre recherche et définira les objectives spécifiques que nous nous sommes fixés.

Contexte de la recherche

L'optimisation sans contraintes s'est développée rapidement grâce aux progrès de la puissance de calcul et des algorithmes associés. Ses applications sont vastes, allant de la conception de structures en ingénierie à la résolution de problèmes d'ajustement de modèles mathématiques. Malgré ces avancées, des défis subsistent, notamment, en ce qui concerne la convergence des méthodes itératives, la gestion des points selles, et l'efficacité des algorithmes pour les problèmes de grande dimension.

Importance de l'optimisation sans contraintes dans le monde réel

L'optimisation sans contraintes représente un outil essentiel dans la modélisation et la résolution de problèmes mathématiques complexes. Son utilité s'étend à divers secteurs tels que l'ingénierie, la finance, la logistique et la recherche opérationnelle, où elle est appliquée pour relever des défis concrets. L'évolution constante des méthodes itératives dans ce domaine peut apporter des avantages substantiels en matière de résolution de problèmes réels, permettant ainsi d'optimiser les processus et de maximiser les performances dans un large éventail d'applications.

Objectives de la thèse

Cette thèse a pour objective principal d'explorer, d'analyser et d'améliorer les méthodes itératives appliquées à l'optimisation sans contraintes. Nous chercherons à comprendre les fondements des méthodes existantes. Alors, nous présentons des propositions d'amélioration et d'évaluer leur performance au moyen d'études de cas et d'exemples concrets. En conclusion, notre ambition est de contribuer au développement des connaissances dans ce domaine et à l'amélioration des outils disponibles pour les chercheurs et les praticiens, dans le but ultime de favoriser des avancées significatives et durables.

Aperçu du plan

Le plan de cette thèse se décompose comme suit **:**

Le premier chapitre de cette thèse se concentre sur l'optimisation sans contraintes. Il commence par décrire la stratégie adoptée pour aborder ce domaine, en mettant en lumière des techniques telles que la recherche linéaire exacte et inexacte (incluant la recherche linéaire d'Armijo, la recherche linéaire de Goldstein et la recherche linéaire de Wolfe), ainsi que la stratégie de région de confiance. En outre, il examine en détail les conditions nécessaires et suffisantes du premier et du second ordre pour garantir l'existence d'un minimum local ou d'un minimum local strict.

Dans le deuxième chapitre, différentes méthodes d'optimisation sans contraintes basées sur le calcul de gradient sont présentées et analysées en détail. Chacune de ces méthodes est examinée selon sa définition, ses avantages et ses inconvénients, ainsi que ses propriétés de convergence. La principale distinction entre ces méthodes réside dans la manière dont elles calculent la direction de descente d_k Pour ce qui est du calcul du pas α_k, la méthode de Wolfe (standard) est généralement privilégiée. Parmi les méthodes discutées, on trouve la descente la plus rapide, les méthodes de Newton, les méthodes de quasi-Newton, les méthodes de quasi-Newton à mémoire limitée, les méthodes de Newton tronqué, les méthodes de gradient conjugué, les méthodes de région de confiance et les méthodes de $p-$régularisation.

Dans le troisième chapitre, en se basant sur la comparaison entre les divers résultats de convergence générale pour les méthodes de gradient conjugué non linéaires énoncent que, sous certaines conditions, l'algorithme converge vers un point stationnaire de la fonction objectif. Tels que le type de convergence le plus couramment étudié pour les méthodes d'optimisation, y compris les méthodes de gradient conjugué non linéaires, et la convergence vers un point stationnaire de la fonction objectif, et leurs concepts. En s'intéresse ici, par la convergence sous

la recherche linéaire de Wolfe forte et de Wolfe standard. Finalement, en fait un critique faible et robuste sur ces résultats de convergence dans divers situation.

Dans le quatrième chapitre, On donne des profils de performance, c'est-à-dire des fonctions de distribution pour une métrique de performance, comme un outil pour évaluer et comparer les logiciels d'optimisation. On montre, ici, que les profils de performance combinent les meilleures caractéristiques et d'autres outils d'évaluation des performances, qui nous aidons de construire un programme de comparaison entre les divers méthodes d'optimisation sans contraintes.

Dans le dernier chapitre, correspond à l'article intitulé "An Improved PRP Conjugate Gradient Method for Optimization Computation" par B. Barrouk, M. Belloufi, R. Benzine, et T. Bechouat dans l'International Journal of Nonlinear Analysis and Applications. Ici, nous avons introduire une nouvelle méthode itératives appliquées à l'optimisation sans contraintes et testé numériquement cette nouvelle méthode en faisant une étude comparative avec quelques méthode existant dans la littérature. En termine la thèse par une conclusion générale, quelques perspectives et une liste bibliographique.

CHAPITRE 1

Présentation de l'optimisation sans contrainte

L'optimisation sans contrainte implique la minimisation d'une fonction dépendant de plusieurs variables réelles, sans aucune restriction quant à leurs valeurs. Lorsque le nombre de variables est important. Ce chapitre explore les principales méthodes de gradient utilisées pour résoudre ces problèmes d'optimisation sans contraintes. Ces méthodes sont itératives, débutant par une estimation initiale des variables et générant une série d'estimations améliorées jusqu'à la convergence vers un ensemble de valeurs. Pour vérifier si cet ensemble de valeurs constitue réellement la solution du problème, alors, les conditions d'optimalité sont utilisées. Dans le cas contraire, ces conditions d'optimalité, celles-ci peuvent être exploitées pour améliorer l'estimation de la solution en cours. Les algorithmes présentés dans ce chapitre reposent sur l'utilisation de la fonction objectif de ses dérivées premières et secondes.

1.1 Position du problème

Soit le problème d'optimisation sans contrainte suivant :

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1.1.1}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction à valeur réelle de n variables, continu sur $\mathbb{R}^n$. Le but est de trouver un minimiseur local de cette fonction, soit un point x^*, de sorte que :

$$f(x^*) \leq f(x) \text{ pour tout } x \text{ proche de } x^* \tag{1.1.2}$$

Si $f(x^*) < f(x)$ pour tout x^* proche de x, alors x^* est appelé un minimiseur local strict de la fonction f. Souvent, f est appelé la fonction objectif, Cependant $f(x^*)$ est le minimum ou la valeur minimale.

Le problème de minimisation locale est différent du problème de minimisation globale, où un minimiseur global, c'est-à-dire un point x^* tel que

$$f(x^*) \leq f(x) \text{ pour tout } x \in \mathbb{R}^n \tag{1.1.3}$$

Ce chapitre traite uniquement des problèmes de minimisation locale.

La fonction f dans(1.1.1) peut avoir n'importe quelle expression algébrique et nous supposons qu'elle est deux fois continûment différentiable sur $\mathbb{R}^n$. Notons $\nabla f(x)$ le gradient de f et $\nabla^2 f(x)$ son hessienne.

Pour résoudre (1.1.1), de nombreuses méthodes sont connues, voir : Luenberger (1973) [Lue73], (1984) [Lue84],Gill, Murray et Wright (1981) [MWG19], Bazaraa, Sherali et Shetty (1993) [BSS13], Bertsekas (1999) [Ber97], Nocedal et Wright. (2006) [NW99], Sun et Yuan (2006) [SY06] , Bartholomew-Biggs (2008) [BB08], Andrei (1999) [And99], (2009) [And09a], (2015) [And15]. En général, pour résoudre (1.1.1), les méthodes d'optimisation sans contraintes mettent en œuvre l'une des deux stratégies suivantes : la recherche linéaire et la région de confiance. Ces deux stratégies sont utilisées pour résoudre (1.1.1) .

1.1.1 Technique de la recherche linéaire

Dans la technique de la méthode de recherche linéaire, l'algorithme associé une direction d_k et la recherche dans cette direction à partir de l'itération actuelle x_k, une nouvelle itération avec une valeur de fonction inférieure. Plus précisément, en commençant par un point initial x_0, les itérations sont déduite de la manière suivante :

$$x_{k+1} = x_k + \alpha_k d_k;\ k = 0, 1, ... \tag{1.1.4}$$

où $d_k \in \mathbb{R}^n$ est la direction de descente qui permet de réduire les valeurs de la fonction f, et $\alpha_k \in \mathbb{R}^n$ est la longueur de pas choisie à l'aide d'une procédure de recherche linéaire. Une

condition préalable est que la direction de descente d_k à l'itération k soit satisfaite. Dans la section 1.3, nous établissons la caractérisation algébrique des directions de descente comme suit :

$$d_k^T g_k \leq 0, \tag{1.1.5}$$

ce qui est un critère très important concernant l'efficacité d'un algorithme. Dans (1.1.5), $g_k = \nabla f(x_k)$ est le gradient de f au point x_k. Pour garantir la convergence globale, il est parfois que la direction de descente d_k vérifie la condition de descente suffisante :

$$g_k^T d_k \leq -c \left\| g_k \right\|^2, \tag{1.1.6}$$

où c est une constante positive.

1.1.2 Technique de région de confiance

Dans la méthode de la région de confiance, le principe est d'exploiter les données collectées sur la fonction de minimisation f pour construire une fonction modèle m_k dont le comportement près du point courant x_k est similaire à celui de la fonction objectif réelle f. Autrement dit, l'étape p est déterminée en résolvant approximativement le sous-problème suivant :

$$\min_p m_k(x_k + p), \tag{1.1.7}$$

où le point $x_k + p$ se trouve à l'intérieur de la région de confiance. Si l'étape p ne donne pas une réduction suffisante, il s'ensuit que la région de confiance est trop grande. Dans ce cas, la région de confiance est réduite et le modèle m_k dans (1.1.7) est résolu. Habituellement, la région de confiance est une boule définie par $\|p\|_2 < \Delta$, où le paramètre Δ désigné le rayon de cette région. Il convient de noter que des régions de confiance elliptiques ou rectangulaires peuvent également être utilisées.

Habituellement, le modèle m_k dans (1.1.7) est défini comme une approximation quadratique de la fonction minimisante f :

$$m_k(x_k + p) = f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T B_k P \tag{1.1.8}$$

où B_k est le hessien $\nabla^2 f(x)$, soit une approximation de celui-ci. Observez que chaque fois que la taille de la région de confiance, c'est-à-dire le rayon de la région de confiance, est réduite

après un échec de l'itération actuelle, alors le pas de x_k au nouveau point sera plus court et pointe, généralement, dans une direction différente du point précédent.

La recherche linéaire et la région de confiance représentent deux méthodes distinctes, divergeant dans leur processus de prise de décision pour sélectionner la direction de descente et la taille du pas afin de progresser vers l'itération suivante. Dans la recherche linéaire, le processus début par la sélection d'une direction d_k, suivie de la détermination d'une distance appropriée le long de cette direction, représentée par le pas α_k. En revanche, dans la méthode de la région de confiance, la première étape consiste à choisir la distance maximale, c'est-à-dire le rayon de la région de confiance Δ_k. Ensuite, une direction p_k et un pas sont déterminés afin d'obtenir la meilleure amélioration des valeurs de fonction sous cette contrainte de distance. Si cette étape n'est pas satisfaisante, alors la mesure de distance Δ_k est réduite et le processus est répété.

Il est possible d'utiliser différentes techniques pour déterminer la direction de la descente. Plusieurs des aspects les plus importants seront abordés dans ce chapitre. Discutons maintenant des principales procédures de détermination de la taille du pas pour une optimisation sans contrainte à l'aide d'une stratégie de recherche linéaire. Ensuite, on ce donne une démonstration des méthodes d'optimisation sans contraintes.

1.2 Recherche linéaire

La recherche linéaire constitue une approche employée en optimisation pour détecter le minimum ou le maximum d'une fonction en déplaçant itérativement le long d'une direction donnée. La méthode de recherche linéaire implique de choisir une direction de descente et de minimiser (ou maximiser) la fonction le long de cette direction. Il existe différentes stratégies de recherche linéaire, parmi ces méthodes, la plus utilisée est la recherche linéaire exacte.

Voici une discription générale de la méthode de recherche linéaire :

• *direction de descente* **:** Dans l'espace des variables et à partir du point actuel on sélectionnez une direction de descente. Diverses techniques, telles que la méthode du gradient ou la méthode de Newton, peuvent être utilisées pour déterminer cette direction.

• *Recherche Linéaire* **:** Une fois que la direction de descente est identifiée, menez une recherche le long de cette direction pour déterminer le pas optimal. Cela revient fondamentalement à minimiser une fonction à une seule variable, cette fonction étant définie par la projection de la

fonction objectif sur la direction de descente.

• *Pas Optimal* : Déterminer le pas qui optimise la fonction le long de la direction sélectionnée. Généralement, ce pas est calculé en résolvant un problème d'optimisation unidimensionnelle

• *Mise à Jour* : Mettez à jour la position actuelle en déplaçant dans la direction optimale avec le pas optimal trouvé à l'étape précédente.

• *Critère d'arrêt* : Répétez les étapes précédentes jusqu'à ce qu'un critère d'arrêt soit satisfait, tel qu'une tolérance d'erreur prédéfinie ou un nombre maximal d'itérations atteint.

Il est à noter que la recherche linéaire peut être incorporée en tant qu'élément d'algorithmes d'optimisation plus complexes, comme la méthode du gradient, la méthode de Newton, ou des algorithmes quasi-Newton.

On distingue généralement deux principales catégories de méthodes de recherche linéaire tel que la recherche linéaire exacte.et la recherche linéaire inexacte.

1.2.1 Recherche linéaire exacte (R.L.E)

La recherche linéaire exacte est basée sur l'hypothèse que la fonction à optimiser est convexe dans la direction de descente. Cependant, lorsque la fonction n'est pas convexe, on recourt généralement à des méthodes de recherche linéaire inexactes. Ces techniques peuvent être une recherche le long d'une direction avec des critères de convergence moins d'efficacité.

Supposons que la fonction minimisante f soit suffisamment continue sur $\mathbb{R}^n$. Concernant le pas α_k qui doit être utilisé dans (1.1.4), les valeurs de fonction la plus grande réduction est obtenue lorsque l'on utilisée la recherche linéaire exacte. dans laquelle :

$$\alpha_k = \arg\min_{\alpha \geq 0} f(x_k + \alpha d_k) \tag{1.2.1}$$

En d'autres termes, la recherche de droite exacte détermine un pas α_k comme solution de l'équation :

$$\nabla f(x_k + \alpha_k d_k)^\intercal d_k = 0 \tag{1.2.2}$$

1.2.2 Recherche linéaire inexacte (R.L.I)

La recherche linéaire exacte est généralement évitée dans les algorithmes d'optimisation. parce que ce n'est pas pratique. Au lieu de cela, une recherche linéaire inexacte est souvent utilisée.

De nombreuses méthodes de recherche de lignes inexactes ont été proposées : Goldstein (1965) [Gol65], Armijo (1966) [Arm66], Wolfe (1969, 1971) [Wol69]; [Wol71], Powell (1976) [Pow76], Lemaréchal (1981)[Lem81] , Shanno (1983) [Sha83], Dennis et Schnabel (1983) [DJS96], Al-Baali et Fletcher (1984) [ABF86], Hager (1989) [Hag89] , Moré et Thuente (1990) [MT94], Lukšan (1992) [Luk92], Potra et Shi (1995) [PS95], Hager et Zhang (2005) [HZ05a], Gu et Mo (2008) [GM08], Ou et Liu (2017) [OL17], et plein d'autres. Les défis liés à la recherche d'une bonne taille de pas par une recherche linéaire inexacte consistent à éviter que la taille du pas soit trop longue ou trop courte. Par conséquent, les méthodes de recherche linéaires inexactes se concentrent sur : une bonne sélection initiale de la taille des pas, des critères qui garantissent que α_k ne sont ni trop longs ni trop courts et la construction d'une séquence de mises à jour qui satisfait aux exigences ci-dessus.

Généralement, les méthodes de recherche linéaire inexactes s'appuient fréquemment sur des interpolations polynomiales quadratiques ou cubiques des valeurs de la fonction unidimensionnelle.$\varphi_k(\alpha) = f(x_k + \alpha d_k)$ $\alpha \geq 0$. Pour minimiser l'approximation polynomiale de $\varphi_k(\alpha)$, les procédures de recherche linéaire inexactes génèrent une série de pas jusqu'à ce que l'une de ces valeurs du pas satisfasse certaines conditions d'arrêt.

Maintenant, examinons les différentes méthodes de recherche linéaire inexacte qui sont utilisées.

Recherche linéaire d'Armijo (R.L.A)

La méthode de recherche linéaire d'Armijo, parfois appelée règle de décroissance arrière (backtracking line search) d'Armijo, est une méthode couramment utilisée dans les algorithmes d'optimisation pour déterminer la longueur du pas (step size) optimale lors de la recherche le long d'une direction donnée. Cette méthode est associée à des algorithmes de descente de gradient pour la minimisation d'une fonction objectif. Son objectif principal est de garantir que la réduction de la fonction objectif lors du déplacement le long de la direction de descente soit suffisamment significative. Voici comment fonctionne cette méthode :

Initialisation : Choisissez une longueur de pas initiale,

$t > 0$, un paramètre de décroissance,

$0 < \beta < 1$ (généralement proche de 1), est un paramètre de contrôle,

$0 < \alpha < 1$ (également appelé coefficient d'Armijo).

Examen du critère d'Armijo : Nous vérifions si la condition d'Armijo est remplie. Cette

condition est habituellement formulée comme suit :

$$f(x+td) \leq f(x) + \alpha_t \nabla f(x),$$

où $f(x)$ est la fonction objectif, x est le point actuel, d est la direction de descente, α_t est la longueur de pas, et $\nabla f(x)$ est le gradient de la fonction objectif à x.

Réduction de la Longueur de Pas : Si la condition d'Armijo n'est pas satisfaite, réduisez la longueur de pas α_t en multipliant par le facteur de décroissance β et répétez l'évaluation de la condition d'Armijo. Continuez à réduire la longueur de pas jusqu'à ce que la condition d'Armijo soit satisfaite.

Mise à Jour : Une fois que la condition d'Armijo est satisfaite, mettez à jour la position en utilisant la longueur de pas α_t trouvée.

La méthode de recherche linéaire d'Armijo assure que la fonction objectif est réduite de manière significative dans chaque itération, tout en entraînant des pas excessivement petits. Cela contribue à la convergence des algorithmes d'optimisation.

Il est important que la méthode d'Armijo est valable parmi plusieurs règles de recherche linéaire, et selon les exigences spécifiques de l'algorithme d'optimisation, d'autres méthodes comme la recherche linéaire de Wolfe peuvent être préférées.

Une des approches de recherche linéaire les plus simples et efficaces est la méthode du retour en arrière appelée règle de décroissance arrière (Ortega & Rheinboldt, 1970) [OR00]. Cette procédure considère les scalaires suivants : $0 < \gamma < 1$, et $0 < s_k = -g_k^T d_k / \|g_k\|^2$ et suit les étapes suivantes basées sur la règle d'Armijo :

Algorithme 1.2.1. *Recherche linéaire d'Armijo*

Etape 1.	*Considérons la direction de descente d_k pour f en x_k. Définir un $\alpha = s_k$*
Etape 2.	*Si : $f(x_k + \alpha d_k) > f(x_k) + \gamma \alpha g_k^T d_k$, on pose $\alpha = \alpha\beta$*
Etape 3.	*On pose $\alpha_k = \alpha$*

Observez que cette recherche linéaire nécessite que la réduction obtenue dans f soit au moins une fraction c fixée de la réduction promise par l'approximation de Taylor de premier ordre de f en x_k. Généralement, $c = 0.0001$ et $b = 0.8$, ce qui signifie qu'une petite partie de la diminuation prédite par l'approximation linéaire de f au point actuel est acceptée. Observez que, lorsque $d_k = -g_k$, alors $s_k = 1$.

Théorème 1.2.1. *(Terminaison du retour en arrière d'Armijo)*
Soit f continûment différentiable de gradient $g(x)$ lipschitzienne continue de constante $L > 0$, c'est-à-dire, $\|g(x) - g(y)\| \leq L \|x - y\|$ pour tout x , y dans l'ensemble $S = \{x : f(x) \leq f(x_0)\}$. Soit d_k une direction de descente en x_k, c'est-à-dire $g_k^T d_k < 0$. Alors pour γ fixé dans $(0, 1)$ on a :

1. La condition d'Armijo $f(x_k + \alpha d_k) \leq f(x_k) + \gamma \alpha g_k^T d_k$ est satisfaite pour tout $\alpha \in [0, \alpha_k^{\max}]$, où

$$\alpha_k^{\max} = \frac{2(\gamma - 1) g_k^T d_k}{L \|d_k\|_2^2}$$

2.Pour toute τ fixé dans $(0, 1)$ le pas généré par la recherche linéaire d'Armijo se termine par :

$$\alpha_k \geq \min \left\{ \alpha_k^0, \frac{2\tau(\gamma - 1) g_k^T d_k}{L \|d_k\|_2^2} \right\},$$

où α_k^0 est le pas initial à l'itération k.

• Remarquons qu'en pratique la constante de Lipschitz L est inconnue. Par conséquent, $\alpha_k^{\max}$ et α_k ne peuvent pas simplement être calculés via les formules explicites données par le théorème 1.2.1

Recherche linéaire de Goldstein (R.L.G)

Une méthode de recherche linéaire inexacte est définie par Goldstein (1965) [Gol65] , où α_k est choisi de manière à satisfaire les conditions suivantes :

$$\delta_1 \alpha_k g_k^T d_k \leq f(x_k + \alpha_k d_k) - f(x_k) \leq \delta_2 \alpha_k g_k^T d_k \tag{1.2.3}$$

où : $0 < \delta_2 < \frac{1}{2} < \delta_1 < 1$.

Recherche linéaire de Wolfe (R.L.W)

Les critères de recherche linéaire les plus couramment utilisés pour déterminer la taille du pas sont les critères de recherche linéaire de Wolfe standard.(Wolfe, 1969, 1971) ([Wol69]; [Wol71]) :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k d_k^T g_k \tag{1.2.4}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma d_k^T g_k \tag{1.2.5}$$

où : $0 < \rho < \sigma < 1$.

