People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Setif 1 University Ferhat Abbas
Faculty of Sciences
Department of Mathematics

جامعة سطيف 1 فرحات عباس
كلية العلوم
قسم الرياضيات

Doctoral Thesis

submitted in fulfillment of the requirements for obtaining the degree of Doctor in Mathematics

Specialty: Nonlinear Optimization

Theme:

New contributions to nonlinear optimization methods

Presented by

Mme. **Asma Maiza**

Thesis defended on 01/22/2026, in front of the jury composed of:

Mr.	**ACHACHE Mohamed**	Prof	Setif 1 University Ferhat Abbas	President
Mr.	**ZIADI Raouf**	MCA	Setif 1 University Ferhat Abbas	Supervisor
Mme.	**LEULMI Assma**	Prof	Setif 1 University Ferhat Abbas	Co-supervisor
Mr.	**YAHIA Djabrane**	Prof	Biskra University Mohamed Khider	Examiner
Mr.	**LAKHDARI Imad Eddine**	Prof	Biskra University Mohamed Khider	Examiner
Mme.	**DAILY Chafia**	MCA	Setif 1 University Ferhat Abbas	Examiner

2025/2026

Acknowledgments

I would like to thank Doctor **Raouf Ziadi** and Professor **Assma Leulmi** for supervising this thesis. I express my deepest respect and sincere appreciation.

It is my great pleasure to thank Professor **Mohamed Achache** for kindly agreeing to examine this work and for the honor of chairing the jury.

I would also like to thank Professors **Djabrane Yahia** and **Imad Eddine Lakhdari** from University of Biskra, as well as Doctor **Chafia Daily**, for agreeing to examine this thesis and for the honor of serving on the jury.

Finally, I express my sincere gratitude to my family and friends for their help and support throughout all the years of my education.

Dedication

To my beloved **parents**, my mother and my father, for your unconditional love, endless sacrifices, and unwavering faith in me. You are the foundation upon which all my achievements stand.

To my dear husband, **Sabri**, for standing by my side with patience, respect, and support, and for sharing with me both the burdens and hopes of this journey.

To my precious children, **Alaa**, **Yahya**, and **Ayoub**, the light of my life and the deepest source of my strength. Every step of this journey was taken with your names in my heart, and every success is a promise to the future I dream of for you.

To my two **sisters**, for your constant encouragement, understanding, and silent presence.

Contents

Introduction **5**

1 Fundamental notions **6**

1.1 Optimization problems 6

1.1.1 Convergence Rates 8

1.2 Convexity 10

1.3 Unconstrained optimization problem 12

1.3.1 Existence and uniqueness results 13

1.3.2 Descent Direction 14

1.3.3 General scheme of unconstrained optimization algorithms 15

1.3.4 Sufficient optimality conditions 17

1.3.5 Necessary and sufficient conditions (convex case) 17

2 Unconstrained optimization methods **19**

2.1 Line search 19

2.1.1 Exact line search 20

2.2 Uncertainty interval 20

2.2.1 Inexact line search 21

2.2.2 Armijo method 22

2.2.3 Goldstein-Price rule 23

2.2.4 Wolfe line search 24

2.2.5 Strong wolfe line search 24

2.3 Algorithms for solving unconstrained optimization 26

2.3.1 Gradient method 26

2.3.2 Newton method 27

2.3.3 Quasi-Newton methods 28

2.3.4 Theory and algorithm 29

3 Conjugate Gradient Methods 33

3.1 Linear conjugate gradient methods 33

3.1.1 Conjugate directions method 35

3.1.2 Conjugate gradient method 37

3.2 Nonlinear conjugate gradient method 39

3.2.1 Zoutendijk conditions for global convergence 40

4 Two new conjugate gradient algorithms for nonlinear unconstrained optimization 44

4.1 The proposed conjugate gradient algorithms 47

4.1.1 The MZ algorithm 47

4.1.2 The hPB algorithm 48

4.2 The sufficient descent property of the proposed approaches 50

4.3 The convergence analysis of the proposed approaches 52

4.3.1 The global convergence of the proposed approaches 54

4.4 Numerical implementations 56

4.4.1 Application on some typical test functions 56

4.4.2 Application for restoring images 59

5 An improved PRP conjugate gradient method for unconstrained optimization 73

5.1 The inexact line search conditions and the corresponding algorithm . . . 75

5.2 The sufficient descent property of PRP-NLS algorithm 76

5.3 The convergence analysis of PRP-NLS algorithm 78

5.3.1 The global convergence of PRP-NLS algorithm 78

5.4 Computational experiments 81

Conclusion 85

List of Algorithms

1 The safety interval algorithm 21
2 Armijo line search 22
3 Goldstein-Price line search 23
4 Weak Wolfe line search 25
5 The strong Wolfe line search 25
6 Gradient algorithm 27
7 Newton algorithm 28
8 DFP algorithm 31
9 BFGS algorithm 32
10 Conjugate direction algorithm 36
11 Conjugate gradient algorithm (quadratic case) 38
12 The nonlinear conjugate gradient algorithm 41
13 The MZ algorithm 47
14 The hPb algorithm 49
15 The PRP-NLS algorithm 75

