People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

university Mohamed Chérif Messaadia–Souk Ahras

Laboratory of Computer Science and Mathematics

Faculty of Sciences and Technology

Thesis

Presented to obtain the diploma of 3rd Cycle Doctorate in Mathematics: non parametric statistiques

Title

New hybrid conjugate gradient method for mode function estimation

Presented by

Oumertem Lemya

Before the following jury:

Name and surname	grade	university	quality
1- BELLOUFI Mohammed	Professor	U.Souk-ahras	President
2- CHAIB Yacine	MCA	U.Souk-ahras	Reporter
3- SELLAMI Badreddine	Professor	U.Souk-ahras	Examiner
4- ZARAI Abderrahmane	Professor	U.L.T-Tébessa	Examiner
5- BOUALI Tahar	MCA	U.L.T-Tébessa	Examiner

University Year: 2024-2025.

بسم الله الرحمن الرحيم

Acknowledgements

All praise and gratitude are due to Allah alone. He has blessed us with success and granted us the patience to overcome the challenges we encountered in completing this humble endeavor.

First and foremost, I would like to express my deepest gratitude to my supervisor, **Dr. Chaib Yacine**, for their invaluable guidance, continuous support, and encouragement throughout the course of my PhD journey. Their expertise and constructive feedback have been instrumental in shaping this work and bringing it to fruition.

I would also like to thank the members of my thesis committee, **Prof. Belloufi Mohammed**, **Prof. Sellami Badreddine** from Souk Ahras University and **Prof. Zarai Abderrahmane**, **Dr. Bouali Tahar** from Tébessa University, for their insightful comments and suggestions, which have greatly contributed to the improvement of this research.

My heartfelt thanks go to my colleagues and friends who provided moral support, shared their knowledge, and stood by me during the challenging moments of this journey, especially **Gueffasa Imen** and **Meansri Khaoula**.

I would like to extend my heartfelt thanks to the Mathematics Department at Souk Ahras University, including all its staff, the department head, and the esteemed professors, for their warm welcome and their efforts in facilitating every procedure we encountered.

Finally, I express my sincere gratitude to everyone who has contributed, in any way, to the development of this work.

Dedication

I dedicate this research to my dear father, my beloved mother, and my esteemed father-in-law. I express my deep gratitude to all of them for their continuous support and invaluable sacrifices, which have been the reason behind my academic success. Their sincere efforts and unwavering support have been among the most important factors in achieving this accomplishment, and I am deeply grateful to them for everything they have done for me throughout my academic journey.

*To my beloved husband, **Harchani Ala Eddine**, I dedicate this humble research as a heartfelt token of my deepest gratitude for his steadfast support throughout my academic journey. His emotional and financial encouragement has been invaluable in allowing me to achieve my scholarly aspirations.*

*I dedicate this research to those whose presence has blessed my life and whose joyful laughter has filled my years with happiness—my dear son **Mohamed**, as well as the beautiful **Noor El-Kamar** and **Acil**.*

*To those who rejoice in our success, share our hopes and dreams, and feel sorrow when we face challenges, I dedicate this research to my dear siblings, and I especially thank and pray for the two honorable families: the **Oumertem** and **Harchani** families, for their continuous support and unlimited encouragement.*

To all my friends who have always been a source of inspiration and motivation at every step of my academic journey.

.

CONTENTS

Abstract

Unconstrained optimization is a technique used to find the best possible solution or optimal value for a given problem, often in the context of minimizing or maximizing a function. These methods are fundamental in various fields such as engineering, economics, machine learning, and operations research.

Conjugate gradient methods are very important methods for solving unconstrained optimization problems, especially when the dimension is large. In this thesis, based on the hybrid conjugate gradient method, a new family of gradient methods are proposed for solving unconstrained optimization.

By using Wolfe line-search conditions, these changes aim to improve the algorithms' convergence features and accelerate the descent direction. Our studies' numerical results offer strong evidence of the robustness and efficiency of these new methods compared to existing conjugate gradient methods.

We conducted thorough numerical studies to validate our proposed methods. We have demonstrated through numerical tests that the suggested methods are more effective and perform better than the combined algorithms after demonstrating their convergence using experimental functions.

Additionally, the proposed algorithms were expanded to address challenges in nonparametric statistics, specifically focusing on the problems of the mode function.

Keywords: Conjugate gradient method, Global convergence, Inexact line search, Numerical comparisons, Mode function, Kernel estimator.

Resumé

L'optimisation sans contrainte est une technique utilisée pour trouver la meilleure solution possible ou la valeur optimale pour un problème donné, souvent dans le contexte de la minimisation ou de la maximisation d'une fonction. Ces méthodes sont fondamentales dans divers domaines tels que l'ingénierie, l'économie, l'apprentissage automatique et la recherche opérationnelle.

Les méthodes de gradient conjuguées sont des méthodes très importantes pour résoudre les problèmes d'optimisation sans contrainte, en particulier lorsque la dimension est grande. Dans cette thèse, en se basant sur la méthode hybride de gradient conjuguée, une nouvelle famille de méthodes de gradient est proposée pour résoudre les problèmes d'optimisation sans contrainte.

En utilisant les conditions de recherche linéaire de Wolfe, ces modifications visent à améliorer les caractéristiques de convergence des algorithmes et à accélérer la direction de descente. Les résultats numériques de nos études offrent des preuves solides de la robustesse et de l'efficacité de ces nouvelles méthodes par rapport aux méthodes de gradient conjuguées existantes.

Nous avons mené des études numériques approfondies pour valider nos méthodes proposées. Nous avons démontré, à travers des tests numériques, que les méthodes suggérées sont plus efficaces et performantes que les algorithmes combinés, après avoir démontré leur convergence à l'aide de fonctions expérimentales.

De plus, les algorithmes proposés ont été étendus pour aborder des défis en statistiques non paramétriques, en se concentrant spécifiquement sur les problèmes liés à la fonction de mode.

Mots clés: Méthode de gradient conjuguées, Convergence globale, Recherche linéaire inexacte, Comparaisons numériques, Fonction de mode, Estimateur à noyau.

الملخص

الأمثلة غير مقيدة هي تقنية تستخدم لإيجاد أفضل حل ممكن أو القيمة المثلى لمشكلة معينة، و غالبا في سياق إيجاد القيم الحدية الصغرى أو الكبرى للدالة. تعد هذه الطرق أساسية في مجالات مختلفة مثل الهندسة، الاقتصاد، التعليم الآلي، و بحوث العمليات.

تعد طرق التدرج المترافق من أهم الطرق لحل مسائل الأمثلة غير مقيدة(الحرة)، خاصة عندما يكون البعد كبيرا. في هذه الأطروحة، و استنادا إلى طريقة التدرج المترافق الهجينة، تم اقتراح عائلة جديدة من طرق التدرج المترافق لحل مشكلات الأمثلة غير مقيدة.

من خلال استخدام شروط بحث وولف القوي، تهدف هذه التعديلات إلى تحسين تقارب الخوارزميات و تسريعه. أظهرت النتائج العددية لدراستنا أدلة قوية على متانة و كفاءة هذه الطرق الجديدة مقارنة بطرق التدرج المترافق القديمة.

قمنا بإجراء دراسات عددية شاملة للتحقق من صحة طرقنا المقترحة. أثبتنا من خلال الاختبارات العددية أن الطرق المقترحة أكثر فعالية و تؤدي أداء أفضل من الخوارزميات القديمة بعد إثبات تقاربها باستخدام دوال تجريبية.

بالإضافة إلى ذلك، تم توسيع الخوارزميات المقترحة لمعالجة المشكلات في الإحصاء الغير وسيطي، مع التركيز بشكل خاص على مشكلات دالة المنوال.

__الكلمات الرئيسية__: طريقة التدرج المترافق، التقارب الشامل، البحث الخطي الغير دقيق، المقارنات العددية، دالة المنوال، مقدر النواة.

General Introduction

Unconstrained optimization refers to the process of finding the maximum or minimum of an objective function without any restrictions or constraints on the variables. It is a fundamental tool used in various fields, including applied mathematics, engineering, medicine, economics, computer science, operations research, and other sciences.

Optimization helps improve processes, design efficient systems, make informed decisions, and solve complex problems by finding the best solution within a continuous and unbounded parameter space. The field of unconstrained optimization has a rich history, rooted in mathematics and practical problem-solving, and has evolved over centuries with contributions from numerous mathematicians, engineers, and scientists. The origins of optimization concepts can be traced back to ancient civilizations, where mathematicians studied geometric problems and extrema. For example, the Greeks worked on maximizing and minimizing areas and volumes.

The 20th century saw a surge in interest in optimization, particularly with the advent of computers. This period focused on developing efficient numerical methods. It should be noted that these advances were mainly achieved in Great Britain. After the 1950s, when conjugate gradient methods [31] and quasi-Newton methods were introduced, nonlinear programming experienced significant development, with many contributions emerging. Notable contributors include G. Zoutendijk ([60], 1960), C. W. Carroll ([Fletcher] and M. J. D. Powell ([24], 1963), A. A. Goldstein ([29], 1965), and A. V. Fiacco and G. P. McCormick ([22], 1968) in the area of nonlinear programming. Additionally, important work was done by E. Polak and G. Ribière ([43], 1969), B. T. Polyak ([44], 1969), and Goldstein and Price ([28], 1969).

In this work, we will discuss one of these methods, the conjugate gradient method (CGM). The conjugate gradient method is used to solve systems of linear equations, particularly those of the form $Ax = b$, where A is a symmetric positive definite matrix and b is a given vector. This method is used to find the solution x that minimizes the quadratic form $\frac{1}{2}x^T Ax = b^T x$.

The Conjugate gradient method is particularly useful for solving large sparse systems that arise in various applications, such as the numerical solutions to partial differential equations.

Many nonlinear conjugate gradient methods are known. They differ from each other in two ways: the manner in which the search direction is updated and the procedure used to compute the stepsize along this direction. The main requirement for the search direction in conjugate gradient methods is that it satisfies the descent or sufficient descent condition. The stepsize is computed using the Wolfe line search conditions or some variants of them.

The unconstrained minimization problem is a type of mathematical optimization problem where the objective is to find the values of variables that minimize a given objective function. Consider the following problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, and the vector x represents the variable we aim to optimize.

A nonlinear conjugate gradient method is usually designed by the iterative form:

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

where x_k is the current iterate point, and α_k is a step length which is determined by some line search. The search direction d_k is generated recurrently by the rule:

$$d_{k+1} = -g_{k+1} + \beta_k d_k \text{ and } d_0 = -g_0,$$

where β_k is a scalar.

The most important standard conjugate gradient methods are:

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \quad \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{y_k^T d_k}, \quad \beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-g_k^T d_k},$$
$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad \beta_k^{HS} = \frac{g_{k+1}^T y_k}{y_k^T d_k}, \quad \beta_k^{LS} = \frac{g_{k+1}^T y_k}{-g_k^T d_k}.$$

The corresponding methods are called FR (Fletcher-Reeves) [23], DY (Dai-Yuan) |15], CD (conjugate descent) [25], PRP (Polak-Ribiére-Polyak) [43,44], HS (Hestenes-Stiefel) [31] and LS [Liu-Storey) [37] conjugate gradient method, respectively.

The methods FR, DY, and CD have good convergence properties; however, their numerical performances are modest and impacted by jamming, according to numerical investigations using standard conjugate gradient methods. On the other hand, the computational performances of HS, PRP, and LS methods are better, even though their convergence properties are weaker.

The conjugate gradient method was discovered in 1952 by Hestenes and Stiefel for the minimization of strictly convex quadratic functions. Several mathematicians have since extended it to the nonlinear case. The algorithms and the convergence study of various versions of the conjugate gradient algorithm for nonquadratic functions were discussed by Fletcher and Reeves (1964) [23], Polak and Ribière (1969), and Polyak (1969)[43,44].

If the objective function is a strongly convex quadratic and the line search is exact, then, in theory, all choices for the search direction in standard conjugate gradient algorithms are equivalent. However,

for non-quadratic functions, each choice of the search direction results in standard conjugate gradient algorithms with very different performances.
There are two methods for combining standard schemes to create hybrid conjugate gradient algorithms:

- The first combination is based on the concept of projection. When a criterion is satisfied, one of the two standard conjugate gradient methods is used. This is the idea behind these methods. The other standard conjugate gradient method from the pair can be used as soon as the criterion is violated.

- The second class of the hybrid conjugate gradient methods is based on the convex combination of standard methods. These methods work by selecting two standard methods and combining them in a convex manner. The parameter in the convex combination is then calculated using either the Newton search direction or the conjugacy condition.

- In general, hybrid methods are more efficient and more robust than the standard ones:
In the following, we will discuss an important part of this thesis which is the application of conjugate gradient method in nonparametric estimation.
Nonparametric estimation refers to a set of statistical techniques used to estimate an unknown distribution or relationship without assuming a specific parametric form for the underlying population. Unlike parametric methods, which make strong assumptions about the functional form (e.g., assuming a normal distribution or a linear relationship), nonparametric methods are more flexible and can adapt to the structure present in the data.
Here are some common nonparametric estimation techniques:

- Kernel Density Estimation (KDE): is a non-parametric way to estimate the probability density function (PDF) of a random variable. It's a useful method for understanding the underlying distribution of data, especially when the form of the distribution is unknown. KDE is widely used in statistics, machine learning, and data visualization.

- Histogram Estimation: is a non-parametric method used to estimate the probability distribution of a dataset. It involves dividing the range of data into a series of intervals, known as bins, and then counting the number of data points that fall into each bin. This provides a simple visual representation of the data distribution, commonly depicted as a bar graph.

- Empirical Cumulative Distribution Function (ECDF): is a statistical tool used to estimate the cumulative distribution function (CDF) of a random variable based on a given sample. It provides a non-parametric way to summarize the distribution of the data.

- Nonparametric Regression: is a type of regression analysis that makes fewer assumptions about the form of the relationship between the independent and dependent variables. Unlike parametric regression models (such as linear regression), which assume a specific functional form (like a straight line or a polynomial), nonparametric regression does not assume any particular shape for the relationship. Instead, it allows the data to determine the model's structure, which can be more flexible in capturing complex patterns.

- Rank-Based Methods: are statistical techniques used to analyze and interpret data by assigning ranks to the data points instead of using their raw values. These methods are particularly useful when the data does not meet the assumptions required for parametric methods, such as normality or equal variances.

Nonparametric estimation is a versatile tool in statistics and data analysis, particularly in situations where the underlying data distribution is not well-defined or when the data does not conform to common parametric assumptions. It allows researchers and analysts to derive insights and make informed decisions based on the observed data, while avoiding potentially restrictive assumptions that might not accurately capture the true characteristics of the data.

The structure of this thesis is as follows:

We start with an introduction and then divide the content into five chapters:

-**Chapter 1**: Contains basic concepts useful in the following, especially differential calculs and convex analysis.

-**Chapter 2**: In this chapter, we provide an overview of the conjugate gradient method and its convergence, present the results of global convergence and the properties of classical hybrid methods and we focus on the part of the nonparametric statistics.

-**Chapter 3**: In this part, we construct a new hybrid conjugate gradient method related to the HS and DY methods.

-**Chapter 4**: We present in this chapter the proposing of two modified conjugate gradient mlethods.

-**Chapter 5**: This part contains a new conjugate gradient method as a convex combination of NHS and FR methods.

For our new methods (presented in Chapters 3, 4, and 5), we prove the sufficient descent conditions, find that the global convergence is established, and we report efficient and reliable numerical performance of each of them. Also, we discuss application of these methods in nonparametric estimation.

CHAPTER 1

Preliminary results

In this part, based on references [6], [10] and [54] we give some generalities on the problems of unconstrained minimization.

1.1 Definitions

1.1.1 Gradient and Hessian

Definition 1.1 *[54]*

-Note by

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

the gradient of f at point $x = (x_1, ..., x_n)$.
-Hessian of f is called the symmetric matrix of order $n \times n$

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

▶ The matrix H is called positive semi-definite if

$$\forall x \in \mathbb{R}^n : \ x^T H(x) \geq 0. \tag{1.3}$$

▶ H is said to be positive-definite if

$$\forall x \in \mathbb{R}^n, \ x \neq 0 : x^T H(x) > 0. \tag{1.4}$$

1.1.2 Continuity and Differentiability

Definition 1.2 *(Norm) [54]*

We have the norm if and only if it satisfies the following properties, and we denoted $\| \,.\, \|$:
1- $\| x \| \geq 0, \forall x \in \mathbb{R}^n$, $\| x \| = 0 \Leftrightarrow x = 0$.
2- $\| \alpha x \| = |\alpha| \, \| x \|$, $\forall \alpha \in \mathbb{R}$ *and* $\forall x \in \mathbb{R}^n$.
3- $\| x_1 + x_2 \| \leq \| x_1 \| + \| x_2 \|, \forall x_1, x_2 \in \mathbb{R}^n$.

Definition 1.3 *[54] A function $f : \mathbb{R}^n \to R$ is said to be* ***continuous*** *at $\bar{x} \in \mathbb{R}^n$,if*

$$\forall \varepsilon > 0, \exists \delta > 0 : \forall x \in N_\delta(\bar{x}) \Rightarrow f(x) \in N\varepsilon(f(\bar{x})). \tag{1.5}$$

If f is **continuous** at every point in an open set $S \subset \mathbb{R}^n$, then f is said to be **continuous** on S.

Definition 1.4 *[54]*

- A continuous function $f : \mathbb{R}^n \to R$ is said to be **continuously differentiable** at $x \in \mathbb{R}^n$ if $\nabla f(x)$ exists and is continuous.
- If f is **continuously differentiable** at every point of an open set $S \subset \mathbb{R}^n$, then f is said to be **continuously differentiable** on S and denoted by $f \in C^1(S)$.
- A **continuously differentiable** function $f : \mathbb{R}^n \to R$ is called **twice continuously differentiable** at $x \in \mathbb{R}^n$ if $[\nabla^2 f(x)]_{ij}$ exists and is continuous, $i, j = 1, \cdots, n$.
- If f is **twice continuously differentiable** at every point in an open set $S \subset \mathbb{R}^n$, then f is said to be twice continuously differentiable on S and denoted by $f \in C^2(S)$.

Definition 1.5 *[54] (Directional derivative) Let $f : \mathbb{R}^n \to R$ be continuously differentiable on an open set $S \subset \mathbb{R}^n$. Then for $x \in S$ and $d \in \mathbb{R}^n$, the directional derivative of f at x in the*

direction d is defined as:

$$f\,'(x, d) = \lim_{\theta \to 0} \frac{f(x + \theta d) - f(x)}{\theta} = \nabla f(x)^T d. \tag{1.6}$$

1.1.3 Convexity

Convex sets and functions

Definition 1.6 *Let the set $S \subset \mathbb{R}^n$. If, for any $x_1, x_2 \in S$, we have*

$$\lambda x_1 + (1-\lambda)x_2 \in S, \forall \lambda \in [0,1], \tag{1.7}$$

then S is said to be a **convex** set.

Definition 1.7 *Let $S \subset \mathbb{R}^n$ be a non-empty convex set. A function $f : S \subset \mathbb{R}^n \to R$ is a convex function if and only if for all $x_1, x_2 \in S$, we have*

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2), \text{ for all } \alpha \in [0,1]. \tag{1.8}$$

▶ It is said that: $f : S \subset \mathbb{R}^n \to R$ is strictly convex if for all $x_1, x_2 \in S$ with $x_1 \neq x_2$, we have

$$f(\alpha x_1 + (1-\alpha)x_2) < \alpha f(x_1) + (1-\alpha) f(x_2), \text{ for all } \alpha \in]0,1[. \tag{1.9}$$

▶ It is said that: $f : S \subset \mathbb{R}^n \to R$ is strongly convex or uniformly convex of module $\delta > 0$ $(\delta - convex)$, if for all $x_1, x_2 \in S$, we have

$$f((1-\alpha)x_1 + \alpha x_2) \leq (1-\alpha) f(x_1) + \alpha f(x_2) - \frac{\delta}{2}\alpha(1-\alpha) \parallel x_1 - x_2 \parallel^2, \text{ for all } \alpha \in]0,1[. \tag{1.10}$$

▶ It is said that: $f : S \subset \mathbb{R}^n \to R$ is quasi-convex if for all $x_1, x_2 \in S$, we have

$$f[\alpha x_1 + (1-\alpha)x_2] \leq max[f(x_1), f(x_2)], \text{ for all } \alpha \in]0,1[. \tag{1.11}$$

▶ It is said that: $f : S \subset \mathbb{R}^n \to R$ is strictly quasi-convex if for all $x_1, x_2 \in S$ with $f(x_1) \neq f(x_2)$, we have

$$f[\alpha x_1 + (1-\alpha)x_2] < max[f(x_1), f(x_2)], \text{ for all } \alpha \in]0,1[. \tag{1.12}$$

A convex function can also be described by an epigraph.