Proposition 1.2.1. *La première condition* (1.2.4)*, nommée condition d'Armijo, garantit une réduction suffisante de la valeur de la fonction objectif, tandis que la deuxième condition* (1.2.5)*, appelée condition de courbure, garantit des pas courts inacceptables. Il convient de mentionner qu'un pas calculé par les conditions de recherche linéaire de Wolfe* (1.2.4) *et* (1.2.5) *peut ne pas être suffisamment proche d'un minimiseur de* $\varphi_k(\alpha) = f(x_k + \alpha d_k)$*. Dans de telles circonstances, il est possible d'utiliser les conditions de recherche linéaire de Wolfe fortes, qui consistent en* (1.2.4) *et, au lieu de* (1.2.5)*, la version améliorée suivante :*

$$\left|\nabla f(x_k + \alpha_k d_k)^T d_k\right| \leq -\sigma d_k^T g_k. \tag{1.2.6}$$

Proposition 1.2.2. *Selon* (1.2.6)*, lorsque* σ *tend vers* 0*, le pas qui satisfait les conditions*(1.2.4) *et* (1.2.6) *tend à être le pas optimal. Remarquez que si un pas* α_k *satisfait aux conditions de recherche linéaire de Wolfe fortes, alors il satisfait également aux conditions de Wolfe standards.*

Preuve. Si la fonction f est continûment différentiable et que d_k une direction de descente au point x_k, supposons que f est bornée inférieurement le long du rayon $\{x_k \alpha d_k : \alpha > 0\}$. Dans ce cas, pour tout ρ et σ tels que $0 < \rho < \sigma < 1$, il existe un intervalle de pas α qui satisfait à la fois les conditions de Wolfe et les conditions de Wolfe fortes.
Puisque $\varphi_k(\alpha) = f(x_k + \alpha d_k)$ est borné inférieurement pour tout $\alpha > 0$, la droite $l(\alpha) = f(x_k) + \alpha\rho\nabla f(x_k)^T d_k$ doit couper le graphe de φ au moins une fois. Soit $\alpha' > 0$ être la plus petite valeur d'intersection de α, c'est-à-dire :

$$f(x_k + \alpha' d_k) = f(x_k) + \alpha'\rho\nabla f(x_k)^T d_k < f(x_k) + \rho\nabla f(x_k)^T d_k \tag{1.2.7}$$

Ainsi, une réduction suffisante est garantie pour tous les α dans l'intervalle $0 < \alpha < \alpha'$. Selon le théorème des valeurs moyennes, il existe α'' dans l'intervalle $(0, \alpha')$ tel que

$$f(x_k + \alpha' d_k) - f(x_k) = \alpha'\rho\nabla f(x_k + \alpha'' d_k)^T d_k \tag{1.2.8}$$

Puisque $\rho < \sigma$ et $\nabla f(x_k)^T d_k < 0$, d'après (1.2.7) et (1.2.8) on obtient :

$$\nabla f(x_k + \alpha'' d_k)^T d_k = \rho\nabla f(x_k)^T d_k > \sigma\nabla f(x_k)^T d_k \tag{1.2.9}$$

Ainsi, α'' satisfait à la fois les conditions de recherche linéaire de Wolfe (1.2.4) et (1.2.5), et ces inégalités sont strictes. En vertu de l'hypothèse de régularité sur f, il existe un intervalle autour de α'' dans lequel les conditions de Wolfe sont satisfaites. Étant donné que $\nabla f(x_k + \alpha'' d_k)^T d_k < 0$,

cela implique que les conditions de recherche linéaire de Wolfe fortes (1.2.4) et (1.2.6) sont également satisfaites dans le même intervalle. □

Proposition 1.2.3. *Supposons que d_k soit une direction de descente et que ∇f satisfasse la condition de Lipschitz*

$$\|\nabla f(x) - \nabla f(x_k)\| \leq L \|x - x_k\|$$

pour tout x sur le segment reliant x_k et x_{k+1}, où L est une constante. Si la recherche linéaire satisfait aux conditions de Goldstein, alors

$$\alpha_k \geq \frac{1-\delta_1}{L} \frac{\left|g_k^T d_k\right|}{\|d_k\|^2} \tag{1.2.10}$$

Si la recherche linéaire satisfait aux conditions standard de Wolfe, alors :

$$\alpha_k \geq \frac{1-\sigma}{L} \frac{\left|g_k^T d_k\right|}{\|d_k\|^2} \tag{1.2.11}$$

Preuve. Si les conditions de Goldstein sont vérifiées, alors d'après (1.2.3) et le théorème de la valeur moyenne nous avons :

$$\begin{aligned} \delta_1 \alpha_k g_k^T d_k &\leq f(x_k + \alpha_k d_k) - f(x_k) \\ &= \alpha_k \nabla f(x_k + \zeta d_k)^T d_k \\ &\leq \alpha_k g_k^T d_k + L\alpha_k^2 \|d_k\|^2 , \end{aligned}$$

où $\zeta \in [0, \alpha_k]$. De l'inégalité ci-dessus, nous obtenons (1.2.10).En soustrayant $g_k^T d_k$ des deux côtés de (1.2.5) et en utilisant la condition de Lipschitz,on a :

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

Mais d_k est une direction de descente et $\sigma < 1$, donc (1.2.11) découle de l'inégalité (1.2.10). □

Recherche linéaire de Wolfe généralisée (R.L.W.G)

Dans la recherche de droite de Wolfe généralisée, la valeur absolue dans (1.2.6) est remplacée par une paire d'inéquations :

$$\sigma_1 d_k^T g_k \leq d_k^T g_{k+1} \leq -\sigma_2 d_k^T g_k \tag{1.2.12}$$

où $0 < \rho < \sigma_1 < 1$ et $\sigma_2 \geq 0$ Le cas particulier dans lequel $\sigma_1 = \sigma_2 = \sigma$ correspond à la recherche linéaire de Wolfe forte.

1.3 Conditions d'optimalité pour une optimisation sans contrainte

Dans cette section, notre objectif est de déterminer les conditions sous lesquelles une solution au problème (1.1.1) existe. Nous cherchons à discuter des concepts clés et des résultats fondamentaux de l'optimisation sans contrainte, connue sous le nom de conditions d'optimalité. Les conditions nécessaires et suffisantes pour l'optimalité seront présentées. De très bons livres montrant ces conditions sont connus : Bertsekas (1999) [Ber97], Nocedal et Wright (2006) [NW99], Sun et Yuan (2006) [SY06], Chachuat (2007)[Cha07], Andrei (2017) [A$^+$17], etc. Pour formuler les conditions d'optimalité, il faut et il est nécessaire d'introduire quelques concepts qui caractérisent une direction d'amélioration le long de laquelle les valeurs de la fonction f diminuent.

Définition 1.3.1. *(**Direction de descente**). Soit $f : \mathbb{R}^n \to \mathbb{R}$ est continu en x. Un vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f en x^* s'il existe $\delta > 0$ tel que $f(x^* + \lambda d) < f(x^*)$, $\forall \lambda \in (0, 1)$. Le cône des directions de descente en x^*, noté $C_{dd}(x^*)$ est donné par :*

$$C_{dd}(x^*) = \{d : \exists \delta > 0 \text{ tel que : } f(x^* + \lambda d) < f(x^*),\ \forall \lambda \in (0, \delta)\}.$$

On suppose que f soit une fonction différentiable. Pour obtenir une caractérisation algébrique d'une direction de descente pour f en x^*, définissons l'ensemble :

$$C_0(x^*) = \left\{d : \nabla f(x^*)^T d < 0\right\}$$

Le résultat suivant montre que tout $d \in C_0(x^*)$ est une direction de descente en x^*.

Proposition 1.3.1. *(Représentation algébrique d'une direction de descente). Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction différentiable en x^*. $\exists\ d$ tel que $\nabla f(x^*)^T d < 0$, alors d est une direction de descente pour f en x^*, c'est-à-dire : $C_0(x^*) \subseteq C_{dd}(x^*)$.*

Preuve. Puisque f est différentiable en x^*, par conséquent

$$f(x^* + \lambda d) = f(x^*) + \lambda \nabla f(x^*)^T d + \lambda \|d\| o(\lambda d),$$

où $\lim_{\lambda \to 0} o(\lambda d) = 0$. donc

$$\frac{f(x^* + \lambda d) - f(x^*)}{\lambda} = \nabla f(x^*)^T d + \|d\| o(\lambda d)$$

Puisque $\nabla f(x^*)^T d < 0$ et $\lim_{\lambda \to 0} o(\lambda d) = 0$, alors il existe un $\delta > 0$ de sorte que $\nabla f(x^*)^T d + \|d\| o(\lambda d) < 0$ pour tout $\lambda \in (0, \delta)$. □

Théorème 1.3.1. *(Conditions nécessaires de premier ordre pour un minimum local). Supposons que $f : \mathbb{R}^n \to \mathbb{R}$ est différentiable en x^*. Si x^* est un minimum local, alors $\nabla f(x^*) = 0$.*

Preuve. Supposons que $\nabla f(x^*) \neq 0$. Si nous considérons $d = -\nabla f(x^*)$, alors $\nabla f(x^*)^T d = -\|\nabla f(x^*)\|^2$. D'après la proposition 1.3.1, il existe un $\delta > 0$ tel que pour tout $\lambda \in (0, \delta)$, $f(x^* + \lambda d) < f(x^*)$. Mais ceci est en contradiction avec l'hypothèse selon laquelle x^* est un minimum local pour f. □

Remarquez que la condition nécessaire ci-dessus représente un système de n équations algébriques non linéaires. Tous les points x^* qui résolvent le système $\nabla f(x) = 0$ sont appelés points stationnaires. De toute évidence, les points stationnaires ne doivent pas nécessairement tous être des minimums locaux. Il pourrait très bien s'agir de maxima locaux ou même de points de selle. Pour définir un minimum local de façon précise, des conditions nécessitant une restriction plus stricte impliquant la matrice hessienne de la fonction f sont requises.

Théorème 1.3.2. *(Conditions nécessaires de second ordre pour un minimum local).*
Soit $f : \mathbb{R}^n \to \mathbb{R}$ est deux fois différentiable au point x^*. Si x^* est un minimum local, alors $\nabla f(x^*) = 0$ et $\nabla^2 f(x^*)$ est semi-défini positif.

Preuve. Supposons une direction arbitraire d. Ensuite, en exploitan la différentiabilité de f en x^* nous obtenons

$$f(x^* + \lambda d) = f(x^*) + \lambda \nabla f(x^*)^T d + \frac{1}{2}\lambda^2 d^T \nabla^2 f(x^*) d + \lambda^2 \|d\|^2 o(\lambda d)$$

où $\lim_{\lambda \to 0} o(\lambda d) = 0$. Puisque x^* est un minimum local, $\nabla f(x^*) = 0$. Ainsi,

$$\frac{f(x^* + \lambda d) - f(x^*)}{\lambda^2} = \frac{1}{2} d^T \nabla^2 f(x^*) d + \|d\|^2 o(\lambda d)$$

Puisque x^* est un minimum local, pour λ suffisamment petit, $f(x^* + \lambda d) \geq f(x^*)$. pour $\lambda \to 0$, il résulte de l'égalité ci-dessus que $d^T \nabla^2 f(x^*) > 0$. Puisque d est une direction arbitraire, alors $\nabla^2 f(x^*)$ est semi-défini positif. □

Dans les théorèmes ci-dessus, nous avons présenté les conditions nécessaires pour qu'un point x^* soit un minimum local, c'est-à-dire que ces conditions doivent être satisfaites à chaque solution minimale locale. Toutefois, Un point qui satisfait à ces conditions nécessaires ne garantit pas automatiquement qu'il soit un minimum local. Les théorèmes suivants énoncent les conditions suffisantes pour un minimum global, à condition que la fonction objectif soit convexe sur $\mathbb{R}^n$ La démonstration du théorème suivant est possible, ce qui met en évidence l'importance de la convexité dans l'optimisation non linéaire globale.

Théorème 1.3.3. *(Conditions suffisantes de premier ordre pour un minimum local strict). Soit $f : \mathbb{R}^n \to \mathbb{R}$ est différentiable en x^* et convexe sur $\mathbb{R}^n$. Si $\nabla f(x^*) = 0$; alors x^* est un minimum global de f sur $\mathbb{R}^n$.*

Preuve. Puisque f est convexe sur $\mathbb{R}^n$ et différentiable en x^* alors de la propriété des fonctions convexes donnée par la proposition 1.2.3 il s'ensuit que pour tout $x \in \mathbb{R}^n$: $f(x) \geq f(x^*) + \nabla f(x^*)^T(x - x^*)$. Mais x^* est un point stationnaire, c'est-à-dire $f(x) \geq f(x^*)$ pour tout $x \in \mathbb{R}^n$. □

Le théorème suivant donne les conditions suffisantes du second ordre caractérisant un point minimum local pour les fonctions strictement convexes au voisinage du point minimum.

Théorème 1.3.4. *(Conditions suffisantes de second ordre pour un minimum local strict). Soit $f : \mathbb{R}^n \to \mathbb{R}$ est deux fois différentiable au point x^*. Si $\nabla f(x^*) = 0$ et $\nabla^2 f(x^*)$ est défini positif, alors x^* est un minimum local de f.*

Preuve. Puisque f est deux fois différentiable, pour tout $d \in \mathbb{R}^n$, on peut écrire :

$$f(x^* + d) = f(x^*) + \nabla f(x^*)^T d + \frac{1}{2} d^T \nabla^2 f(x^*) d + \|d\|^2 d(o)$$

où $\lim_{d \to 0} o(d) = 0$. Soit λ la plus petite valeur propre de $\nabla^2 f(x^*)$. Puisque $\nabla^2 f(x^*)$ est défini positif, il s'ensuit que $\lambda > 0$ et $d^T \nabla^2 f(x^*) d \geq \lambda \|d\|^2$. Par conséquent, puisque $\nabla f(x^*) = 0$; nous pouvons écrire :

$$f(x^* + d) - f(x^*) \geq \left[\frac{\lambda}{2} + o(d) \right] \|d\|^2$$

Puisque $\lim_{d \to 0} o(d) = 0$, alors il existe un $\eta > 0$ de sorte que $|o(d)| < \frac{\lambda}{4}$ pour tout $d \in B(0, \eta)$, où $B(0, \eta)$ est la boule ouverte de rayon η centrée en 0. D'où

$$f(x^* + d) - f(x^*) \geq \frac{\lambda}{4} \|d\|^2 > 0$$

pour tout $d \in B(0,\eta)\setminus\{0\}$, c'est-à-dire que x^* est un minimum local strict de la fonction f. □

Si nous supposons que f est deux fois continûment différentiable, nous observons que, puisque $\nabla^2 f(x^*)$ est défini positif, alors $\nabla^2 f(x^*)$ est défini positif dans un petit voisinage de x^* et donc f est strictement convexe dans un petit voisinage de x^*. Ainsi, x^* est un minimum local strict, c'est l'unique minimum global sur un petit voisinage de x^*.

CHAPITRE 2

Les méthodes d'optimisation sans contrainte

Dans ce chapitre, nous exposons quelques-unes des méthodes d'optimisation sans contraintes les plus importantes qui reposent sur le calcul de gradient, en insistant sur leur définition, leurs avantages et inconvénients, ainsi que sur leurs propriétés de convergence. La principale variation entre ces techniques se trouve dans la manière dont la direction de descente d_k est calculée. Quant au calcul du pas α_k, la méthode de Wolfe (standard) est largement adoptée. Les méthodes suivantes sont abordées : la descente la plus rapide, les méthodes de Newton, de quasi-Newton, de quasi-Newton à mémoire limitée, de Newton tronqué, de gradient conjugué, de région de confiance et de $p-$régularisation.

2.1 Méthodes à directions de descente

2.1.1 Principe général de la méthode

Soit le problème d'optimisation sans contrainte (1.1.1) , nous désignons également par $\nabla f(x)$ et $\nabla^2 f(x)$ respectivement le gradient et le hessien de f en x par rapport à ce produit scalaire, notre intérêt se porte sur des algorithmes basés sur le concept de direction de descente.

2.1.2 Convergence de la méthode

Nous allons examiner le rôle de la recherche linéaire inexacte dans le processus de convergence des algorithmes utilisant des directions de descente.

Condition de Zoutendijk : La condition de Zoutendijk est satisfait pour la recherche linéaire si pour tout $k \geq 1$, $\exists C > 0$ telle que

$$f(x_{k+1}) \leq f(x_k) - C \left\|\nabla f(x_k)\right\|^2 \cos^2 \theta_k \tag{2.1.1}$$

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

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

Le lemme suivant explique comment introduire la condition de Zoutendijk.

Lemme 2.1.1. *Si la suite $\{x_k\}$ générée par un algorithme d'optimisation vérifie la condition de Zoutendijk(2.1.1) et si la suite $\{f(x_k)\}$ est minorée, alors*

$$\sum_{k\geq 1} \left\|\nabla f(x_k)\right\|^2 \cos^2 \theta_k < \infty \tag{2.1.2}$$

Théorème de Zoutendijk : Le principe fondamental utilisé par les différentes variantes du gradient conjugué avec une recherche linéaire inexacte est énoncé dans le théorème suivant de Zoutendijk**.**

Hypothèse 2.1.1. *Soit $f : \mathbb{R}^n \to \mathbb{R}$. On dit que f vérifie la condition $\mathcal{H}1$ si f est contiument différentiable dans un voisinage $V(\Gamma)$ de $\Gamma = \{x \in \mathbb{R}^n : f(x) \leq f(x_1) : x_1 \in \mathbb{R}^n$ point initial$\}$ et si $\nabla f(x)$ vérifie la condition de Lipschitz dans $V(\Gamma)$, c'est à dire, il existe une constante $L > 0$ telle que*

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| : \text{pour tout } x, y \in V(\Gamma)$$

Théorème 2.1.1. ([*Gil07*]) *Considérons une méthode itérative quelconque générant une suite $\{x_k\}$de la forme :*

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

d_k étant une direction de descente et α_k est une recherche linéaire inéxacte de Wolfe. Supposons aussi que f vérifie l'hypothèse $\mathcal{H}1$. Alors on a

$$\sum_{k=0}^{\infty} \frac{\left(g_k^T d_k\right)^2}{\left\|d_k\right\|^2} < \infty \tag{2.1.3}$$

2.2 Méthode de la plus forte pente (Méthode du gradient)

La méthode de base pour l'optimisation sans contrainte est la descente la plus rapide, connue sous le nom de méthode de Cauchy (1847) [C+47]. Cette méthode consiste à choisir la direction de descente de la manière suivante :

$$d_k = -g_k \tag{2.2.1}$$

Au point x_k, la direction du gradient négatif est bon choix de direction de descente pour un minimum de f. Cependant, dès que nous avançons dans cette direction, elle cesse d'être la meilleure et continue de se détériorer jusqu'à devenir orthogonale à g_k, c'est-à-dire que la méthode commence à faire de petits pas sans faire de progrès significatifs jusqu'au minimum. C'est son inconvénient majeur, les étapes nécessaires sont trop longues, c'est-à-dire qu'il y a d'autres points z_k sur le segment de droite reliant x_k et x_{k+1} , où $-\nabla f(z_k)$ présente une meilleure nouvelle direction de descente que $-\nabla f(x_k)$. La méthode de descente la plus rapide converge globalement sous une grande variété de procédures de recherche de lignes inexactes. Cependant, sa convergence n'est que linéaire et elle est fortement affectée par un mauvais conditionnement (Akaike, 1959) [Aka59]. Le taux de convergence de cette méthode dépend fortement de la distribution des valeurs propres du Hessien de la fonction minimisante.

Théorème 2.2.1. *Soit f deux fois continument différentiable. Si le hessien $\nabla^2 f(x^*)$ de la fonction f est défini positif et a la plus petite valeur propre $\lambda_1 > 0$ et la plus grande valeur propre $\lambda_n > 0$, alors la série de valeurs objectives $\{f(x_k)\}$ générée par l'algorithme de descente la plus rapide converge vers $f(x^*)$ linéairement avec un rapport de convergence pas plus grand que*

$$\left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 = \left(\frac{k-1}{k+1}\right)^2 \tag{2.2.2}$$

i.e,

$$f(x_{k+1}) \leq \left(\frac{k-1}{k+1}\right)^2 f(x_k) \tag{2.2.3}$$

où $k = \dfrac{\lambda_n}{\lambda_1}$ est le numéro de condition du Hessian.

C'est l'une des meilleures estimations que nous puissions obtenir pour la descente la plus rapide dans certaines conditions. Pour les fonctions fortement convexes pour lesquelles le gradient est

continu de Lipschitz, Nemirovsky et Yudin (1983) [NY83] définissent l'estimation globale du taux de convergence d'une méthode itérative comme $f(x_{k+1}) - f(x^*) \leq c.h(x_1 - x^*, m, L, k)$, où h est une fonction, c est une constante, m est une limite inférieure de la plus petite valeur propre du hessien $\nabla^2 f(x)$, L est la constante de Lipschitz et k est le numéro d'itération. Plus la vitesse à laquelle h converge vers 0 est rapide lorsque $k \to \infty$, plus l'algorithme est efficace. Les avantages de la méthode de descente la plus rapide sont multiples. Tout d'abord, elle converge de manière globale vers le minimum local à partir de n'importe quel point de départ initial x_0 . De plus, de nombreuses autres méthodes d'optimisation passent vers la descente la plus rapide lorsqu'elles rencontrent des difficultés à progresser suffisamment vers la solution. En revanche, il présente les inconvénients suivants. Ce n'est pas invariant à l'échelle, c'est-à-dire que changer le produit scalaire sur $\mathbb{R}^n$ changera la notion de gradient. De plus, elle est généralement très (très) lente, c'est-à-dire que sa convergence est linéaire. Numériquement, il arrive souvent qu'elle ne converge pas du tout. Une accélération de la méthode de descente la plus rapide avec retour en arrière a été donnée par Andrei (2006) [And09b] et discutée par Babaie-Kafaki et Rezaee (2018) [BKR18].

Algorithme 2.2.1. ***(Algoithme du Gradient)***

1. Initialisation : $k = 0$ *: choix de* x_0 *et de* $\rho_0 > 0$.

2. Itération k : $x_{k+1} = x_k - \rho_k \nabla f(x_k)$.

3. Critère d'arrêt

- *Si* $\|x_{k+1} - x_k\| < \varepsilon$ *, STOP.*
- *Sinon, on pose* $k = k + 1$ *et on retourne à* 2.