Introduction

Optimization is a fundamental branch of mathematics concerned with the modeling, analysis, and resolution of problems involving the minimization or maximization of a given function. These problems may be either constrained or unconstrained and are addressed using analytical or numerical techniques. The objective functions under consideration can be either linear or nonlinear in nature.

The origins of optimization theory can be traced back to the 18th century, with the pioneering work of Taylor, Newton, and Lagrange, particularly in the context of series expansions. Later, Cauchy introduced one of the earliest optimization techniques—the gradient method—for solving unconstrained problems. Although these early contributions laid the foundation for the field, progress remained relatively gradual for some time.

Significant advancements emerged after the Second World War, with many key developments stemming from research conducted in Great Britain. Scientists have explored various techniques, such as Newton's method, quasi-Newton methods, and conjugate gradient (CG) methods, to find the most efficient way to solve a particular problem. The conjugate gradient methods (CG) have become a favored thanks to their straightforward iterative process and low memory requirements; they are also widely employed in numerous applications, in particular in image processing, neural networks, grid computing, molecular physics and so on. The first conjugate gradient method was initially proposed by Hestenes and Stiefel in 1952 [40] to minimize strictly convex quadratic functions.

In the decades that followed, researchers extended the conjugate gradient framework to address nonlinear optimization problems. In 1964, Fletcher and Reeves [19] introduced the Fletcher–Reeves method. Next, in 1969, by the independent development of the Polak–Ribière–Polyak method, attributed to Polak [44], Ribière, and Polyak [45].

Subsequent improvements included Fletcher's Conjugate Descent Method in 1987 and the Dai–Yuan method proposed by Dai and Yuan in 1999 [15], among several other variants.

In the last two decades, substantial efforts have been dedicated to developing efficient nonlinear conjugate gradient methods tailored for unconstrained nonlinear optimization. Motivated by these developments, in this thesis, we have proposed three contributions for enhancing the resolution of such an optimization problem where the cost function is nonlinear. Indeed, we have proposed an improved version of the PRP method. On the other hand, we developed two novel approaches for unconstrained minimization problems involving nonlinear and differentiable cost functions. The structure of the thesis is organized as follows:

Chapter 1 introduces the fundamental notions essential to the theoretical foundation of the thesis, defining optimization problem, convergence rates, and so on. This chapter establishes the mathematical tools and concepts that underpin the subsequent algorithmic developments.

Chapter 2 delves into the unconstrained optimization problem in detail. We present some of the well-known line-search techniques for determining a step size. Also, it covers existence and uniqueness results, explores various descent directions, and presents a general scheme for unconstrained optimization algorithms. The chapter also includes a rigorous discussion of optimality conditions—both sufficient and necessary—with a special section on the convex case. The chapter concludes with an overview of algorithmic strategies for solving unconstrained minimization problems, highlighting gradient descent, Newton method, and quasi-Newton methods.

Chapter 3 focuses on conjugate gradient methods, beginning with the linear conjugate gradient algorithm and the computation of the step size. It then moves to nonlinear conjugate gradient algorithms, providing a detailed analysis of convergence conditions, such as those proposed by Zoutendijk. This chapter strengthens the theoretical foundations and highlights the practical considerations of employing conjugate gradient methods in optimization.

Chapter 4 presents our main contributions by developing two new conjugate gradient methods for solving nonlinear unconstrained optimization problems by introducing new conjugate gradient parameters, named hPB [33] and MZ [34]. This chapter

analyses the convergence properties of the proposed approaches under a new inexact line search conditions. A numerical experiments are also conducted to affirm the efficiency of the proposed approaches.

Chapter 5 introduces an improved version of the PRP (Polak-Ribière-Polyak) conjugate gradient method for unconstrained nonlinear optimization, by adapting a modified conditions to determine a step size. In this chapter, we study the convergence properties of the PRP conjugate gradient method under the new conditions. The generated descent direction and the convergence properties of the proposed method are studied, where the global convergence is proven under mild assumptions. The proposed approach is evaluated on various test functions, and a comparison with similar recent PRP variants is conducted.