Definition 1.8 *(Epigraph)*

▶ Let $S \subset \mathbb{R}^n$ be a non-empty set. Let $f : S \subset \mathbb{R}^n \to \mathbb{R}$. The epigraph of f, denoted by *epi* f, is a subset of $\mathbb{R}^{n+1}$ defined by

$$epi\ f = \{(x, \alpha) : f(x) \leq \alpha,\ x \in S,\ \alpha \in \mathbb{R}\}. \tag{1.13}$$

▶ The *hypograph* of f, denoted by *hyp* f, is a subset of $\mathbb{R}^{n+1}$ defined by

$$hyp\ f = \{(x, \alpha) : f(x) \geq \alpha,\ x \in S,\ \alpha \in \mathbb{R}\}. \tag{1.14}$$

1.2 Optimality Conditions

Definition 1.9 *We consider the unconstrained optimization problem (P).*

▶ $\bar{x} \in \mathbb{R}^n$ is called **global minimum** of the problem (P) if

$$f(\bar{x}) \leq f(x), \ \forall x \in \mathbb{R}^n.$$

• $\bar{x} \in \mathbb{R}^n$ is called **strict global minimum** of the problem (P) if

$$f(\bar{x}) < f(x), \ \forall x \in \mathbb{R}^n, \ \text{with } \bar{x} \neq x.$$

▶ $\bar{x} \in \mathbb{R}^n$ is called **local minimum** of the problem (P) if there is a $V_\epsilon(\bar{x})$ neighbourhood of $\bar{x}$ such that

$$f(\bar{x}) \leq f(x), \ \forall x \in V\varepsilon(\bar{x}).$$

• $\bar{x} \in \mathbb{R}^n$ is called **strict local minimum** of the problem (P), if there is a $V_\epsilon(\bar{x})$ neighbourhood of $\bar{x}$ such that

$$f(\bar{x}) < f(x), \ \forall x \in V\epsilon(\bar{x})\text{with } \bar{x} \neq x.$$

Definition 1.10 *[6] Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuous at $\bar{x}$. A vector $d \in \mathbb{R}^n$ is a descent direction for f at $\bar{x}$ if there exists $\delta > 0$ so that*

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

Theorem 1.1 *[6] Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable function at $\bar{x}$. If there exists a vector d so that $\nabla f(\bar{x})^T d < 0$, then d is called a descent direction for f at $\bar{x}$.*

Proof. From the differentiability of f at $\bar{x}$, we have

$$f(\bar{x} + \alpha d) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T d + \alpha ||d|| \circ (\alpha d),$$

where $\lim_{\alpha \to 0} \circ (\alpha d) = 0$, Therefore,

$$\frac{f(\bar{x} + \alpha d) - f(\bar{x})}{\alpha} = \nabla f(\bar{x})^T d + ||d|| \circ (\alpha d).$$

From $\nabla f(\bar{x})^T d < 0$ and $\lim_{\alpha \to 0} \circ (\alpha d) = 0$, it follows that there exists a $\delta > 0$ so that $\nabla f(\bar{x})^T d + ||d|| \circ (\alpha d) < 0$ for all $\alpha \in [0, \delta]$. ■

1.2.1 Necessary Optimality Conditions

First Order Necessary Condition

Theorem 1.2 *[6] Let $f : \mathbb{R}^n \to \mathbb{R}$ a function continuously differentiable at $\bar{x} \in \mathbb{R}^n$. If $\bar{x}$ is a local minimum of the problem (P), then we have*

$$\nabla f(\bar{x}) = F(\bar{x}) = 0.$$

Second Order Necessary Condition

Theorem 1.3 *[6] Let $f : \mathbb{R}^n \to \mathbb{R}$ a function twice differentiable at $\bar{x} \in \mathbb{R}^n$. If $\bar{x}$ is a local minimum of the problem (P), then $F(\bar{x}) = 0$ and the hessian matrix of f at $\bar{x}$ is positive semi definite.*

1.2.2 Sufficient Optimality Conditions

First Order Sufficient Condition

Theorem 1.4 *[6] Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at the point $\bar{x}$ and convex on $\mathbb{R}^n$. If $\nabla f(\bar{x}) = 0$ then $\bar{x}$ is a global minimum of f on $\mathbb{R}^n$.*

Second Order Sufficient Condition

Theorem 1.5 *[6] Let $f : \mathbb{R}^n \to \mathbb{R}$ a function twice differentiable at $\bar{x} \in \mathbb{R}^n$. If $\bar{x}$ is a global minimum of the problem (P), then $F(\bar{x}) = 0$ and the hessian matrix of f at $\bar{x}$ is positive semi definite.*

CHAPTER 2

General Properties of Unconstrained Optimization

2.1 Line Search

There are two main classes of methods that are interested in one-dimensional optimization.

2.1.1 Exact Line Search

As we seek to minimize f, it seems natural to seek to minimize the criterion long of d_k and thus determine the step α_k as the solution of the problem

$$min\ \varphi_k(\alpha) = \min f(x_k + \alpha d_k),\ \ \alpha \geq 0.$$

This is called the Cauchy rule and the step determined by this rule is called Cauchy's step or optimal step.
In some cases, we prefer the smallest stationary point of h_k which decreases this function:

$$\alpha_k = inf\{\alpha \geq 0 : \varphi_k(\alpha) < \varphi_k(0)\}.$$

This is called the Curry Rule and the step determined by this rule is called the Curry Step.
Somewhat imprecisely, these two rules are sometimes referred to as exact linear research.
▶ These two rules are only used in special cases, for example when φ_k is quadratic.
▶ The exact word takes its meaning in the fact that if f is quadratic the solution of the linear search is obtained exactly and in a finite number of iterations.
The disadvantages of exact linear searches
For an arbitrary nonlinear function:

• there may be not found Cauchy's step or Curry'step.
• the determination of these steps generally requires a lot of calculation time and cannot in any case be done with infinite precision.
• the possible additional efficiency of an algorithm by an exact linear search does not, in general, compensate for the time lost in determining such a step.
• convergence results allow other types of rules (inexacte linear search), which are less time consuming.

The Uncertainty Interval

In most modern optimization algorithms, one never does an exact linear search, because finding α_k means that one will have to calculate a large number of times the h_k function and this can be a deterrent from the point of view of the calculation time.
In practice, we are looking for a value of α which ensures a sufficient decrease of f.

Remark 2.1 *We designate $[\alpha_m, \alpha_n]$ as a safety interval if it organizes the values of α in the following order.*

• If $\alpha < \alpha_m$ then α is considered too small.
• If $\alpha_m \leq \alpha \leq \alpha_n$ then α is satisfying.
• If $\alpha > \alpha_n$ then α is considered too large.

Algorithm 1. (basic algorithm)

Algorithm 1. (basic algorithm)
Step 1. $\alpha_m = \alpha_n = 0$, choose $\alpha_1 > 0$, set $k = 1$ and go to step 1.
Step 2.
• If α_k is suitable, put $\alpha^* = \alpha_k$ and stop.
• If α_k too small, we take $\alpha_{m,k+1} = \alpha_k, \alpha_n = \alpha_n$, and go to step 3.
• If α_k too large, we take $\alpha_{n,k+1} = \alpha_k, \alpha_m = \alpha_m$, and go to step 3.
Step 3.
• If $\alpha_{n,k+1} = 0$ determine $\alpha_{k+1} \in]\alpha_{m,k+1}, +\infty[$.
• If $\alpha_{n,k+1} \neq 0$ determine $\alpha_{k+1} \in]\alpha_{m,k+1}, \alpha_{n,k+1}[$.
Replace k with $k+1$ and go to step 2.

2.1.2 Inexact Line Search

We consider the situation that is typical for the application of linear research technique within the multidimensional main method.

The objective of this section is to present the main tests.

Armijo Method [1966]

Armijo method [7] was the first non-exact linear search method, it gives as follows:
Suppose that d_k the descent direction, we can say that α_k verified the Armijo condition, if we have

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \rho \nabla^T f(x_k) d_k, \quad \rho \in]0, 1[,$$

or

$$\varphi(\alpha_k) \leq \varphi(0) + \alpha_k \rho \varphi\prime(0), \quad \rho \in]0, 1[.$$

We can use the following algorithm for the Armijo method.

Algorithm 2. (Armijo rule)
Step 1. $\alpha_{m,1} = \alpha_{n,1} = 0$, choose $\alpha_1 > 0, \rho \in]0, 1[$, set $k = 1$ and go to step 2.
Step 2.
• If $\varphi_k(\alpha_k) \leq \varphi_k(0) + \alpha_k \rho \varphi\prime(0)$, stop($\alpha^* = \alpha_k$).
• If $\varphi_k(\alpha_k) > \varphi_k(0) + \alpha_k \rho \varphi\prime(0)$, then $\alpha_{n,k+1} = \alpha_{n,k}$, $\alpha_{m,k+1} = \alpha_k$ and go to step 3.
Step 3.
• If $\alpha_{n,k+1} = 0$ determine $\alpha_{k+1} \in]\alpha_{m,k+1}, +\infty[$.
• If $\alpha_{n,k+1} \neq 0$ determine $\alpha_{k+1} \in]\alpha_{m,k+1}, \alpha_{n,k+1}[$.
replace k with $k + 1$ and go to step 2.

Theorem 2.1 *Consider the function $\varphi : \mathbb{R}_+^* \to \mathbb{R}$, which is defined as $\varphi_k(\alpha) = f(x_k + \alpha d_k)$. Supposed that φ_k is continuous and bounded below, if d_k is a descent direction at x_k, and $\rho \in]0, 1[$. In that case, the set of steps satisfying the Armijo rule is non empty.*

Proof. See [27]. ■

Goldstein-Price rule (1967)

The primary drawback of the Armijo technique is that, despite being strictly positive, the steps degenerate and eventually approach zero randomly, as a result, the procedure is unable to proceed. There are versions of the procedure that are not affected by this drawback. Based on the so-called Goldstein criteria, there is a criterion that is comparable to Armijo's but allows for the one-step determination of an acceptable α_k.

According to Goldstein conditions, we have

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \rho \nabla^T f(x_k) d_k, \quad \rho \in]0, 1[,$$

$$f(x_k + \alpha_k d_k) \geq f(x_k) + \delta \alpha_k \nabla^T f(x_k) d_k, \quad \delta \in]\varrho, 1[,$$

stated another way

$$\varphi(\alpha_k) \leq \varphi(0) + \alpha_k \rho \varphi\prime(0), \quad \rho \in]0, 1[,$$

$$\varphi(\alpha_k) \geq \varphi(0) + \delta \alpha_k \varphi\prime(0), \quad \delta \in]\varrho, 1[,$$

where ϱ and δ are two reals such us $0 < \varrho < \delta < 1$.

Theorem 2.2 *Consider the function $\varphi : \mathbb{R}_+ \to \mathbb{R}$ is defined as $\varphi_k(\alpha) = f(x_k + \alpha d_k)$, which is continuous and bounded below, if d_k is a descent direction at x_k, and $\delta \in]\varrho, \frac{1}{2}[$. In that case, the set of steps satisfying the Goldstein rule is non empty.*

Wolfe line search(1969)
The conditions of "Goldstein end Price rule" may exclude a minimum which may be a disadvantage. Wolfe conditions ([51]) do not have this disadvantage.
Given two real ρ and σ such as $0 < \rho < \sigma < 1$, these conditions are

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \rho \nabla^T f(x_k) d_k, \tag{2.1}$$

$$\nabla^T f(x_k + \alpha_k d_k) d_k \geq \sigma \nabla^T f(x_k) d_k. \tag{2.2}$$

This means

$$\varphi_k(\alpha) \leq \varphi_k(0) + \alpha_k \rho \varphi\prime(0),$$

$$\varphi\prime_k(\alpha) \geq \sigma \varphi\prime(0).$$

Algorithm 3. (Wolfe rule)
Step 1. $\alpha_{m,1} = \alpha_{n,1} = 0$, choose $\alpha_1 > 0, \rho \in]0,1[, \rho < \sigma < 1$, set $k = 1$ and go to step 2.
Step 2.
• If $\varphi_k(\alpha_k) \leq \varphi_k(0) + \alpha_k \rho \varphi\prime(0)$ and $\varphi\prime_k(\alpha) \geq \sigma\varphi\prime(0)$, stop($\alpha^* = \alpha_k$).
• If $\varphi_k(\alpha_k) \leq \varphi_k(0) + \alpha_k \rho \varphi\prime(0)$,then $\alpha_{n,k+1} = \alpha_k$, $\alpha_{m,k+1} = \alpha_{m,k}$ and go to step 3.
• If $\varphi\prime_k(0)\alpha_k \geq \sigma\varphi\prime(0)\alpha_k$, then $\alpha_{n,k+1} = \alpha_{n,k}$, $\alpha_{m,k+1} = \alpha_k$.
Step 3.
• If $\alpha_{n,k+1} = 0$ determine $\alpha_{k+1} \in]\alpha_{m,k+1}, +\infty[$.
• If $\alpha_{n,k+1} \neq 0$ determine$\alpha_{k+1} \in]\alpha_{m,k+1}, \alpha_{n,k+1}[$.
Replace k with $k + 1$ and go to step 2.

Strong Wolfe line search

For some algorithms (for example the non-linear conjugate gradient) it is sometimes necessary to have a more restrictive condition than (2.2).
We have strong Wolfe conditions

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

and

$$\left|\nabla^T f(x_k + \alpha_k d_k) d_k\right| \leq \sigma \left|\nabla^T f(x_k) d_k\right| \leq -\sigma \nabla^T f(x_k) d_k.$$

The strong Wolfe conditions will therefore be:

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

$$\left|\nabla^T f(x_k + \alpha_k d_k) d_k\right| \leq -\sigma \nabla^T f(x_k) d_k.$$

We can say

$$\varphi_k(\alpha) \leq \varphi_k(0) + \alpha_k \rho \varphi\prime(0),$$

$$\varphi\prime_k(\alpha) \leq -\sigma \varphi\prime(0),$$

where $0 < \rho < \sigma < 1$.

2.1.3 Zoutendijk Theorem

It is said that a linear search rule satisfies the Zoutendijk condition if there is a constant $C > 0$ such that for any index $k \geq 1$ one has

$$f(x_{k+1}) \leq f(x_k) - C \left\|\nabla f(x_k)\right\|^2 \cos^2 \theta_k, \tag{2.3}$$

where θ_k is the angle d_k makes with -$\nabla f(x_k)$, defined by

$$\cos \theta_k = \frac{-\nabla^T f(x_k) d_k}{\|\nabla f(x_k)\| \, \|d_k\|}. \tag{2.4}$$

Proposition 2.1 *If the $\{x_k\}$ sequence generated by an optimization algorithm checks the Zoutendijk condition (2.3) and the $\{f(x_k)\}$ sequence is minimized, then*

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

Proof. By summing the inequality (2.3), we have

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

The series is therefore convergent since there is a constant C' such that for all k, $f(x_k) \geq C'$. ∎

2.2 Descent Direction Methods

The general diagram of a descent direction method is given as follows

$$\begin{cases} x_0 \in \mathbb{R}^n, & \text{given,} \\ x_{x+1} = x_k + \alpha_k d_k, & \text{for all } k \geq 0, \end{cases}$$

where $\alpha_k \in \mathbb{R}_+^*$ the step-size and d_k the descent direction are chosen such that

$$f(x_k + \alpha_k d_k) \leq f(x_k).$$

The step-size and direction descent can be fixed or changed with each iteration.
Now, we define the descent direction.

2.2.1 Descent Direction

Definition 2.1 *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable at $x \in \mathbb{R}^n$. If there exists a vector $d \in \mathbb{R}^n$, such that*

$$\langle \nabla f(x), d_k \rangle < 0,$$

then d_k is called a descent direction of f at x.

Descent Direction Algorithm

algorithm 4. (Descent Direction Method)
Step 1. Suppose $x_0 \in \mathbb{R}^n, \epsilon > 0$ (quite small). Set $k = 0$.
Step 2. Evaluate the descent direction d_k.
Step 3. Determine the step-size factor $\alpha_k > 0$, by the line search.
Step 4. Evaluate new iteration $x_{k+1} = x_k + \alpha_k d_k (k = k + 1)$.
Step 5. Stop criterion
If $\|x_{k+1} - x_k\| < \epsilon$ stop, replace $k = k + 1$ and go to step 2.

2.3 Some Descent Methods

2.3.1 Newton Method

The basic idea of Newton's method for unconstrained optimization is to iteratively use the quadratic approximation $q^{(k)}$ to the objective function f at the current iterate x_k and to minimize the approximation $q^{(k)}$.

Let $f : R^n \to R$ be a twice continuously differentiable, $x_k \in R^n$ and the Hessian $\nabla^2 f(x)$ positive definite. We model f at the current point x_k by the quadratic approximation $q^{(k)}$

$$f(x_{k+s}) \approx q^{(k)}(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s, \tag{2.5}$$

where $s = x - x_k$. The search direction of the Newton method is computed as

$$d_k = -\left[\nabla^2 f(x_k)\right]^{-1} g_k. \tag{2.6}$$

Therefore, the Newton method is defined as

$$x_{k+1} = x_k - \alpha_k [\nabla^2 f(x_k)]^{-1} g_k, \tag{2.7}$$

where α_k is the stepsize. For the last Newton method(2.7), we see that d_k is a descent direction if and only if $[\nabla^2 f(x_k)]^{-1}$ is a positive definite matrix.

2.3.2 Quasi Newton Method

This method is inspired by Newton's algorithm, but without calculating the Hessian matrix of f, nor its inverse. The idea is to replace (2.7) by

$$x_{k+1} = x_k - \lambda_k S_k g_k,$$

where λ_k is a parameter provided by a linear search along the direction $d_k = -S_k g_k$, S_k is a symmetric, positive definite approximation of $\nabla^2 f(x_k)$. Take $f \in C^2(R^n)$, by development of f in a neighborhood of x_k, we find

$$[\nabla^2 f(x)]^{-1}[\nabla f(x) - \nabla f(x_k)] \simeq (x - x_k).$$

If f is quadratic, then the approximations are exact. In particular, $x = x_{k+1}$ and if S_k was a good approximation of Hessian, we should have

$$S_k[\nabla f(x_{k+1}) - \nabla f(x)] = (x_{k+1} - x_k),$$

but since x_{k+1} is evaluated after S_k, it is unlikely that this equation will be satisfied, even approximately. So we impose that S_{k+1} satisfies this equation exactly, hence

$$S_{k+1}[\nabla f(x_{k+1}) - \nabla f(x)] = (x_{k+1} - x_k),$$

this equation is called the quasi-Newton condition.

2.3.3 The Gradient Method

Cauchy introduced the gradient method in 1847, establishing it as one of the simplest and most crucial minimization techniques for unconstrained optimization [48]. Assume that f is continuously differentiable in the vicinity of x_k and the gradient $\nabla f(x_k)$ is not equal to zero.
From the Taylor expansion

$$f(x) = f(x_k) + (x - x_k)^T g_k + \circ \left\| (x - x_k) \right\|,$$

if $x - x_k = \alpha d_k$, then the direction d_k satisfying

$$d_k^T g_k < 0, \quad \text{for all } k,$$

is called a descent direction (descent condition) that is such that $f(x) < f(x_k)$.

Fixing α, it follows that the smaller the value $d_k^T g_k$ is the faster the function value decreases. By the Cauchy-Schwartz inequality

$$\left|d_k^T g_k\right| \leq \|d_k\| \, \|g_k\| \, .$$

We have that the value $d_k^T g_k$ is the smallest if and only if $d_k = -g_k$. Therefore $-g_k$ is the steepest descent direction.

The iterative scheme of the steepest descent method is

$$x_{k+1} = x_k - \alpha_k g_k.$$

2.4 The global convergence of conjugate gradient method and nonparametric estimation

2.4.1 The global convergence

The ability of an optimization algorithm to identify the best solution regardless of the starting point selected is known as global convergence. Stated otherwise, a globally convergent algorithm guarantees the discovery of the best solution, irrespective of the initial starting point.

Types of convergence

In the context of optimization, convergence can take many distinct forms. An optimization algorithm's approach to a solution over time is described by one of numerous convergence modes. Here are some of the most prevalent convergence modes.

Let be $\{x_k\}_{k\in\mathbb{N}}$ a sequence in $\mathbb{R}^n$ that converges to x^*.

▶ If

$$\lim_{k\to+\infty} \sup \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \upsilon \prec 1.$$

Then, we say $\{x_k\}_{k\in\mathbb{N}}$ converges linearly to x^*, with υ as the associated convergence rate.

▶ If

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \to 0 \text{ when } k \to +\infty,$$

then, the convergence is said to be superlinear.

Conjugate gradient method

Conjugate gradient method is an iterative technique for solving optimization problems without constraints. Although it can be extended to non-quadratic functions, its primary application is in the minimization of convex quadratic functions. Conjugate gradient methods are typically applied to very big problems that are difficult to solve with direct approaches.
One important characteristic of the conjugate gradient method consists in the notion that its directions are mutually conjugate. First, let us define the term "conjugate".