2.3 Méthode à directions conjuguées

Donnons les définitions suivantes :

Définition 2.3.1. *Soit A une matrice symétrique $n \times n$, définie positive. On dit que deux vecteurs x et y de $\mathbb{R}^n$ sont $A-$conjugués (ou conjugués par rapport à A) s'ils vérifient*

$$x^T A y = 0 \tag{2.3.1}$$

On dit que q fonction quadratique de n variables si : $q(x) = \frac{1}{2} x^T A x + b^T x + c$; avec $x \in \mathbb{R}^n$ et $A \in \mathcal{M}_{n \times n}$.

2.3.1 Description de la méthode

Cette méthode est de la forme suivante :

$$x_0 \text{ donné}$$

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.3.2}$$

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

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

Calcul de α_k

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

$$\alpha_k = \frac{-d_k^T (Ax_k + b)}{d_k^T A d_k} \tag{2.3.3}$$

Construire les directions d_k

Les directions A-conjuguées $d_0, ..., d_k$ peuvent être générées à partir d'un ensemble de vecteurs linéairement indépendants $\xi_0, ..., \xi_k$ en utilisant la procédure dite de Gram-Schmidt, de telle sorte que pour tout i entre 0 et k, le sous-espace généré par $d_0, ..., d_i$ soit égale au sous-espace généré par $\xi_0, ..., \xi_i$.

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

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

2.4 Méthodes du gradients conjugués

Le gradient conjugué est une méthode itérative utilisée pour résoudre efficacement des systèmes linéaires et des problèmes d'optimisation non linéaires sans contraintes. Plutôt que de calculer explicitement la solution, cette méthode converge vers la solution en utilisant des directions de recherche conjuguées. Elle est particulièrement efficace pour les problèmes de grande taille et présente des avantages en termes de mémoire et de calcul par rapport à d'autres méthodes traditionnelles.

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

Elles reposent sur le concept des directions conjuguées parce que les gradients successifs sont orthogonaux entre eux et aux directions précédentes.
L'idée initiale était de trouver une suite de directions de descente permettant de résoudre le problème (1.1.1).

2.4.1 Méthode du gradient conjugué pour les fonctions quadratiques

Construction de la méthode

On suppose ici que la fonction à minimiser est quadratique sous la forme : $q(x) = \frac{1}{2}x^T Ax + b^T x + c$. L'algorithme consiste à générer une suite d'itérés $\{x_k\}$ sous la forme :

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.4.1}$$

Si l'on note $g_k = \nabla f(x_k)$, l'algorithme prend la forme suivante :
Pour les directions d_k on a :

$$\begin{cases} d_1 = -\nabla q(x_1) \\ d_{k+1} = -\nabla q(x_{k+1}) + \beta_{k+1} d_k \end{cases} \tag{2.4.2}$$

Les coefficientsβ_{k+1} sont sélectionnés de manière à garantir la conjugaison de d_k avec toutes les directions précédentes. En d'autres termes :
Les coefficients β_{k+1} étant choisis de telle manière que d_k soit conjuguée avec toutes les directions précédentes. autrement dit :

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

cela nous permet de conclure que : où Par conséquent,
on en déduit :

$$\beta_{k+1} = \frac{\nabla^T q(x_{k+1}) A d_k}{d_k^T A d_k} = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}$$

Concernant le coefficient α_k, nous avons :
Pour le pas α_k on a :

$$\alpha_k = \min f(x_k + \alpha d_k),\ \alpha > 0 \tag{2.4.3}$$

on en déduit : **Cela nous conduit à :**

$$\alpha_k = \frac{-d_k^T g_k}{d_k^T A d_k} = -\frac{1}{A d_k} g_k \frac{d_k^T}{d_k^T} = \frac{-d_k^T g_k}{d_k^T A d_k} \tag{2.4.4}$$

Le pas α_k obtenu ainsi s'appelle le pas optimal. **Le coefficient α_k obtenu de cette manière est appelé le pas optimal.**

Différentes formules de β_{k+1} Diverses expressions pour β_{k+1}

Un examen rapide du tableau 2.1 montre qu'à l'exception de la méthode de Daniel (1967), qui nécessite l'évaluation du Hessien à chaque itération, le numérateur du paramètre de mise à jour β_k est soit $\|g_{k-1}\|^2$, soit $g_{k+1}^T y_k$ et le dénominateur est soit $\|g_k\|^2$ou $d_k^T y_k$ ou $d_k^T g_k$. Ici,$y_k = g_{k+1} - g_k$. Principalement, ces deux choix pour le numérateur et les trois choix pour le dénumérateur conduisent à six choix différents pour β_k. Si la fonction f est quadratique fortement convexe et que la recherche linéaire est exacte, alors en théorie, tous les choix ci-dessus pour le paramètre de mise à jour β_k présentés dans le tableau 2.1 sont équivalents. Pour les fonctions objectives non quadratiques, chaque choix pour β_k conduit à des algorithmes avec des performances numériques différentes (nombre d'itérations, nombre de fonctions et ses évaluations de gradient ou temps CPU). Par conséquent, dans ce qui suit, les propriétés de convergence globale des méthodes standards de gradient conjugué avec le numérateur $\|g_{k+1}\|^2$ pour le paramètre de mise à jour β_k (FR, CD et DY) et avec $g_{k+1}^T y_k$ au numérateur de β_k (HS, PRP et LS) seront présentés séparément. De manière générale, la théorie de la convergence pour les méthodes de numérateur $\|g_{k+1}\|^2$ est mieux développée que la théorie des méthodes de numérateur $g_{k+1}^T y_k$ de β_k. Cependant, les méthodes avec $g_{k+1}^T y_k$ au numérateur de β_k fonctionnent mieux en pratique que les méthodes avec $\|g_{k+1}\|^2$ au numérateur de β_k. L'algorithme général des méthodes standard de gradient conjugué est le suivant.

Tel que : $y_k = g_{k+1} - g_k$.

Tableau 2.1 Choix de β_k dans les méthodes standards de gradient conjugué

$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$	Méthode originale du gradient conjugué linéaire d'Hestènes et Stiefel (1952) [HS+52].
$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$	Première méthode de gradient conjugué non linéaire par Fletcher et Reeves (1964) [FR64].
$\beta_k^{D} = \frac{g_{k+1}^T \nabla^2 f(x_k) d_k}{d_k^T \nabla^2 f(x_k) d_k}$	Proposé par Daniel (1967). Cette méthode du gradient conjugué nécessite l'évaluation du Hessien [FR64].
$\beta_k^{PRP} = \frac{g_k^T y_k}{\|g_{k-1}\|^2}$	Proposé par Polak et Ribière (1969) et par Polyak (1969) [PR69],[Pol69].
$\beta_k^{PRP} = \max\left\{0, \frac{g_k^T y_k}{\|g_{k-1}\|^2}\right\}$	Proposé par Powell (1984) [Pow84] et analysé par Gilbert et Nocédal (1992) [GN92a].
$\beta_k^{CD} = -\frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}}$	Proposée par Fletcher (1987) connue sous le nom de méthode CD (descente conjuguée) [Fle00].
$\beta_k^{LS} = -\frac{g_{k+1}^T y_k}{d_k^T g_k}$	Proposé par Liu et Storey (1991) [LS91a].
$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}$	Proposé par Dai et Yuan (1999) [DY99].
$\beta_k^{HZ} = \left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k}\right)^T \frac{g_{k+1}}{d_k^T y_k}$	Gradient conjugué variante Hager-Zhang(HZ) (HZ) (2005) [HZ05b].
$\beta_k^{*} = \frac{g_k^T \left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1}\right)}{\|g_{k-1}\|^2}$	Gradient conjugué variante de Z. Wei (2006) [WYL06a].
$\beta_k^{MN} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}}{\mu \left\|g_k^T d_{k-1}\right\| + \|g_{k-1}\|^2}$	Gradient conjugué variante de H. Fan, Z. Zhu et A. Zhou (2003) [HZ05c].
$\beta_k^{RMIL} = \frac{g_k^T (g_k - g_{k-1})}{\|d_{k-1}\|^2}$	Proposé par Rivaie-Mustafa-Ismail-Leong (RMIL) [MMA+15].

2.4.2 Méthode du gradient conjugué pour les fonctions non quadratiques

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

Dans cette section, notre intérêt se porte sur la minimisation d'une fonction $f : \mathbb{R}^n \to \mathbb{R}$ qui n'est pas forcément quadratique.

$$\min f(x);\ x \in \mathbb{R}^n \tag{2.4.5}$$

Les méthodes du gradient conjugué génèrent des suites $\{x_k\}_{k=0,1,2...}$de la forme suivante :

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.4.6}$$

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

$$d_k = \begin{cases} -g_k \ \text{ si } k = 1 \\ -g_k + \beta_k d_{k-1} \ \text{ si } k \geq 2 \end{cases} \tag{2.4.7}$$

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

$$\beta_k^{HS}, \beta_k^{FR}, \beta_k^{D}, \beta_k^{PRP}, \beta_k^{CD}, \beta_k^{LS}, \beta_k^{DY}, \beta_k^{DL}, \beta_k^{HZ}, \beta_k^{*}, \beta_k^{MN}, \beta_k^{RMIL}$$

Algorithme 2.4.1. *Algorithme de La méthode du gradient conjugué*

Etape 0 : (initialisation)

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

Poser $k = 0$ et aller à l'étape 1.

Etape 1 :

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

Si non aller à l'étape 2.

Etape 2 :

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

α_k : calculer par la recherche linéaire

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

où

$$\beta_{k+1} : \textit{défini selon la méthode}$$

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

2.5 Méthode de Newton pour l'optimisation

Une autre approche souvent utilisée pour l'optimisation est la méthode de Newton. Elle cherche à trouver le minimum (ou maximum) d'une fonction en se basant sur ses dérivées premières et secondes. Cette méthode peut converger rapidement, mais nécessite généralement le calcul des dérivées secondes à chaque itération, ce qui peut être coûteux en termes de calcul. Des variantes telles que la méthode de Newton tronquée sont souvent employées pour atténuer cette complexité.

2.5.1 Description de la méthode

La méthode de Newton est classée parmi les techniques de descente, s'efforçant de minimiser le développement en série de Fourier du second ordre de la fonction. $f(X) \in C^2$ suivant la quantité d

$$f(x+d) = f(x) + \nabla f(x)\, d + \frac{1}{2} d^T H_f(x)\, d$$

Voici la forme de d_k qui minimise la partie droite :

$$d_k = -\left[H_f(x_k)\right]^{-1} \nabla f(x_k)$$

Puis, nous vérifions qu'elle est inférieure ou égale à une certaine précision ε

$$x_{k+1} = x_k - \left[H(x_k)\right]^{-1} \nabla f(x_k).$$

Algorithme 2.5.1. *(Algorithme de Newton pour l'optimisation)*

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

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

$$x_{k+1} = x_k - \left[H_f(x_k)\right]^{-1} \nabla f(x_k).$$

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

2.6 Méthode de quasi-Newton pour l'optimisation

La méthode de Newton nécessite l'inversion du Hessien, une opération qui peut être coûteuse et entraîner une instabilité numérique si la matrice est mal conditionnée. C'est pourquoi, dans de telles situations, on préfère utiliser la méthode Quasi-Newton.
Cette méthode propose de remplacer l'inverse de la matrice Hessienne $[H_f(x_k)]^{-1}$ par une matrice symétrique et semi-définie positive.$Q_f(x_k)$, noté Q_k On peut alors approximer le Hessien par :

$$H_f(x_k)(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k)$$

Lorsqu'on multiplie cette équation par l'inverse du Hessien, on constate que la matrice Q_{k+1} doit satisfaire la condition suivante :

$$Q_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = (x_{k+1} - x_k)$$

Ainsi, nous évitons la nécessité d'inverser une matrice, mais nous devons résoudre un système d'équations linéaires à la place.

Remarque 2.6.1. *Cette technique est très utilisée dans beaucoup de problèmes, cependant, elle nécessite un nombre d'itérations très important si le problème est mâle conditionné ou si une mauvaise estimée initiale du Hessien est utilisée.*

La première méthode quasi-Newton, appelée DFP, a été découverte en 1959 par Davidson, puis exploitée par Fletcher et Powell dans la décennie suivante. Cette méthode se base sur la variation du gradient entre deux itérations. Une autre forme récursive largement utilisée est celle appelée BFGS (Broyden, Fletcher, Goldfarb et Shanno), développée en 1970. Contrairement à la résolution d'un système d'équations linéaires, cette méthode utilise des formules algébriques telles que la multiplication de vecteurs et de matrices. Ainsi, nous avons :

$$Q_{k+1} = (I - \rho_k dx_k y_k^T) Q_k (I - \rho_k dx_k y_k^T)^T + dx_k \rho_k dx_k^T$$

Dans le cas de problèmes de grande dimension, stocker toutes les données est impossible. Une version à empreinte mémoire réduite de cette méthode, appelée L-BFGS (Limited-memory BFGS), a été développée pour pallier cette limitation.

CHAPITRE 3

Résultats de convergence générale pour les méthodes de gradient conjugué non linéaires

3.1 Introduction

En se basant sur la comparaison des divers résultats de convergence générale pour les méthodes de gradient conjugué non linéaire, il est généralement établi que sous certaines conditions, l'algorithme converge vers un point stationnaire de la fonction objectif. Ce type de convergence, qui est le plus couramment étudié pour les méthodes d'optimisation, incluant les méthodes de gradient conjugué non linéaire, est fondamental pour évaluer la performance des algorithmes. Dans ce contexte, En s'intéresse à la convergence sous la recherche linéaire de Wolfe forte et Wolfe standard, qui sont des critères de recherche de pas utilisés dans les méthodes d'optimisation. Enfin, en fait un critique faible et robuste sur ces résultats de convergence dans divers situation. Pour résoudre le problème d'optimisation non linéaire sans contrainte

$$\min_{x\in\mathbb{R}^n} f(x) \tag{3.1.1}$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction continûment différentiable, tout algorithme de gradient conjugué non linéaire génère la série $\{x_k\}$ de la forme

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.1.2}$$

où α_k est le pas obtenu par recherche linéaire et d_k est la direction de descente calculée par

$$d_{k+1} = -g_{k+1} + \beta_k d_k \tag{3.1.3}$$

pour $k \geq 0$, où β_k est le paramètre de gradient conjugué et $g_k = \nabla f(x_k)$. Dans les méthodes de gradient conjugué, $d_0 = -g_0$

Une stratégie courante pour la détermination de la taille du pas, qui joue un rôle clé dans l'efficacité (le performance) des algorithmes d'optimisation sans contrainte, consiste à accepter un pas positif α_k satisfaisant les conditions standard de recherche linéaire de Wolfe.

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k d_k^T g_k \tag{3.1.4}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma d_k^T g_k \tag{3.1.5}$$

où $0 < \rho < \sigma < 1$. La recherche linéaire de Wolfe forte est souvent utilisée dans la l'application de méthodes de gradient conjugué. Ceux-ci sont donnés par(3.1.4) et on a :

$$\left|\nabla f(x_k + \alpha_k d_k)^T d_k\right| \leq -\sigma d_k^T g_k \tag{3.1.6}$$

où encore $0 < \rho < \sigma < 1$. Observez que si $\sigma = 0$, alors la recherche linéaire de Wolfe forte se réduit à la recherche de Wolfe exacte. Dai et Yuan $(1999, 2001)$[DY99] ont prouvé que la recherche linéaire de Wolfe standard (3.1.4) et (3.1.5) assure la convergence et peut donc être utilisée avec succès dans les implémentations actuelles dans les programmes informatiques des méthodes de gradient conjugué.

Les méthodes non linéaires du gradient conjugué ont une très belle théorie, avec de nombreux résultats importants sur leur convergence. C'est l'argument principal pour lequel ces méthodes sont intensément utilisées dans la résolution d'applications pratiques d'optimisation sans contraintes. Ce chapitre est dédié à la présentation des principaux théorèmes de convergence des méthodes non linéaires du gradient conjugué, en supposant que leurs directions de descente sont la descente. Une brève présentation des types de convergence des séries générées par les algorithmes d'optimisation est tout d'abord discutée. Ensuite, le concept de gradient conjugué non linéaire est détaillé, en poursuivant avec la convergence de la méthode du gradient conjugué sous la recherche linéaire de Wolfe forte, puis sous la recherche linéaire de Wolfe standard.

3.2 Types de convergence

Partant d'un point initial x_0, chaque méthode d'optimisation sans contrainte génère une série $\{x_k\}$ de points qui, espérons-le, converge vers une solution du problème.

L'objectif de l'analyse de convergence des algorithmes d'optimisation sans contraintes est d'étudier les propriétés de la série $\{x_k\}$ sous réserve de sa convergence vers une solution du problème ou vers un point stationnaire, de voir le taux de convergence de la série et de comparer les performances de convergence de différents algorithmes.

Le taux de convergence est une caractérisation locale d'un algorithme soumise à son efficacité à résoudre un problème. Par méthodes de convergence locale, nous entendons que le point initial x_0 est proche d'un minimiseur local x^* du problème auquel le les conditions d'optimalité sont saturées .

La série $\{x_k\}$ converge vers un point x^* si

$$\lim_{k\to+\infty} \|x_k - x^*\| = 0 \tag{3.2.1}$$

Cependant, dans des situations pratiques, la solution x_k n'est pas connue et il n'est donc pas possible d'utiliser (3.2.1) comme test de convergence. Une possibilité de voir la convergence de $\{x_k\}$ est de calculer la limite

$$\lim_{k\to+\infty} \|x_{k+1} - x_k\| = 0 \tag{3.2.2}$$

Malheureusement, le critère (3.2.2) ne peut garantir la convergence de $\{x_k\}$. Par conséquent, l'étude de convergence globale des algorithmes d'optimisation sans contrainte tente de prouver la limite suivante

$$\lim_{k\to+\infty} \|g_k\| = 0 \tag{3.2.3}$$

ce qui garantit que x_k est proche de l'ensemble des points stationnaires où $g_k = \nabla f(x_k)$, ou la limite

$$\liminf_{k\to+\infty} \|g_k\| = 0 \tag{3.2.4}$$

ce qui garantit qu'au moins une sous-série de $\{x_k\}$ est proche de l'ensemble des points stationnaires. En d'autres termes, si les itérations $\{x_k\}$ restent dans une région limite, alors (3.2.3) dit que chaque point de cluster de $\{x_k\}$ sera un point stationnaire de f, tandis que (3.2.4) signifie qu'il existe au moins un point de cluster qui est un point stationnaire. point de f . Notons x_k ce qui signifie que la série $\{x_k\}$ converge vers x , c'est-à-dire que(3.2.1) est vérifié. Dans ce

qui suit, la q-convergence, qui signifie convergence-quotient, et la r-convergence, qui signifie convergence-racine, sont introduites. Plus de détails peuvent être trouvés, par exemple, dans Ortega et Rheinboldt (1970)[OR00], Potra (1989) [Pot89], Sun et Yuan (2006) [SY06] , Cătinaş (2019).[Căt19]

Définition 3.2.1. *Soit*$\{x_k\}$ *une suite de* $\mathbb{R}^n$ *et* $x^* \in \mathbb{R}^n$. *Alors, on dit que :*

1. $x_k \to x^*$ $q-$*quadratiquement si* $x_k \to x^*$ *et il existe* $K > 0$ *qui est indépendant du nombre itératif* K *de sorte que*

$$\|x_{k+1} - x^*\| \leq K \|x_k - x^*\|^2 \tag{3.2.5}$$

2. $x_k \to x^*$ $q-$*superlinéaire de* $q-$*ordre ,* $p > 1$ *si* $x_k \to x^*$ *et il existe* $K > 0$ *qui est indépendant du nombre itératif* k, *de sorte que :*

$$\|x_{k+1} - x^*\| \leq K \|x_k - x^*\|^p \tag{3.2.6}$$

3. $x_k \to x^*$ $q-$*superlinéaire si*

$$\lim_{k \to +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0 \tag{3.2.7}$$

4. $x_k \to x^*$ $q-$*linéairement avec le* $q-$*facteur* $\sigma \in (0, 1)$ *si*

$$\|x_{k+1} - x^*\| \leq \sigma \|x_k - x^*\| \tag{3.2.8}$$

pour k *suffisamment grand.*

Souvent, p est appelé $q-$ordre et K est $q-$facteur . Notez que la convergence $q-$linéaire $(p = 1)$ signifie que asymptotiquement le point x_{k+1} se rapproche de x avec $\log_{10} K$ chiffres plus corrects que x_k : Par conséquent, dans ce cas, le nombre de chiffres corrects augmente linéairement en fonction du nombre d'itérations. D'autre part, pour une convergence $q-$superlinéaire de $q-$ordre p, le nombre de chiffres corrects supplémentaires augmente asymptotiquement d'un $p-$facteur , c'est-à-dire que le nombre de chiffres corrects est exponentiel en fonction du nombre d'itérations, et donc de la convergence est rapide.

Une séquence q-superlinéairement convergente est également q-linéairement convergente avec le facteur q r pour tout $r > 0$. Une séquence $q-$quadratiquement convergente est $q-$superlinéairement convergente avec un ordre q égale à 2.

La motivation pour introduire le taux de $q-$convergence était de comparer la vitesse de convergence des algorithmes. On voit que le taux de $q-$convergence dépend plus de p et moins

de K. Supposons qu'il existe deux série $\{x_k\}$ et $\{\hat{x}_k\}$, où leur$q-$ordre et le $q-$facteur sont $\{p, K\}$ et $\left\{\hat{p}, \hat{K}\right\}$, respectivement. Si $p > \hat{p}$, alors la série $q-$ordre p converge plus rapidement que la série $q-$ordre $\hat{p}$: En d'autres termes, les séeie convergentes $q-$quadratiquement convergent finalement plus rapidement que les série convergentes $q-$superlinéaire et $q-$linéaire. Lorsque$p = \hat{p}$, c'est-à-dire que les série ont le même $q-$ordre de convergence, si $K < \hat{K}$, alors la séquence $\{x_k\}$ est plus rapide que $\{\hat{x}_k\}$. Habituellement, nous nous intéressons aux série convergentes $q-$superlinéaire et $q-$quadratiquement. Dans ce cas, on dit que la série a un taux de convergence rapide.