Finally, for clarity, this thesis includes three original contributions, covering Chapters 4 and 5. The remainder of the document consists of a recent review of existing work.

The thesis concludes with a general summary of the various optimization methods discussed, along with perspectives for future research in this field.

1

Fundamental notions

This chapter gives an introduction to the basics of optimization and will help build the tools required for our in-depth understanding in the later chapters. Some fundamental concepts will be touched which will be required for further studies in optimization

1.1 Optimization problems

In mathematics, an optimization problems are a type of mathematical programming where the objective is to find the best solution among all possible feasible solutions. The goal is to optimize (maximize or minimize) a certain objective function, subject to a set of constraints. The optimization problem (CP) is formulated in the following way:

$$\begin{cases} \text{Minimize} \quad f(x) & \text{(cost function)} \\ \text{subject to} & \\ \qquad g_i(x) \leq 0 \quad \text{for } i = 1, \ldots, n_{\text{ineq}} & \text{(inequality constraints)} \\ \qquad h_j(x) = 0 \quad \text{for } j = 1, \ldots, n_{\text{eq}} & \text{(equality constraints)} \end{cases} \tag{CP}$$

Here, $F \subset \mathbb{R}^n$ denotes the feasible set, and x is the optimization variable (or decision variable). We assume that: We assume that

- the functions f, g_i, and $h_j : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable;
- the number of inequality and equality constraints is finite (possibly zero).

Furthermore, we assume that $F = \mathbb{R}^n$, i.e., we consider only unconstrained optimization problems.

Definition 1.1. (Elementary notions).

(i) The set

$$F := \{x \in \mathbb{R}^n | g_i(x) \leq 0, \, \forall i = 1, \ldots, n_{\text{ineq}}, \, h_j(x) = 0 \, \forall j = 1, \ldots, n_{\text{eq}}\}. \tag{1.1}$$

associated with an optimization problem (CP) is termed the feasible set. Any $x \in F$ is termed a feasible point.

(ii) The inequality $g_i(x) \leq 0$ is called active at a point x if $g_i(x) = 0$. It is called inactive in case $g_i(x) < 0$. It is called violated if $g_i(x) > 0$ holds.

(iii) The value

$$f^* := \inf_{x \in F} f(x).$$

is termed the infimal value of problem (CP).

(iv) In case $F = \emptyset$, the problem (CP) is said to be infeasible. In that case, we have $f^* = \infty$. In case $f^* = -\infty$, the problem is said to be unbounded.

(v) A point $x^* \in F$ is a global minimizer or globally optimal solution of (CP) if

$$f(x^*) \leq f(x) \quad \text{for all} \quad x \in F.$$

Equivalently, $x^* \in F$ is a global minimizer if $f(x^*) = f^*$. In this case, the infimal value f^* is also referred to us as the global minimum or globally optimal value of (CP).

(vi) A global minimizer x^* is strict in case

$$f(x^*) < f(x) \quad \text{for all} \quad x \in F, \; x \neq x^*.$$

(vii) A point $x^* \in F$ is a local minimizer or locally optimal solution of (CP) if there exists a neighborhood $U(x^*)$ such that

$$f(x^*) \leq f(x) \quad \text{for all} \quad x \in F \cap U(x^*).$$

In this case, $f(x^*)$ is also referred to us as a local minimum or a locally optimal value of (CP).

(viii) A local minimizer x^* is strict in case

$$f(x^*) < f(x) \quad \text{for all} \quad x \in F \cap U(x^*),\ x \neq x^*.$$

(ix) An optimization problem (CP) is solvable if it has at least one global minimizer, i.e., if the optimal value is attained at some point. Otherwise, the problem is unsolvable.

Definition 1.2. (Classification of optimization problems).

(i) An optimization problem (CP) is said to be unconstrained in case $n_{\text{ineq}} = n_{\text{eq}} = 0$. Otherwise, it is said to be equality constrained and/or inequality constrained.

(ii) Inequality constraints of the simple kind

$$l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n$$

with bounds $l_i, u_i \in \mathbb{R}$ are called bound constraints.

(iii) When f is a quadratic polynomial and g_i and h_j are affine linear functions, (CP) is called a quadratic optimization problem or a quadratic program (QP).

(iv) In the general case, i.e., when the problem (CP) is not a quadratic program, we refer to it as a nonlinear optimization problem or nonlinear program (NLP).

1.1.1 Convergence Rates

We denote vector-valued sequence $\{x_k\}_{k\in\mathbb{N}}$, rather than (x_k), in order to avoid a conflict of notation with the components of a vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$.