Definition 2.2 *Let M be a square symmetric matrix of order n. The directions $d_0, \dots, d_k$ are said to be $M-$conjugate if,*

$$d_i^T M d_j = 0,\ 0 \leq i,\ j \leq k,\ i \neq j.$$

Corollaire 2.1 *Let $d_0, \dots, d_k$ be a system of $M-$conjugate directions (M symmetric and positive definite). Then, $d_0, \dots, d_k$ forms a basis for $\mathbb{R}^n$.*

Linear conjugate gradient method Let's consider the quadratic (linear) problem

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^t M x - b^t x + c,$$

where $M \in \mathbb{R}^n$ is a symmetric and positive definite matrix, $b \in \mathbb{R}^n$ is a vector and $c \in \mathbb{R}$ is a constant. In addition, we have

$$\nabla f(x) = Mx - b \text{ and } \nabla^2 f(x) = M.$$

Note that the method terminates if $\nabla f(x) = 0$. If we define $g_x = \nabla f(x_k)$, the linear conjugate gradient method (for quadratic functions) consists of generating an iterative sequence $\{x_k\}_{k \in \mathbb{N}}$ in the form

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

where $\alpha_k \in \mathbb{R}_+^*$ is the step size (obtained through exact or inexact line search) and $d_k \in \mathbb{R}^n$ is a search direction at iteration k wanted in the form

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

where $g_k = \nabla f(x_k)$ and the coefficients β_k are chosen in such a way that d_k is conjugate with all previous directions, i.e.

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

Therefore, we deduce

$$\begin{aligned} d_{k+1}^T M d_k &= 0 \Longrightarrow (-g_{k+1} + \beta_k d_k)^T M d_k = 0 \\ &\Longrightarrow -g_{k+1}^T M d_k + \beta_k d_k^T M d_k = 0 \\ &\Longrightarrow \beta_k = \frac{g_{k+1}^T M d_k}{d_k^T M d_k}. \end{aligned}$$

So, α_k is an optimal solution to the one-dimensional minimization problem

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

If and only if α_k satisfies $f'(\alpha_k) = 0$. However

$$\begin{aligned} f'(\alpha_k) &= 0 \Longrightarrow d_k^T \nabla f(x_{k+1}) = 0 \\ &\Longrightarrow d_k^T (M x_{k+1} - b) = 0 \\ &\Longrightarrow d_k^T (M(x_k + \alpha_k d_k) - b) = 0, \end{aligned}$$

then,

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

with $g_k = g(x_k) = \nabla f(x_k) = M x_k - b$.

Nonlinear conjugate gradient method An expansion of the conjugate gradient (CG) approach for unconstrained nonlinear optimization problems is the nonlinear conjugate gradient (NCG) method. The nonlinear conjugate gradient approach is intended for more broad objective functions, whereas the linear conjugate gradient method is mainly intended for tackling quadratic optimization issues.

The NCG technique makes use of several conjugate search directions. In order to preserve conjugacy with regard to the nonlinear function's metric, these directions are calculated. Formulas like Polak-Ribière, Fletcher-Reeves and others are frequently used in the search for these conjugate directions. This approach is used to resolve nonlinear optimization issues that are not constrained

$$\min_{x \in \mathbb{R}^n} f(x), \tag{2.8}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a continuously differentiable function.
Any nonlinear conjugate gradient algorithm generates a sequence $\{x_k\}_{k\in\mathbb{N}}$ in the form

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.9}$$

where α_k is the distance to travel along the conjugate search direction obtained through line search and d_k is the search direction calculated by

$$\begin{cases} d_0 = -g_0, & \\ d_{k+1} = -g_{k+1} + \beta_k d_k, & k \geq 1, \end{cases} \tag{2.10}$$

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

The general result of global convergence for the nonlinear conjugate gradient method

The convergence of the nonlinear CG method depends on the properties of the objective function and the chosen line search strategy. Proper initialization and suitable termination criteria are also important for obtaining reliable results.

For optimization strategies that use line search techniques, descent directionality is a basic prerequisite that guarantees advancement toward the best solution.

Theorem 2.3 *Let the sequences $\{d_k\}_{k\geq 0}$ and $\{g_k\}_{k\geq 0}$ be generated by conjugate gradient algorithm. Then, the search direction d_k satisfies the sufficient descent condition, i.e.*

$$g_k^T d_k \leq -c \|g_k\|^2, \text{ for all } k \geq 0, \tag{2.11}$$

where $c > 0$.

We provide the following assumptions about the function f in order to establish the general convergence results of any technique specified by (2.9) and (2.10). These assumptions are essential to guaranteeing the algorithm's convergence.

Assumption A. The level set

$$S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\},$$

is bounded.

Assumption B. In some open convex neighborhood $\mathcal{N}$ of S, the function f is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathcal{N}. \tag{2.12}$$

These assumptions imply that there exists a positive constant $\Gamma \geq 0$ such that

$$\| \nabla f(x) \| \leq \Gamma, \text{for all } x \in \mathcal{N}. \tag{2.13}$$

For unconstrained optimization algorithms in the nonlinear situation, the following Zoutendijk Lemma is required to demonstrate global convergence results [60].

Lemme 2.1 *Suppose that x_0 is a starting point for which assumptions **A** and **B** hold. Consider any method in the form (2.9) and (2.10), where d_k is a descent direction and the step-size α_k satisfies the standard Wolfe conditions (2.1) and (2.2), then*

$$\sum_{k=0}^{\infty} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < \infty. \tag{2.14}$$

It is trivial to deduce from it is easy to get from (2.11) that the Zoutendijk condition (2.14) is equivalent to the following inequality

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \tag{2.15}$$

It is demonstrated by Dai et al. [13] that the following is true for every conjugate gradient method using a strong Wolfe line search.

Theorem 2.4 *Let assumptions **A** and **B** hold. Consider the method (2.9) and (2.10), where d_k is a descent direction and α_k is obtained by the strong Wolfe line search. If*

$$\sum_{k\geq 0} \frac{1}{\|d_k\|^2} = \infty, \tag{2.16}$$

then

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

This Lemma is also necessary to demonstrate the conjugate gradient method's convergence.

Lemme 2.2 *Let assumptions **A** and **B** hold. If d_k is a descent direction and α_k satisfies the strong Wolfe condition. Then*

$$\alpha_k \geq \frac{(1-\sigma)\left|g_k^T d_k\right|}{L\|d_k\|^2}. \tag{2.17}$$

Proof. See the proof of Lemma 3.2 in Liu and Li [36] . ■

Remark 2.2 *According to the assumptions of the beginning of this section and (2.11), it is easy to obtain that* $g_k^T d_k \neq 0$ *for all* $k \geq 0$*. Thus,* $\alpha_k = 0$ *does not satisfy the strong Wolfe condition. This indicates that* α_k *obtained by any conjugate gradient method is not equal to zero, i.e., there exists a constant* $\lambda > 0$, *such that*

$$\alpha_k \geq \lambda, \quad \text{for all } k \geq 0. \tag{2.18}$$

Dai et al. [13] presented an important Theorem that expands upon the Zoutendijk condition and is relevant to universal conjugate gradient algorithms utilizing the strong Wolfe line search (SWLS). Here is how this theorem is presented.

Theorem 2.5 *Suppose that assumptions **A** and **B** hold. Consider any CG method in the form (2.9) and (2.10), in which the steplength* α_k *determined by the strong Wolfe line search. Then, either*

$$\lim_{k\to\infty} \inf \|g_k\| = 0, \tag{2.19}$$

or

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \tag{2.20}$$

Classical conjugate gradient methods

★ Methods where the term $\|g_{k+1}\|^2$ appears in the numerator of β_k

- Fletcher-Reeves (FR) method

The Fletcher-Reeves (FR) approach, in particular, was proposed by Roger Fletcher and Christine M. Reeves in 1964 [23]. Their work helped formalize the conjugate gradient method by introducing a coefficient that measures the conjugacy between successive search directions. This conjugacy was defined to ensure descent along the conjugate direction. The Fletcher-Reeves method's search direction is given by

$$d_{k+1} = -g_{k+1} + \beta_k^{FR} d_k,$$

where β_k^{FR} is defined as a conjugation coefficient that is computed using the FR rule, as follows

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

In 1970, Zoutendijk [60] demonstrated that the FR method associated with exact line search is globally convergent. Powell [45] highlighted the weakness of the Fletcher-Reeves method compared to Zoutendijk's results. Al-Baali [1] extended this result to the strong Wolfe line search (2.1) and (2.3) with $\sigma < \frac{1}{2}$.

- Conjugate Decent (CD) Method

This method proposed by Fletcher in 1987, is defined as follows

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

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

- Dai-Yuan (DY) method

In 1999, Dai and Yuan [15] developed the Dai-Yuan method, β_k is equivalent to

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^t y_k}, \quad y_k = g_{k+1} - g_k.$$

This approach differs significantly from both the CD method and the Fletcher-Reeves method. In both (2.1) and (2.2), the DY technique consistently produces descent directions using the standard Wolfe line search. Furthermore, under extremely general assumptions about the objective function f, it is globally convergent. All that is needed for f to satisfy assumption **B** is for it to be continuously differentiable and for the gradient to be Lipschitzian.

★ **Methods where the term $g_{k+1}^T y_k$ appears in the numerator of β_k**

- Polak-Ribière and Polyak (PRP) Method

The Polak-Ribiére and polyak method was introduced in 1969 by Polak and Ribiére and Polyak (PRP) as a nonlinear conjugate gradient method. It significantly contributed to the theory of unconstrained optimization methods and has become a widely used technique for solving nonlinear optimization problems. The search direction in this method is given by

$$d_{k+1} = -g_{k+1} + \beta_k^{PRP} d_k,$$

where β_k^{PRP} is a conjugation coefficient calculated according to the PRP rule, it is defined as follows

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

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

♦ Powell [45] showed in 1977 that, under the condition that the step size goes to zero, the PRP technique for a nonlinear function f can converge globally. When Lipschitz continuity is assumed and accurate line search is used, this convergence is guaranteed.

♦ Powell demonstrated that the PRP approach can deviate endlessly without coming close to a solution in 1984 by presenting a difficult situation with three variables. This emphasizes how crucial it is that the step size go closer to zero in order to guarantee convergence.

♦ However, in 1997, Dai [12] showed that even in strong convexity and with a small enough $\delta \in]0,1[$, the PRP method can fail by producing an ascending search direction. Later, in 1993, Yuan [57] established the global convergence of the PRP method for strongly convex objective functions, especially when combined with Wolfe's line search.

♦ Even in situations involving exact line search, Dai, Han, Liu, Sun, Yin, and Yuan [13] stressed in their 1999 follow-up research the importance of defining the level set for the PRP method's convergence. As a result, the PRP method's convergence is not certain. Nonetheless, it has demonstrated remarkable efficacy in resolving extensive, unrestricted optimization problems.

Hestenes and Stiefel (HS) Method

The early conjugate gradient methods were given by Michael R. Hestenes and Eduard Stiefel in 1952 set the basis for the creation of the Hestenes-Stiefel (HS) method. These approaches were first designed to effectively handle linear systems but were quickly expanded to unconstrained optimization problems.

The formula for the search direction in the HS method can be represented as follows for a given iteration

$$d_{k+1} = -g_{k+1} + \beta_k^{HS} d_k,$$

where d_k is the search direction at iteration k, g_k is the gradient of the objective function at iteration k and β_k^{HS} is a conjugation coefficient calculated according to the Hestenes-Stiefel rule, is defined as follows

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

where $y_k = g_{k+1} - g_k$, the HS method was developed to converge efficiently, particularly for quadratic objective functions. Nevertheless, a number of factors could effect convergence, therefore, careful parameter adjustment is essential to acquire the best results in different unconstrained optimization settings.

Liu and Storey (LS) Method

In 1991, Jun Liu and Craig Storey introduced the Liu-Storey (LS) method. The method's main goal was to effectively optimize ridge regression models while adhering to a non-negativity requirement. The LS approach has been expanded and used to solve a variety of optimization issues over time.

When optimization can be broken down into univariate subproblems, it is especially well-known for its conceptual simplicity and efficacy. The search direction in this method is given by

$$d_{k+1} = -g_{k+1} + \beta_k^{LS} d_k,$$

where β_k^{LS} is a coefficient is defined as follows

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

where $y_k = g_{k+1} + g_k$. In the context of exact line search, it is observed that the LS method is equivalent to the PRP method. Liu and Storey have examined this approach, demonstrating its global convergence.

2.4.2 Nonparametric estimation

Mode estimation

A sample's "central tendency" can be determined using a number of parameters, such as the mean, median, and mode. The mode is the only option that keeps its definition when the variable of interest changes from a quantitative to a qualitative form, even if the mean is the most frequently chosen option. In real-world data, these measurements can differ dramatically even while they coincide for certain distributions. A fascinating discovery in classical statistics is the actual connection between these parameters for continuous distributions, as shown by Pearson [42].
moyenne - mediane

$$moyenne - mode \approx 3(moyenne - mediane)$$

Although mode estimation first appeared in the 1960s, Eddy [21] points out that issues with density function estimation and interpretability caused it to lose favor. Still, there's a resurgence of inquiry, with writers like Parzen [41] laying the foundation for kernel estimators-based univariate mode estimation. This was extended to multivariate densities by Yamato [55], and asymptotic normality proofs were added by Konakov [35] and Samanta [47]. In this work, we explore different mode estimators thanks to this newfound interest.

Parzen estimator

Given a sample $X_1, ..., X_n$ from an unknown distribution of probability density $f(.)$ bounded and continuous on $\mathbb{R}$. It is assumed that density f has a single mode, represented by θ and specified by

$$f(\theta) = \max_{x \in \mathbb{R}^n} f(x).$$

A kernel estimator of the mode θ is defined as the random variable $\theta_{n,p}$, which maximizes the Rosenblatt-Parzen kernel estimator $f_n(x)$ of $f(x)$, is a kernel estimator of the mode θ

$$f_n\left(\theta_{n,p}\right) = \max_{x\in\mathbb{R}^n} f_n\left(x\right),$$

where

$$f_n\left(x\right) = \frac{1}{nh_n^n}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right).$$

The kernel K represents a p.d.f. on $\mathbb{R}^n$ and the bandwidth h_n is a sequence of positive real numbers that goes to zero as n goes to infinity.

Chernoff estimator

Let a_n be a sequence of positive real numbers decreasing towards zero with n goes to infinity, the Chernoff estimator denoted $\theta_{n,c}$, is defined as being the midpoint of any interval of length $2a_n$ containing the maximum of observations $X_1, \ldots.., X_n$.

Venter estimator

Let b_n be a sequence of natural integers tending towards ∞ when n goes to infinity. The Venter estimator denoted $\theta_{n,v}$, is defined as being the midpoint of the shortest interval containing b_n observations among $X_1, \ldots.., X_n$.

Nonparametric conditional models

Studying the links between two random variables is a very important question in statistics. It often happens that the variable of interest denoted Y is not free to the extent that its realization depends closely on the realization of a second variable X. In this case, we speak of the presence of covariates or explanatory variables. In practice, this case is often the most natural situation, the reason having led to resuming work on the three location parameters. Although the conditional median has become a special case of the study of the conditional quantile, regression is older and more widely used than the conditional mode.

CHAPTER 3

A new hybrid HS-DY conjugate gradient algorithm with application in mode function

Abstract Conjugate gradient methods are an important class of methods for unconstrained optimization, especially for large-scale problems. Recently, they have been much studied. In this paper, a new hybrid conjugate gradient algorithm is proposed and analyzed. The proposed method inherits the features of the HS, DY and NHS conjugate gradient methods. The method can generate the descent direction at every iteration, moreover, this property doesn't depend on any line search. Under the strong Wolfe line search, the global convergence of the proposed method is established. The numerical results also show the feasibility and effectiveness of our algorithm. Furthermore, the proposed algorithm EHD was extended to solve problem of mode function.

The new hybrid HS-DY conjugate gradient method

The optimization model is a needful mathematical problem since it has been connected to different fields such as economics, engineering and physics. Today there are many optimization algorithms, such as Newton, quasi-Newton and bundle algorithms. Note that these algorithms fail to solve large-scale optimization problems because they need to store and calculate relevant matrices. In contrast, conjugate gradient (CG) method is one of iterative techniques prominently used in solving unconstrained optimization problems due to its simplicity, low memory storage and good convergence analysis. In this work, we consider the unconstrained optimization problem

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

where f is continuously differentiable and bounded from below and its gradient $g_k = \nabla f(x_k)$ is available.

Conjugate gradient methods are very important methods for solving (3.1), especially when the dimension n is large. The iterative process of a conjugate gradient method for solving (3.1) is given by

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

where x_k is the current iterate point and d_k is the search direction generated by the following rule

$$d_0 = -g_0;\ d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{3.3}$$

where β_k is a parameter known as the conjugate gradient coefficient. The step-length α_k is very important for global convergence of conjugate gradient methods, one often requires the line search to satisfy the standard Wolfe conditions

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k, \tag{3.4}$$

and

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k. \tag{3.5}$$

Also, the strong Wolfe conditions consist of (3.4) and

$$\left|g_{k+1}^T d_k\right| \leq -\sigma g_k^T d_k. \tag{3.6}$$

where $0 < \delta < \sigma < 1$.
Now, we denote $y_k = g_{k+1} - g_k$, $\|.\|$ the Euclidean norm and $s_k = x_{k+1} - x_k$.
The scalar β_k is chosen so that the methods (3.2) and (3.3) reduces to the linear conjugate gradient method in the case when f is convex quadratic and exact line search, since the gradient are mutually orthogonal and the parameters β_k in these methods are equal. For general nonlinear function, however, a different formula for scalar β_k result in distinct nonlinear conjugate gradient methods. Some of these methods as Polak- Ribière and Polyak (PRP) method [43, 44], Hestenes-Stiefel (HS) method [31] and Liu-Storey (LS) method [37]

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \; \beta_k^{HS} = \frac{g_{k+1}^T y_k}{y_k^T d_k}, \; \beta_k^{LS} = \frac{g_{k+1}^T y_k}{-g_k^T d_k},$$

in general may not be convergent, but they often have better computational performances.
Moreover, although Fletcher-Reeves (FR) method [23], Dai-Yuan (DY) method [15] and Conjugate Decent (CD) proposed by Fletcher [25]

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \; \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{y_k^T d_k}, \; \beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-g_k^T d_k}.$$

These methods have strong convergence properties, but they may not perform well in practice due to jamming [2] and [5].
Naturally, people try to devise some new methods, which have the advantages of these two kinds of methods. Touati-Ahmed and Storey [49] introduced one of the first hybrid conjugate gradient algorithms, where the parameter β_k is computed as

$$\beta_k^{TaS} = \min\left\{\beta_k^{FR}, \; \beta_k^{PRP}\right\}.$$

The authors proved that β_k^{TaS} has good convergence properties and numerically outperforms both the β_k^{FR} and β_k^{PRP} algorithms. Soon afterwards, Hu and Storey [32], Gilbert and Nocedal [26] further studied other hybrid schemes about PRP and FR methods. Dai and Yuan [16] combined DY method with HS method, proposing the following two hybrid methods

$$\beta_k^{hDY} = \max\left\{-c\beta_k^{DY}, \; \min\left\{\beta_k^{HS}, \; \beta_k^{DY}\right\}\right\},$$

$$\beta_k^{hDYz} = \max\left\{0, \; \min\left\{\beta_k^{HS}, \; \beta_k^{DY}\right\}\right\},$$

where $c = \frac{1-\sigma}{1+\sigma}$. For the standard Wolfe conditions (3.4) and (3.5), under the Lipschitz continuity of the gradient, Dai and Yuan [16] established the global convergence of these hybrid computational schemes.
Another hybrid conjugate gradient is a convex combination of the different conjugate gradient algorithms. Recently, Andrei [3] introduced a new hybrid conjugate gradient method based on HS and

DY methods (denoted as HYBRID method) for solving unconstrained optimization problem (3.1), calculating the parameter β_k^c as a convex combination of β_k^{HS} and β_k^{DY} i.e.

$$\beta_k^c = (1-\theta_k)\,\beta_k^{HS} + \theta_k \beta_k^{DY},$$

where θ_k is a scalar parameter satisfying $0 \leq \theta_k \leq 1$. Convergence with the standard Wolfe condition was established. In 2009, this author [5] presented a new hybrid conjugate gradient algorithm between PRP and DY methods (denoted as CCOMB method) with the β_k is obtained by

$$\beta_k^c = (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{DY}.$$

Under the strong Wolfe line search, he proved the global convergence of this method. Recently, Liu and Li [36] proposed another hybrid conjugate gradient method as a convex combination of LS and DY method (denoted as HLSDY method) given by

$$\beta_k^{HLSDY} = (1-\theta_k)\,\beta_k^{LS} + \theta_k \beta_k^{DY}.$$

The global convergence was established under strong Wolfe line search. Numerical result show that the method is efficient for the standard unconstrained problems in a CUTE library [4] .
In 2019, Mtagulwa and Kaelo [40] introduced another hybrid and three-term conjugate gradient method which computes β_k^{EPF} as

$$\beta_k^{EPF} = \begin{cases} \beta_k^{PRP}, & \text{if } \|g_{k+1}\|^2 > \left|g_{k+1}^T g_k\right| \\ (1-\theta_k)\,\beta_k^{NPRP} + \theta_k \beta_k^{FR}, & \text{otherwise} \end{cases},$$

where β_k^{NPRP} given in Zhang [58] by

$$\beta_k^{NPRP} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}\left|g_{k+1}^T g_k\right|}{\|g_k\|^2},$$

and direction d_k defined as

$$d_0 = -g_0;\ d_{k+1} = -\left(1 + \beta_k^{EPF}\frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2}\right) g_{k+1} + \beta_k^{EPF} d_k.$$

The authors proved this method has global convergence under the strong Wolfe line search conditions.
This chapter aims to propose new hybrid conjugate gradient algorithm. We establish, under a strong Wolfe line search, convergence properties of the proposed conjugate gradient method. Numerical results show that the EHD method is efficient and robust and outperforms as seven conjugate gradient methods famous. Finally, an application of our method in nonparametric mode estimator is also considered.