Définition 3.2.2. *Une méthode itérative de calcul de x^* est dite localement ($q-$quadratiquement, $q-$superlinéaire, $q-$linéaire, etc.) convergente si les itérations x_k générées par la méthode convergent vers x^* ($q-$quadratiquement, $q-$ superlinéaire, $q-$linéaire, etc.), à condition que le point initial de la méthode itérative soit suffisamment proche de x^* .*

Une autre mesure du taux de convergence d'une série $\{x_k\}$, qui est plus faible que le taux de $q-$convergence , est le taux de $r-$convergence . La motivation de l'introduction du taux de $r-$convergence est la suivante. Il existe des situations où la précision de l'itération peut être améliorée au moyen de certaines procédures externes à l'algorithme, par exemple en évaluant la fonction objectif et son gradient avec une précision croissante. Dans de tels cas, il n'y a aucune garantie que la précision de l'itération augmente de manière monotone, mais seulement que la précision des résultats s'améliore à un rythme déterminé par l'amélioration de la précision dans les évaluations de gradient de fonction. Une autre situation est celle où certaines série convergent encore assez rapidement, mais dont la vitesse de convergence est variable. Le concept de taux de $r-$convergence rend compte de ces situations (Kelley, 1999) [Kel99].

Définition 3.2.3. *Soit $\{x_k\}$ une suite de $\mathbb{R}^n$ et $x^* \in \mathbb{R}^n$. soit*

$$r_\delta = \begin{cases} \lim\limits_{k\to+\infty} \sup \|x_k - x^*\|^{\frac{1}{k}}, & si\ \delta = 1 \\ \lim\limits_{k\to+\infty} \sup \|x_k - x^*\|^{\frac{1}{\delta^k}}, & si\ \delta > 1 \end{cases} \tag{3.2.9}$$

Si $r_1 = 0$, alors $\{x_k\}$ est dit r_1-superlinéairement convergent vers x^* .

Si $0 < r_1 < 1$, alors $\{x_k\}$ est dit r_1-linéairement convergent vers x^* .

Si $r_1 \geq 1$, alors $\{x_k\}$ est dit r_1-sublinéairement convergent vers x^* .

De même, si $r_2 = 0$, $0 < r_2 < 1$ et $r_2 \geq 1$, respectivement, alors x^* est dit convergent r_2-superquadratiquement, r_2-quadratiquement et r_2-subquadratiquement vers x^* , respectivement.

Observez que lorsque $\{x_k\}$ converge vers x^* , alors il existe toujours un indice $k_0 \geq 0$ tel que $0 \leq \|x_k - x^*\| < 1$, pour tout $k \geq k_0$: Donc, pour tout $\delta \geq 1$, nous avons $0 \leq r_\delta < 1$ Dans ce cas, δ est appelé $r-$ordre et r_δ est $r-$facteur Plus $r-$ordre est élevé, plus la série $\{x_k\}$ converge rapidement. Lorsque deux séries ont le même $r-$ordre , plus le $r-$facteur est petit, plus la série correspondante converge rapidement (Sun & Yuan, 2006 [Yua99]). Une autre définition du taux de $r-$convergence , qui présente le lien avec le taux de $q-$convergence , est la suivante.

Définition 3.2.4. *Soit $\{x_k\}$ une série de $\mathbb{R}^n$ et $x^* \in \mathbb{R}^n$. La série $\{x_k\}$ converge vers x^* $r-$(quadratiquement, superlinéairement, linéairement) s'il existe une série $\{\gamma_k\}$ de $\mathbb{R}$ convergeant $q-$(quadratiquement, superlinéairement, linéairement) vers zéro de sorte que*

$$\|x_k - x^*\| \leq \gamma_k$$

La séquence $\{x_k\}$ converge de manière $r-$superlinéaire avec $r-$ordre $p > 1$ si la séie $\{x_k\}$ converge vers zéro $q-$superlinéaire avec $q-$ordre p

En général, pour analyser la convergence des algorithmes, le taux de $q-$convergence est utilisé. Un algorithme avec un taux de convergence $q-$superlinéaire ou $q-$quadratique est considéré comme un bon algorithme. Par conséquent, nous nous intéressons à la conception d'algorithmes superlinéaires ou quadratiquement convergents.

3.3 Résultats généraux de convergence pour les méthodes de gradient conjugué non linéaires

Comme nous l'avons déjà vu en (3.1.2), $d_0 = -g_0$. La sélection $d_0 = -g_0$ est critique dans les algorithmes de gradient conjugué. Une propriété très importante de la méthode du gradient conjugué linéaire est qu'elle se termine après au plus n itérations si $f(x)$ est une fonction quadratique convexe et si la première direction de descente est $d_0 = -g_0$. Cependant, pour une fonction non linéaire générale, elle ne peut être approchée de près par une fonction quadratique qu'après un certain nombre d'itérations. Par conséquent, l'analyse locale ne peut pas s'appliquer pour montrer la terminaison quadratique car dans ce cas $d_k \neq -g_k$ pour $k > 1$, en raison des itérations précédentes. Crowder et Wolfe (1969) ont donné un exemple tridimensionnel montrant que même pour une quadratique fortement convexe, le taux de convergence est linéaire si la

direction de descente initiale n'est pas la direction qui garantit une descente optimale. Powell (1976) a obtenu un résultat plus fort, montrant que si la fonction objectif est quadratique convexe et si la direction de descente initiale est une direction de descente arbitraire, alors soit l'optimalité est atteinte en au plus $n+1$ itérations, soit le taux de convergence n'est que linéaire. Yuan (1993) a donné une étude théorique montrant que la méthode du gradient conjugué appliquée aux fonctions quadratiques convexes ne converge toujours que linéairement si la terminaison finie ne se produit pas. Certains détails sur cette sélection de d_0 sont également donnés par Andrei (2011) [And11]

Un algorithme de gradient conjugué donné par (3.1.2) et (3.1.3) génère une série $\{x_k\}$. L'intérêt est de voir dans quelles conditions cette suite converge vers la solution x^* du problème (3.1.1). Puisque l'algorithme donné par (3.1.2) et (3.1.3) ne dépend que du paramètre β_k, il s'ensuit que l'intérêt est de voir les valeurs de ce paramètre pour lequel l'algorithme est convergent.

Une exigence importante pour les méthodes d'optimisation basées sur la recherche linéaire est que la direction de descente doit être une direction descendante. La direction de decsnte d_k satisfait la propriété de descente, c'est-à-dire qu'il s'agit d'une direction de descente si pour tout $k = 1, 2, ...$

$$d_k^T g_k < 0 \tag{3.3.1}$$

Pour les méthodes du gradient conjugué, de (3.1.3), il s'ensuit que :

$$d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2 + \beta_k g_{k+1}^T d_k \tag{3.3.2}$$

Maintenant, si la recherche linéaire est exacte, c'est-à-dire si ${}_k g_{k+1}^T d_k = 0$, alors $\mathrm{g} d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2$. Par conséquent, d_{k+1} est une direction de descente si $g_{k+1} \neq 0$ Cependant, pour la recherche linéaire inexacte, cela peut ne pas être vrai. En utilisant le redémarrage avec $d_{k+1} = -g_{k+1}$, cette situation peut être corrigée. La direction de descente d_k satisfait la propriété de descente suffisante, c'est-à-dire qu'elle est une direction de descente suffisante si

$$d_k^T g_k < -c\|g_k\|^2 \tag{3.3.3}$$

pour tout $k = 1\ , 2, ...$; où $c > 0$ est une constante.

Les propriétés de convergence d'une méthode de recherche linéaire, comme le gradient conjugué non linéaire, peuvent être étudiées en mesurant l'efficacité de la direction de decente et de la

longueur du pas. La qualité d'une direction de decente d_k peut être déterminée en étudiant l'angle entre g_k et la direction de descente d_k définie par

$$\cos(\theta) = \frac{-d_k^T g_k}{\|g_k\| \|d_k\|} \tag{3.3.4}$$

Pour établir les résultats généraux de convergence de toute méthode des formes (3.1.2) et (3.1.3), les hypothèses de base suivantes sur la fonction objectif sont introduites.

Hypothèses GC :

1. L'ensemble de niveaux $S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ est borné, c'est-à-dire qu'il existe une constante $B > 0$ telle que $\|x\| \leq B$ pour tout x dans l'ensemble de niveaux.

2. Dans un certain voisinage N de l'ensemble de niveaux, f est continûment différentiable et son gradient est continu de Lipschitz, c'est-à-dire qu'il existe une constante $L > 0$ telle que

$$\|g(x) - g(y)\| \leq L \|x - y\|, \text{ pour tout } x, y \in N \tag{3.3.5}$$

Notez que ces hypothèses impliquent qu'il existe une constante Γ telle que $\|g(x)\| \leq \Gamma$, pour tout x de l'ensemble de niveaux S. La partie 1 de L'hypothèse GC n'est pas nécessaire dans tous les cas. Seule l'hypothèse selon laquelle f est bornée inférieument de l'ensemble de niveaux peut être utilisée pour l'analyse de convergence globale.

Sous l'hypothèse GC, le théorème suivant, dû à Zoutendijk (1970) [Zou70] et Wolfe (1969, 1971)([Wol69], [Wol71]), est essentiel pour prouver les résultats de convergence globale des algorithmes d'optimisation sans contrainte, y compris celui à gradient conjugué.

Théorème 3.3.1. *Supposons que f soit borné inférieument dans $\mathbb{R}^n$ et que f soit continûment dérivable dans un voisinage N de l'ensemble de niveaux $S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. Supposons également que le gradient soit Lipschitzien, c'est-à-dire qu'il existe une constante $L > 0$ de sorte que (3.3.5) est satisfait pour tout $x, y \in N$. Considérons toute itération de la forme (3.1.2), où d_k est une direction de descente et α_k satisfait les conditions de recherche linéaire Wolfe (3.1.4) et (3.1.5). Alors*

$$\sum_{k=1}^{\infty} \cos^2 \theta \|g_k\|^2 < \infty \tag{3.3.6}$$

Preuve. De (3.1.5) il résulte que

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

En revanche, la continuité Lipschitzienne (3.3.5) donne

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

La combinaison de ces deux relations donne donc

$$\alpha_k \geq \frac{(\sigma - 1)}{L} \frac{g_k^T d_k}{\|d_k\|^2} \tag{3.3.7}$$

Maintenant, en utilisant la première condition de Wolfe (3.1.4) et (3.3.7), il en résulte que

$$f_{k+1} \leq f_k + \rho \frac{(\sigma - 1)}{L} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} \tag{3.3.8}$$

De la définition (3.3.4) de $\cos\theta_k$, il s'ensuit que (3.3.8) peut s'écrire

$$f_{k+1} \leq f_k + c \cos\theta_k \|g_k\|^2 \tag{3.3.9}$$

où $c = \dfrac{\rho\,(1-\sigma)}{L}$. En additionnant (3.3.9) pour $k \geq 1$ et en considérant que f est borné en dessous, on obtient (3.3.6). □

La relation (3.3.6) est appelée condition de Zoutendijk, et à partir de (3.3.4), elle peut être réécrite comme

$$\sum_{k=1}^{\infty} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < \infty \tag{3.3.10}$$

Il est intéressant de voir comment la condition de Zoutendijk est utilisée pour obtenir des résultats de convergence globale (Nocedal, 1992) [Noc92]. Supposons que l'itération (3.1.2) soit telle que

$$\cos\theta_k \geq \delta > 0 \tag{3.3.11}$$

pour tout k. Alors, de (3.3.6), on à :

$$\lim_{k\to\infty} \|g_k\| = 0 \tag{3.3.12}$$

En d'autres termes, si la direction de descente générée par une méthode d'optimisation sans contrainte n'a pas tendance à être orthogonale au gradient, alors la série de gradients correspondante converge vers zéro. Pour les méthodes de recherche linéaire (3.1.2), la limite (3.3.12) est le meilleur type de résultat de convergence globale pouvant être obtenu. Nous ne pouvons pas garantir que la méthode converge vers les minimisateurs de la fonction f, mais seulement

qu'elle converge vers des points stationnaires. Les implications de la condition de Zoutendijk sont les suivantes.

1. Pour la descente la plus rapide avec la recherche linéaire de Wolfe, $\cos\theta_k = 1$ pour tout k. Ainsi, la méthode de descente la plus rapide n'est globalement convergente que si les pas sont correctement calculés.

2. Considérons les méthodes de type Newton, où la direction de descente est calculée comme $d_k = -B_k^{-1} g_k$, où B_k est une matrice symétrique non singulière, ($B_k = I, B_k = \nabla^2 f(x_k)$ ou B_k est une approximation définie symétrique et positive du hessien $\nabla^2 f(x_k)$). En supposant que le numéro de condition des matrices B_k est uniformément borné, c'est-à-dire pour tout k, $\|B_k\| \left\|B_k^{-1}\right\| \leq \Delta$, où $\Delta > 0$ est une constante, alors de (3.3.4) on à :

$$\begin{aligned} \cos\theta_k &= -\frac{g_k^T d_k}{\|g_k\| \, \|d_k\|} = \frac{g_k^T B_k^{-1} g_k}{\|g_k\| \left\|B_k^{-1} g_k\right\|} \\ &\geq \frac{1}{\|g_k\|} \frac{\|g_k\|^2}{\|B_k\|} \frac{1}{\left\|B_k^{-1}\right\| \|g_k\|} \\ &= \frac{1}{\|B_k\| \left\|B_k^{-1}\right\|} \geq \frac{1}{\Delta} \end{aligned}$$

Ainsi,$\cos\theta_k \geq \dfrac{1}{\Delta}$, c'est-à-dire qu'il est borné en s'éloignant de 0. Ainsi, la méthode de Newton ou les méthodes quasi-Newton sont globalement convergentes si les matrices B_k sont définies positives (condition de descente), si leur numéro de condition est borné et si la recherche linéaire satisfait aux conditions de Wolfe. Notons que la condition (3.3.11) est cruciale pour obtenir ces résultats.

3. Pour les méthodes du gradient conjugué, il n'est pas possible de montrer la limite (3.3.12), mais seulement un résultat plus faible, c'est-à-dire

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \tag{3.3.13}$$

Ce type de résultat est également obtenu à partir de la condition de Zoutendijk. En effet, supposons que (3.3.13) ne soit pas vérifié, c'est-à-dire que les gradients restent bornés loin de zéro. En d'autres termes, supposons qu'il existe une constante $c > 0$ telle que pour tout k

$$\|g_k\| \geq \gamma \tag{3.3.14}$$

Dans ce cas, toujours à partir de la condition de Zoutendijk (3.3.6), alors on à

$$\cos\theta_k \to 0 \tag{3.3.15}$$

Par conséquent, l'algorithme ne peut échouer au sens de (3.3.14) que si la série $\{\cos\theta_k\}$ converge vers zéro. Par conséquent, pour établir (3.3.13), il suffit de montrer qu'une sous-suite $\{\cos\theta_k\}$ de la série $\{\cos\theta_k\}$ est bornée en dehors de zéro.

Présentons maintenant quelques conditions sur β_k, qui déterminent la convergence des algorithmes de gradient conjugué. Supposons que $\beta_k \geq 0$ et que la direction de descente d_k soit une direction de descente, c'est-à-dire $g_k^T d_k < 0$. A ce moment, nous cherchons à trouver un β_k qui produit une direction de descente d_{k+1}, c'est-à-dire une direction qui satisfait

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k g_{k+1}^T d_k < 0 \tag{3.3.16}$$

Proposition 3.3.1. *Suppose that $\beta_k > 0$, si*

$$\beta_k \leq \frac{\|g_{k+1}\|^2}{d_k^T y_k} \tag{3.3.17}$$

alors d_{k+1} est une direction de descente pour la fonction f

Théorème 3.3.2. *Supposons que l'hypothèse CG soit vérifiée. Soit $\{x_k\}$ la série généré par les algorithmes (3.1.2) et (3.1.3), où β_k satisfait (3.3.17). Alors, l'algorithme soit détermine un point stationnaire, soit converge dans le sens où*

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

Gilbert et Nocedal (1992) [GN92b] ont prouvé la convergence de la méthode du gradient conjugué sous la condition de descente suffisante (3.3.3). En fait, cette condition (3.3.3) est souvent implicite ou requise dans de nombreuses analyses de convergence des méthodes du gradient conjugué, par exemple voir : Hestenes et Stiefel (1952) [HS+52], Al-Baali (1985), Touati-Ahmed et Storey (1990). , Hu et Storey (1991), Grippo et Lucidi (1997)

3.3.1 Convergence sous la recherche linéaire de Wolfe forte

Le théorème suivant montre que si β_k est choisi pour satisfaire la condition (3.3.17) pour tout k, alors sous les conditions de recherche linéaire de Wolfe fortes (3.1.4) et (3.1.6) la direction (3.1.3) satisfait la condition de descente suffisante (3.3.3)

Théorème 3.3.3. *Supposons que x_0 soit un point initial et que l'hypothèse GC soit vérifiée. Soit $\{x_k\}$ la série générée par l'algorithme de gradient conjugué (3.1.2) et (3.1.3). Si β_k est tel*

que

$$\begin{aligned} \|g_{k+1}\|^2 &\geq \beta_k d_k^T y_k, \\ \beta_k &> 0 \end{aligned}$$

et le pas α_k satisfait les conditions de Wolfe fortes (3.1.4) et (3.1.6), alors l'algorithme de gradient conjugué satisfait la condition de descente suffisante (3.3.3) avec $c = \dfrac{1}{1+\sigma}$

Théorème 3.3.4. *Supposons que l'hypothèse GC soit vérifiée. Considérons n'importe quelle méthode générale de gradient conjugué donnée par (3.1.2) et (3.1.3), où la taille du pas α_k est déterminée par la recherche linéaire de Wolfe forte (3.1.4) et (3.1.6). Alors, soit*

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \tag{3.3.18}$$

où

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty \tag{3.3.19}$$

Corollaire 3.3.1. *Supposons que l'hypothèse GC soit valable et considère toute méthode du gradient conjugué donnée par (3.1.2) et (3.1.3), où la taille du pas α_k est déterminée par la recherche linéaire de Wolfe forte (3.1.4) et (3.1.6). Si*

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^t}{\|d_k\|^2} = \infty \tag{3.3.20}$$

pour tout $t \in (0, 4)$, alors la méthode converge dans le sens où (3.3.18) est réalisé.

Le théorème suivant, qui introduit la condition Nocedal, présente un résultat de convergence générale pour toute méthode de gradient conjugué (3.1.2) et (3.1.3) sous la recherche forte de droite de Wolfe (3.1.4) et (3.1.6). Principalement, le théorème dit que si $\|d_k\|^2$ augmente au plus linéairement, c'est-à-dire si $\|d_k\|^2 \leq c_1 k + c_2$ pour tout k, où c_1 et c_2 sont des constantes, alors une méthode de gradient conjugué avec une forte recherche de droite de Wolfe est globalement convergent. Le théorème est prouvé par Dai (2011).[Dai11] Voir également (Nocedal, 1996).

Théorème 3.3.5. *Supposons que l'hypothèse GC soit vérifiée. Considérons n'importe quelle méthode de gradient conjugué (3.1.2) et (3.1.3) avec d_k satisfaisant $g_k^T d_k < 0$ et avec une recherche linéaire de Wolfe forte (3.1.4) et (3.1.6). Alors la méthode est globalement convergente si*

$$\sum_{k=1}^{\infty} \frac{1}{\|d_k\|^2} = \infty \tag{3.3.21}$$

Théorème 3.3.6. *Supposons que l'hypothèse GC soit vérifiée. Considérons une méthode de gradient conjugué donnée par* (3.1.2) *et* (3.1.3)*, où la direction de descente d_k est la descente, c'est-à-dire $g_k^T d_k < 0$. Considérons que la taille du pas est déterminée par les conditions de recherche linéaire de Wolfe fortes*(3.1.4) *et* (3.1.6)*. Si β_k satisfait*

$$\sum_{k=1}^{\infty} \prod_{j=1}^{k} \beta_j^{-2} = \infty, \tag{3.3.22}$$

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

Le théorème présédente montre que la convergence globale de tout algorithme de gradient conjugué est obtenue si les paramètres du gradient conjugué β_k satisfont à la condition (3.3.22) et que la taille du pas est déterminée par la recherche linéaire de Wolfe forte. Principalement, il est basé sur la condition de Zoutendijk. Il est à noter que dans le théorème présédente, c'est la condition de descente (3.3.1) qui est utilisée et non la condition de descente suffisante (3.3.3).

3.3.2 Convergence sous la recherche linéaire de Wolfe standard

Dai (2010) a prouvé que la conclusion du théorème 3.3.6 pour la convergence globale de toute méthode de gradient conjugué est également valable dans le cadre de la recherche linéaire de Wolfe standard. Ce résultat est basé sur la proposition suivante prouvée par Dai et Yuan (2003).

Proposition 3.3.2. *Considérez n'importe quelle méthode de gradient conjugué* (3.1.2) *et* (3.1.3)*. Définissons Φ_k et t_k comme suit :*

$$\Phi_k^2 = \begin{cases} \|g_k\|^2 \ \ si\ k = 0 \\ \prod_{j=0}^{k-1} \beta_j^{-2} \ \ si\ k \geq 1 \end{cases} \tag{3.3.23}$$

et

$$t_k = -2 \sum_{i=0}^{k} \frac{g_i^T d_i}{\Phi_i^2} - \sum_{i=0}^{k} \frac{\|g_i\|^2}{\Phi_i^2} \tag{3.3.24}$$

Proposition 3.3.3. *Supposons que $\{a_i\}$ et $\{b_i\}$ soient deux suites de nombres réels positifs, satisfaisant :*

$$b_k \leq c_1 + c_2 \sum_{i=1}^{k} a_i, \ \ pour\ tout\ k$$

où c_1, c_2 deux constante positive. Si la somme $\sum_{k\geq 1} a_k$ est divergente, alors $\sum_{k\geq 1} \frac{a_k}{b_k}$ est également divergent.

Le théorème suivant, prouvé par Dai (2010), montre que la condition (3.3.22) sur β_k est suffisante pour la convergence globale de toute méthode de gradient conjugué (3.1.2) et (3.1.3).

Théorème 3.3.7. *Supposons que l'hypothèse GC soit vérifiée. Considérons une méthode de gradient conjugué donnée par* (3.1.2) *et* (3.1.3)*, où d_k la direction de descente, c'est-à-dire $d_k^T g_k < 0$ et la taille du pas est déterminée par les conditions de recherche linéaire de Wolfe standard* (3.1.4) *et* (3.1.5)*. Si β_k satisfait* (3.3.22)*, alors* $\lim_{k\to\infty} \inf \|g_k\| = 0$.