3.1 Modified HS-DY hybrid conjugate gradient method

In this section, we construct a new hybrid conjugate gradient method relating to the HS and DY methods. Dai and Yuan [15] proved that the DY method always generate descent directions and converges globally with the Wolfe line conditions (3.4) and (3.5). On the other hand, the HS method is generally regarded to be one of the most efficient conjugate gradient methods, but their convergence property is not so good.
In the latest years, many works have devoted their time and effort to come up with new formulae in order to increase the efficiency and effectiveness of the DY and HS methods.
Yao et al. [56] gave a variant of the HS method which we call the MHS method. The parameter β_k in the MHS method is given by

$$\beta_k^{MHS} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|} g_{k+1}^T g_k}{y_k^T d_k}.$$

If $\sigma < \frac{1}{3}$ in the strong Wolfe line search (3.6), Yao et al. [56] proved that the MHS method also can produce sufficient descent direction and global convergence. More recently, Zhang [58] took a little modification to the MHS method and constructed the NHS method as follows

$$\beta_k^{NHS} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|} \left|g_{k+1}^T g_k\right|}{y_k^T d_k}.$$

Under the strong Wolfe line search (3.6) with the parameter σ is restricted in $\left(0, \frac{1}{2}\right)$, it has been shown that the NHS method can generate sufficient descent directions and converges globally.
Motivated by the ideas on the hybrid methods [3] and [40], this paper introduce a new hybrid choice for parameter β_k as follows

$$\beta_k^{EHD} = \begin{cases} \beta_k^{HS} \quad , & \text{if } \|g_{k+1}\|^2 > \left|g_{k+1}^T g_k\right|, \\ (1-\theta_k)\,\beta_k^{NHS} + \theta_k \beta_k^{DY} \quad , & \text{otherwise}, \end{cases} \tag{3.7}$$

where θ_k is a scalar parameter satisfying $0 \leq \theta_k \leq 1$ and the direction d_k defined as

$$d_0 = -g_0; \;\; d_{k+1} = -\left(1 + \beta_k^{EHD} \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2}\right) g_{k+1} + \beta_k^{EHD} d_k. \tag{3.8}$$

For convenience, we call this method as EHD method.

3.1.1 The conjugate condition

In conjugate gradient method, the traditional conjugacy condition $d_{k+1}^T y_k = 0$, plays an important role in the convergence analyses and numerical calculation. To select the parameter θ_k we consider the following Lemma.

Lemme 3.1 *If the conjugacy condition $d_{k+1}^T y_k = 0$ is satisfied at every iteration, we get*

$$\theta_k = \frac{\eta - \zeta \beta_k^{NHS}}{\zeta \mu}, \tag{3.9}$$

where $\eta = y_k^T g_{k+1}$, $\zeta = y_k^T d_k - \eta \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2}$ *and* $\mu = \frac{\frac{\|g_{k+1}\|}{\|g_k\|}\left|g_{k+1}^T g_k\right|}{y_k^T d_k}$.

Proof. If $\|g_{k+1}\|^2 \leq \left|g_{k+1}^T g_k\right|$, we have $\beta_k^{EHD} = \beta_k^{NHS} + \theta_k\left(\beta_k^{DY} - \beta_k^{NHS}\right)$, then from (3.8) we get

$$d_{k+1} = -g_{k+1} + \left[\beta_k^{NHS} + \theta_k\left(\beta_k^{DY} - \beta_k^{NHS}\right)\right]\left[d_k - \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2} g_{k+1}\right]. \tag{3.10}$$

We multiply both sides of the relation (3.10) by the vector y_k^T, we obtain

$$\theta_k = \frac{y_k^T g_{k+1} - \beta_k^{NHS}\left[y_k^T d_k - \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2} y_k^T g_{k+1}\right]}{\left(\beta_k^{DY} - \beta_k^{NHS}\right)\left[y_k^T d_k - \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2} y_k^T g_{k+1}\right]}.$$

From the above equality of β_k^{DY} and β_k^{NHS}, after some algebra, we get the result. ■

Remark 3.1 *Having in view the relation* (3.9), *we define*

$$\theta_k = \begin{cases} 0 & \textit{if} \;\; \frac{\eta - \zeta\beta_k^{NHS}}{\zeta\mu} \leq 0 \;\textit{or}\; \zeta\mu = 0, \\ \frac{\eta - \zeta\beta_k^{NHS}}{\zeta\mu} & \textit{if} \;\; 0 < \frac{\eta - \zeta\beta_k^{NHS}}{\zeta\mu} < 1, \\ 1 & \textit{if} \;\; \frac{\eta - \zeta\beta_k^{NHS}}{\zeta\mu} \geq 1. \end{cases} \tag{3.11}$$

3.1.2 EHD Algorithm and the sufficient descent condition

The framework of the proposed EHD algorithm is given as follows

Step 1: Initialization.

Choose an initial point $x_0 \in \mathbb{R}^n$ and the parameters $0 < \delta < \sigma < 1$. Compute $f(x_0)$ and g_0. Set $d_0 = -g_0$.

Step 2: Test for continuation of iterations.

If $\|g_k\|_\infty \leq 10^{-6}$, then stop. Otherwise, go to the next step.

Step 3: Line search.

Compute α_k by the strong Wolfe line searches (3.4) , (3.6) and update the variables $x_{k+1} = x_k + \alpha_k d_k$.

Step 4: Compute θ_k using (3.11).

Step 5: Compute β_k^{EHD} using (3.7).

Step 6: Compute the search direction. If the restart criterion of Powell condition

$$\left|g_{k+1}^T g_k\right| > 0.2 \left\|g_{k+1}\right\|^2, \tag{3.12}$$

is satisfied, then set $d_{k+1} = -g_{k+1}$, otherwise generate d_{k+1} by (3.8) .

Step 7: Set $k = k + 1$ and go to Step 2.

Now, we prove that it generates search direction d_k obtained by new hybrid conjugate gradient method satisfying in some condition the sufficient descent conditions.

Theorem 3.1 *Let the sequences $\{d_k\}_{k\geq 0}$ and $\{g_k\}_{k\geq 0}$ be generated by EHD method. Then the search direction d_k satisfies the sufficient descent for all k*

$$g_k^T d_k = -\left\|g_k\right\|^2. \tag{3.13}$$

Proof. *Multiplying (3.8) by g_{k+1}^T from the left, we get*

$$g_{k+1}^T d_{k+1} = -\left(1 + \beta_k^{EHD} \frac{d_k^T g_{k+1}}{\left\|g_{k+1}\right\|^2}\right) \left\|g_{k+1}\right\|^2 + \beta_k^{EHD} g_{k+1}^T d_k.$$

So, we can get

$$g_{k+1}^T d_{k+1} = -\left\|g_{k+1}\right\|^2.$$

Hence true for $k \geq 1$. The proof is completed. ■

3.2 Global convergence

To analyze the global convergence property of our hybrid method, the following Assumptions are required. These assumptions have been used extensively in the literature for the global convergence analysis of conjugate gradient methods.
Assumption A. The level set

$$S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\},$$

is bounded.
Assumption B. In some open convex neighborhood $\mathcal{N}$ of S, the function f is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathcal{N}. \tag{3.14}$$

These assumptions imply that there exists a positive constant $\Gamma \geq 0$ such that

$$\| \nabla f(x) \| \leq \Gamma, \text{for all } x \in N. \tag{3.15}$$

The following result was essentially proved by Dai et al. [13].

Lemme 3.2 *Let Assumptions **A** and **B** hold. Let the sequence $\{x_k\}_{k\geq 0}$ be generated by* (3.2) *and search direction d_k is a descent direction, and α_k is received from the strong Wolfe line search. If*

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

then

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

The following Lemma gives some interesting properties of the EHD method.

Lemme 3.3 *Let Assumptions **A** and **B** hold. If d_k is a descent direction and α_k satisfies the standard Wolfe condition* (3.5)*. Then*

$$\alpha_k \geq \frac{(1-\sigma) \| g_k \|^2}{L \| d_k \|^2}. \tag{3.16}$$

Proof. *See the proof of Lemma 3.2 in Liu and Li* [36]. ■

Remark 3.2 *From (3.6) and (3.13), the step-size α_k obtained in the EHD algorithm satisfies (3.16). This indicates, the step size α_k obtained in EHD method is not equal to zero, i.e., there exists a constant $\lambda > 0$, such that*

$$\alpha_k \geq \lambda, \quad \text{for all } k \geq 0. \tag{3.17}$$

The following Theorem establishes the global convergence of EHD method with the strong Wolfe line search.

Theorem 3.2 *Suppose that Assumptions **A** and **B** hold. Consider the sequences $\{g_k\}_{k\geq 0}$ and $\{d_k\}_{k\geq 0}$ generated by EHD algorithm. Then this method converges in the sense that*

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{3.18}$$

Proof. *For the sake of contradiction, assume that (3.18) doesn't hold. Then there exists a positive constant γ such that*

$$\|g_k\| \geq \gamma, \quad \text{for all } k. \tag{3.19}$$

We have for the definition of β_k^{NHS} and Cauchy Schwarz inequality, that

$$0 \leq \beta_k^{NHS} \leq \beta_k^{DY}. \tag{3.20}$$

From (3.7) and (3.20), we have

$$\left|\beta_k^{EHD}\right| \leq \left|\beta_k^{HS}\right| + \beta_k^{DY}.$$

For all k sufficiently large. By using (3.6) and from the sufficient descent condition we obtain

$$d_k^T y_k = d_k^T (g_{k+1} - g_k) \geq (1-\sigma) \|g_k\|^2. \tag{3.21}$$

So, using (3.19) we get

$$d_k^T y_k \geq (1-\sigma)\gamma^2. \tag{3.22}$$

On the other hand, using the Cauchy Schwarz inequality, (3.14) and (3.15), we obtain

$$\left|g_{k+1}^T y_k\right| \leq \|g_{k+1}\| \, \|y_k\| \leq \Gamma L D,$$

where D is a diameter of the level set $\mathcal{N}$.

Now we use (3.22)*, we have*

$$\left|\beta_k^{HS}\right| \leq \frac{\Gamma LD}{(1-\sigma)\gamma^2}. \tag{3.23}$$

On the other side,

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} \leq \frac{\Gamma^2}{(1-\sigma)\gamma^2}. \tag{3.24}$$

From (3.23) *and* (3.24) *, we have*

$$\left|\beta_k^{EHD}\right| \leq \frac{\Gamma}{(1-\sigma)\gamma^2}\left(LD+\Gamma\right) = E. \tag{3.25}$$

Thus, it follows from (3.8) *that*

$$\|d_{k+1}\| \leq \| g_{k+1} \| + \left|\beta_k^{EHD}\right| \left(\frac{\left|d_k^T g_{k+1}\right|}{\|g_{k+1}\|} + \frac{\| s_k \|}{\alpha_k} \right).$$

Cauchy Schwarz inequality, (3.17) *and* (3.25) *yields*

$$\|d_{k+1}\| \leq M,$$

where $M = \Gamma + 2E\frac{D}{\lambda}$.
By take the summation $k \geq 0$*, we get*

$$\sum_{k\geq 0} \frac{1}{\|d_k\|^2} = \infty.$$

So, applying Lemma 3.1, we conclude that which impels that (3.18) *is true. This is a contradiction with* (3.19)*, so we have proved* (3.18)*.* ∎

3.3 Numerical Experiments

In this section, we present some numerical experiments obtained with the new proposed conjugate gradient method with the hybridization parameter β_k given by (3.7). The test problems have been taken to the CUTE library [4] and [9] . All the algorithms have been coded in MATLAB 2013 and compiler settings on the PC machine (2.5 GHz, 3.8 GB RAM memory) with windows XP operating system. We compare the computational results of our method (EHD method) against the NHS [58], DY [15], hDYz [16], CCOMB [5], HYBRID [3], HLSDY [36] and CG_DESCENT [30] methods. In this numerical result, all algorithms implement the strong Wolfe line search condition with $\delta = 10^{-4}$ and $\sigma = 10^{-3}$. The iteration is terminated if one of the following conditions is satisfied (i) $\|g_k\|_\infty < 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector, (ii) The number of iterations exceeded 2000, (iii) The computing time is more than 500 s. We show the performance difference clearly between our method EHD and seven conjugate gradient algorithms. We choose the performance profile introduced by Dolan and Morè [17] to compare the performance according to the number of iteration and CPU time with rule as follows. Let S is the set of methods and P is the set of the test problems with n_p , n_s are the number of the test problems and the number of the methods, respectively. For each problem $p \in P$ and solver $s \in S$, denote $\tau_{p,s}$ be the computing time of iteration or CPU time required to solve problems $p \in P$ by solver $s \in S$. Then comparsion between different solvers based on the performance ratio is given by

$$r_{p,s} = \frac{\tau_{p,s}}{\min\left\{\tau_{p,i},\ 1 \le i \le n_s\right\}}.$$

Suppose that a parameter $r_M \ge r_{p,s}$ for all problem and solvers chosen, and $r_M = r_{p,s}$ if and only if solver s does not solve problem p. The overall evaluation of performance of the solvers is then given by the performance profile function given by

$$F_s(t) = \frac{size\left\{p:\ 1 \le p \le n_p,\ r_{p,s} \le t\right\}}{n_p},$$

where $t \ge 1$ and $size\left\{p:\ 1 \le p \le n_p,\ r_{p,s} \le t\right\}$ is the number of elements in the set $\{p:\ 1 \le p \le n_p,\ r_{p,s} \le$ function $F_s : [1, \infty[\ \rightarrow [0, 1]$ is the distribution function for the performance ratio. The value of $F_s(1)$ is the probability that the solver will win the rest of the solvers.

In this numerical study, Table 3.1 lists the names of the test functions and Table 3.2 shows the performance of the eight methods which gives the number of the test problems (N°), the dimension of functions (Dim), the total number of iterations (NI), the CPU time in seconds (CPU) and 'INF' indicates that the algorithm failed to solve the problem. Table 3.2, Figure 3.1 and Figure 3.2 give a performance comparison of the EHD method with those for the number of iterations and the CPU time. From these Figures and Table 3.2, we can see that the new method EHD performs better than NHS [58], DY [15], hDYz [16], CCOMB [5], HYBRID [3], CG_DESCENT [30] and HLSDY [36] methods, for the given test problems. These obtained preliminary results are indeed encouraging.

Number	function	Number	function
1	Beale	21	Himmelbleau
2	Booth	22	Liarwhd
3	Branin	23	Penalty
4	Lion	24	Perquadratic
5	Matyas	25	Power
6	Almost Perturbed Quadratic	26	Qing
7	Almost Perturbed Quartic	27	Quadratic
8	Alpine1	28	Quartic
9	Chung	29	Rastring
10	DIAG	30	Raydan1
11	Diag-aup1	31	Raydan2
12	Diagonal1	32	Ridge
13	Diagonal2	33	Rosenbrock
14	Diagonal4	34	Schwefel
15	Dixon	35	Schwefel220
16	Engval1	36	Schwefel221
17	Exponential	37	Schwefel223
18	Extended Hiebert	38	Styblinski
19	Greinwak	39	Sumsquares
20	Hager	40	Zakharov

Table 3.1: The test functions.

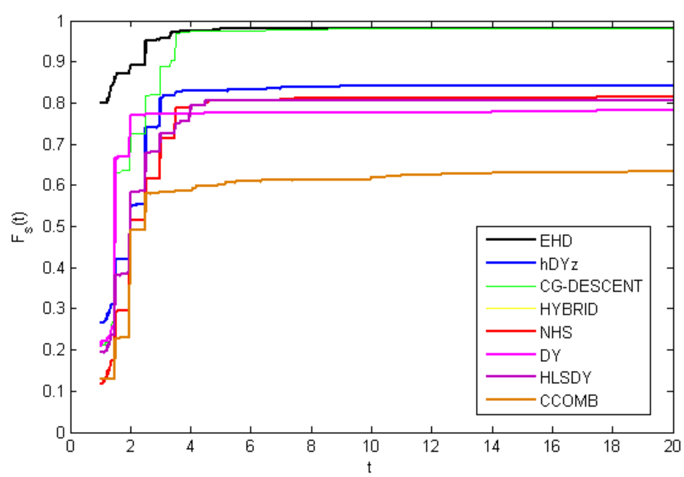

Figure 3.1: Performance profile on the number of iterations.

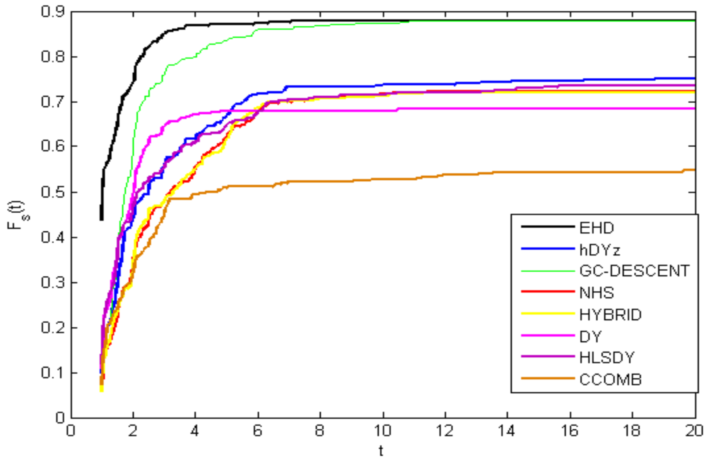

Figure 3.2: Performance profile on the CPU time.

N°	Dim	EHD		hDYz		CG-DESCENT		NHS		HYBRID		DY		HLSDY		CCOMB	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU
1	2	15	0.0780	29	0.1410	10	0.0780	11	0.0630	11	0.0620	10	0.0470	80	0.3600	105	0.5310
2	2	6	0.0310	6	0.0320	8	0.0470	7	0.0320	7	0.0320	6	0.0320	8	0.0470	60	0.3440
3	2	8	0.0470	14	0 .0780	16	0.0780	10	0.0480	10	0.0620	8	0.0620	15	0.0940	88	0.5620
4	2	2	0.0150	2	0.0150	3	0.0160	3	0.0160	3	0.0160	3	0.0160	3	0.0160	5	0.0310
5	2	2	0.0150	2	0.0160	2	0.0160	3	0.0160	3	0.0320	3	0.0160	2	0.0160	2	0.0160
6	70 200 300	10 14 22	0.0310 0.0480 0.0620	840 1999 1999	2.4370 20.9230 30.6070	10 13 25	0.0310 0.0460 0.0670	84 109 194	0.2500 0.8920 2.2540	84 100 190	0.2510 0.8300 2.2020	86 140 190	0.2650 1.1410 2.2120	271 333 1999	0.8130 2.7850 32.9200	647 1999 1999	2.0680 20.7530 33.7580
7	800	2	0.0470	3	0.0940	4	0.0780	3	0.1100	3	0.0930	3	0.0940	3	0.01090	3	0.1120
8	70 300	6 11	0.1590 0.2750	42 I NF	7.4280 INF	5 11	0.1400 0.2780	1999 20	195.5290 1.8090	1999 24	198.0890 1.8200	109 INF	9.8520 INF	8 15	0.2000 0.6440	1999 INF	329.8640 INF
9	100	5	0.0310	5	0.0320	5	0.0310	9	0.0400	9	0.0350	7	0.0870	INF	INF	6	0.0330
10	200	2	0.0150	4	0.0310	5	0.0310	4	0.0160	4	0.0310	3	0.0320	4	0.0160	4	0.0160
11	600	4	0.0470	5	0.1100	6	0.0940	6	0.0930	6	0.0940	4	0.0610	4	0.0620	5	0.0680
12	3000	1	0.0180	2	0.0320	1	0.0220	2	0.0300	2	0.0210	233	5.4850	4	0.0210	1999	20.3680
13	500	2	0.0110	3	0.0160	3	0.0200	4	0.0250	4	0.0250	4	0.0280	2	0.1390	2	0.1050
14	5000	6	0.1410	2	0.0990	4	0.1110	6	0.3760	5	0.3670	3	0.1970	4	0.2030	3	0.4100
15	2000	4	0.0160	4	0.0820	4	0.0510	4	0.0205	4	0.2080	3	0.0320	5	0.0160	5	0.0310
16	50	2	0.0150	5	0.0770	5	0.0360	5	0.0160	5	0.0160	3	0.0160	7	0.0980	1999	28.2610
17	3000	3	0.1250	1999	28.3400	5	0.1400	3	0.1270	3	0.1260	4	0.1880	5	0.1410	4	0.1460
18	120	3	0.0150	5	0.0310	6	0.0320	5	0.0310	5	0.0310	4	0.0160	4	0.0460	80	1.2810
19	1000	1	0.0160	1	0.0630	1	0.0470	1	0.0460	1	0.0470	1	0.0460	2	0.0310	2	0.0160
20	2000	2	0.0150	4	0.0310	4	0.0160	3	0.0160	3	0.0310	3	0.0310	4	0.0320	5	0.0620

Table 3.2: Numerical results of the eight methods.

N°	Dim	EHD		hDYz		CG-DESCENT		NHS		HYBRID		DY		HLSDY		CCOMB	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU
21	200	3	0.0160	6	0.0780	4	0.0470	6	0.0630	7	0.0940	4	0.0310	4	0.0310	6	0.0320
22	80	2	0.0150	5	0.0310	7	0.0310	8	0.0320	7	0.0320	3	0.0160	7	0.0310	4	0.0160
23	20 2000	2 2	0.0630 0.0620	5 5	0.1470 0.1230	2 2	0. 0680 0.1090	7 7	0.1560 0.3130	7 6	0.1600 0.2800	12 11	0.2810 0.6730	INF INF	INF INF	5 5	0.0940 0.1420
24	200 400	25 18	0.0320 0.0470	INF INF	INF INF	20 15	0.0310 0.0430	273 428	1.2960 3.7650	276 425	1.3280 3.8270	755 INF	4.2650 INF	261 390	1.2650 3.4990	1999 1999	17.5680 31.4430
25	10 500	23 4	0.0160 0.0150	8 1999	0.0150 42.3770	24 5	0.0310 0.0310	8 1999	0.0150 41.9810	8 1999	0.0150 41.8280	11 INF	0.0160 INF	INF 10	INF 0.0310	59 1999	0.9690 41.8080
26	150	3	0.0150	6	0.0310	4	0.0160	8	0.0320	8	0.0310	4	0.0160	9	0.0470	5	0.0310
27	200 1000	15 20	0.0480 0.0630	INF INF	INF INF	12 18	0.0470 0.0620	305 389	4.6080 8.2010	319 367	4.6600 8.2480	323 390	4.9550 8.3900	296 381	4.7960 8.0190	1431 1999	31.3520 66.7380
28	800	2	0.0310	4	0.0470	2	0.5620	5	0.0470	5	0.0460	3	0.0320	5	0.0470	4	0.0320
29	200	9	0.0780	215	2.0620	6	0.2030	6	0.0460	6	0.0480	9	0.0630	INF	INF	35	0.3280
30	20 1000	5 9	0.0160 0.0310	8 30	0.0310 0.0470	9 8	0.0310 0.0320	35 1999	0.5270 720.4940	35 INF	0.3510 INF	35 INF	0.5310 INF	56 INF	0.5930 INF	84 1999	0.9690 631.3020
31	5000	5	0.1100	8	0.0940	4	0.0780	8	0.0940	9	0.0970	31	4.1930	7	0.0720	653	109.3600
32	400	2	0.0620	2	0.1090	2	0.0630	2	0.0940	2	0.0930	1030	41.6090	146	0.4220	1999	22.3730
33	10	7	0.0160	6	0.0160	6	0.0930	7	0.0310	7	0.0320	82	0.0780	86	0.0940	7	0.0160
34	40	11	0.0470	3	0.0160	10	0.0160	3	0.0160	3	0.0160	4	0.0420	INF	INF	1999	71.3760
35	2000	2	0.0310	2	0.0470	3	0.0940	3	0.1400	3	0.0780	3	0.0620	3	0.0670	1999	199.9170
36	2000	3	0.0310	3	0.1860	2	0.0180	4	0.0470	4	0.0470	3	0.0940	3	0.0320	1999	42.4450
37	500	2	0.0150	2	0.0160	2	0.0320	3	0.0310	3	0.0160	3	0.0310	3	0.0160	3	0.0160
38	5000	4	0.2829	14	3.4680	5	0.2890	15	2.4210	15	2.4840	62	18.8240	37	10.0450	5	0.3680
39	200	226	1.5460	553	5.1890	224	1.4550	158	1.4210	159	1.6710	230	1.5840	810	9.2390	INF	INF
40	30	6	0.0150	6	0.0160	6	4.7490	8	0.0320	8	0.0310	13	0.0460	7	0.0320	6	0.0160

Table 3.2: (Continued).

3.4 Application in mode function

The conjugate gradient method has played an important role in solving large scale unconstrained optimization problems that may arise in statistics nonparametric [39], portfolio selection [8] and image restoration problems [38].
Estimation nonparametric has received a great deal of attention in both theoretical and applied statistics literature. For the historical and mathematical survey, we refer the reader to Sager [46]. In statistics, it is always interesting to study the central tendency of the data, that is usually quantified using the location parameters (mean, mode, median). The problem of estimating the mode function of a probability density function (p.d.f.) has taken considerable attention in the past for both independent and dependent data, and a number of distinguished papers deal with this topic. For example, Parzen [41] and Eddy [21] for estimation of the unconditional mode in the independent and identically distributed (i.i.d.) case.
In this section, we consider the problem of estimating the mode of a multivariate unimodal probability density f with support in $\mathbb{R}^n$ from i.i.d. standard normal random variables $X_1, ..., X_n$ with common probability density function f. This problem has been investigated in numerous paper. To quote a few of them, Konakov [35] and Samanta [47]. We assume that density f has an unique mode denoted by θ and defined by

$$f(\theta) = \max_{x\in\mathbb{R}^n} f(x). \tag{3.26}$$

A kernel estimator of the mode θ is defined as the random variable $\hat{\theta}$ which maximizer the kernel estimator $f_n(x)$ of $f(x)$, that is

$$f_n\left(\hat{\theta}\right) = \max_{x\in\mathbb{R}^n} f_n(x), \tag{3.27}$$

where

$$f_n(x) = \frac{1}{nh_n^n}\sum_{i=1}^{n} K\left(\frac{x-X_i}{h_n}\right). \tag{3.28}$$

The bandwidth (h_n) is a sequence of positive real numbers which goes to zero as n goes to infinity and the kernel K is a p.d.f. on $\mathbb{R}^n$.
In this simulation, we choose between two different types of kernel: while standard Gaussian kernel defined by

$$K\left(x\right)=\frac{1}{\left(2\pi\right)^{\frac{n}{2}}}\exp\left(-\frac{1}{2}\sum_{j=1}^{n}x_j^2\right),$$

and Epanechnikov kernel obtained by

$$K\left(x\right)=\left(\frac{3}{4}\right)^{n}\prod_{j=1}^{n}\left(1-x_j^2\right).$$

The selection of the bandwidth h is an important and basic problem in kernel smothing techniques. In this simulation, we choose the optimal bandwidth by the cross-validation method.
In this context, we employ our proposed method to solve the problem (3.27) under strong Wolfe line search technique and compare the computational results of the EHD method against the CG_DESCENT method [30]. We choose some initial points and we obtain the result as in the Table 3.3 According to these results, it is clear that the EHD method more efficient than CG_DESCENT method based on the number of iterations and CPU time for solving the problem (3.27).

$STAT_page - 0001$

Kernel	Point initial	Dim	EHD		GC-DESCENT	
			NI	CPU	NI	CPU
Gaussian	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		120	8	10.5020	27	18.8140
		200	2	7.1660	3	8.9070
	(0.025,...,0.025)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		50	17	3.7430	70	6.6610
		250	5	[illegible]	1	[illegible]
		400	2	16.5270	3	23.0360
	(-1.01,...,-1.01)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Epanechnikov	(0.25,...,0.25)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		160	5	17.1650	3	5.7560
		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	(-0.45,...,-0.45)	45	4	[illegible]	[illegible]	[illegible]
		120	2	2.5300	4	4.0630
		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		70	16	6.6420	2	0.0260
		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		120	3	3.9400	19	23.0000
		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	(0.005,...,0.005)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Table 3.3: The simulation result of EHD and GC-DESCENT methods for solving problem (3.27).

$5.png$

Conclusion

We have presented a new hybrid conjugate gradient algorithm. The proposed method possesses a good descent search direction at each iteration and this is independent of the line search. The global convergence properties of the proposed method have been established under strong Wolfe line search conditions. We present the computational evidence that the performance of our method EHD is better than to some well-known conjugate gradient methods. The practical applicability of our method is also explored in nonparametric estimation of the mode function.

CHAPTER 4

Two modified conjugate gradient method for unconstrained optimization problems with application in mode function

In recent years, much effort has been placed by many researchers, on the construction of some modification of the conjugate gradient methods combined with good numerical performance and that has global convergence properties.

Based on the DY conjugate gradient method, Huang [33] proposed a new conjugate gradient formula, which is the same as the DY formula with the exact line search. Under the previous formula, a new conjugate gradient algorithm with the Wolfe line search was proposed, where β_k is given by

$$\beta_k^{MDY} = \frac{\|g_{k+1}\|^2 - \frac{(g_{k+1}^T d_k)^2}{\|d_k\|^2}}{y_k^T d_k}.$$

In 2012, Dai and Wen [14] gave a variant of the HS method, such that β_k defined by

$$\beta_k^{DHS} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}|g_{k+1}^T g_k|}{y_k^T d_k + \mu |g_{k+1}^T d_k|}.$$

Under the standard Wolfe line search, Dai and Wen [14] proved that the DHS method possesses sufficient descent property and global convergence.

Recently, Du et al. [20] proposed a two modified conjugate gradient methods, denoted by VLS* and NVLS*.

The parameter β_k in VLS* method is given by

$$\beta_k^{VLS^*} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}|g_{k+1}^T g_k|}{-g_k^T d_k},$$

and the scalar β_k in NVLS* method is defined as

$$\beta_k^{NVLS^*} = \frac{\|g_{k+1}\|^2 - \frac{|g_{k+1}^T g_k|}{\|g_k\|^2} g_{k+1}^T g_k}{-g_k^T d_k}.$$

The authors proved that VLS* and NVLS* methods possess the sufficient descent conditions and global convergence property by using the strong Wolfe line search with $\sigma < 1$ and $0 < \sigma < \frac{1}{2}$, respectively.

In 2020, Zhu et al. [59] proposed a modified DY conjugate gradient method (called DDY1 method) where the β_k^1 is defined by

$$\beta_k^1(\mu) = \begin{cases} \dfrac{\|g_{k+1}\|^2 - \frac{\mu\left(g_{k+1}^T d_k\right)^2}{\|g_{k+1}\|\|g_k\|\|d_k\|^2} g_{k+1}^T g_k}{y_k^T d_k} & if \qquad g_{k+1}^T g_k \geq 0, \\ \dfrac{\|g_{k+1}\|^2 + \frac{\mu\left(g_{k+1}^T d_k\right)^2}{\|g_{k+1}\|\|g_k\|\|d_k\|^2} g_{k+1}^T g_k}{y_k^T d_k} & if \qquad g_{k+1}^T g_k < 0. \end{cases}$$

By using the standard Wolfe line search, the modified DY method satisfies the global convergence and the sufficient descent direction where $0 \leq \mu \leq 1$.

According to the above ideas, we introduce the two novels $\boldsymbol{\beta}_k$ which are defined as β_k^{OCB1} and β_k^{OCB2}.

First, we present the OCB1 method by the formula

$$OCB1 : d_{k+1} = -g_{k+1} + \beta_k^{OCB1} d_k; d_0 = -g_0, \tag{4.1}$$

where β_k^{OCB1} is written as follows

$$\beta_k^{OCB1} = \frac{\|g_{k+1}\|^2 - \frac{(g_{k+1}^T d_k)^2}{\|d_k\|^2}}{|g_k^T d_k| + \mu_1 |g_{k+1}^T d_k|}, \qquad \mu_1 > 1 \ . \tag{4.2}$$

We determine the OCB2 method by

$$OCB2 : d_{k+1} = -g_{k+1} + \beta_k^{OCB2} d_k; d_0 = -g_0, \tag{4.3}$$

where the formula of β_k^{OCB2} is given by

$$\beta_k^{OCB2} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|} |g_{k+1}^T g_k|}{\mu_2 |g_{k+1}^T d_k|}, \qquad \mu_2 > 1 \ . \tag{4.4}$$

The most important and new thing in this work is the application of our methods in non-parametric statistics, where we are the first to use and access in this field.
The aim of this chapter is to propose two new conjugate gradient methods. By using the strong Wolfe line search, we establish the convergence properties of our new methods. Numerical results show that these methods are efficient and robust compared by five conjugate gradient methods famous. Finally, an application of our methods in non-parametric mode estimator is also considered.

4.1 Convergence properties of OCB1

4.1.1 The OCB1 algorithm

The framework of the OCB1 algorithm is given as follows

Step 1. Given an initial point $x_0 \in \mathbb{R}^n$ and the parameters $0 < \delta < \sigma < 1$. Compute $f(x_0)$ and g_0. Set $d_0 = -g_0$.
Step 2. If $\|g_k\| \leq 10^{-6}$, then stop.
Step 3. Compute α_k by the strong Wolfe line searches (3.4) and (3.6) and update the variables $x_{k+1} = x_k + \alpha_k d_k$.
Step 4. Compute β_k by the formula (4.2).
Step 5. Compute the search direction by the formula (4.1).
Step 6. Set $k = k + 1$ and go to step 2.

4.1.2 The sufficient descent direction

In this subsection, we demonstrate that the OCB1 method satisfies the sufficient descent condition, we immediately obtain the following result.
Theorem 4.1: Let the sequences $\{d_k\}_{k\geq 0}$ and $\{g_k\}_{k\geq 0}$ be generated by OCB1 algorithm. Then the search direction d_k satisfies the sufficient descent condition, i.e.

$$g_k^T d_k \leq -c_1 \|g_k\|^2, \ \forall \ k \geq 0. \tag{4.5}$$

Proof: The following proof is by induction. For $k = 0$, it holds $d_0 = -g_0$ then $g_0^T d_0 \leq -\|g_0\|^2$, we conclude that the sufficient descent condition holds for $k = 0$. Now, we assume that (4.5) holds for k and prove that for $k + 1$.

By multiplying (4.1) by g_{k+1}^T from the left and replacing β_k^{OCB1} by the formula (4.2), we obtain

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2 - \frac{(g_{k+1}^T d_k)^2}{\|d_k\|^2}}{|g_k^T d_k| + \mu_1 |g_{k+1}^T d_k|} g_{k+1}^T d_k. \tag{4.6}$$

Based on the definition of β_k^{OCB1} and Cauchy Schwarz inequality, we have

$$0 \leq \beta_k^{OCB1} \leq \frac{1}{\mu_1} \frac{\|g_{k+1}\|^2}{|g_{k+1}^T d_k|}. \tag{4.7}$$

Now, from (4.6) and (4.7), we get

$$g_{k+1}^T d_{k+1} \leq -c_1 \|g_{k+1}\|^2, \tag{4.8}$$

where $c = 1 - \frac{1}{\mu_1}$. The proof is now complete.

4.1.3 The global convergence

Now, we establish the convergence results and the following assumptions are necessary in the analysis of the global convergence properties of the conjugate gradient methods.

Assumption A. The level set

$$S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\},$$

.

is bonded.

Assumption B. In some open convex neighborhood $\mathcal{N}$ of S, the function f is continuously differentiable and its gradient $g_k = \nabla f(x_k)$ is Lipschitz continuous, namely, there exists a constant $L > 0$, such that:

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathcal{N}. \tag{4.9}$$

this assumption imply that there exists a positive constant $\Gamma \geq 0$, such that

$$\| \nabla f(x) \| \leq \Gamma, \text{ for all } x \in \mathcal{N}. \tag{4.10}$$

Dai et al [13] proved the following result, which is necessary in the analysis of global convergence of OCB1 method.

Lemma 4.1: Let Assumptions **A** and **B** hold. Consider the method (3.2) and (3.3), where d_k is a descent direction, and α_k is obtained by the strong Wolfe line search. If

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

then

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

Now, we need also this Lemma to prove the convergence of our methods.

Lemma 4.2: Let assumptions **A** and **B** hold. If d_k is a descent direction and α_k satisfies the standard Wolfe condition (3.5). Then

$$\alpha_k \geq \frac{(1-\sigma) \parallel g_k^T d_k \parallel^2}{L \parallel d_k \parallel^2}. \tag{4.11}$$

Proof: See the proof of Lemma 3.2 in Liu and Li [36].

Remark 4.1: From (3.6), (4.5) and (4.9), the stepsize α_k obtained in the OCB1 and OCB2 algorithms satisfies (4.11). This indicates that the step size α_k obtained in the OCB1 and OCB2 methods is not equal to zero, i.e., there exists a constant $\lambda > 0$, such that

$$\alpha_k \geq \lambda \ \ , \text{for all } k \geq 0. \tag{4.12}$$

The following Theorem establishes the global convergence of our new method OCB1 with the strong Wolfe line search.

Theorem 4.2: Suppose that assumptions **A** and **B** hold and consider the iterative method in the form (3.2) and (3.3). Let $\{x_k\}_{k\geqslant 0}$ be generated by OCB1 algorithm, where α_k satisfies the strong Wolfe conditions, then the following condition holds

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{4.13}$$

Proof: We prove by contradiction. Suppose that (4.13) doesn't hold, there exists a constant γ such that

$$\|g_k\| \geq \gamma \quad \text{, for all } k. \tag{4.14}$$

From (4.2), we get

$$\beta_k^{OCB1} \leq \frac{\|g_{k+1}\|^2}{|g_k^T d_k|}.$$

Now, by using (4.5), (4.10) and (4.14), we obtain

$$\beta_k^{OCB1} \leq \frac{\Gamma^2}{c_1 \gamma^2} = E. \tag{4.15}$$

Thus, it follows from (4.1), (4.10), (4.12) and (4.15) that

$$\|d_{k+1}\| \leq \| g_{k+1} \| + \beta_k^{OCB1} \frac{\|x_{k+1} - x_k\|}{\alpha_k} \leqslant M,$$

where

$$M = \Gamma + E\frac{D}{\lambda},$$

and D is a diameter of the level set $\mathcal{N}$.
By take the summation $k \geq 0$, we get

$$\sum_{k \geq 0} \frac{1}{\|d_k\|^2} = \infty.$$

So, applying Lemma 2.1, we conclude that which impels that (4.13) is true. This is a contradiction with (4.14), so we have proved (4.13). ∎

4.2 Convergence properties of OCB2

4.2.1 The OCB2 algorithm

The steps of the OCB2 algorithm are the same as those of OCB1 algorithm, but step 4 is replaced by: Compute β_k by the formula (4.4).
Step 5 is replaced by: Compute the search direction by the formula (4.3).

4.2.2 The sufficient descent condition

Now, we prove that the search direction d_k defined by (4.3) satisfying the sufficient descent condition with the strong Wolfe line search.

Theorem 4.3: Suppose that Assumptions **A** and **B** hold and let the sequences $\{d_k\}_{k\geq 0}$ and $\{g_k\}_{k\geq 0}$ be generated by OCB2 algorithm. For all $k \geq 0$, then we have

$$g_k^T d_k \leq -c_2 \left\| g_k \right\|^2. \tag{4.16}$$

Proof: The following proof is by induction. For $k = 0$, then $g_0^T d_0 \leq -\left\| g_0 \right\|^2$, so condition (3.1) holds true for $k = 0$.
Now let us suppose that (4.16) holds for k and prove that for $k+1$.
We multiply (4.3) by g_{k+1}^T from the left, we get

$$g_{k+1}^T d_{k+1} = -\left\| g_{k+1} \right\|^2 + \frac{\left\| g_{k+1} \right\|^2 - \frac{\| g_{k+1} \|}{\| g_k \|} |g_{k+1}^T g_k|}{\mu_2 |g_k^T d_k|} g_{k+1}^T d_k. \tag{4.17}$$

On the other hand, we first need to simplify β_k^{OCB2} by using the Cauchy Schwarz inequality as follows

$$0 \leq \beta_k^{OCB2} \leq \frac{\left\| g_{k+1} \right\|^2}{\mu_2 |g_k^T d_k|}. \tag{4.18}$$

Now, by using (3.6), (4.17) and (4.18), we can obtain

$$g_{k+1}^T d_{k+1} \leq -c_2 \left\| g_{k+1} \right\|^2,$$

where $c_2 = \left(1 - \frac{\sigma}{\mu_2}\right)$. Hence, **Theorem 4.3** is proved. ■

4.2.3 The global convergence

In order to prove the global convergence, we have the following Lemma, which proven by Zoutendijk [60].

Lemma 4.3: Suppose that Assumptions **A** and **B** hold. Consider the iterative method in the form (3.2) and (3.3), where d_k is a descent direction and α_k satisfies the standard Wolfe conditions, then the Zoutendijk condition

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty \tag{4.19}$$

holds.