Les théorèmes ci-dessus présentent la condition nécessaire et suffisante sur le paramètre de gradient conjugué β_k, à savoir (3.3.22), pour la convergence globale de toute méthode générale de gradient conjugué sous la recherche linéaire de Wolfe standard.

Dans ce qui suit, deux propriétés qui établissent certaines conditions sur le paramètre β_k pour assurer la convergence de la méthode du gradient conjugué correspondante sont discutées. La première est due à Gilbert et Nocedal (1992). La seconde a été développée par Dai (2010).

Propriété (*) Gilbert et Nocedal (1992)

Pour prouver la convergence de la méthode du gradient conjugué, Gilbert et Nocedal (1992) ont introduit ce que l'on appelle la propriété (*). L'idée est qu'en plus de $\beta_k > 0$, cela nécessite que β_k soit petit lorsque le pas $s_k = x_k x_{k-1}$ est petit. Formellement, cette propriété est la suivante :

Propriété (*) Considérons n'importe quelle méthode de gradient conjugué (3.1.2) et (3.1.3). Supposons que pour tout $k \geq 0$, $0 < \gamma \leq \|g_k\| \leq \Gamma$. Sous cette hypothèse, nous disons que la méthode a la propriété (*) s'il existe des constantes $b > 1$ et $\lambda > 0$ de sorte que pour tout k

$$|\beta_k| \leq b$$

et

$$\|s_k\| \leq \lambda \Rightarrow |\beta_k| \leq \frac{1}{2b}$$

Gilbert et Nocedal (1992) ont prouvé que si les gradients sont bornés à partir de zéro et si la méthode a la propriété (*), alors une fraction des pas ne peut pas être trop petite. Par conséquent, la propriété (*) détermine la convergence des méthodes de gradient conjugué.

Théorème 3.3.8. *Supposons que l'hypothèse GC soit vérifiée et considérons toute méthode de gradient conjugué* (3.1.2) *et* (3.1.3) *avec les propriétés suivantes : $\beta_k \geq 0$ pour tout k, le Zoutendijk, la descente suffisante et les conditions de propriété (*) sont vérifiées. Alors,* $\lim_{k\to\infty} \inf \|g_k\| = 0$.

Sous l'hypothèse GC, les Polak–Ribiére–Polyak (PRP) et les Hestenes–Stiefel (HS) ont la Propriété (*). Si β_k a la propriété (*), alors $|\beta_k|$ et $\beta_k^+ = \max\{0, \beta_k\}$. Par conséquent, de nombreux autres choix de $\beta_k > 0$ conduisent à des algorithmes avec Propriété (*).

Propriété (#) Dai (2010)

Cette propriété a été introduite par Dai (2010) comme une généralisation de Property (*). L'idée était d'assouplir les limites de β_k sous réserve d'une série positive et uniformément délimitée.

Propriété (#) Considérons une méthode de gradient conjugué donnée par (3.1.2) et (3.1.3) et supposons que pour tout $k \geq 0$, $0 < \gamma \leq \|g_k\| \leq \Gamma$. Sous cette hypothèse, nous disons que la méthode a la propriété (#) s'il existe une suite $\{\varphi_k\}$ positive et uniformément bornée et les constantes $b \geq 1$ et $\lambda > 0$, de sorte que pour tout k

$$|\beta_k| \leq b\frac{\varphi_k}{\varphi_{k-1}}$$

et

$$\|s_k\| \leq \lambda \Rightarrow |\beta_k| \leq \frac{1}{b}\frac{\varphi_k}{\varphi_{k-1}}$$

La propriété (#) a la propriété (*) comme cas particulier, c'est-à-dire que si la propriété (*) est vraie, alors la propriété (#) doit être vraie avec $\varphi_k = 1$. Le théorème suivant montre comment le théorème 3.3.7 et la propriété (#) peuvent être utilisé pour analyser la convergence globale des méthodes de gradient conjugué.

Théorème 3.3.9. *Supposons que l'hypothèse GC soit vérifiée et considérons toute méthode de gradient conjugué* (3.1.2) *et* (3.1.3)*, où β_k a la propriété (#) avec $b = 1$. Supposons que d_k est la direction de descente , c'est-à-dire $d_k^T g_k < 0$ pour tout k. Si le pas α_k satisfait aux conditions standard de Wolfe* (3.1.4) *et* (3.1.5)*, alors* $\lim_{k\to\infty} \inf \|g_k\| = 0$.

Remarquer la différence entre la convergence des méthodes de gradient conjugué avec Property (*) et avec Property (#). Dans le théorème 3.3.8, la convergence est prouvée si la propriété (*) et la condition de descente suffisante sont satisfaites. En revanche, dans le théorème 3.3.9, la convergence est prouvée si la propriété (#) et seules les conditions de descente sont satisfaites. En utilisant la propriété (*), Gilbert et Nocedal (1992) ont prouvé la convergence des algorithmes de gradient conjugués Hestenes – Stiefel (HS) et Polak – Ribiére – Polyak (PRP). En revanche, en utilisant la propriété (#), Dai (2010) a prouvé la convergence de Fletcher-Reeves (FR), Polak–Ribiére–Polyak (PRP), Dai–Yuan (DY), ainsi que des méthodes hybrides FR -Méthodes de gradient conjugué PRP et DY-HS.

3.4 Critique des résultats de convergence

L'ingrédient le plus important pour prouver la convergence globale des méthodes de gradient conjugué est la condition de Zoutendijk, initialement donnée par Zoutendijk ([Zou70]) (1970) et Wolfe (1969, 1971) ([Wol69]; [Wol71]).

Normalement, la direction d_k est choisie de telle sorte qu'il s'agisse d'une direction de descente, à savoir $d_k^T g_k < 0$, si $g_k \neq 0$. En utilisant la première condition standard de recherche linéaire de Wolfe (3.1.4), il s'ensuit que :

$$\alpha_k \geq c\frac{-d_k^T g_k}{\|d_k\|^2} \tag{3.4.1}$$

où c est une constante positive. Puisque la direction est de descente, il s'ensuit que $\alpha_k > 0$. Par ailleurs, en utilisant (3.4.1), à partir de (3.1.4),on à :

$$f(x_k) - f(x_{k+1}) \geq \rho c\frac{\left(-d_k^T g_k\right)^2}{\|d_k\|^2} \tag{3.4.2}$$

Par conséquent, si $\{f(x_k)\}$est borné en dessous, alors de (3.4.2), on à :

$$\sum_{k=1}^{\infty} \frac{\left(-d_k^T g_k\right)^2}{\|d_k\|^2} < \infty, \tag{3.4.3}$$

Mais

$$\cos^2\theta_k = \frac{\left(-d_k^T g_k\right)^2}{\|d_k\|^2 \|g_k\|^2}$$

où θ_k est l'angle entre la direction de descente la plus rapide et la direction de descente d_k. Donc, d'après (3.4.3),

$$\sum_{k=1}^{\infty} \|g_k\|^2 \cos^2\theta_k < \infty \tag{3.4.4}$$

Si $f(x)$ est borné inférieure. Alors (3.4.4) est exactement la condition de Zoutendijk (3.3.10), qui implique la convergence de la méthode.

Par conséquent, à partir des développements ci-dessus, les résultats généraux de convergence suivants peuvent être établis pour les algorithmes de recherche linéaire [voir Wolfe (1969, 1971) ([Wol69]; [Wol71]), Zoutendijk (1970) ([Zou70])].

Théorème 3.4.1. *Soit $\{x_k\}$ la série générée par un algorithme de recherche linéaire sous la recherche linéaire exacte ou sous toute recherche linéaire inexacte qui satisfait (3.4.2). Si*

$$\sum_{k=1}^{\infty} \cos^2\theta_k = \infty$$

alors la suite $\{x_k\}$ *est convergente dans le sens où* $\lim_{k\to\infty} \inf \|g_k\|^2 = 0$ *De plus, s'il existe une constante positive* η *telle que* $\cos^2\theta_k \geq \eta$ *pour tout* k*, alors la suite est convergente dans le sens où* $\lim_{k\to\infty} \inf \|g_k\| = 0$.

Dans les algorithmes de gradient conjugué, on suppose que le paramètre β_k est choisi de telle sorte que la condition de descente suffisante $d_k^T g_k \leq -c\|g_k\|^2$ soit satisfaite pour une constante positive c. Par exemple, le théorème 3.3.8, prouvé par Gilbert et Nocedal (1992), montre que si la condition de descente suffisante et la propriété (*) sont satisfaites, alors sous l'hypothèse GC, l'algorithme général du gradient conjugué est convergent, soit $\lim_{k\to\infty} \inf \|g_k\| = 0$. D'autre part, le théorème 3.3.9, prouvé par Dai (2010), montre que si la condition de descente et la propriété (#) sont satisfaites, alors sous l'hypothèse GC, l'algorithme général du gradient conjugué est convergent dans le sens où $\lim_{k\to\infty} \inf \|g_k\| = 0$.

Les preuves sur la convergence des méthodes du gradient conjugué sont principalement basées sur l'estimation de

$$\sum_{k=1}^{\infty} \cos^2\theta_k = \sum_{k=1}^{\infty} \frac{1}{\|d_k\|^2} \left(\frac{\left(d_k^T g_k\right)^2}{\|g_k\|^2} \right), \tag{3.4.5}$$

Si $\dfrac{\left(d_k^T g_k\right)^2}{\|g_k\|^2}$ est borné en dehors de zéro, alors du théorème 3.3.10, il s'ensuit que la condition

$$\sum_{k=1}^{\infty} \frac{1}{\|d_k\|^2} = \infty, \tag{3.4.6}$$

connue sous le nom de condition Nocedal, implique la convergence de la méthode du gradient conjugué. Par conséquent, dans l'analyse de convergence d'une méthode de gradient conjugué, une technique largement utilisée consiste à dériver une contradiction en établissant la relation ci-dessus s'il existe une constante positive γ telle que $\|g_k\| \geq \gamma$, pour tout k. Observez que sous le caractère borné de $\|g_k\|$, le caractère borné de $\left(\dfrac{\left(d_k^T g_k\right)^2}{\|g_k\|^2} \right)$ est équivalent à la condition de descente suffisante (3.3.3).

Les résultats de convergence sur les méthodes de gradient conjugué obtenus par Dai et Yuan (1996) montrent que la condition de descente suffisante $d_k^T g_k \leq -c\|g_k\|$ n'est pas toujours nécessaire [voir aussi (Yuan, 1998)]. Au lieu de cela, cette condition doit être satisfaite au sens moyen, c'est-à-dire que la valeur moyenne de $-d_k^T g_k$, toutes les deux itérations consécutives, doit être bornée à partir de zéro. En d'autres termes, la condition suffisante $d_k^T g_k \leq -c\|g_k\|$

peut être remplacée par

$$\frac{\left(d_k^T g_k\right)^2}{\|g_k\|^4} + \frac{\left(d_{k+1}^T g_{k+1}\right)^2}{\|g_{k+1}\|^4} \geq c, \text{ pour tout } k > 0 \tag{3.4.7}$$

Théorème 3.4.2. *Si* $\{f(x_k)\}$ *est borné en dessous,*$\{\beta_k\}$ *est borné et* (3.4.6) *est vrai, alors la série* $\{x_k\}$ *générée par le gradient conjugué* (3.1.2) *et* (3.1.3) *sous la recherche linéaire de Wolfe forte* (3.1.4) *et* (3.1.6) *converge dans le sens où* $\lim_{k\to\infty} \inf \|g_k\| = 0$

Le théorème ci-dessus, prouvé par Yuan (1998) [Yua99], montre qu'une technique essentielle pour prouver la convergence des méthodes de gradient conjugué est d'essayer d'obtenir des limites sur le taux d'augmentation de$\|d_k\|$ de telle sorte que (3.4.6) soit vérifié. Pour estimer les bornes de $\|d_k\|$, une méthode directe consiste à utiliser (3.1.3) de manière récursive. Cela implique donc une certaine inégalité sur β_k. En d'autres termes, les résultats de convergence des méthodes du gradient conjugué s'établissent sous certaines inégalités sur β_k, ce qui est tout à fait normal car la direction de descente dans les méthodes de gradient conjugué dépend du paramètre β_k. Dans cette présentation, de telles conditions sur β_k sont données par (3.3.17) ou (3.3.22).

Etude numérique. Dans l'analyse de convergence sur les méthodes de gradient conjugué, la recherche linéaire de Wolfe standard et les conditions de recherche linéaire de Wolfe fortes sont utilisées. Afin de voir l'efficacité des conditions de recherche minéaire de Wolfe et de faire une comparaison entre la recherche linéaire de Wolfe standard et la recherche minéaire de Wolfe forte, considérons l'expérience numérique suivante utilisant l'ensemble de 80 problèmes de test d'optimisation sans contrainte de notre collection UOP. (Andrei, 2018). Nous rapportons les résultats numériques obtenus avec une implémentation Fortran de la méthode du gradient conjugué de Hestenes et Stiefel (HS) avec à la fois la recherche linéaire de Wolfe standard et la recherche linéaire de Wolfe forte. La recherche linéaire standard de Wolfe utilise les implémentations de Shanno (1983) avec quelques modifications mineures supplémentaires d'Andrei (1995). La recherche linéaire de Wolfe forte utilise les implémentations de Moré et Thuente (1994).

La direction de descente dans la méthode du gradient conjugué HS est déterminée comme dans (3.1.3), où le paramètre β_k est calculé comme $\beta_k = \frac{g_{k+1}^T y_k}{d_k^T y_k}$. Pour chaque problème test, dix expériences numériques ont été considérées avec un nombre de variables $n = 1000, 2000, ..., 10000$.

Ainsi, un certain nombre de 800 problèmes de tests d'optimisation sans contraintes ont été résolus. Les comparaisons des algorithmes sont données dans le cadre de la remarque 1.1.
La figure 3.1 montre le profil de performance de Dolan et Moré (2002) de l'algorithme de gradient conjugué HS avec recherche linéaire de Wolfe standard par rapport à HS avec recherche linéaire de Wolfe forte. Sur 800 problèmes considérés dans cette expérience numérique, le critère suivant qui consiste que la performance de ALG1 était meilleure que la performance de ALG2 si

$$\left|f_i^{ALG1} - f_i^{ALG2}\right| \leq 10^{-3} \tag{3.4.8}$$

tel que f_i^{ALG1}, f_i^{ALG2} les valeurs optimales trouvées par ALG1 et ALG2 pour le problème $i \geq 1$ respectivement. n'est valable que pour 760 problèmes. Le côté gauche de la figure 3.1 (petites valeurs de s) donne le pourcentage de problèmes de test, sur 760, pour lesquels un algorithme est plus efficace (plus rapide) ; le côté droit (grandes valeurs de s) donne le pourcentage de problèmes de test qui ont été résolus avec succès par chacun des algorithmes. Observez que le HS avec recherche linéaire de Wolfe standard surpasse le HS avec recherche linéaire de Wolfe forte dans la grande majorité des problèmes et que les différences sont substantielles.
En comparant HS avec recherche linéaire de Wolfe standard par rapport à HS avec recherche linéaire de Wolfe forte en fonction du nombre d'itérations (voir Tableau 3.1), nous remarquons que HS avec recherche linéaire de Wolfe standard était meilleur dans 351 problèmes (c'est-à-dire qu'il a atteint le nombre minimum de itérations dans 351 problèmes). HS avec une recherche linéaire de Wolfe forte était meilleur dans 220 problèmes et ils ont réalisé le même nombre d'itérations dans 99 problèmes. En ce qui concerne le temps CPU, nous voyons que HS avec une recherche linéaire de Wolfe standard était meilleur dans 352 problèmes et HS avec une recherche linéaire de Wolfe forte était meilleur dans 114 problèmes, etc.

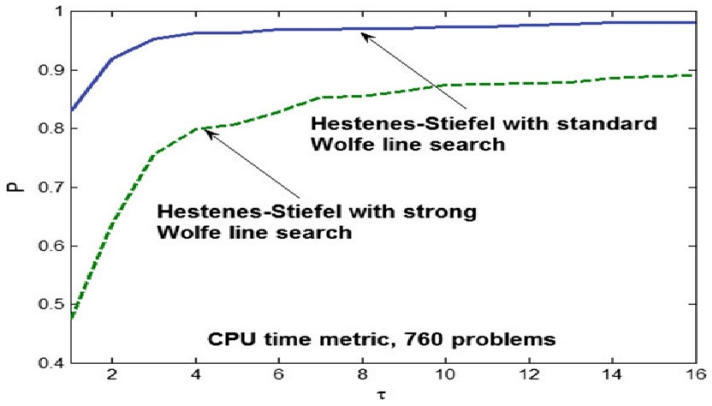

FIGURE 3.1 – Profils de performances du gradient conjugu Hestenes-Stiefel avec recherche linaire de Wolfe standard par rapport au gradient conjugu Hestenes-Stiefel avec recherche linaire de Wolfe forte, bas sur le temps CPU.

Tableau 3.1 Performances du gradient conjugué Hestenes-Stiefel avec recherche linéaire de Wolfe standard par rapport au gradient conjugué Hestenes-Stiefel avec recherche linéaire de Wolfe forte.

	Wolfe standard	**Wolfe forte**	=
Nombre de itéeation	351	220	99
Nombre de fg	467	201	2
CPU	352	114	204

Il est évident que le gradient conjugué avec une recherche linéaire de Wolfe standard est plus efficace que celui de Wolfe forte, selon trois critères principaux : le nombre d'itérations, le nombre d'évaluations de la fonction et du gradient, et le temps CPU. La méthode du gradient conjugué basé sur la recherche linéaire, ce qui la rend très efficace par rapport aux autres méthodes de minimisation comme celles de Newton ou quasi-Newton.

CHAPITRE 4

Évaluation des logiciels d'optimisation à l'aide de profils de performance

4.1 Historique et introduction

L'évaluation des logiciels d'optimisation est devenue un sujet de plus en plus visible dans ces derniers années. Les efforts de recherche, notamment ceux de Hans Mittlemann [Mit17], ont mis en lumière les lacunes dans de nombreux logiciels, conduisant ainsi à des améliorations substantielles. Cependant, Mittlemann n'est pas le seul à s'intéresser à ce domaine ; d'autres chercheurs ont également contribué à l'évaluation et à l'analyse des performances des algorithmes d'optimisation. Par exemple, des travaux récents tels que [BSV00], [BDF97], [BBM00a], [Num97], [CGT96], [Mit99] et [VS99] ont apporté de nouvelles perspectives. L'un des principaux défis de cette évaluation comparative réside dans l'interprétation des données collectées. Les chercheurs utilisent souvent des tableaux pour présenter les performances des différentes solutions sur divers problèmes, les dernieres travaux sont basés sur des mesures telles que le temps CPU ou le nombre d'itérations. Cependant, ces tableaux peuvent être difficiles à interpréter, surtout lorsqu'ils concernent de grands ensembles de données.

L'interprétation des résultats de ces tableaux n'est pas le meilleur choix, et avec la quantité considérable de données générées lors des évaluations comparatives utilisant de vastes ensembles

de tests, les chercheurs ont été confrontés au défi d'explorer différentes approches pour analyser ces données de manière efficace. Certains ont recours à des méthodes telles que la prise de moyennes ou le calcul des totaux cumulatifs des performances de chaque solution pour chaque métrique, comme le soulignent les études citées [BSV00], [Num97] et [CGT96].

On donne des profils de performance, c'est-à -dire des fonctions de distribution pour une métrique de performance, comme un outil pour évaluer et comparer les logiciels d'optimisation. Nous montrons que les profils de performance combinent les meilleures caractéristiques d'autres outils d'évaluation des performances. La plupart des chercheurs choisissent de ne signaler le nombre d'échecs que dans un tableau séparé.

Dans le même but, l'approche consistant à évaluer les solutions en fonction du ratio entre leur temps d'exécution et le temps optimal, voir [BDF97]. L'approche du rapport évite la plupart des difficultés que nous avons discutées, fournissant des informations sur l'amélioration en pourcentage et éliminant les effets négatifs. Le principal inconvénient de cette approche réside dans le choix arbitraire des limites définissant les extrémités.

Dans ce chapitre, on introduire les profils de performance comme un outil pour évaluer et comparer la performance des logiciels d'optimisation. Le profil de performance pour une solution est la fonction de distribution (cumulative) d'une métrique de performance, et en utilise le rapport du temps de calcul de solution par rapport au meilleur temps de tous les solutions comme métrique de performance.

Et en passe à l'analyse de l'ensemble de tests et des solutions utilisés dans les résultats de l'évaluation comparative ultérieure qui est nécessaire pour comprendre les limites du processus d'évaluation comparative.

Et après, en suivi la démonsration de l'utilisation des profils de performance avec les résultats [DMM04] obtenus avec la version 2.0 de l'ensemble de tests COPS [BBM00b]. On montre que les profils de performance éliminent l'influence d'un petit nombre de problèmes sur le processus d'évaluation comparative et la sensibilité des résultats associés au classement des solutions. Les profils de performance fournissent un moyen de visualiser la différence de performance attendue parmi de nombreux solutions, tout en évitant les choix arbitraires de paramètres et la nécessité d'écarter les échecs des solutions.