Obviously, if the sufficient descent condition (4.16) is satisfied, then the Zoutendijk condition (4.19) implies that

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < +\infty. \tag{4.20}$$

Now, we can establish the global convergence result of OCB2 method by using the following result.

Theorem 4.4: Suppose that Assumptions **A** and **B** hold. Consider any CG method in the form (3.2) and (3.3), with the parameter $\beta_k = \beta_k^{OCB2}$, in which the step length α_k is determined to satisfy the SWLS condition (3.4) and (3.6), where d_k is a descent search direction. Then this method converges in the sense that

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{4.21}$$

Proof: To prove Theorem 3.2, we use contradiction. That is, if equation (4.21) is not true, then we can find a positive constant r, such that

$$\|g_k\| \geq r \quad , \text{for all } k. \tag{4.22}$$

From the formulas (4.10), (4.16), (4.18) and (4.22), we get

$$\beta_k^{OCB2} \leq P, \tag{4.23}$$

where $P = \frac{\Gamma^2}{c_2 r^2}$.

So, by using (4.3), (4.10), (4.12) and (4.23), we find

$$\|d_{k+1}\| \leq \| g_{k+1} \| + \beta_k^{OCB2} \frac{\|x_{k+1} - x_k\|}{|\alpha_k|} \leqslant M_1,$$

where

$$M_1 = \Gamma + P\frac{D}{\lambda},$$

and D is a diameter of the level set $\mathcal{N}$.
By take the summation $k \geq 0$, we get

$$\sum_{k\geq 0} \frac{1}{\|d_{k+1}\|^2} = +\infty. \tag{4.24}$$

On the other side, according to (4.20) and (4.22), we get that

$$r^2 \sum_{k\geq 0} \frac{1}{\|d_k\|^2} \leqslant \sum_{k\geq 0} \frac{\| g_k \|^4}{\|d_k\|^2} < +\infty,$$

which contradicts with (4.24). Consequently, (4.22) does not hold, and the the equality (4.21) is proven. ■

4.3 Numerical Experiments

In this section, in order to evaluate the efficiency of our two new methods, we present some numerical experiments. In this numerical study, we used 40 test problems have been taken to the CUTE library [4] and [9] collection. We let the OCB1 and OCB2 methods be compared with the NVLS*[20], DDY1[16], DHS[14], MDY[33] and CD[25]. All the algorithms have been coded in MATLAB 2013 and compiler settings on the PC machine (2.5 GHz, 3.8 GB RAM) with windows XP operating system. In this numerical results, all algorithms implement the strong Wolfe line search condition with $\delta = 10^{-3}$ and $\sigma = 10^{-1}$. The iteration is terminated if one of the following conditions is satisfied (i) $\|g_k\|_\infty < 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector, (ii) The number of iterations exceeded 2000, (iii) The computing time is more than 500s. We show the performance difference clearly between our new methods and five conjugate gradient algorithms famous.
We use the performance profile introduced by Dolan and Morè [17], to compare the performance according to the number of iterations and CPU time with the rule as follows. Let S is the set of methods and P is the set of the test problems with n_p , n_s are the number of the test problems and the number of the methods, respectively. For each problem $p \in P$ and solver $s \in S$, denote

$\tau_{p,s}$ be the number of iteration or CPU time required to solve problems $p \in P$ by solver $s \in S$. Then comparsion between different solvers based on the performance ratio is given by

$$r_{p,s} = \frac{\tau_{p,s}}{\min\left\{\tau_{p,i},\ 1 \leq i \leq n_s\right\}}.$$

Suppose that a parameter $r_M \geq r_{p,s}$ for all problem and solvers chosen, and $r_M = r_{p,s}$ if and only if solver S does not solve problem p. The overall evaluation of performance of the solvers is then given by the performance profile function given by

$$F_s\left(t\right) = \frac{size\left\{p:\ 1 \leq p \leq n_p,\ r_{p,s} \leq t\right\}}{n_p},$$

where $t \geq 1$ and $size\left\{p:\ 1 \leq p \leq n_p,\ r_{p,s} \leq t\right\}$ is the number of elements in the set $\{p:\ 1 \leq p \leq n_p,\ r_{p,s} \leq$ This function $F_s : [1, \infty[\rightarrow [0, 1]$ is the distribution function for the performance ratio. The value of $F_s\left(1\right)$ is the probability that the solver will win the rest of the solvers.

In this numerical study, **Table 4.1** lists the names of the test functions. **Table 4.2** shows the performance of the seven methods which gives the number of the test problems (**N°**), the dimension of functions (**Dim**), the total number of iterations (**NI**) and the CPU time in seconds (**CPU**) and 'INF' indicates that the algorithm failed to solve the problem.

The Figure 4.1 and Figure 4.2 give a performance comparison of our new methods with the other methods which is based on the number of iterations and the CPU time, respectively. From these Figures and Table 4.2, we can see that the new methods perform better than NVLS* [20], DDY1 [16], DHS [14], MDY [33] and CD [25] methods for the given test problems. These preliminary results are actually positive.

Number	Function	Number	Function
1	BEALE	21	NONSCOMP
2	BOOTH	22	Penalty
3	BRANIN	23	Perquadratic
4	LION	24	Powe
5	Almost Perturbed Quartic	25	Qing
6	Chung	26	Quadratic
7	CUBE	27	Quartic
8	DIAG	28	QUARTICM
9	DIAG-AUP1	29	Rastring
10	Diagonal1	30	Raydan1
11	Diagonal4	31	Rosenbrock
12	DIXON	32	SCHWEFEL
13	ENGVAL1	33	Schwefel221
14	Exponential	34	Schwefel223
15	FLETCHER	35	Staircases1
16	Hager	36	Staircases2
17	Himmelbleau	37	Staircases3
18	LIARWHD	38	Sumsquares
19	Linear perturbed	39	TRIDIA
20	NONDIA	40	TR-Sumofquadratics

Table 4.1: The test functions.

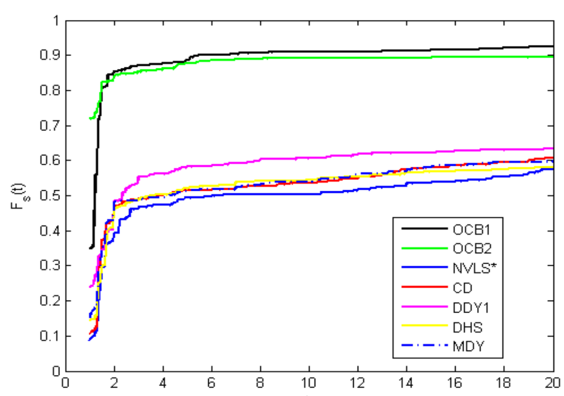

Figure 4.1: Performance profile on the number of iterations.

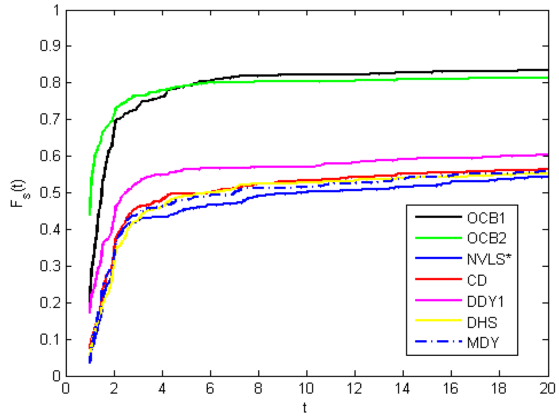

Figure 4.2: Performance profile on the CPU time.

N°	Dim	OCB1		OCB2		NVLS*		DDY1		DHS		MDY		CD	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU
1	2	5	0.0310	5	0.0310	INF	INF	59	0.1730	INF	INF	53	0.3440	46	0.2970
2	2	7	0.0470	6	0.0470	16	0.0780	28	0.1250	18	0.0780	25	0.0780	21	0.0620
3	2	5	0.0310	4	0.0150	INF	INF	5	0.0310	INF	INF	47	0.2030	65	0.2970
4	2	5	0.0310	4	0.0310	125	0.5940	4	0.0310	674	3.2670	1004	6.4360	6	0.0320
5	10000	3	1.0470	3	1.0310	3	1.0770	3	1.0780	3	1.0940	3	1.0630	3	1.1710
6	1200	4	0.7190	4	0.7340	70	3.9990	32	1.9990	99	5.5150	99	5.0460	54	2.9520
	1500	4	1.0940	4	1.0780	106	8.1080	26	2.3740	103	7.7180	103	7.7750	48	3.8430
7	100	4	0.0150	3	0.0160	INF	INF	INF	INF	1999	19.0020	INF	INF	INF	INF
	500	4	0.0310	3	0.0320	INF	INF	INF	INF	1999	37.4610	INF	INF	INF	INF
	800	4	0.0960	3	0.780	INF	INF	INF	INF	1999	84.8660	INF	INF	INF	INF
	1000	4	0.1720	3	0.1720	INF	INF	INF	INF	1999	160.126	INF	INF	INF	INF
	1500	4	0.2660	3	0.2500	INF	INF	INF	INF	1999	235.708	INF	INF	INF	INF
8	800	3	0.0630	3	0.0620	4	0.0780	3	0.0620	4	0.0780	4	0.0780	4	0.0780
9	300	4	0.0630	3	0.0310	4	0.0630	3	0.0470	5	0.0630	5	0.0470	4	0.0630
	600	4	0.0620	3	0.0320	4	0.0630	3	0.0460	5	0.0630	5	0.0630	4	0.0460
10	800	4	0.0780	3	0.0620	1806	63.4270	INF	INF	1999	75.1640	INF	INF	31	0.3750
	1000	4	0.0870	3	0.0620	INF	INF	INF	INF	1999	87.2370	INF	INF	37	0.4850
	2000	4	0.1560	3	0.1400	INF	INF	INF	INF	1999	165.062	INF	INF	53	1.6250
	10000	4	0.8120	3	0.7490	INF	INF	INF	INF	INF	INF	INF	INF	313	28.3700
	20000	4	1.5620	3	1.7500	INF	INF	INF	INF	INF	INF	INF	INF	369	67.2590
11	100	8	0.0470	3	0.0150	8	0.0310	7	0.0160	3	0.0150	3	0.0160	4	0.0160
12	90000	2	6.4850	3	5.2490	INF	INF	5	7.9820	INF	INF	INF	INF	86	216.0210
13	1000	3	0.0630	3	0.0630	4	0.0930	3	0.0620	5	0.1250	5	0.0940	4	0.0930
14	1000	2	0.0310	2	0.0310	4	0.1400	4	0.0700	2	0.0590	2	0.2370	2	0.1990
15	600	2	0.0470	2	0.0470	3	0.0620	4	0.0780	3	0.0470	3	0.0470	3	0.0310
16	800	6	0.1250	4	0.0780	152	3.5510	INF	INF	1999	76.4620	82	1.9210	104	2.5620
	900	6	0.1400	4	0.0780	114	2.7330	INF	INF	326	10.2470	97	2.9250	76	1.9370
17	200	4	0.0270	3	0.0200	5	0.0380	5	0.0480	5	0.0410	4	0.0370	5	0.0360
18	600	3	0.0310	3	0.0470	4	0.0620	3	0.1030	4	0.0620	4	0.0630	4	0.0620
19	1000	5	0.0940	4	0.0780	480	6.9200	673	12.9350	1999	64.5740	283	5.7640	INF	INF
20	500	3	0.0310	3	0.0310	3	0.0310	2	0.0310	4	0.0470	4	0.0470	3	0.0310

Table 4.2: Numerical results of the seven methods.

N°	Dim	OCB1		OCB2		NVLS*		DDY1		DHS		MDY		CD	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU
21	10	5	0.0160	4	0.0160	INF	INF	INF	INF	1999	1.9450	INF	INF	INF	INF
22	50	4	0.0050	3	0.0050	5	0.0070	5	0.0100	5	0.0080	5	0.0070	5	0.0070
	500	3	0.0340	3	0.0360	4	0.0490	4	0.0490	4	0.0500	4	0.0500	4	0.0490
	1500	3	0.0820	3	0.0800	4	0.1170	4	0.1200	4	0.1170	4	0.1170	4	0.1190
23	500	5	0.0620	4	0.0460	1569	19.6890	1698	23.6240	1999	34.584	965	4.7800	516	5.7490
	700	5	0.0790	4	0.0570	1625	27.8980	1542	26.8800	1999	45.632	536	7.8750	547	8.1590
	1500	5	0.1910	4	0.1460	INF	INF	INF	INF	1999	119.981	448	43.0910	1101	51.164
24	400	5	0.0570	4	0.0370	INF	INF	INF	INF	1999	34.6610	INF	INF	INF	INF
	1000	5	0.0990	4	0.0800	INF	INF	INF	INF	1999	73.8780	INF	INF	INF	INF
	5000	5	0.4920	4	0.4080	INF	INF	INF	INF	1999	370.975	INF	INF	INF	INF
25	50	4	0.0230	3	0.0170	5	0.0280	6	0.0310	6	0.0340	6	0.0330	6	0.0320
	1000	4	0.1410	3	0.0960	5	0.1850	6	0.2280	6	0.2300	6	0.2300	6	0.2290
	10000	4	1.4890	3	0.9050	5	1.8230	6	2.0900	6	2.2990	6	2.2750	6	2.2390
26	6000	5	0.6250	4	0.4370	1582	243.981	INF	INF	1999	399.675	905	137.856	1025	163.978
27	8000	6	1.8430	5	1.4100	239	20.3080	63	6.1080	245	24.4180	232	21.9030	133	14.802
	10000	6	2.2180	4	1.32590	209	21.6380	43	4.5610	251	31.8050	234	28.4920	137	18.557
28	400	5	0.0470	4	0.0240	885	12.9770	INF	INF	822	12.0590	INF	INF	INF	INF
	800	5	0.0620	4	0.0470	1625	47.4060	INF	INF	1491	43.3370	INF	INF	INF	INF
	1000	5	0.0940	4	0.0620	1990	74.5080	INF	INF	1799	69.7680	INF	INF	INF	INF
29	2000	7	1.2190	28	4.0460	294	47.7480	30	3.6240	49	7.2640	49	7.3260	61	8.9820
30	1000	6	0.1250	4	0.0780	INF	INF	1224	39.4990	1999	72.3550	INF	INF	INF	INF
31	2000	3	0.1410	3	0.1400	4	0.1870	4	0.2030	5	0.2500	5	0.2340	4	0.1720
32	200	3	0.0320	3	0.0160	4	0.0310	5	0.0310	3	0.0310	3	0.0310	3	0.0310
33	1500	1	0.0310	1	0.0310	INF	INF	3	0.0780	1	0.0310	1	0.0320	INF	INF
34	200	3	0.0150	3	0.0150	3	0.0160	4	0.0160	3	0.0160	3	0.0160	4	0.0160
35	6000	6	3.6600	5	0.9410	INF	INF	INF	INF	1999	322.457	INF	INF	INF	INF
	20000	6	2.6320	5	2.5030	INF	INF	INF	INF	INF	INF	INF	INF	INF	INF
36	600	6	0.0840	5	0.0740	1193	15.6080	1329	13.3820	1999	33.3480	1591	23.3550	1201	15.791
	1000	6	0.1160	5	0.0960	1982	54.4500	INF	INF	1999	52.2230	1210	24.6030	1832	45.428
37	3000	6	0.3605	5	0.3050	INF	INF	INF	INF	1999	162.973	INF	INF	INF	INF
	6000	6	0.7360	5	0.5840	INF	INF	INF	INF	1999	328.816	INF	INF	INF	INF
	10000	6	1.1840	5	1.0240	INF	INF	INF	INF	1999	516.844	INF	INF	INF	INF
38	100	6	0.0470	5	0.0310	94	0.2820	98	0.3280	181	0.8280	72	0.3280	72	0.3290
	500	6	0.1250	5	0.1100	240	3.3590	236	3.0310	766	16.1930	153	2.8120	171	3.1860
	1000	6	0.2340	5	0.1870	357	9.4510	388	11.3430	1587	68.4620	236	7.9050	266	9.0450
39	50	7	0.0150	5	0.0140	INF	INF	INF	INF	1999	5.5400	INF	INF	INF	INF
	100	6	0.0240	5	0.0210	INF	INF	INF	INF	1999	10.9710	INF	INF	INF	INF
40	40	4	0.0150	5	0.0150	5	0.0160	5	0.0160	5	0.0160	5	0.0160	6	0.0150

Table 4.2: (Continued).

4.4 Application in mode function

Estimation non-parametric has received a great deal of attention in both theoretical and applied statistics literature. For the historical and mathematical survey, we refer the reader to Sager [46]. In statistics, it is always interesting to study the central tendency of the data, that is usually quantified using the location parameters (mean, mode, median). The problem of estimating the mode function of a probability density function (pdf) has taken considerable attention in the past for both independent and dependent data, and a number of distinguished papers deal with this topic. For example, Parzen [41] and Eddy [21] for estimation of the unconditional mode in the independent and identically distributed (i.i.d.) case.

In this section, we consider the problem of estimating the mode of a multivariate unimodal probability density f with support in $\mathbb{R}^n$ from i.i.d. standard normal random variables $X1, ..., Xn$ with common probability density function f. This problem has been investigated in numerous paper. To quote a few of them, Konakov [35] and Samanta [47] . We assume that density f has an unique mode denoted by θ and defined by

$$f(\theta) = \max_{x \in \mathbb{R}^n} f(x). \tag{4.25}$$

A kernel estimator of the mode θ is defined as the random variable $\hat{\theta}$ which maximizes the kernel estimator $f_n(x)$ of $f(x)$, that is

$$f_n\left(\hat{\theta}\right) = \max_{x \in \mathbb{R}^n} f_n(x), \tag{4.26}$$

where

$$f_n(x) = \frac{1}{nh_n^n} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right). \tag{4.27}$$

The bandwidth (h_n) is a sequence of positive real numbers which goes to zero as n goes to infinity and the kernel K is a p.d.f. on $\mathbb{R}^n$.

In this simulation, we choose between two different types of kernel: while Gaussian kernel defined by

$$K(x) = \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\sum_{j=1}^{n} x_j^2\right),$$

and Epanechnikov kernel obtained by

$$K(x) = \left(\frac{3}{4}\right)^n \prod_{j=1}^{n} \left(1 - x_j^2\right).$$

The selection of the bandwidth h is an important and basic problem in kernel smothing techniques. In this simulation, we choose the optimal bandwidth by the cross-validation method.
In this context, we employ our proposed methods to solve the problem (4.26) under strong Wolfe line search technique and compare the computational results of the OCB1 method against the NVLS* [20] and DHS [14] methods. On the other hand, we compare the computational results of the OCB2 method against the DDY1 [16] and MDY [33] methods. We choose some initial points and we obtain the result as in the Tables 4.3 and 4.4. According to these results, it is clear that the OCB1 and OCB2 methods more efficient than the other methods based on the number of iterations and CPU time for solving the problem (4.26).

$STAT_page - 0001$

Kernel	Dim	OCB1		NVLS*		DHS	
		NI	CPU	NI	CPU	NI	CPU
Gaussian	10	6	0.0320	6	0.0780	INF	INF
	20	6	0.1150	INF	INF	463	8.6790
	60	10	2.0120	12	2.2690	52	9.0080
	80	7	2.3900	INF	INF	71	12.0440
Epanechnikov	10	5	0.0620	29	0.3510	18	0.0940
	20	3	0.0940	4	0.2030	301	5.5820
	100	3	7.0140	3	6.6230	27	12.0440

Table 4.3: The simulation result of OCB1, NVLS* and DHS methods for solving problem (4.26).

Kernel	Dim	OCB2		DDY1		MDY	
		NI	CPU	NI	CPU	NI	CPU
Gaussian	10	6	0.0310	95	0.7960	INF	INF
	20	5	0.0940	32	0.5940	157	3.6530
	30	6	0.2800	19	0.7600	419	17.3290
	40	5	0.4280	25	1.7850	53	3.8250
	60	6	1.3050	49	8.0610	18	3.2480
Epanechnikov	10	7	0.1100	11	0.0620	INF	INF
	20	4	0.1880	24	0.4370	23	0.4220
	100	2	1.3430	21	9.3320	142	62.6180

Table 4.4: The simulation result of OCB2, DDY1 and MDY methods for solving problem (4.26).

$_10.png$

Conclusion

In this chapter, we have proposed two new conjugate gradient methods, which we called OCB1 and OCB2. Under the strong Wolfe line search condition, the sufficient descent condition and the global convergence properties were established. Based on the numerical results, it can be observed that our modified methods are more efficient and robust than the other methods. The practical applicability of our methods is also explored in nonparametric estimation of the mode function.