4.2 Évaluation des performances

Pour créer des résultats de source, on utilise un algorithme sur un ensemble de problèmes et enregistrons des informations comprenant le nombre d'itérations et le temps de calcul. Dans cette section, on utilise les profils de performances comme moyen d'évaluer et de comparer les performances d'un ensemble des algorithmes $\mathcal{A}$ sur un ensemble de problèmes $\mathcal{P}$. On suppose que nous avons n_A algorithmes et n_P problèmes. En utilisant le temps de calcul comme mesure de performance. Pour chaque problème P et algorithme A, ont défini :

$$t_{P,A} = \text{temps de calcul requis pour résoudre le problème } P \text{ par l'algorithme } A.$$

La notation des $t_{P,A}$ se fait à chaque fois en fonction d'évaluation de la performance étudiée. On utilise le ratio de performance comme outil de comparaison, en comparant les performances sur le problème P en utilisant l'algorithme A avec la meilleure performance de tout algorithme à ce problème, tel que :

$$r_{P,A} = \frac{t_{P,A}}{\min\left\{t_{P,A} : A \in \mathcal{A}\right\}}$$

On choisit un paramètre $r_M \geq r_{P,A}$ pour tout P, A, et $r_{P,A} = r_M$ si et seulement si, l'algorithme A ne résout pas le problème P. On démontre que le choix de r_M n'affecte pas l'évaluation des performances. Si l'on veut obtenir une évaluation globale des performances des solutions, on définit une probabilité pour l'algorithme $A \in \mathcal{A}$ telle que le rapport de performance $r_{P,A}$ soit compris dans le facteur $\tau \in \mathbb{R}$ du meilleur rapport possible. La fonction ρ_A est la fonction de répartition (cumulative) du ratio de performance et sa formule est la suivante :

$$\rho_A(\tau) = \frac{1}{n_P}\text{card}\left\{P \in \mathcal{P} : r_{P,A} \leq \tau\right\}$$

Si l'ensemble de problèmes $\mathcal{P}$ est suffisamment grand et représente des problèmes d'application, alors on choisit les algorithmes avec la plus grande probabilité. Le profil de performance à également utilisé pour une représentation graphique d'une mesure de performance par rapport l'ensemble $\mathcal{P}$. Tel que $\rho_A : \mathbb{R} \mapsto [0, 1]$ est une fonction constante non décroissante, continue à droite à chaque point d'arrêt. La valeur $\rho_A(1)$ est la probabilité que la solution soit meilleure que le reste des solutions. Ainsi, si l'on s'intéresse uniquement au nombre de succès, il suffit de comparer les valeurs de $\rho_A(1)$ pour tous les algorithmes $A \in \mathcal{A}$. La définition du profil de performances pour les valeurs supérieures nécessite une certaine prudence. On suppose que

$r_{P,A} \in [1, r_M]$ et que $r_{P,A} = r_M$ si le problème P est résolu par A. En conséquence, $\rho_A(r_M) = 1$, et

$$\rho_A^* = \lim_{\tau \to \bar{r}_M} \rho_A(\tau)$$

ρ_A^* est la probabilité que l'algorithme résoudre un problème. Ainsi, si on intéresse uniquement aux algorithmes ayant une forte probabilité de succès, nous devons alors comparer les valeurs de ρ_A^* pour toutes les solutions $A \in \mathcal{A}$ et choisir les solutions ayant la plus grande valeur. La valeur de ρ_A^* peut être facilement visible dans un profil de performance car ρ_A est éliminé pour de grandes valeurs de τ ; c'est-à-dire que $\rho_A(\tau) = \rho_A^*$ pour $\tau \in [r_A, r_M)$ pour certains $r_A < r_M$. Une propriété importante des profils de performance est la stabilité par rapport aux variations sur un nombre limité de problèmes. Cette stabilité repose sur l'observation que les valeurs de ρ_A et de $\hat{\rho}_A$, déterminées respectivement par les temps observés $t_{P,A}$ et $\hat{t}_{P,A}$, ne sont pas significativement affectées par les résultats de quelques problèmes spécifiques, où

$$\hat{t}_{P,A} = t_{P,A}, \ P \in \mathcal{P} \backslash \{q\},$$

pour un problème $q \in \mathcal{P}$, alors $\hat{r}_{P,A} = r_{P,A}$ pour $P \in \mathcal{P} \backslash \{q\}$. Puisque seul le rapport $\hat{r}_{P,A}$ change pour tout $A \in \mathcal{A}$,

$$|\rho_A(\tau) - \hat{\rho}_A(\tau)| \leq \frac{1}{n_P}, \ \tau \in \mathbb{R}$$

pour $A \in \mathcal{A}$. De plus, $\hat{\rho}_A(\tau) = \rho_A(\tau)$ pour $\tau < \min\{r_{P,A}, \hat{r}_{P,A}\}$ ou $\tau \geq \max\{r_{P,A}, \hat{r}_{P,A}\}$. Ainsi, si n_P est raisonnablement grand, le résultat sur un problème particulier q n'affecte pas grandement les profils de performances ρ_A. Non seulement les profils de performance sont relativement insensibles dans les résultats sur un petit nombre de problèmes, mais ils sont également largement insensibles aux petits changements dans les résultats sur de nombreux problèmes. Le théorème suivant démontre que de petits changements de $r_{P,A}$ à $\hat{r}_{P,A}$, s'entraînent une erreur proportionnellement petite entre ρ_A et $\hat{\rho}_A$.

Théorème 4.2.1. *Soient r_i et $\hat{r}_i$ pour $1 \leq i \leq n_P$ des ratios de performance pour une algorithme , et soient ρ et $\hat{\rho}$, respectivement, les profils de performance définis par ces ratios. Si*

$$|r_i - \hat{r}_i| \leq \varepsilon, \ 1 \leq i \leq n_P \tag{4.2.1}$$

pour un certain ε, alors

$$\int_1^\infty |\rho(t) - \hat{\rho}(t)| \, dt \leq \varepsilon$$

Preuve. étant donné que les profils de performance ne dépendent pas de l'ordre des données, nous pouvons supposer que $\{r_i\}$ est monotone croissante. Nous pouvons réorganiser la série $\{\hat{r}_i\}$ de manière à ce qu'elle soit également monotone croissante, et (4.2.1) est toujours vérifiée. Ces réorganisations garantissent que $\rho(t) = \dfrac{i}{n_P}$ pour $t \in [r_i, ri_{+1})$, avec un résultat similaire pour $\hat{\rho}$. Nous montrons maintenant que pour tout entier k avec $1 \leq k \leq n_P$,

$$\int_1^{s_k} |\rho(t) - \hat{\rho}(t)|\, dt \leq k \left(\frac{\varepsilon}{n_P}\right) \tag{4.2.2}$$

où $s_k = \max(r_k, \hat{r}_k)$, et

$$|r_i - \hat{r}_i|\,, \quad 1 \leq i \leq k, \quad \text{et } \hat{r}_i = r_i, \quad k < i < n_P \tag{4.2.3}$$

La démonstration est achevée lorsque $k = n_P$. Le cas $k = 1$ découle directement de la définition d'un profil de performance, alors supposons que (4.2.2) soit vraie pour des données de performance telles que (4.2.3) soit vraie. Nous prouvons maintenant que (4.2.1) est vraie pour $k+1$ en démontrant que

$$\int_{s_k}^{s_{k+1}} |\rho(t) - \hat{\rho}(t)|\, dt \leq \left(\frac{\varepsilon}{n_P}\right), \quad k < n_P \tag{4.2.4}$$

les formules (4.2.2) et (4.2.4) montrent que (4.2.2) est vraie pour $k+1$. Nous présentons la démonstration pour le cas où $\hat{r}_i \leq r_k$. Un argument similaire peut être fait pour $r_k \leq \hat{r}_k$. Si $\hat{r}_i \leq r_k$, alors $s_k = r_k$ et $\hat{r}_i \leq r_k \leq r_{k+1}$. L'argument dépend de la position de $\hat{r}_{k+1}$ et utilise à plusieurs reprises le fait que $\rho(t) = k/n_P$ pour $t \in [r_k, r_{k+1})$, avec un résultat similaire pour $\hat{\rho}$.

Si $r_{k+1} \leq \hat{r}_{k+1}$, alors $\rho(t) = \hat{\rho}(t)$ dans $[r_k, r_{k+1})$. Notons également que $|\rho(t) - \hat{\rho}(t)| = 1/np$ dans $[r_{k+1}, \hat{r}_{k+1})$. Ainsi, (4.2.4) est vérifiée avec $s_{k+1} = \hat{r}_{k+1}$.

Le cas où $\hat{r}_{k+1} \leq r_{k+1}$ utilise l'observation que $\hat{r}_i = r_i \geq r_{k+1}$ pour $i > k+1$. Si $r_k \leq \hat{r}_{k+1} \leq r_{k+1}$, alors $\rho(t) = \hat{\rho}(t)$ dans $[r_k, \hat{r}_{k+1})$, et $|\rho(t) - \hat{\rho}(t)| = 1/n_p$ dans $[\hat{r}_{k+1}, r_{k+1})$. Ainsi, (4.2.4) est vérifiée. D'autre part, si $\hat{r}_k \leq \hat{r}_{k+1} \leq r_k$, alors il suffit de noter que $|\rho(t) - \hat{\rho}(t)| = 1/n_P$ dans $[r_k, r_{k+1})$ pour conclure que (4.2.4) est vérifiée.

Nous avons montré que (4.2.2) est vraie pour tous les entiers k avec $1 \leq k \leq n_P$. En particulier, le cas $k = n_P$ donne notre résultat car $\rho(t) = \hat{\rho}(t)$ pour $t \in [s_{n_P}, \infty)$ □

4.3 Données de référence

Les données temporelles utilisées pour calculer les profils de performances dans les sections 4 et 5 sont générées avec l'ensemble de tests COPS, qui comprend dix-sept applications différentes, toutes des modèles dans le langage de modélisation AMPL [FGK03]. Le choix de l'ensemble de problèmes de test $\mathcal{P}$ est toujours source de désaccord car il n'existe pas une manière générale pour choisir les problèmes. Les problèmes COPS sont sélectionnés pour être intéressants et difficiles, mais ces critères sont subjectifs. Pour chacune des applications de l'ensemble COPS, nous utilisons quatre instances de l'application obtenues en faisant varier un paramètre dans l'application, par exemple le nombre de points de grille dans une discrétisation. Le tableau 4.1 donne les quartiles pour trois paramètres du problème : le nombre de variables n, le nombre de contraintes et le rapport $(n - n_e)/n$, où n_e est le nombre de contraintes d'égalité. Dans l'optimisation, $n - n_e$ sont les degrés de liberté du problème, car il s'agit d'une limite supérieure du nombre de variables libres à la solution.

Tableau 4.1 .Données problématiques pour l'ensemble de test COPS

	COPS complète					COPS sous-ensemble				
	min	q_1	q_2	q_3	max	min	q_1	q_2	q_3	max
Nbr. variables	48	400	1000	2402	5000	100	449	899	2000	4815
Nbr contraintes	0	150	498	1598	5048	51	400	800	1601	4797
Degrés de liberté	0	23	148	401	5000	0	5	99	201	1198
Deg. liberté (%)	0.0	1.0	33.2	100.0	100.0	0.0	0.4	19.8	33.1	49.9

Les données du Tableau 4.1 sont assez représentatives de la distribution de ces paramètres dans l'ensemble de tests et montrent qu'au moins trois quarts des problèmes ont un nombre de variables n dans l'intervalle $[400, 5000]$. Notre objective était d'éviter les problèmes où n était dans l'intervalle $[1, 50]$, car d'autres ensembles de problèmes de référence ont tendance à avoir une prépondérance de problèmes avec n dans le même intervalle. La principale différence entre l'ensemble complet COPS et le sous-ensemble COPS est que le sous-ensemble COPS est plus contraint avec $n_e \geq n/2$ pour tous les problèmes. Une autre caractéristique du sous-ensemble COPS est que les contraintes d'égalité résultent de l'approximation par différence ou collocation d'équations différentielles. Les solutions que nous évaluons ont des exigences différentes. MINOS et SNOPT utilisent uniquement des informations de premier ordre, tandis que LANCELOT

et LOQO ont besoin d'informations de second ordre. L'utilisation d'informations de second ordre peut réduire le nombre d'itérations, mais le coût par itération augmente généralement. De plus, l'obtention d'informations de second ordre est plus coûteuse. MINOS et SNOPT sont spécifiquement conçus pour des problèmes avec un nombre modeste de degrés de liberté, alors que ce n'est pas le cas de LANCELOT et LOQO.

4.4 Étude de cas : problèmes de contrôle optimal et d'estimation des paramètres

On examine maintenant les performances de LANCELOT [CGT13], MINOS [MS95], SNOPT [GMS97] et LOQO [DM02] sur le sous-ensemble des problèmes de contrôle optimal et d'estimation de paramètres dans l'ensemble de tests COPS [BBM00b]. Les figures 4.1 et 4.2 montrent les profils de performance dans différentes intervalles pour représenter divers domaines d'intérêt. Notre objectif est de montrer comment les profils de performances fournissent des informations objectives pour l'analyse d'un vaste ensemble de tests. La figure 4.1 présente les profils de performance des quatre solutions pour des petites valeurs de τ. En montrant les ratios des temps de résolution, et on élimine l'effet que pourraient avoir les différences de temps d'exécution long de chaque solution. Cela permet une évaluation plus équilibrée et équitable des performances relatives de chaque méthode.

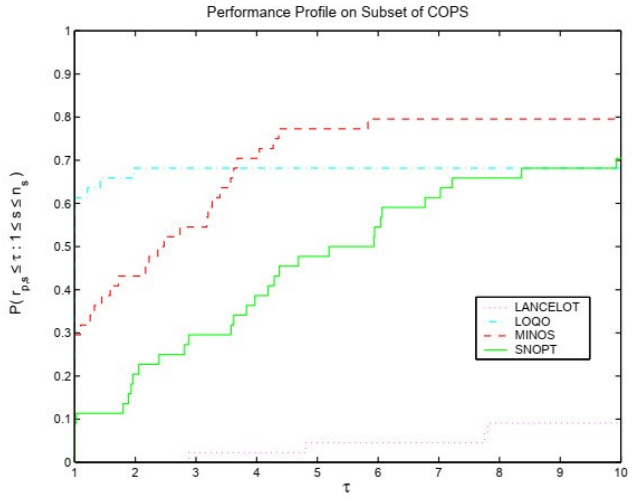

FIGURE 4.1 – Profil de performances sur [0,10]

Nous ne pensons pas qu'il soit nécessaire d'éliminer les problèmes liés aux tests. Pour cela, l'algorithme est d'une importance maximale pour résoudre des problèmes où une ou plusieurs algorithmes échouent. En particulier, $1 - \rho_A(\tau)$ est la probabilité que l'algorithme A ne résoudre pas le problème P dans un temps inferieure où égale τ.

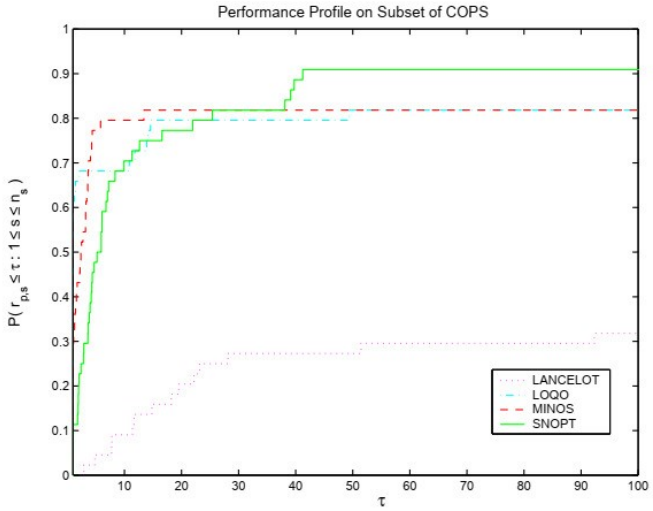

FIGURE 4.2 – Profil de performances sur [0,100].

D'après cette figure, il est clair que LOQO a le plus de succès (à la plus grande probabilité d'être la solution optimale) et que la probabilité que LOQO soit le gagnant sur un problème donné est d'environ 0,61. Si nous choisissons d'être dans un facteur 4 de la meilleure algorithme comme champ d'intérêt (voir figure 4.1), alors soit LOQO, soit MINOS suffiraient ; mais le profil de performance montre que la probabilité que ces deux algorithme puissent résoudre un problème dans un facteur 4 de la meilleure algorithme n'est que d'environ 70%. SNOPT a un nombre de succès inférieur à LOQO ou MINOS, mais ses performances deviennent beaucoup plus performantes si nous étendons notre τ d'intérêt à 7.

La figure 4.2 montre les profils de performances dans l'intervalle $[1, 100]$ pour les quatre problèmes. Si nous cherchons la solution qui résout 75% des problèmes avec la plus grande efficacité, alors MINOS se distingue clairement. Cependant, si nous privilégions des critères de réussite plus rigoureux, SNOPT devient plus pertinent, car il parvient à résoudre plus de 90% de ce sous-ensemble COPS. Cela est clairement démontré par la hauteur de son profil de performance pour $\tau > 40$. Ce graphique affiche la possibilité d'écarts importants dans les ratios de performance pour un pourcentage substantiel de problèmes. Un autre point intéressant est que LOQO,

MINOS et SNOPT ont chacun la meilleure probabilité $\rho_a(\tau)$ pour τ dans un certain intervalle, avec des performances similaires dans l'intervalle $[15, 40]$.

Un constat qui ressort de ces figures est le manque de cohérence des valeurs quartiles des ratios temporels. Les trois meilleures algorithmes partagent un ratio minimum de 1, et LOQO et MINOS partagent également des valeurs de premier quartile de 1. En d'autres termes, ces deux algorithmes sont les meilleures solutions sur au moins 25% des problèmes. LOQO surpasse la valeur médiane de MINOS avec 1 contre$2, 4$, mais MINOS revient avec un ratio de troisième quartile de $4, 3$ contre $13, 9$ pour LOQO, SNOPT mélangeant encore les résultats en battant également LOQO avec $12, 6$. En examinant les graphiques 4.1 et 4.2, on constate que la progression entre quartiles ne se déroule pas nécessairement de manière linéaire ; Par conséquent, nous perdons réellement des informations si nous ne fournissons pas les données complètes. De plus, le rapport maximum serait r_M pour nos tests, et aucune valeur alternative évidente n'existe. Cependant, au lieu de fournir uniquement des valeurs quartiles, le profil de performances fournit beaucoup plus d'informations sur les forces et les faiblesses d'un algorithme. Nous avons vu qu'au moins deux graphiques peuvent être nécessaires pour examiner les performances des algorithmes. Même en étendant τ à 100, nous ne parvenons pas à capturer les données complètes de performances pour LANCELOT et LOQO. Comme dernière option, nous affichons une échelle logarithmique des profils de performances. De cette façon, nous pouvons montrer toutes les activités qui ont lieu avec $\tau < r_M$ et saisir toutes les implications de nos données de test concernant la probabilité des algorithmes de résoudre avec succès un problème. Puisque nous nous intéressons également au comportement de τ proche de l'unité, nous utilisons une base de 2 pour l'échelle. En d'autres termes, nous traçons

$$\tau \to \frac{1}{n_P} \text{ size } \{P \in \mathcal{P} : \log_2(r_{P,A}) \leq \tau\}$$

Dans la figure 4.3. Ce graphique révèle toutes les caractéristiques des deux graphiques précédents et capture ainsi les performances de toutes les algorithmes. L'inconvénient est que l'interprétation du graphique n'est pas aussi intuitive, puisque nous utilisons une échelle logarithmique. Les figures 4.1 et 4.2 sont mappées à une nouvelle échelle pour refléter toutes les données, nécessitant au moins l'intervalle $[0, \log_2(1043)]$ de la figure 4.3 pour inclure le plus grand $r_{P,A} < r_M$. Nous étendons légèrement l'intervalle pour montrer la stagnation de toutes les algorithmes. La nouvelle figure contient toutes les informations des deux autres figures et montre en outre que chacun

des algorithmes échoue sur au moins 8% des problèmes. Il ne s'agit pas d'une performance déraisonnable pour l'ensemble de tests COPS car ces problèmes ont généralement été jugés difficiles.

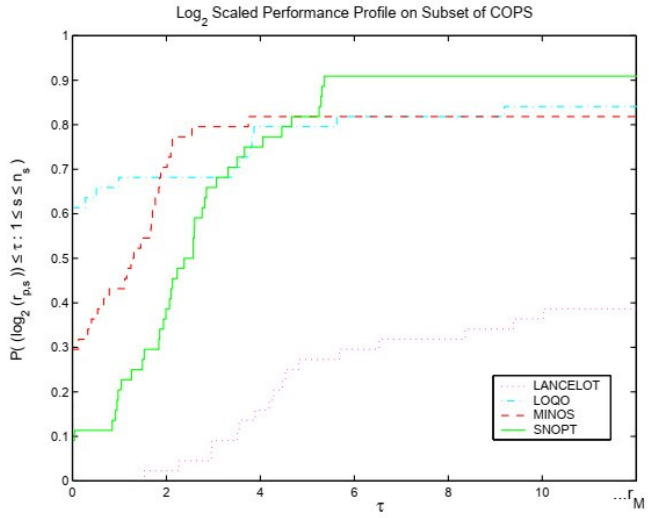

FIGURE 4.3 – Profil de performances sur une échelle log_2 .

Remarque 4.4.1. *dans le cas de COPS complet on peut introduire la fonction log dans la base* 2 *pour faire une comparaison efficace. (Voir la figure 4.4)*

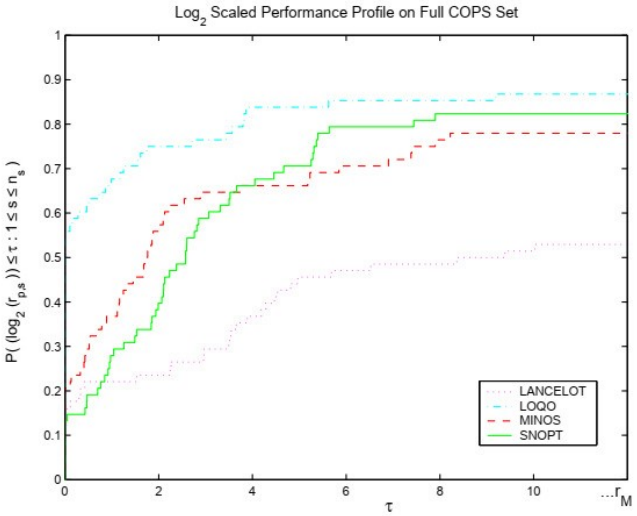

FIGURE 4.4 – Profil de performances pour l'ensemble COPS complet.

La figure 4.4 donne une indication claire des performances relatives de chaque algorithme, cette figure montre que les profils de performance éliminent l'influence indue d'un petit nombre de

problèmes sur le processus d'analyse comparative et la sensibilité des résultats associés au classement des algorithmes. De plus, les profils de performances fournissent une estimation de la différence de performances attendue entre les algorithmes.