CHAPTER 5

A new conjugate gradient method for unconstrained optimization as a convex combination

Hestens-Stiefel [31] proposed the first formula for solving the quadratic functions in 1952. Fletcher and Reeves [23] presented the first formula for non-linear functions in 1964. Zoutendijk [60] and AL-Baali [1] had proved that the FR method is convergent globally with different line searches, where the convergence properties of FR with exact line search were obtained by Zoutendijk [60], but AL-Baali [1] proved that FR method is globally convergent with the strong Wolfe line search when $\sigma < \frac{1}{2}$ and proved that FR method satisfied the sufficient descent condition.
The last years, many works try to devise some new methods with good numerical performance and that have global convergence properties. Touati-Ahmed and Storey [49] introduced the first hybrid conjugate gradient algorithm, where the β_k is computed as

$$\beta_k^{TaS} = \min\left\{\beta_k^{FR},\ \beta_k^{PRP}\right\}.$$

The authors proved that TaS method has good convergence properties and numerically outperforms both the FR and PRP algorithms. Hu and Storey [32] introduced another hybrid conjugate gradient method which the parameter β_k is obtained by

$$\beta_k^{HuS} = \max\left\{0, \min\left\{\beta_k^{FR},\ \beta_k^{PRP}\right\}\right\}.$$

Another hybrid conjugate gradient methods is a convex combination of the different conjugate gradient methods. Recently, Andrei [3] introduced a new hybrid conjugate gradient method based on HS and DY method (denoted as HYBRID method) for solving unconstrained optimization problems (1.1), where β_k is given as a convex combination of β_k^{HS} and β_k^{DY} i.e.

$$\beta_k^C = (1-\theta_k)\,\beta_k^{HS} + \theta_k \beta_k^{DY},$$

where θ_k is a scalar parameter satisfying $0 \leq \theta_k \leq 1$. Convergence with the standard Wolfe conditions was established. Djordjevic [18] introduced a hybrid conjugate gradient method FRPRPCC, where the parameter β_k is computed as a convex combination of β_k^{PRP} and β_k^{FR} i.e.

$$\beta_k^{hyb} = (1-\theta_k)\,\beta_k^{PRP} + \theta_k \beta_k^{FR}.$$

This author has proved the global convergence of this new hybrid conjugate gradient method with strong Wolfe line search. In 2018, this author also studied the global convergence of the $HLSFR$ method [19] under the strong Wolfe line search, such that

$$\beta_k^{hyb} = (1-\theta_k)\,\beta_k^{HS} + \theta_k \beta_k^{FR}.$$

Numerical results show that this method is efficient for the standard unconstrained problems in CUTE library [4].

In the recent years, many of the variants of the original conjugate gradient methods had been studied by many authors. In 2006, Wei et al. [50] gave a variant of the PRP and is denoted by WYL method, where the parameter β_k is yielded by

$$\beta_k^{WYL} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|} g_{k+1}^T g_k}{\|g_k\|^2}.$$

Huang et al. [34] proved that WYL method satisfies the sufficient descent condition and convergence globally under the strong Wolfe line search with the parameter $\sigma < \frac{1}{4}$. Yao et al. [56] gave a variant of the HS method which is called the MHS method. The parameter β_k in this method is given by

$$\beta_k^{MHS} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|} g_{k+1}^T g_k}{y_k^T d_k}.$$

Under the strong Wolfe line search (3.6), Yao et al. [56] proved that the MHS method also can produce sufficient descent direction and global convergence with the parameter $\sigma < \frac{1}{3}$. In 2009, Zhang [58] shows that NHS method satisfies the sufficient descent condition and converge globally if the strong Wolfe line search is used and the parameter σ is restricted in $\left(0, \frac{1}{2}\right)$, where the parameter β_k is designated by

$$\beta_k^{NHS} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}|g_{k+1}^T g_k|}{d_k^T y_k}.$$

The aim of this chapter is to propose new hybrid conjugate gradient as a convex combination of NHS and FR conjugate gradient algorithms. By using the strong Wolfe line search, we establish the convergence properties of our new method. Numerical results show that the new method is efficient and robust, and outperforms as five CGMs famous. Finally, an application of our method in nonparametric mode estimator is also considered.

5.1 Convex combination

In this section, we combine NHS and FR methods to get $hNHSFR$ method, which the parameter β_k in our new method, denoted as β_k^{hNHSFR}, is computed as a convex combination of β_k^{NHS} and β_k^{FR},i.e.

$$\beta_k^{hNHSFR} = (1 - \theta_k)\, \beta_k^{NHS} + \theta_k \beta_k^{FR}, \tag{5.1}$$

where θ_k is a scalar parameter satisfying $0 \leq \theta_k \leq 1$, which we have to determined. Observe that if $\theta_k = 0$, then $\beta_k^{hNHSFR} = \beta_k^{NHS}$, and if $\theta_k = 1$, then $\beta_k^{hNHSFR} = \beta_k^{FR}$. On the other hand, if $0 < \theta_k < 1$, then β_k^{hNHSFR} is a convex combination of β_k^{NHS} and β_k^{FR}.
The search direction d_k of our new method is defined by

$$d_0 = -g_0,\; d_{k+1} = -g_{k+1} + \beta_k^{hNHSFR} d_k. \tag{5.2}$$

5.1.1 The conjugate condition

In the conjugate gradient method, the traditional conjugacy condition $d_{k+1}^T y_k = 0$, plays an important role in the convergence analyses and numerical calculation. To select the parameter θ_k, we consider the following result.
Lemma 5.1: If the conjugacy condition $d_{k+1}^T y_k = 0$ is satisfied at every iteration, we get

$$\theta_k = \frac{\eta - 1 + \mu}{\vartheta - 1 + \mu}, \tag{5.3}$$

where $\eta=\frac{y_k^T g_{k+1}}{\|g_{k+1}\|^2}$, $\mu=\frac{\left|g_{k+1}^T g_k\right|}{\|g_{k+1}\|\|g_k\|}$ and $\vartheta=\frac{y_k^T d_k}{\|g_k\|^2}$.

Proof: We multiply both sides of the relation (5.2) by the vector y_k^T and computing β_k by (5.1), we obtain

$$\theta_k=\frac{g_{k+1}^T y_k-\beta_k^{NHS} d_k^T y_k}{\left(\beta_k^{FR}-\beta_k^{NHS}\right) d_k^T y_k}.$$

From the formula of β_k^{NHS} and β_k^{FR}, we get

$$\theta_k=\frac{g_{k+1}^T y_k-\frac{\|g_{k+1}\|^2-\frac{\|g_{k+1}\|}{\|g_k\|}\left|g_{k+1}^T g_k\right|}{d_k^T y_k} d_k^T y_k}{\left(\frac{\|g_{k+1}\|^2}{\|g_k\|^2}-\frac{\|g_{k+1}\|^2-\frac{\|g_{k+1}\|}{\|g_k\|}\left|g_{k+1}^T g_k\right|}{-d_k^T y_k}\right) d_k^T y_k}.$$

After some algebra, we obtain

$$\theta_k=\frac{\frac{y_k^T g_{k+1}}{\|g_{k+1}\|^2}-1+\frac{\left|g_{k+1}^T g_k\right|}{\|g_{k+1}\|\|g_k\|}}{\frac{y_k^T d_k}{\|g_k\|^2}-1+\frac{\left|g_{k+1}^T g_k\right|}{\|g_{k+1}\|\|g_k\|}}.$$

Remark 2.1: From the relation (5.3), θ_k is given as follows

$$\theta_k=\begin{cases} 0 & if \quad \frac{\eta-1+\mu}{\vartheta-1+\mu}\leq 0 \text{ or } \vartheta-1+\mu=0, \\ 1 & if \quad \frac{\eta-1+\mu}{\vartheta-1+\mu}\geq 1, \\ \frac{\eta-1+\mu}{\vartheta-1+\mu} & else. \end{cases} \tag{5.4}$$

5.1.2 Algorithm and the sufficient descent condition

The framework of the proposed $hNHSFR$ algorithm is given as follows

Step 1. Initialization.

Select an initial point $x_0\in\mathbb{R}^n$, choose the parameters, $0<\delta<\sigma<\frac{1}{2}$. Set $d_0=-g_0$.

Step 2. Test for continuation of iterations.

If $\|g_k\|\leq 10^{-6}$, then stop. Otherwise go to step 3.

Step 3. Line search.

Compute α_k by the strong Wolfe line searches (3.4) and (3.6) and update the variables $x_{k+1}=x_k+\alpha_k d_k$.

Step 4. Compute θ_k using (5.4).

Step 5. Compute β_k.

If $0<\theta_k<1$, then compute β_k by (5.1).

If $\theta_k\geq 1$, then compute β_k by the formula of β_k^{FR}.

If $\theta_k \leq 0$, then compute β_k by the formula of β_k^{NHS}.
Step 6. Compute the search direction and generate $d_{k+1} = -g_{k+1} + \beta_k^{hNHSFR} d_k$.
Step 7. Set $k = k + 1$ and go to step 2.

Descent condition

In this subsection, we demonstrate that $hNHSFR$ method satisfies the sufficient descent condition, we immediately obtain the following result.
Theorem 5.1: Let the sequences $\{d_k\}_{k\geq 0}$ and $\{g_k\}_{k\geq 0}$ be generated by $hNHSFR$ algorithm. Then the search direction d_k satisfies the sufficient descent condition

$$g_k^T d_k \leq -c \|g_k\|^2, \ \forall \ k \geqslant 0. \tag{5.6}$$

Proof: The following proof is by induction. For $k = 0$, it holds $d_0 = -g_0$ then $g_0^T d_0 = -\|g_0\|^2$, we conclude that the sufficient descent condition holds for $k = 0$. Now, we assume that (5.6) holds for k and prove that for $k + 1$. From (5.1) and (5.2), we have

$$d_{k+1} = -g_{k+1} + ((1 - \theta_k)\, \beta_k^{NHS} + \theta_k \beta_k^{FR}) d_k.$$

Thus, we can obtain

$$d_{k+1} = \theta_k d_{k+1}^{FR} + (1 - \theta_k)\, d_{k+1}^{NHS}. \tag{5.7}$$

Multipling (5.7) by g_{k+1}^T from the left, we get

$$g_{k+1}^T d_{k+1} = \theta_k g_{k+1}^T d_{k+1}^{FR} + (1 - \theta_k)\, g_{k+1}^T d_{k+1}^{NHS}. \tag{5.8}$$

First, let $\theta_k = 0$, then $d_{k+1} = d_{k+1}^{NHS}$. Remind that

$$g_{k+1}^T d_{k+1}^{NHS} = -\|g_{k+1}\|^2 + \beta_k^{NHS} g_{k+1}^T d_k. \tag{5.9}$$

We have from the definition of β_k^{NHS} and the Cauchy Schwarz inequality that

$$0 \leq \beta_k^{NHS} \leq \beta_k^{DY}. \tag{5.10}$$

So, from (5.9) and (5.10) we have

$$g_{k+1}^T d_{k+1}^{NHS} \leq -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2}{d_k^T y_k} \left| g_{k+1}^T d_k \right|. \tag{5.11}$$

By using (3.6) and from the sufficient descent condition, we get

$$y_k^T d_k \geqslant -(1-\sigma) g_k^T d_k. \tag{5.12}$$

Using (5.11), (5.12) and from the second strong Wolfe line search condition, we concluded

$$g_{k+1}^T d_{k+1}^{NHS} \leq -c_1 \left\| g_{k+1} \right\|^2, \tag{5.13}$$

where $c_1 = \frac{2\sigma-1}{\sigma-1}$, so there is a constant $c_1 > 0$ with $\sigma < \frac{1}{2}$.
Secondly, let $\theta_k = 1$, then $d_{k+1} = d_{k+1}^{FR}$.
Let's remind to the fact that the sufficient descent condition holds for FR method in the presence of the strong Wolfe condition [25]. So, there exists a constant $c_2 > 0$, such that

$$g_{k+1}^T d_{k+1}^{FR} \leq -c_2 \left\| g_{k+1} \right\|^2. \tag{5.14}$$

where $c_1 = c_2$.
Now, we assume that $0 < \theta_k < 1$, then there exist two constant a_1 and a_2 positive such as $0 < a_1 \leq \theta_k \leq a_2 < 1$. From the relation (5.11), (5.13) and (5.14), we get

$$g_{k+1}^T d_{k+1} \leqslant -c \left\| g_{k+1} \right\|^2,$$

such that $c = a_1 c_1 + (1-a_2) c_1$. □

5.2 Global convergence

Now, we establish the convergence results and the following assumptions are necessary in the analysis of the global convergence properties of the conjugate gradient method.
Assumption A. The level set

$$S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\},$$

is bounded.
Assumption B. In some open convex neighborhood $\mathcal{N}$ of S, the function f is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$ such that:

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathcal{N}. \tag{5.15}$$

These assumptions imply that there exists a positive constant $\Gamma \geq 0$ such that

$$\| \nabla f(x) \| \leq \Gamma, \text{for all } x \in \mathcal{N}. \tag{5.16}$$

The following Lemma is needed in the analysis of global convergence of our new conjugate gradient method, which is proved by Dai et al [13].

Lemma 5.2: Let Assumptions A and B hold. Consider the method (3.2) and (3.3), where d_k is a descent direction and α_k is obtained by the strong Wolfe line search. If

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

then

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

Now, we need also this Lemma to prove the convergence of our methods.

Lemma 5.3: Let Assumptions **A** and **B** hold. If d_k is a descent direction and α_k satisfies the standard Wolfe condition (3.5). Then

$$\alpha_k \geq \frac{(1-\sigma) \mid g_k^T d_k \mid}{L \parallel d_k \parallel^2}. \tag{5.17}$$

Proof: See the proof of Lemma 3.2 in Liu and Li [36].

Remark 5.1: From (3.6) and (5.6), the stepsize α_k obtained in the $hNHSFR$ algorithm satisfies (5.17). This indicates, the step size α_k obtained in the $hNHSFR$ method is not equal to zero, i.e., there exists a constant $\lambda > 0$, such that

$$\alpha_k \geq \lambda, \qquad \text{for all } k \geq 0. \tag{5.18}$$

The following Theorem establishes the global convergence of our new method with the strong Wolfe line search.

Theorem 5.2: Suppose that Assumptions A and B hold and consider the iterative method in the form (3.2) and (3.3). Let $\{x_k\}_{k\geq 0}$ be generated by $hNHSFR$ algorithm, then the following condition holds

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{5.19}$$

Proof: We prove by contradiction.Suppose that (5.19) doesn't hold, there exists a constant r, such that

$$\|g_k\| \geq r \qquad \text{for all } k. \tag{5.20}$$

From (5.1), we have

$$\left|\beta_k^{hNHSFR}\right| \leq \beta_k^{NHS} + \beta_k^{FR}. \tag{5.21}$$

Now, from the sufficient descent condition, (5.10), (5.16) and (5.20), we can get

$$\beta_k^{NHS} \leq \frac{\Gamma^2}{c(1-\sigma)r^2}.$$

So,

$$\beta_k^{hNHSFR} \leq \frac{\Gamma^2}{c(1-\sigma)r^2} + \frac{\Gamma^2}{r^2} = E.$$

Applying (5.2), we find that

$$\|d_{k+1}\| \leq \| g_{k+1} \| + \left|\beta_k^{hNHSFR}\right| \frac{\|x_{k+1} - x_k\|}{\alpha_k} \leq M,$$

where

$$M = \Gamma + E\frac{D}{\lambda},$$

and D is a diameter of the level set $\mathcal{N}$.
By take the summation $k \geq 0$, we get

$$\sum_{k\geq 0} \frac{1}{\|d_k\|^2} = \infty.$$

So, applying Lemma 5.2, we conclude that (5.19) is true. This is a contradiction with (5.20), so we have proved (5.19). □

5.3 Numerical Experiments

In this section, we present some numerical experiments obtained with our new proposed conjugate gradient method to show that the method is efficient for the unconstrained optimization problems. The test problems have been taken to the CUTE library [4] and [9]. All the algorithms have been coded in MATLAB 2013 and compiler settings on the PC machine (2.5 GHz, 3.8 GB RAM memory) with windows XP operating system. Five conjugate gradient methods NHS [58], Tas [49], HuS [32], $FRPRPCC$ [18], and $HHSFR$ [19] are compared with our method in numerical performance. The parameters in strong Wolfe line search are chosen as $\sigma = 10^{-3}$ and $\delta = 10^{-4}$.The iteration is terminated if one of the following conditions is satisfied (i) $\|g_k\|_\infty < 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector, (ii) The number of iterations exceeded 2000, (iii) The computing time is more than 500 s.

We use the performance profile introduced by Dolan and Morè [17] to compare the performance according to number iteration and CPU time to rule as follows. Let S is the set of methods and P is the set of the test problems with n_p , n_s are the number of the test problems and the number of the methods, respectively. For each problem $p \in P$ and solver $s \in S$, denote $\tau_{p,s}$ be the number of iterations or CPU time required to solve problems $p \in P$ by solver $s \in S$. Then comparison between different solvers based on the performance ratio is given by

$$r_{p,s} = \frac{\tau_{p,s}}{\min\left\{\tau_{p,i},\ 1 \leq i \leq n_s\right\}}.$$

Suppose that a parameter $r_M \geq r_{p,s}$ for all problems and solvers chosen, and $r_M = r_{p,s}$ if and only if solver S does not solve problem p. The overall evaluation of the performance of the solvers is then given by the performance profile function given by:

$$F_s(t) = \frac{size\left\{p:\ 1 \leq p \leq n_p,\ r_{p,s} \leq t\right\}}{n_p},$$

where $t \geq 1$ and $size\left\{p:\ 1 \leq p \leq n_p,\ r_{p,s} \leq t\right\}$ is the number of elements in the set $\left\{p:\ 1 \leq p \leq n_p,\ r_{p,s} \leq t\right\}$. This function $F_s : [1, \infty[\rightarrow [0, 1]$ is the distribution function for the performance ratio. The value of $F_s(1)$ is the probability that the solver will win the rest of the solvers.

In this numerical study, Table 5.1 lists the names of the test functions and Table 5.2 shows the performance of the six methods. Table5.2, Figure 5.1 and Figure 5.2 give a performance comparison of the $hNHSFR$ method with those for the number of iterations and the CPU time.

From these Figures and Table 5.2, we can see that our new method performs better and effective than NHS [58], TaS [49], HuS [32], $FRPRPCC$ [18], and $HHSFR$ [19] methods.

Number	function	Number	function
1	BEALE	21	LIARWHD
2	BOOTH	22	NONDIA
3	BRANIN	23	NONSCOMP
4	LION	24	Penalty
5	MATYAS	25	Power
6	Almost Perturbed Quadratic	26	Qing
7	Chung	27	Quadratic
8	CUBE	28	Quartic
9	DIAG	29	QUARTICM
10	DIAG-AUP1	30	Rastring
11	Diagonal1	31	Raydan1
12	Diagonal2	32	Raydan2
13	Diagonal4	33	Rosenbrock
14	DIXON	34	SCHWEFEL
15	ENGVAL1	35	Schwefel223
16	Exponential	36	Schwefel221
17	ExtendedHiebert	37	Sphere
18	FLETCHER	38	TRIDIAGONAL
19	Hager	39	TR-Sumofquadratics
20	Himmelbleau	40	ZAKHAROV

Table 5.1: The test functions.

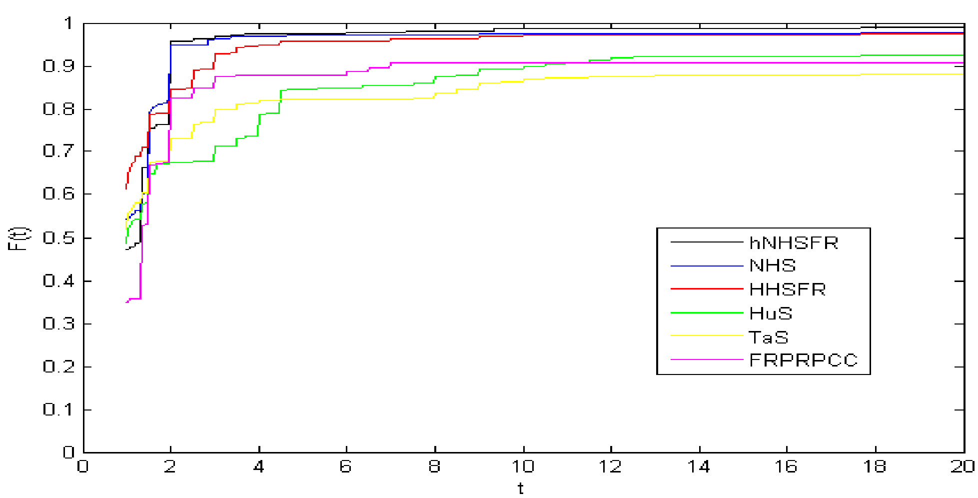

Figure 5.1: Performance profile on the number of iterations.

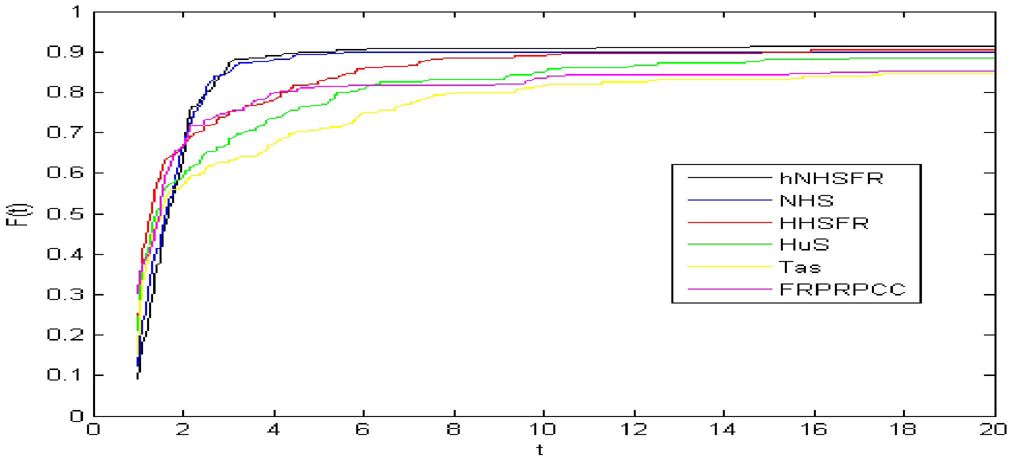

Figure 5.2: Performance profile on the CPU time.

N°	Dim	hNHSFR		TaS		FRPRPCC		HuS		HHSFR		NHS	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU
1	2	2	0.0150	2	0.0080	2	0.0160	2	0.0160	2	0.0160	2	0.0150
2	2	14	0.0310	20	0.0630	13	0.0470	17	0.0320	14	0.0470	19	0.0630
3	2	14	0.0470	15	0.0630	14	0.0620	14	0.0630	29	0.0780	37	0.1250
4	2	14	0.0930	16	0.1090	27	0.1720	15	0.0940	31	0.2030	43	0.5190
5	2	2	0.0150	2	0.0150	2	0.0150	3	0.0160	2	0.0160	2	0.0160
6	50 100	163 130	0.3440 0.8750	234 533	0.5310 2.1090	179 241	0.2810 0.7190	177 232	0.3750 1.1710	864 1999	2.2340 10.6540	404 357	0.8280 1.3280
7	40	3	0.0160	4	0.0160	3	0.0150	3	0.0160	3	0.0160	3	0.0160
8	50 800	2 2	0.0160 0.0620	2 2	0.0160 0.0630	2 2	0.0160 0.0630	2 2	0.0160 0.1230	2 2	0.0470 0.0630	2 2	0.0310 0.0630
9	200	4	0.0150	4	0.0310	4	0.0310	4	0.0310	4	0.0310	4	0.0320
10	800 1000	3 3	0.0470 0.0620	3 3	0.0470 0.0720	3 3	0.0470 0.0620	3 3	0.0560 0.0630	3 3	0.0470 0.0620	3 3	0.0630 0.0630
11	100	114	0.4370	140	0.4370	434	1.7340	433	23.2310	691	3.2330	INF	INF
12	600	143	1.5620	468	2.7180	467	2.5470	146	1.6250	567	8.2640	155	1.7500
13	8000 12000	2 2	0.1560 0.2180	5 6	0.3120 0.3130	5 5	0.2190 0.8280	5 10	0.6410 2.3580	3 3	0.2060 0.3620	2 2	0.1720 0.2190
14	40	5	0.0150	INF	INF	1999	5.3420	1999	5.8660	5	0.0160	5	0.0160
15	50	3	0.0150	3	0.0320	3	0.0160	3	0.0310	3	0.0160	3	0.0150
16	2000	2	0.0470	4	0.4090	3	0.0940	34	1.4060	2	0.0780	2	0.0780
17	500	3	0.0310	3	0.0470	2	0.0150	3	0.0470	3	0.0320	3	0.0470
18	200	3	0.0150	INF	INF	3	0.0310	3	0.0160	3	0.0160	3	0.0470
19	120	31	0.1250	32	0.0630	42	0.1410	36	0.1560	32	0.1400	40	0.1880
20	80 100	3 3	0.0460 0.0780	21 24	0.1720 0.3440	19 19	0.1870 0.3280	8 8	0.2000 0.1880	3 3	0.0470 0.0790	3 3	0.0460 0.0790
21	400	3	0.0150	3	0.0160	3	0.0160	3	0.0150	3	0.0160	3	0.0160
22	200	3	0.0150	3	0.0160	2	0.0160	3	0.0150	3	0.0160	3	0.0150
23	5000 10000	3 3	0.0150 0.0160	3 3	0.0160 0.0470	2 2	0.0160 0.0160	3 3	0.0320 0.0310	3 3	0.0310 0.0310	3 3	0.0310 0.0470
24	200 1000	8 7	0.0160 0.0460	23 16	0.0780 0.0620	13 16	0.0310 0.0620	8 27	0.0310 0.2350	8 10	0.0310 0.0780	8 8	0.1700 0.0800
25	10 100	30 78	0.0150 0.0460	71 181	0.0320 0.1090	65 136	0.0160 0.0620	36 69	0.0320 0.0470	166 585	0.0940 0.5000	30 62	0.0160 0.0470
26	50	163	2.4210	187	2.0000	246	2.6870	180	2.6710	537	8.2480	163	2.4220
27	200 500	129 222	0.7240 2.2960	194 334	0.7810 2.8120	193 293	0.8120 2.7060	137 212	0.7660 2.4990	461 1340	2.6720 17.4360	134 223	0.7810 2.3120
28	200	195	0.5630	55	0.2650	354	0.9290	366	1.0150	196	0.7650	40	0.1940
29	400	1077	16.8080	INF	INF	INF	INF	916	13.7460	1085	17.0430	INF	INF
30	2000	7	0.3720	22	1.2030	3	0.0940	100	5.3970	6	0.2940	22	1.2030
31	20	26	0.0310	35	0.0310	32	0.0310	35	0.0310	26	0.0310	30	0.0310
32	10000 20000	2 2	0.2810 0.5780	3 3	0.3750 0.7970	3 3	0.3280 0.7660	3 3	0.3750 0.8450	2 2	0.2970 0.6400	3 3	0.3590 0.7810
33	200	3	0.0150	2	0.0150	2	0.0160	3	0.0160	3	0.0150	3	0.0160
34	500	5	0.0630	9	0.2350	10	0.1250	3	0.0630	5	0.0630	6	0.1470
35	2000	3	0.0150	3	0.0310	3	0.0160	3	0.0150	3	0.0160	3	0.0310
36	1500 2000	2 2	0.0160 0.0150	3 3	0.0620 0.1090	3 3	0.0930 0.0940	3 3	0.0470 0.0780	2 2	0.0160 0.0160	3 3	0.0470 0.0620
37	50000	2	1.0160	3	3.57770	3	3.7490	3	0.7960	6	3.2450	2	1.0630
38	40	45	0.1090	53	0.1560	45	0.1250	45	0.1250	74	0.2500	54	0.1570
39	500	7	0.0940	5	0.0940	7	0.1100	7	0.1410	7	0.1100	7	0.1100
40	800	3	0.0620	3	0.0630	4	0.0940	4	0.0940	4	0.0780	3	0.0620

Table 5.2: Numerical results of the six methods.

5.4 Application in mode function

Estimation nonparametric has received a great deal of attention in both theoretical and applied statistics liter ature. For the historical and mathematical survey, we refer the reader to Sager [46]. In statistics, it is always interesting to study the central tendency of the data, that is usually quantified using the location parameters (mean, mode, median). The problem of estimating the mode function of a probability density function (p.d.f.) has taken considerable attention in the past for both independent and dependent data, and a number of distinguished papers deal with this topic. For example, Parzen [41] and Eddy [21] for estimation of the unconditional mode in the independent and identically distributed (i.i.d.) case.

In this section, we consider the problem of estimating the mode of a multivariate unimodal probability density f with support in $\mathbb{R}^n$ from i.i.d. standard normal random variables $X_1, ..., X_n$ with common probability density function f. This problem has been investigated in numerous paper. To quote a few of them, Konakov [35] and Samanta [47]. We assume that density f has an unique mode denoted by θ and defined by

$$f(\theta) = \max_{x \in \mathbb{R}^n} f(x). \tag{5.22}$$

A kernel estimator of the mode θ is defined as the random variable $\hat{\theta}$ which maximizer the kernel estimator $f_n(x)$ of $f(x)$, that is

$$f_n\left(\hat{\theta}\right) = \max_{x \in \mathbb{R}^n} f_n(x), \tag{5.23}$$

where

$$f_n(x) = \frac{1}{nh_n^n} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right). \tag{5.24}$$

The bandwidth (h_n) is a sequence of positive real numbers which goes to zero as n goes to infinity and the kernel K is a p.d.f. on $\mathbb{R}^n$.

In this simulation, we choose between two different types of kernel: while standard Gaussian kernel defined by

$$K(x) = \frac{1}{(2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\sum_{j=1}^{n} x_j^2\right),$$

and Epanechnikov kernel obtained by

$$K(x) = \left(\frac{3}{4}\right)^n \prod_{j=1}^{n} \left(1 - x_j^2\right).$$

The selection of the bandwidth h is an important and basic problem in kernel smothing techniques. In this simulation, we choose the optimal bandwidth by the cross-validation method.
In this context, we employ our proposed method to solve the problem (5.23) under strong Wolfe line search technique and compare the computational results of the $hNHSFR$ method against the NHS [58] method. We choose some initial points and we obtain the result as in the Table 5.3. According to these results, it is clear that the $hNHSFR$ method more efficient than NHS method based on the number of iterations and CPU time for solving the problem (5.23).

$STAT_page - 0001$

Kernel	Point initial	Dim	hNHSFR		NHS	
			NI	CPU	NI	CPU
Gaussian	(-0.75,...,-0.75)	15	20	0.2340	60	0.7030
		30	2	0.0780	32	1.3280
	(-1,...,-1)	55	30	4.5930	INF	INF
		60	4	0.7180	416	71.0700
		80	4	1.4220	17	5.4920
	(0.25,...,0.25)	40	2	0.1400	37	3.0490
		80	2	0.6580	15	4.4480
		150	6	3.0300	43	6.2390
	(-0.95,...,-0.95)	55	19	2.6080	207	29.5440
		35	24	1.4530	215	11.9810
		50	54	6.2260	94	10.5910
		90	14	5.0930	18	6.5920
Epanechnikov	(-1.5,...,-1.5)	10	21	0.8350	INF	INF
		20	3	0.1970	24	0.4480
		100	2	0.5340	116	8.3690
	(-2,...,-2)	6	42	0.3970	123	4.2340
		8	58	0.8650	104	3.6640
		10	53	0.6530	142	5.6180
	(-0.001,...,-0.001)	22	30	0.8310	47	1.5410
		30	4	0.3460	7	0.6210
		50	2	0.2680	3	0.4700

Table 5.3: The simulation result of hNHSFR and FR methods for solving problem (5.23).

$_14.png$

Conclusion

In this chapter, we have presented a new hybrid conjugate gradient algorithm wich is a convex combination of NHS and FR methods. The global convergence properties and the sufficient descent condition were established, under strong Wolfe line search condition. Also our new conjugate gradient method is more robust and effective than the other conjugate gradient algorithms.
We also investigate our method's practicality in nonparametric mode function estimation.

General Conclusion

Nonparametric estimation and nonlinear conjugate gradient methods are of interest to us in this thesis. Thus, in order to solve unconstrained optimization problems, we have examined three types of nonlinear conjugate gradient algorithms.

► Study1 presented a new hybrid and three-term conjugate gradient method, known as EHD method. By skillfully combining the finest features of DY and HS algorithms, this novel hybrid conjugate gradient approach gets you to solutions more quickly and effectively than ever before. Our method's search direction is designed to meet the descent requirement in order to guarantee effective optimization. We have demonstrated its globally convergence by employing the strong Wolfe line search. Numerous numerical experiments have produced interesting results, including iteration counts, time measurements, and gradient evaluation numbers. These results clearly show that our proposed three-term conjugate gradient algorithm outperforms competing methods in terms of speed and efficiency. Finally, applying the EHD method in nonparametric statistiques.

► In the second work, Two modified conjugate gradient techniques, known as the OCB1 and OCB2 methods, were presented. Under the SWLS conditions, the sufficient descent condition for our two modified methods (OCB1 and OCB2 methods) has been determined. Also, the OCB1 and OCB2 method's global convergence properties have been validated. Then, it is discovered that the proposed approches are more effective than some alternative approches.

► Third, we introduced a novel conjugate gradient method as a convex combination of NHS and FR methods, where it is called hNHSFR. It is essential to note that, regardless of the flexible conjugate parameter and line search that are selected, the search direction always satisfies the sufficient descent criterion. Under mild assumptions, the hNHSFR method's global convergence characteristics have been established. The results of many numerical experiments show that our approaches are highly reliable, efficient and outperform some of the current approaches in minimizing unconstrained optimization problems.

-Possibility of Practical Use: Research is currently being done to determine whether it can be applied to nonparametric mode function estimation, which points to some exciting potential uses.

BIBLIOGRAPHY

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

[2] N. Andrei, A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan, Studies in Informatics and Control, 17 (2008) 55-70.

[3] N. Andrei, Another hybrid conjugate gradient algorithm for unconstrained optimization, Numer. Algorithms, 47 (2008) 143–156.

[4] N. Andrei, An unconstrained optimization test functions, Advanced Modeling and Optimization, 10 (2008) 147-161.

[5] N. Andrei, New hybrid conjugate gradient algorithms for unconstrained optimization, Encyclopedia of Optimization, (2009) 2560–2571.

[6] N. Andrei. Nonlinear Conjugate Gradient Methods for Unconstrained Optimization. Springer, 2020.

[7] L. Armijo. "Minimization of functions having Lipschitz continuous first partial derivatives". In: Pacific Journal of mathematics 16.1 (1966), pp. 1–3.

[8] A. M. Awwal, I. M. Sulaiman, M. Malik, M. Mamat, P. Kumam, K. Sitthithakerngkiet, A spectral RMIL conjugate gradient method for unconstrained optimization with applications in portfolio selection and motion control, IEEE Access, 9 (2021) 75398-75414.

[9] I. Bongartz, A. R. Conn, N. Gould and P. L. Toint, Constrained and unconstrained testing environment, ACM Trans. Math. Software 21 (1995) 123–160.

[10] S. Boyd and L. Vandenberghe, Convex optimization.Cambridge university press, 2004.

[11] C. W. Carroll, The created reponse surface technique for optimizing nonlinear restrained systems, Operations Res., Vol. 9 (1961), pp. 169-84.

[12] Y. H. Dai, Analyses of conjugate gradient methods, Ph.D. thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1997.

[13] Y. H. Dai, J. Y. Han, G. H. Liu, D. F. Sun, X. Yin and Y. Yuan, Convergence properties of nonlinear conjugate gradient methods, SIAM J. Optim., 10 (1999) 348–358.

[14] Z.F. Dai and F.H. Wen, Another improved wei-yao-liu nonlinear conjugate gradient method with sufficient descent property, Appl. Math. Comput. 218 (2012) 7421-7430.

[15] Y. H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optim., 10 (1999) 177-182.

[16] Y. H. Dai and Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization, Ann. Oper. Res., 103 (2001) 33–47.

[17] E. D. Dolan and J. J. Morè, Benchmarking optimization software with performance profiles, Math. Programming, 91 (2002) 201-213.

[18] S. S. Djordjevic, New hybrid conjugate gradient method as a convex combination of FR and PRP methods, Filomat, 30:11 (2016) 3083-3100.

[19] S. S. Djordjevic, New hybrid conjugate gradient method as a convex combination of HS and FR methods, Applied Mathematics and Computation (JAMC), (2018), 366-378.

[20] X.W. Du, P. Zhang and W. Ma, Some modified conjugate gradient methods for unconstrained optimization, J. Comput and Appl. Math., 305 (2016) 92–114.

[21] W. F. Eddy, Optimum kernel estimators of the mode, Ann. Statist., 8 (1980) 870-882.

[22] A. V. Fiacco and G. P. McCormick, Nonlinear Programming, John Wiley, New York, (1968).

[23] R. Fletcher and C. Reeves, Function minimization by conjugate gradients, Comput. J., 7 (1964) 149-154.

[24] R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization, Computer. J., 6 (1963), pp. 163-168.

[25] R. Fletcher, Practical Methods of Optimization, vol. 1: Unconstrained Optimization, John Wiley and Sons, New York, 1987.

[26] J. C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, SIAM J. Optim., 2 (1992) 21-42.

[27] J. C. Gilbert, Elements of Differentiable Optimization: Theory and Algorithms, Lecture notes, National School of Advanced Techniques, Paris 2007.

[28] A.A. Goldstein and J.F. Price, An effective algorithm for minimzation, Num.Math., 10 (1969), pp. 184-189.

[29] A. A. Goldstein, On steepest descent, SIAM J. on Control A, Vol. 3, No. 1 (1965), pp. 147-151.

[30] W. W. Hager and H. Zhang, Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent., ACM Trans. Math. Software, 32 (2006) 113–137.

[31] M. R. Hestenes and E. L. Stiefel, Methods of conjugate gradients for solving linear systems, J. Research Nat. Bur. Standards, 49 (1952) 409-436.

[32] Y. F. Hu and C. Storey, Global convergence result for conjugate gradient methods, J. Optim. Theory Appl., 71 (1991) 399-405.

[33] H. Huang, A new conjugate gradient method for nonlinear unconstrained optimization problems, J. Henan Univ., 44 (2014) 141-145.

[34] H. Huang, Z. Wei and Y. Shengwei. The proof of the sufficient descent condition of the Wei-Yao-Liu conjugate gradient method under the strong Wolfe-Powell line search. Applied Mathematics and Computation, 189 (2007) 1241–1245.

[35] V. D. Konakov, On the asymptotic normality of the mode of multidimensional distributions, Theory Probab. Appl., 19 (1974) 794 799.

[36] J. K. Liu and S. J. Li, New hybrid conjugate gradient method for unconstrained optimization, Appl. Math. Comput., 245 (2014) 36-43.

[37] Y. Liu and C. Storey, Efficient generalized conjugate gradient algorithms, Theory JOTA, 69 (1991) 129-137.

[38] G. Ma, H. Lin, W. Jin, D. Han, Two modified conjugate gradient methods for unconstrained optimization with applications in image restoration problems, Journal of Applied Mathematics and Computing, (2022) 1-26.

[39] A. E. Mehamdia, Y. Chaib, T. Bechouat, Two modified conjugate gradient methods for unconstrained optimization and applications, RAIRO-Oper. Res., 57 (2023) 333–350.

[40] P. Mtagulwa and P. Kaelo, An efficient modified PRP-FR hybrid conjugate gradient method for solving unconstrained optimization problems, Applied Numerical Mathematics, 145 (2019) 111–120.

[41] E. Parzen, On estimating probability density function and mode, Ann. Math. Statist., 33 (1962) 1065-1076.

[42] K. Pearson, Contributions to the mathematical theory of evolution, II : skew variation in homogeneous material, Philosophical transactions of the royal society of London array, 186 (1895).343-414.

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

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

[45] M. J. D. Powell, Restart procedures for the conjugate gradient method. Mathematical programming, 12(1): 241-254, (1977).

[46] T. W. Sager, An iterative method for estimating a multivariate mode and isopleth, J. Amer. Statist. Assoc., 74 (1975) 329-339.

[47] M. Samanta, Nonparametric estimation of the mode of a multivariate density, South Africain Statist. J., 7 (1973) 109-117.

[48] P. S. Stanimirović et al. "A Survey of gradient methods for solving nonlinear optimization". In: Electronic Research Archive 28.4 (2020), p. 1573.

[49] D. Touati-Ahmed and C. Storey, Efficient hybrid conjugate gradient techniques, J. Optim. Theory Appl., 64 (1990) 379-397.

[50] Z . Wei, S. Yao and L. Liu, The convergence properties of some new conjugate gradient methods, Appl. Math. Comput., 183 (2006) 1341–1350

[51] P. Wolfe, A duality theorem for nonlinear programming, Quart. Appl. Math., 19 (1961), pp. 239-244.

[52] P. Wolfe, Convergence conditions for ascent methods. II : Some corrections, SIAM Review, 11 (1969), pp. 226–235.

[53] P. Wolfe, Convergence conditions for ascent methods. II : Some corrections, SIAM Review, 13 (1971), pp. 185–188.

[54] S. Wenyu and Ya-Xiang Yuan, Optimization theory and methods: nonlinear programming. Vol. 1. Springer Science & Business Media, 2006.

[55] H. Yamato, Sequential estimation of a continuous probability density function and mode, Bull. Math. Statist. Jap., 14 (1972) 1-12.

[56] S. W. Yao, Z. X. Wei and H. Huang, A notes about WYL's conjugate gradient method and its applications, Appl. Math. Comput., 191 (2007) 381–388.

[57] Y. Yuan, Analysis on the conjugate gradient method, Optim. Methods Softw., 2 (1993),19–29.

[58] L. Zhang, An improved Wei–Yao–Liu nonlinear conjugate gradient method for optimization computation, Applied Mathematics and Computation, 6 (2009) 2269-2274.

[59] Z. Zhu, D. Zhang and S. Wang, Two modified DY conjugate gradient methods for unconstrained optimization problems, (2020) 125004.

[60] G. Zoutendijk, Nonlinear programming, computational methods, Integer and Nonlinear Programming, 143 (1970) 37-86.