L'aspect le plus significatif de la figure 4.4, par rapport à la figure 4.3, est que sur cet ensemble de tests, l'algorithme LOQO domine tous les autres algorithmes ; le profil de performance de LOQO se situe au-dessus de tous les autres pour tous les ratios de performance. L'interprétation des résultats de la figure 4.4 est importante. En particulier, ces résultats n'impliquent pas que LOQO est plus rapide sur tous les problèmes. Ils indiquent seulement que, pour tout $\tau \geq 1$, LOQO résout plus de problèmes dans un facteur τ de tout autre temps de résolution. De plus, en examinant $\rho_a(1)$ et $\rho_a(r_M)$, nous pouvons également dire que LOQO est l'algorithme la plus rapide sur environ 58% des problèmes, et que LOQO résout le plus de problèmes (environ 87%) de manière optimale.

4.5 Conclusion

L'analyse du profil de performance représente une méthode essentielle pour évaluer et comparer les performances des algorithmes d'optimisation. Elle permet de mettre en lumière les forces et les faiblesses de chaque méthode en termes de vitesse de convergence, précision des résultats et capacité à éviter les optima locaux. Cette approche intégrative permet de tirer parti des avantages de différents outils d'évaluation des performances, offrant ainsi une vision holistique et éclairée du comportement des algorithmes dans divers contextes d'optimisation.

Chapitre 5

Article de thèse

Publication découlant de cette Thèse

Bachir Barrouk, Mohammed Belloufi, Rachid Benzine, Tahar Bechouat

An improved PRP conjugate gradient method for optimization computation

Int. J. Nonlinear Anal. Appl. Volume 15,Issue 11, 139–147

ISSN: 2008-6822 (electronic)

Received: September 2022

Accepted: October 2023

Available Online: from 16 December 2023

Article URL : http://dx.doi.org/10.22075/ijnaa.2023.28524.3918

Int. J. Nonlinear Anal. Appl. 15 (2024) 11, 139–147
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2023.28524.3918

An improved PRP conjugate gradient method for optimization computation

Bachir Barrouk[a,b,*], Mohammed Belloufi[b], Rachid Benzine[c], Taher Bechouat[b]

[a] *Badji Mokhtar University, Annaba, 23000, Algeria*

[b] *Laboratory Informatics and Mathematics (LiM), Mohamed Cherif Messaadia University, Souk Ahras, 41000, Algeria*

[c] *Superior School of Industrial Technologies, Annaba, 23000, Algeria*

(Communicated by Javad Damirchi)

Abstract

The conjugate gradient method plays a very important role in several fields, to solve problems of large sizes. To improve the efficiency of this method, a lot of works have been done; in this paper we propose a new modification of PRP method to solve a large scale unconstrained optimization problems in relation with strong Wolf Powell Line Search property, when the latter was used under some conditions, a global convergence result was proved. In comparison with other known methods the efficiency of this method proved that it is better in the number of iterations and in time on 90 proposed problems by use of Matlab.

Keywords: Unconstrained optimization, Conjugate gradient method, strong Wolfe line search, Numerical comparisons
2020 MSC: 49M07; 49M10, 90C06

1 Introduction

Nonlinear Conjugate Gradient Methods,is are very convenient to large-scale problems because of their iterations easiness and their very low memory requirements; that is they are designed to solve the following unconstrained optimization problem:

$$\min f(x), x \in \mathbb{R}^n \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth, nonlinear function, and its gradient is denoted by $g(x) = \nabla f(x)$ The iterative formula of the conjugate gradient methods is given by

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k = 0, 1, 2..., \tag{1.2}$$

where x_k is the current iteration point and α_k is the step length, which is computed by carrying out a line search, and d_k is the search direction defined by

$$d_k = \begin{cases} -g_k & \text{if } k = 0 \\ -g_k + \beta_k d_{k-1} & \text{if } k \geq 1 \end{cases} \tag{1.3}$$

*Corresponding author
Email addresses: bachir.barrouk@univ-soukahras.dz (Bachir Barrouk), m.belloufi@univ-Soukahras.dz (Mohammed Belloufi), rabenzine@yahoo.fr (Rachid Benzine), m.belloufi@univ-Soukahras.dz (Taher Bechouat)

Received: September 2022 *Accepted:* October 2023

where $\beta_k$is a scalar, and $g_k = g(x_k)$. Various conjugate gradient methods have been proposed, and they mainly differ in the choice of the parameter β_k..Some well-known formulas for β_k are given below:

$$\beta_k^{HS} = \frac{g_k^T(g_k - g_{k-1})}{(g_k - g_{k-1})^T d_{k-1}} \tag{1.4}$$

$$\beta_k^{FR} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{1.5}$$

$$\beta_k^{PRP} = \frac{g_k^T(g_k - g_{k-1})}{g_{k-1}^T g_{k-1}} \tag{1.6}$$

$$\beta_k^{CD} = -\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}} \tag{1.7}$$

$$\beta_k^{LS} = \frac{g_k^T(g_k - g_{k-1})}{g_{k-1}^T d_{k-1}} \tag{1.8}$$

$$\beta_k^{DY} = \frac{g_k^T g_k}{(g_k - g_{k-1})^T d_{k-1}} \tag{1.9}$$

and some other formulas for β_k based on the β_k^{PRP} are as the following:

$$\beta_k^{WYL} = \frac{g_k^T(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1})}{\|g_{k-1}\|^2} \tag{1.10}$$

$$\beta_k^{RMIL} = \frac{g_k^T(g_k - g_{k-1})}{\|d_{k-1}\|^2} \tag{1.11}$$

$$\beta_k^{LAMR} = \frac{g_k^T(\frac{\|d_{k-1}\|}{\|d_{k-1} - g_k\|} g_k - g_{k-1})}{\frac{\|d_{k-1}\|}{\|d_{k-1} - g_k\|} \|d_{k-1}\|^2} \tag{1.12}$$

The corresponding method is respectively called, (Hestenes-Stiefel [4]), (Fletcher_Revees [3]), (Polak_Ribiére_Polyak ([3]-[9]), (Conjugate Descent [8]), (Liu-Storey [5]), and (Dai_Yuan [17]) conjugate gradient method. The convergence behaviour of the above formula with some line search conditions has been studied by many authors for many years ([8]-[20]).

Mamat , Rivaie and Zabidin [22] proposed a new modification of PRP method called HRM method.

$$\beta_k^{HRM} = \frac{g_k^T(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1})}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2}, 0 < u < 1 \tag{1.13}$$

Mohamed Hamoda, Mohd Rivaie, Mustafa Mamat, Zabidin Salleh. 2015 [7] proposed a new modification of PRP method called HRM method.

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

There are many conjugate gradient methods; a great contribution in this sphere is given by Hagar and Zhang. Different conjugate methods correspond to different values of the scalar parameter β_k. Hybrid conjugate gradient methods as combine different conjugate gradient methods to improve the behavior of these methods, which has been widely studied by many authors, see [2, 22].

In the already-existing convergence analysis and implementations of the conjugate gradient method, the weak Wolfe–Powell (WWP) line search conditions are as follows:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta\alpha_k g_k^T d_k \tag{1.14}$$

$$g_{k-1}^T d_k \geq \sigma g_k^T d_k \tag{1.15}$$

where $0 < \delta < \sigma < 1$ and d_k is a descent direction. The strong Wolfe–Powell conditions consist of (1.14) and,

$$\left|g(x_k + \alpha_k d_k)^T d_k\right| \leq \sigma \left|g_k^T d_k\right| \tag{1.16}$$

Furthermore, the sufficient descent property, namely,

$$g_k^T d_k \leq -c \left\|g_k\right\|^2, \tag{1.17}$$

where c is a positive constant and crucial to ensure the global convergence of the nonlinear conjugate gradient method with the inexact line search techniques ([7] - [16]).

In this paper and depending on the above ideas we propose a new method called BBBB by the modification of PRP conjugate gradient method.

2 New conjugate gradient method

In the last decade, a lot of efforts have been done and devoted to develop new modifications of conjugate gradient methods which don't only pessess strong convergence properties but they are also superior to the classical ones in performance. Such methods are found in ([1] to [13]).

In the present time, Wei et al [18] gave a variant of the PRP method which is called the WYL method. Zhang studied and improved based on WYL method a new conjugate gradient method, called NPRP based on Strong Wolfe line search condition. Moreover, Zhang et al. proposed another modified method called MPRP method, where Dai and Wen [19] proposed a modification of NPRP method called DPRP method,M.Hamoda , M. Mamat , M.Rivaie and S.Zabidin [22] proposed a new modification of PRP method called HRM method. In order to take in the advantages of PRP methods and establish a more efficient and robust algorithm, and inspired by the work of Mohamed Hamoda, Mustafa Mamat, Mohd Rivaie, Zabidin Salleh [3] and Mohamed Hamoda, Mohd Rivaie, Mustafa Mamat, Zabidin Salleh [14], we propose a new hybrid CG method based on PRP methods for solving unconstrained optimization problems with suitable conditions. The parameter β_k in the proposed method is computed bay the formula below:

$$\beta_k^{BBBB} = \frac{g_k^T(g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1})}{u\left\|g_{k-1}\right\|^2 + (1-u)\left\|d_{k-1}\right\|^2 + \left|g_{k-1}^T d_k\right|}, 0 < u < 1 \tag{2.1}$$

where,$BBBB$ denotes Barrouk, Benzine, Belloufi and Bechouat. According to the results obtained by [13], the value of the parameter u can be set to $0 < u < 1$, but in this paper, we will test our new method with an arbitrary value $u = 0.04$.

The algorithm of new CG method used in this paper is given as follow:

Step 1: Given, $x_0 \in \mathbb{R}^n$ $\varepsilon > 0$. Set $d_0 = -g_0$ if $\|g_0\| \leq \varepsilon$ then stop.

Step 2: Compute α_k by (SWP) line search.

Step 3: Let $x_{k+1} = x_k + \alpha_k d_k$, $g_{k+1} = g(x_{k+1})$ if $\|g_{k+1}\| < \varepsilon$ then stop.

Step 4: Compute β_k by formula (2.1) and generate d_{k+1} by (1.3).

Step 5: Set $k = k + 1$ go to Step 2.

The following assumptions are often used in previous studies of the conjugate gradient methods:

Assumption A

$f(x)$ is bounded from below on the level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$, where x_0 is the starting point.

Assumption B

In some neighborhoods of Ω, the objective function is continuously differentiable, and its gradient is Lipschitz continuous, that is, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\| \quad \forall x, y \in N. \tag{2.2}$$

In 1992, Gilbert and Nocedal introduced the property (*) which plays an important role in the studies of CG methods. This property means that the next research direction approaches the steepest direction automatically when a small step-size is generated, and the step-sizes are not produced successively [21].

Property (*)

Consider a conjugate gradient method of the form (1.2) and (1.3). Suppose that, for all $k \geq 0$,

$$0 < \gamma \leq \|g_k\| \leq \overline{\gamma}$$

where γ and $\overline{\gamma}$ are two positive constants. We say that the method has property (*), if there exist constants $b > 0$,$\lambda > 0$, such that for all k, $|\beta_k| \leq b, |S_k| \leq \lambda$ implies $|\beta_k| \leq \frac{1}{2b}$,where $S_k = \alpha_k d_k$..

The following lemma shows that the new method β_k^{BBBB} has the property (*).

Lemma 2.1. Consider the method of form (1.2) and (1.3), Suppose that Assumptions A and B hold, then, the method β_k^{BBBB} has property (*).

Proof . Set $b = \frac{50\overline{\gamma}(\overline{\gamma}+\frac{\overline{\gamma}^2}{\gamma})}{2\gamma^3} > 1$, $\lambda = \frac{\gamma^2}{100Lb\overline{\gamma}}$. By (2.1) and (2.2), We have

$$\left|\beta_k^{BBBB}\right| \leq \frac{\left|g_k^T(g_k - \frac{\|g_k\|}{\|g_{k-1}\|}g_{k-1})\right|}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2 + \left|g_{k-1}^T d_k\right|} \leq \frac{\left|g_k^T(g_k + \frac{\overline{\gamma}}{\gamma}g_{k-1})\right|}{0.04\|g_{k-1}\|^2} \leq \frac{50\overline{\gamma}(\overline{\gamma}+\frac{\overline{\gamma}^2}{\gamma})}{2\gamma^3} = b$$

From assumption B, holds. If $|S_k| \leq \lambda$ then,

$$\begin{aligned}
\left|\beta_k^{BBBB}\right| &\leq \frac{\left(\|g_k - g_{k-1}\| + \left\|g_{k-1} - \frac{\|g_k\|}{\|g_{k-1}\|}g_{k-1}\right\|\right)\|g_k\|}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2 + \left|g_{k-1}^T d_k\right|} \\
&\leq \frac{\left(\|g_k - g_{k-1}\| + \left\|g_{k-1} - \frac{\|g_k\|}{\|g_{k-1}\|}g_{k-1}\right\|\right)\|g_k\|}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2} \\
&\leq \frac{(L\lambda + \|g_{k-1}\| - \|g_k\|)\|g_k\|}{u\|g_{k-1}\|^2} \\
&\leq \frac{2L\lambda\|g_k\|}{0.04\|g_{k-1}\|^2} \leq \frac{100L\lambda\overline{\gamma}}{2\gamma^2} = \frac{1}{2b}.
\end{aligned}$$

The proof is finished. □

3 The global convergence properties

The following theorem shows that the formula BBB with SWP line search process the sufficient descent condition.

Theorem 3.1. Suppose that the sequences $\{g_k\}$ and $\{d_k\}$ are generated by the method of the form (1.2), (1.3) and (2.1), and the step length α_k is determined by the (SWP)) line search (2.1) and (1.14), if $g_k \neq 0$, then the sequence $\{d_k\}$ possesses the sufficient descent condition (1.16).

Proof . By the formulae (2.1), we have the following:

$$\begin{aligned}
\beta_k^{BBBB} &= \frac{g_k^T(g_k - \frac{\|g_k\|}{\|g_{k-1}\|}g_{k-1})}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2 + \left|g_{k-1}^T d_k\right|} \\
&\geq \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|}\left|g_k^T g_{k-1}\right|}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2 + \left|g_{k-1}^T d_k\right|} \\
&\geq \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|}\|g_k\|\|g_{k-1}\|}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2 + \left|g_{k-1}^T d_k\right|} = 0
\end{aligned}$$

thus we get, $\beta_k^{BBBB} \geq 0$. Also,

$$\begin{aligned}\beta_k^{BBBB} &= \frac{g_k^T(g_k - \frac{\|g_k\|}{\|g_{k-1}\|}g_{k-1})}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2 + |g_{k-1}^T d_k|} \\ &\leq \frac{\|g_k\|^2 + \frac{\|g_k\|}{\|g_{k-1}\|}|g_k^T g_{k-1}|}{u\|g_{k-1}\|^2 + (1-u)\|d_{k-1}\|^2 + |g_{k-1}^T d_k|} \\ &\leq \frac{2\|g_k\|^2}{u\|g_{k-1}\|^2} = \frac{2\|g_k\|^2}{0.04\|g_{k-1}\|^2} = \frac{50\|g_k\|^2}{\|g_{k-1}\|^2}.\end{aligned}$$

Hence, we obtain

$$0 \leq \beta_k^{BBBB} \leq \frac{50\|g_k\|^2}{\|g_{k-1}\|^2} \tag{3.1}$$

using (1.16) and (3.1), we get

$$|\beta_{k+1}^{BBBB} g_{k+1}^T d_k| \leq \frac{50\|g_{k+1}\|^2}{\|g_k\|^2}\sigma|g_k^T d_k|. \tag{3.2}$$

By (1.3), we have $d_{k+1} = -g_{k+1} + \beta_{k+1}d_k$

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} = -1 + \beta_{k+1}\frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2}. \tag{3.3}$$

We prove the descent property of $\{d_k\}$ by induction. Since $g_0^T d_0 = -\|g_0\|^2 < 0$, if $g_0 \neq 0$, now suppose that$d_i, i = 1, 2, ..., k$, are all descent direction, that is $g_i^T d_i < 0$ By (3.2), we get

$$|\beta_{k+1}^{BBBB} g_{k+1}^T d_k| \leq \frac{50\|g_{k+1}\|^2}{\|g_k\|^2}\sigma(-g_k^T d_k). \tag{3.4}$$

That is,

$$\frac{\|g_{k+1}\|^2}{\|g_k\|^2}50\sigma g_k^T d_k \leq \beta_{k+1}^{BBBB} g_{k+1}^T d_k \leq -\frac{\|g_{k+1}\|^2}{\|g_k\|^2}50\sigma g_k^T d_k. \tag{3.5}$$

(3.3) and (3.5) deduce,

$$-1 + \frac{50\sigma g_k^T d_k}{\|g_k\|^2} \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -1 - \frac{50\sigma g_k^T d_k}{\|g_k\|^2}.$$

By repeating this process and the fact $g_0^T d_0 = -\|g_0\|^2$, we have,

$$-\sum_{i=0}^{k}(50\sigma)^i \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k-1}\|^2} \leq -2 + \sum_{i=0}^{k}(50\sigma)^i. \tag{3.6}$$

Since $\sum_{i=0}^{k}(50\sigma)^i < \sum_{i=0}^{\infty}(50\sigma)^i$., (3.6) can be written as

$$\frac{-1}{1-50\sigma} \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k-1}\|^2} \leq -2 + \frac{1}{1-50\sigma}. \tag{3.7}$$

By making the restriction $\sigma \in]0, 0.01]$, we have $g_{k+1}^T d_{k+1} < 0$. So by induction $g_k^T d_k < 0$ holds for all $k \geq 0$. Denote $c = 2 - \frac{1}{1-50\sigma}$ the $0 < c < 1$, and (3.7) turns out to be

$$(c-2)\|g_k\|^2 \leq g_k^T d_k \leq -c\|g_k\|^2. \tag{3.8}$$

This implies that (1.17) holds, the proof is complete □

The following condition known as Zoutendijk condition was used to prove the global convergence of nonlinear CG methods ([6],[12])

Lemma 3.2. Suppose that Assumption A and B hold. Consider a CG method of the form (1.2) and (1.3), where d_k satisfies $g_k^T d_k \leq 0$, for all k, and α_k is obtained by (SWP) line search (1.14) and (1.16), then,

$$\sum_{k=0}^{\infty} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < \infty \tag{3.9}$$

The proof had been given in [11, 15, 23], Gilbert and Nocedal introduced the following important theorem:

Theorem 3.3. consider any CG method of the form (1.2) and (1.3), that satisfies the following conditions:

1) $\beta_k \geq 0$

2) The search direction satisfy the sufficient descent

3) The.Zoutendijk condition holds

4) Property(*) holds.

If the Lipschitz and boundedness Assumption hold, then the iterates are globally convergent.

From (1.17),(2.2),(3.7) and Lamma 1, we found that the $BBBB$ method with the parameter $0 < \delta < \sigma < \frac{1}{1000}$ satisfies all four conditions in theorem 1 under the strong Wolfe-Powell line search, so the method is globally convergent.

4 Numerical Experiments

In our numerical experiments we chose sixteen different functions which are a miscture of both small scale and large scale optimization problems. When these functions were tested and a range of the variables lie from 2 to 50000, we arrived to test 90 problems by using Strong Wolfe-Powell line search. The algorithm was implemented by using Matlab R2013b in the same PC with Intel (R) core (TM) i5-3210M, CPU (2.50 GHz), 4 GB RAM, and Windows 7 operating system. To assess the performance of BBBB method, we tested in against some of the classical and modified methods which are PRP, LS, RIM, RMIL and HRM method using the some problems, and assumed that the best method should require fewer iterations and less CPU time.

In order to ossers the efficacy of the new proposed method, we copared it (BBBB method) with PRP method and the other modified methods based on PRP method (LS, RAMI, RMIL, HRM) by using the same problems; and by calculating the number of iterations and CPU time of each problem, the best method is that which requires fewer iterations and less CPU time. All of these algorithms terminated when $\|g_k\|^2 < 10^{-6}$. The step size α_k satisfies the strong Wolfe-Powell condition, with $\delta = 10^{-4}$, and $\sigma = 0.001$. For the HRM method, we chose $\mu = 0.04$, the table 1 below shows the list test functions, the dimensions and the used initial points. In some cases, the calculation stopped because of the line search failure to find the positive step size, and thus it was considered as a failure and for us, we consider the search so when the number of iterations passed 2000 or CPU execution time passed 1000 seconds.

The performance results are shown in Figure 1 and 2 respectively, using a performance profile introduced by Dolan and More ([10]) here we compared the numerical results relatively with CPU time and number of iterations.

By using a strong Wolfe-Powell line search, the performance profile of all methods measured by the number of iterations required is shown in Figure 1, and in Figure 2 when it is based on the CPU time. The profile plots shapes in both Figures 1 and 2 are almost similar. In the left side, by an inspection in the left side of Figures 1 and 2, we observe a clear lowest curve that represents RMIL method, so that method possesses the least performance. The top left side curve for BBBB method indicates that it is the best performer. The curves for methods HRM, PRP, RAMI and LS, fall between the two extreme curves.

The result shown in Figures 1 and 2 indicate evidently that the RMIL method achieved a success rate of only 0.908, while the RAMI method had 0.967, and HRM method scored 0.984, and LS method scored 0.977. Furthermore, the PRP method achieved 0.978 BBBB achieved 0.995 success rate. This result indicates that our method (BBBB) is the best among the other 5 methods. Hence, our new method solved all the test problems successfully, and it is competitive with PRP method and the other methods based on it for unconstrained optimization.

5 Conclusion

In this paper, we proposed a new conjugate gradient method for unconstrained optimization. The results showed that it could satisfy the sufficient descent condition and converge globally if the strong Wolfe-Powell line search was used. Numerical results showed that the BBBB method is efficient for the addressed problems.

N°	Function	Dimension	Initial points
1	Booth	2	$1, 2, 3, 4$
2	Branin	2	$1, 2, 3, 4, 5$
3	Diagonal 1	$20; 30; 40; 50; 70;$ $80; 100; 150$	$1, 2, 3, 4, 5$
4	Diagonal 2	$200; 250; 300; 350;$ $400; 450; 500; 560$	$1, 2, 3, 4, 5$
5	Diagonal 4	$1000, 2000, 5000, 10000,$ $20000, 30000, 40000, 50000$	$15, 20, 25, 30$
6	Hager	$50; 150; 200; 300;$ $400; 500; 1000$	$3, 5, 6, 7$
7	Penalty	$35; 40; 45; 50; 60$	$2, 10, 20, 30$
8	Quadratic	$100; 150; 200; 250;$ $300; 350; 400; 500$	$3, 10, 20, 40$
9	Power	$8; 10; 15; 20; 25,$ $30; 35; 40$	$3, 5, 7, 10$
10	Qing	$50; 100; 300; 500;$ $800; 1000, 1200, 1500$	$15, 20, 30, 40$
11	Quadratic QF1	$100; 200; 300; 500$	$5, 10, 15, 20$
12	Raydan 1	$100; 200; 250; 300$	$-4, -3, -2, -1$
13	Raydan 2	$500; 1000; 2000; 3000; 3500;$ $4000; 4500; 5000$	$-6, -4, 4, 6$
14	Sphere	$5000; 10000; 20000; 30000;$ $35000; 40000; 45000; 50000$	$10, 20, 30, 40$
15	Sumsquares	$50; 80; 100; 200$	$2, 4, 8, 12$

Table 1: List of the problem functions

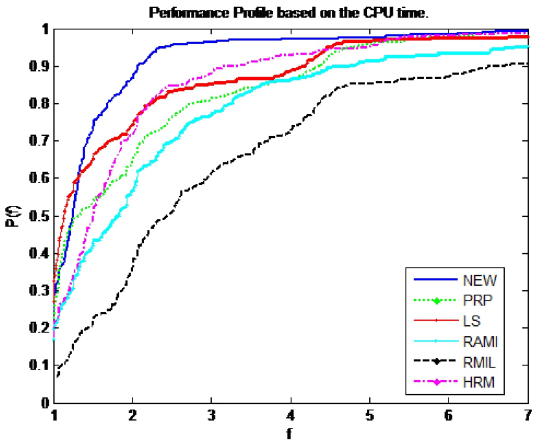

Figure 1: Performance Profile based on the CPU time.

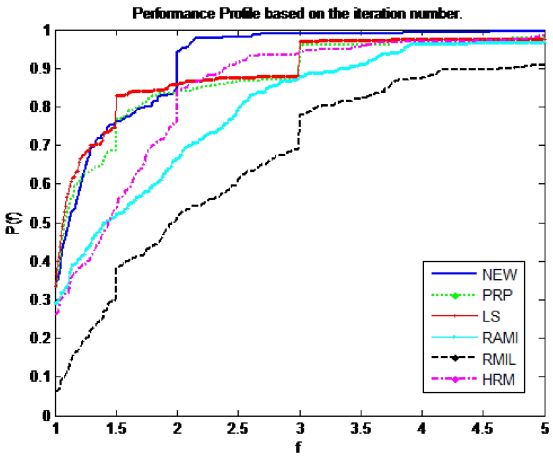

Figure 2: Performance Profile based on the iteration number.

References

[1] M. Al-Baali, *Descent property and global convergence of the Fletcher—Reeves method with inexact line search*, IMA J. Numer. Anal. **5** (1985), no. 1, 121–124.

[2] N. Andrei, *An unconstrained optimization test functions collection*, Adv. Model. Optim **10** (2008), no. 1, 147–161.

[3] E. Blum, *From optimization and variational inequalities to equilibrium problems*, Math. Student **63** (1994), 123–145.

[4] F.H. Clarke, Y.S. Ledyaev, R.J. Stern, and P.R. Wolenski, *Nonsmooth Analysis and Control Theory*, vol. 178, Springer Science & Business Media, 2008.

[5] R. Fletcher and C.M. Reeves, *Function minimization by conjugate gradients*, Comput. J **7** (1964), no. 2, 149–154.

[6] J.C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim. **2** (1992), no. 1, 21–42.

[7] M. Hamoda, M. Mamat, M. Rivaie, and Z. Salleh, *A conjugate gradient method with strong Wolfe-Powell line search for unconstrained optimization*, Appl. Math. Sci. **10** (2016), 721–734.

[8] M.R Hestenes and E. Stiefel, *Methods of conjugate gradients for solving*, J. Res. Nat. Bureau Stand. **49** (1952), no. 6, 409.

[9] D.E. Knuth, *The TEXbook*, Addison Wesley Professional, Massachusetts, 1984.

[10] G. Li, C. Tang, and Z. Wei, *New conjugacy condition and related new conjugate gradient methods for unconstrained optimization*, J. Comput. Appl. Math. **202** (2007), no. 2, 523–539.

[11] Y. Liu and C. Storey, *Efficient generalized conjugate gradient algorithms, part 1: theory*, J. Optim. Theory Appl. **69** (1991), no. 1, 129–137.

[12] Mu. Mamat, M. Rivaie, I. Mohd, and M. Fauzi, *A new conjugate gradient coefficient for unconstrained optimization*, Int. J. Contemp. Math. Sci. **5** (2010), no. 29, 1429–1437.

[13] I.S. Mohammed, M. Mamat, A. Abashar, M. Rivaie, and Z. Salleh, *A modified nonlinear conjugate gradient method for unconstrained optimization*, Appl. Math. Sci. **9** (2015), no. 54, 2671–2682.

[14] T. Nguyen Xuan and T. Phan Nhat, *On the existence of equilibrium points of vector functions*, Numer. Funct. Anal. Optim. **19** (1998), no. 1-2, 141–156.

[15] O. Omer, M. Mamat, and M. Rivaie, *The global convergence properties of a family of conjugate gradient method*

under the strong Wolfe line search, Abstr. Appl. Anal., vol. 2015, 2015.

[16] M.J.D. Powell, *Restart procedures for the conjugate gradient method*, Math. Program. **12** (1977), no. 1, 241–254.

[17] G. Quon, S. Haider, A.G. Deshwar, A. Cui, P.C. Boutros, and Q. Morris, *Computational purification of individual tumor gene expression profiles leads to significant improvements in prognostic prediction*, Genome Med. **5** (2013), no. 3, 1–20.

[18] M. Rivaie, A. Abashar, M. Mamat, and I. Mohd, *The convergence properties of a new type of conjugate gradient methods*, Appl. Math. Sci. **8** (2014), 33–44.

[19] M. Rivaie, M. Mamat, L.W. June, and I. Mohd, *A new class of nonlinear conjugate gradient coefficients with global convergence properties*, Appl. Math. Comput. **218** (2012), no. 22, 11323–11332.

[20] D. Touati-Ahmed and C. Storey, *Efficient hybrid conjugate gradient techniques*, J. Optim. Theory Appl. **64** (1990), no. 2, 379–397.

[21] J. Wang and X. Chi, *Cg global convergence properties with Goldstein linesearch*, Bull. Brazil. Math. Soc. **36** (2005), no. 2, 197–204.

[22] Z. Wei, G. Li, and L. Qi, *New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems*, Appl. Math. Comput. **179** (2006), no. 2, 407–430.

[23] Z. Wei, S. Yao, and L. Liu, *The convergence properties of some new conjugate gradient methods*, Appl. Math. Comput. **183** (2006), no. 2, 1341–1350.

Conclusion

Dans ce travail, la recherche aborde une problématique centrale dans le domaine de l'analyse numérique et de l'optimisation. L'originalité de cette étude réside dans l'exploration approfondie de différentes méthodes itératives pour résoudre efficacement des problèmes d'optimisation sans contraintes, ce qui représente une contribution significative à l'avancement des connaissances dans ce domaine spécialisé.

Les conclusions de cette étude sont exposées minutieusement, accompagnées d'une analyse approfondie qui met en lumière leur pertinence et leur impact sur la résolution de divers problèmes. L'efficacité des nouvelles méthodes explorées est étayée par des expérimentations numériques rigoureuses, soulignant ainsi leur capacité à enrichir les applications et à faire avancer les connaissances dans le domaine spécifique examiné.

Perspectives

Tout d'abord, il serait intéressant d'explorer davantage l'intégration de techniques d'apprentissage automatique dans ces méthodes itératives, afin de bénéficier de leur capacité à généraliser les solutions et à gérer des ensembles de données massifs. De plus, l'extension de ces méthodes à des problèmes d'optimisation avec des contraintes pourrait représenter un défi enrichissant, nécessitant le développement de nouvelles stratégies adaptatives et robustes. Enfin, une exploration approfondie des aspects théoriques sous-jacents à ces méthodes, comme la convergence et la stabilité, pourrait conduire à des améliorations substantielles dans leur efficacité et leur applicabilité pratique.

Bibliographie

[A+17] Neculai Andrei et al. *Continuous nonlinear optimization for engineering applications in GAMS technology*, volume 121. Springer, 2017.

[AB85] Mehiddin Al-Baali. Descent property and global convergence of the fletcher—reeves method with inexact line search. *IMA Journal of Numerical Analysis*, 5(1) :121–124, 1985.

[ABF86] Mehiddin Al-Baali and Roger Fletcher. An efficient line search for nonlinear least squares. *Journal of Optimization Theory and Applications*, 48(3) :359–377, 1986.

[Aka59] Hirotugu Akaike. On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Annals of the Institute of Statistical Mathematics*, 11 :1–16, 1959.

[And99] Neculai Andrei. Programarea matematică avansată. *Teorie, Metode Computaţionale, Aplicaţii. Editura Tehnică-Bucureşti*, 1999.

[And08] Neculai Andrei. An unconstrained optimization test functions collection. *Adv. Model. Optim*, 10(1) :147–161, 2008.

[And09a] N Andrei. Critica raţiunii algoritmilor de optimizare fără restricţii.[criticism of the unconstrained optimization algorithms reasoning]. *Editura Academiei Române, Bucureşti*, 2009.

[And09b] Neculai Andrei. Acceleration of conjugate gradient algorithms for unconstrained optimization. *Applied Mathematics and Computation*, 213(2) :361–369, 2009.

[And11] Neculai Andrei. Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization. *Bulletin of the Malaysian Mathematical Sciences Society. Second Series*, 34(2) :319–330, 2011.

[And15] Neculai Andrei. *Critica ratiunii algoritmilor de optimizare cu restrictii [& 1 CD anexa]*. Editura Academiei Române, 2015.

[Arm66] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1) :1–3, 1966.

[BB08] Michael Bartholomew-Biggs. *Nonlinear optimization with engineering applications*, volume 19. Springer Science & Business Media, 2008.

[BBM00a] Alexander S Bondarenko, David M Bortz, and JJ Moré. Cops : Large-scale nonlinearly constrained optimization problems. Technical report, Argonne National Lab., IL (US), 2000.

[BBM00b] Alexander S Bondarenko, David M Bortz, and JJ Moré. Cops : Large-scale nonlinearly constrained optimization problems. Technical report, Argonne National Lab., IL (US), 2000.

[BDF97] Stephen C Billups, Steven P Dirkse, and Michael C Ferris. A comparison of large scale mixed complementarity problem solvers. *Computational Optimization and Applications*, 7 :3–25, 1997.

[Ber97] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3) :334–334, 1997.

[BKR18] Saman Babaie-Kafaki and Saeed Rezaee. Two accelerated nonmonotone adaptive trust region line search methods. *Numerical Algorithms*, 78 :911–928, 2018.

[Blu94] Eugen Blum. From optimization and variational inequalities to equilibrium problems. *Math. student*, 63 :123–145, 1994.

[BSS13] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming : theory and algorithms*. John wiley & sons, 2013.

[BSV00] Hande Y Benson, David F Shanno, and Robert J Vanderbei. Interior-point methods for nonconvex nonlinear programming : Jamming and comparative numerical testing. *Operations Research and Financial Engineering, Princeton University, ORFE-00-02*, page 59, 2000.

[C$^+$47] Augustin Cauchy et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847) :536–538, 1847.

[Căt19] Emil Cătinaş. A survey on the high convergence orders and computational convergence orders of sequences. *Applied Mathematics and Computation*, 343 :1–20, 2019.

[CDM79] Harlan Crowder, Ron S Dembo, and John M Mulvey. On reporting computational experiments with mathematical software. *ACM Transactions on Mathematical Software (TOMS)*, 5(2) :193–203, 1979.

[CGT96] Andrew R Conn, Nick Gould, and Ph L Toint. Numerical experiments with the lancelot package (release a) for large-scale nonlinear optimization. *Mathematical Programming*, 73(1) :73–110, 1996.

[CGT13] Andrew R Conn, GIM Gould, and Philippe L Toint. *LANCELOT : a Fortran package for large-scale nonlinear optimization (Release A)*, volume 17. Springer Science & Business Media, 2013.

[Cha07] Benoit Chachuat. Nonlinear and dynamic optimization : From theory to practice. 2007.

[Dai11] Yuhong Dai. Convergence analysis of nonlinear conjugate gradient methods. *Optimization and regularization for computational inverse problems and applications*, pages 157–181, 2011.

[DJS96] John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

[DM02] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91 :201–213, 2002.

[DMM04] Elizabeth D Dolan, Jorge J Moré, and Todd S Munson. Benchmarking optimization software with cops 3.0. Technical report, Argonne National Lab., Argonne, IL (US), 2004.

[DY99] Yu-Hong Dai and Yaxiang Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on optimization*, 10(1) :177–182, 1999.

[FGK03] Robert Fourer, David M Gay, and Brian W Kernighan. Ampl. a modeling language for mathematical programming. 2003.

[Fle00] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2000.

[FR64] Reeves Fletcher and Colin M Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2) :149–154, 1964.

[Gil07] JC Gilbert. Eléments d'optimisation différentiable : Théorie et algorithmes, notes de cours. *École Nationale Supérieure de Techniques Avancées, Paris*, 2007.

[GM08] Neng-zhu Gu and Jiang-tao Mo. Incorporating nonmonotone strategies into the trust region method for unconstrained optimization. *Computers & Mathematics with Applications*, 55(9) :2158–2172, 2008.

[GMS97] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt : An algorithm for large-scale constrained optimization. *Report NA97-2, University of California, San Diego*, 1997.

[GN92a] Jean Charles Gilbert and Jorge Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on optimization*, 2(1) :21–42, 1992.

[GN92b] Jean Charles Gilbert and Jorge Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on optimization*, 2(1) :21–42, 1992.

[GN92c] Jean Charles Gilbert and Jorge Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on optimization*, 2(1) :21–42, 1992.

[Gol65] Allen A Goldstein. On steepest descent. *Journal of the Society for Industrial and Applied Mathematics, Series A : Control*, 3(1) :147–151, 1965.

[Hag89] William W Hager. A derivative-based bracketing scheme for univariate minimization and the conjugate gradient method. *Computers & Mathematics with Applications*, 18(9) :779–795, 1989.

[HDZL01] Y H. Dai and L Z. Liao. New conjugacy conditions and related nonlinear conjugate gradient methods. *Applied Mathematics and optimization*, 43 :87–101, 2001.

[Hig02] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[HMRS16] Mohamed Hamoda, Mustafa Mamat, Mohd Rivaie, and Zabidin Salleh. A conjugate gradient method with strong wolfe-powell line search for unconstrained optimization. *Appl. Math. Sci*, 10 :721–734, 2016.

[HS+52] Magnus Rudolph Hestenes, Eduard Stiefel, et al. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

[HZ05a] William W Hager and Hongchao Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, 16(1) :170–192, 2005.

[HZ05b] William W Hager and Hongchao Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, 16(1) :170–192, 2005.

[HZ05c] William W Hager and Hongchao Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, 16(1) :170–192, 2005.

[JBNP90] Richard HF Jackson, Paul T Boggs, Stephen G Nash, and Susan Powell. Guidelines for reporting results of computational experiments. report of the ad hoc committee. *Mathematical programming*, 49 :413–425, 1990.

[Kel99] Carl T Kelley. *Iterative methods for optimization*. SIAM, 1999.

[Lem81] C Lemaréchal. A view of line search. în : Auslander, oettli, j. stoer (eds.) optimization and optimal control, 1981.

[LS91a] Y Liu and C Storey. Efficient generalized conjugate gradient algorithms, part 1 : theory. *Journal of optimization theory and applications*, 69 :129–137, 1991.

[LS91b] Y Liu and C Storey. Efficient generalized conjugate gradient algorithms, part 1 : theory. *Journal of optimization theory and applications*, 69 :129–137, 1991.

[LTW07] Guoyin Li, Chunming Tang, and Zengxin Wei. New conjugacy condition and related new conjugate gradient methods for unconstrained optimization. *Journal of Computational and Applied Mathematics*, 202(2) :523–539, 2007.

[Lue73] David G Luenberger. Introduction to linear and nonlinear programming. *Reading : Addison-Wesley Publishing Company)*, 1973.

[Lue84] David G Luenberger. Introduction to linear and nonlinear programming (2nd ed). *Reading : Addison-Wesley Publishing Company)*, 1984.

[Luk92] Ladislav Lukšan. Computational experience with improved conjugate gradient methods for unconstrained minimization. *Kybernetika*, 28(4) :249–262, 1992.

[Mit99] HD Mittelmann. Benchmarking interior point lp/qp solvers. *Optimization Methods and Software*, 11(1-4) :655–670, 1999.

[Mit17] Hans D Mittelmann. Latest benchmarks of optimization software. In *INFORMS Annual Meeting. Houston, TX*, 2017.

[MMA+15] Ibrahim S Mohammed, Mustafa Mamat, Abdelrhaman Abashar, Mohd Rivaie, and Zabidin Salleh. A modified nonlinear conjugate gradient method for unconstrained optimization. *Applied Mathematical Sciences*, 9(54) :2671–2682, 2015.

[MRMF10] Mustafa Mamat, Mohd Rivaie, Ismail Mohd, and Muhammad Fauzi. A new conjugate gradient coefficient for unconstrained optimization. *Int. J. Contemp. Math. Sciences*, 5(29) :1429–1437, 2010.

[MS95] BA Murtagh and MA Saunders. Minos 5.4 user's guide, rep. sol 83-20r, syst. *Optim. Lab., Stanford Univ., Stanford, Calif*, 1995.

[MT94] Jorge J Moré and David J Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software (TOMS)*, 20(3) :286–307, 1994.

[MWG19] Walter Murray, Margaret H Wright, and Philip E Gill. *Practical optimization*. SIAM-Society for Industrial and Applied Mathematics, 2019.

[Noc92] Jorge Nocedal. Theory of algorithms for unconstrained optimization. *Acta numerica*, 1 :199–242, 1992.

[Num97] A Numerical. Lancelot and minos packages for. 1997.

[NW99] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[NXPN98] Tan Nguyen Xuan and Tinh Phan Nhat. On the existence of equilibrium points of vector functions. *Numerical Functional Analysis and Optimization*, 19(1-2) :141–156, 1998.

[NY83] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[OL17] Yigui Ou and Yuanwen Liu. A memory gradient method based on the nonmonotone technique. *Journal of Industrial & Management Optimization*, 13(2), 2017.

[OMR15] Osman Omer, Mustafa Mamat, and Mohd Rivaie. The global convergence properties of a family of conjugate gradient method under the strong wolfe line search. In *Abstract and Applied Analysis*, 2015.

[OR00] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[Pol69] Boris Teodorovich Polyak. The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics*, 9(4) :94–112, 1969.

[Pot89] FA Potra. On q-order and r-order of convergence. *Journal of Optimization Theory and Applications*, 63(3) :415–431, 1989.

[Pow76] Michael JD Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. *Nonlinear programming*, 9(1) :53–72, 1976.

[Pow77] Michael James David Powell. Restart procedures for the conjugate gradient method. *Mathematical programming*, 12 :241–254, 1977.

[Pow84] Michael JD Powell. Nonconvex minimization calculations and the conjugate gradient method. In *Numerical Analysis : Proceedings of the 10th Biennial Conference held at Dundee, Scotland, June 28–July 1, 1983*, pages 122–141. Springer, 1984.

[PR69] E Pola and G Ribiere. Note sur la convergence de methodes de directions conjugées. *Rev Française Informat Recherche Operationelle, 3e Année*, 16 :35–43, 1969.

[PS95] FA Potra and Y Shi. Efficient line search algorithm for unconstrained optimization. *Journal of Optimization Theory and Applications*, 85 :677–704, 1995.

[RAMM14] Mohd Rivaie, Abdelrhaman Abashar, Mustafa Mamat, and Ismail Mohd. The convergence properties of a new type of conjugate gradient methods. *Appl. Math. Sci*, 8 :33–44, 2014.

[RMJM12] Mohd Rivaie, Mustafa Mamat, Leong Wah June, and Ismail Mohd. A new class of nonlinear conjugate gradient coefficients with global convergence properties. *Applied Mathematics and Computation*, 218(22) :11323–11332, 2012.

[Sha83] DF Shanno. Conmin–a fortran subroutine for minimizing an unconstrained nonlinear scalar valued function of a vector variable x either by the bfgs variable

metric algorithm or by a beale restarted conjugate gradient algorithm. *Private communication, October*, 17 :1983, 1983.

[SY06] Wenyu Sun and Ya-Xiang Yuan. *Optimization theory and methods : nonlinear programming*, volume 1. Springer Science & Business Media, 2006.

[VS99] Robert J Vanderbei and David F Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13 :231–252, 1999.

[WC05] Jian Wang and Xuebin Chi. Cg global convergence properties with goldstein linesearch. *Bulletin of the Brazilian Mathematical Society*, 36(2) :197–204, 2005.

[WLQ06] Zengxin Wei, Guoyin Li, and Liqun Qi. New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems. *Applied Mathematics and computation*, 179(2) :407–430, 2006.

[Wol69] Philip Wolfe. Convergence conditions for ascent methods. *SIAM review*, 11(2) :226–235, 1969.

[Wol71] Philip Wolfe. Convergence conditions for ascent methods. ii : Some corrections. *SIAM review*, 13(2) :185–188, 1971.

[WYL06a] Zengxin Wei, Shengwei Yao, and Liying Liu. The convergence properties of some new conjugate gradient methods. *Applied Mathematics and computation*, 183(2) :1341–1350, 2006.

[WYL06b] Zengxin Wei, Shengwei Yao, and Liying Liu. The convergence properties of some new conjugate gradient methods. *Applied Mathematics and computation*, 183(2) :1341–1350, 2006.

[Yua99] Ya-xiang Yuan. Problems on convergence of unconstrained optimization algorithms. *Numerical Linear Algebra and Optimization,(Science Press, Beijing, New York)*, pages 95–107, 1999.

[Zou70] G Zoutendijk. Nonlinear programming, computational methods. *Integer and nonlinear programming*, pages 37–86, 1970.