A subsequence of $\{x_k\}_{k\in\mathbb{N}}$ generated by a strictly increasing mapping $\ell \mapsto k(\ell) \in \mathbb{N}$ is denoted by $\{x_{k(\ell)}\}$.

In optimization theory, several notions of convergence rates are introduced to characterize the speed at which a sequence approaches its limit, particularly for iterates generated by optimization algorithms.

Definition 1.3 (Q-convergence rates). [1] Let $\{x_k\} \subset \mathbb{R}^n$ be a sequence converging to $x^* \in \mathbb{R}^n$, and let M be an inner product on $\mathbb{R}^n$.

[1] The letter "Q" refers to "quotient".

1. The sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to x^* **Q-linearly** with respect to the M-norm if there exists a constant $c \in (0,1)$ such that

$$\|x_{k+1} - x^*\|_M \leq c\,\|x_k - x^*\|_M \tag{1.2}$$

for all sufficiently large $k \in \mathbb{N}$.

2. The sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to x^* **Q-superlinearly** with respect to the M-norm if

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|_M}{\|x_k - x^*\|_M} = 0. \tag{1.3}$$

3. The sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to x^* **Q-quadratically** with respect to the M-norm if there exists a constant $C > 0$ such that

$$\|x_{k+1} - x^*\|_M \leq C\|x_k - x^*\|_M^2 \tag{1.4}$$

for all sufficiently large $k \in \mathbb{N}$.

Remark 1.4. Q-superlinear and Q-quadratic convergence are independent of the chosen norm (or inner product) on $\mathbb{R}^n$. In contrast, Q-linear convergence may be lost under a change of norm.

Definition 1.5 (R-convergence rates)**.** [2] Let $\{x_k\} \subset \mathbb{R}^n$ be a sequence and $x^* \in \mathbb{R}^n$. Let M be an inner product on $\mathbb{R}^n$.

1. The sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to x^* **R-linearly** with respect to the M-norm if there exists a null sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$ such that

$$\|x_k - x^*\|_M \leq \varepsilon_k \quad \forall k \in \mathbb{N}, \tag{1.5}$$

and $\{\varepsilon_k\}_{k\in\mathbb{N}}$ converges to zero Q-linearly.

2. The sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to x^* **R-superlinearly** with respect to the M-norm if there exists a null sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$ such that

$$\|x_k - x^*\|_M \leq \varepsilon_k \quad \forall k \in \mathbb{N}, \tag{1.6}$$

and $\{\varepsilon_k\}_{k\in\mathbb{N}}$ converges to zero Q-superlinearly.

[2] The letter "R" refers to "root".

3. The sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to x^* **R-quadratically** with respect to the M-norm if there exists a null sequence$\{\varepsilon_k\}_{k\in\mathbb{N}}$ such that

$$\|x_k - x^*\|_M \leq \varepsilon_k \quad \forall k \in \mathbb{N},, \tag{1.7}$$

and $\{\varepsilon_k\}_{k\in\mathbb{N}}$ converges to zero Q-quadratically.

Remark 1.6. R-convergence rates are weaker than their corresponding Q-convergence rates. While Q-convergence measures the reduction of the error at each iteration, R-convergence only requires the existence of a dominating sequence that converges at the prescribed rate.

1.2 Convexity

Convexity plays a very important role in optimization in general. In this class, however, we will rely on it only scarcely. We briefly recall here some elements of convexity.

Definition 1.7. (Convex set). A set $C \subset \mathbb{R}^n$ is convex if for all $x, y \in C$ and $\alpha \in [0, 1]$ imply $\alpha x + (1 - \alpha)y \in C$.

The condition in Definition (1.8) means that the entire line segment between x and y belongs to C.

Definition 1.8. (Convex set). A set $C \subset \mathbb{R}^n$ is convex if for all $x, y \in C$ and all $\alpha \in [0, 1]$, we have

$$\alpha x + (1 - \alpha)y \in C.$$

The condition in Definition (1.8) means that the entire line segment between x and y belongs to C.

Definition 1.9. (Convex function). A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be

(i) **convex** if

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall x, y \in \mathbb{R}^n, \ \forall \alpha \in [0, 1].$$

(ii) **strictly convex** if

$$f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha)f(y), \quad \forall x \neq y, \ \forall \alpha \in (0, 1).$$

(iii) μ-**strongly convex** (or **strongly convex** with parameter $\mu > 0$) if

$$f(\alpha x+(1-\alpha)y)+\frac{\mu}{2}\alpha(1-\alpha)\|x-y\|^2 \leq \alpha f(x)+(1-\alpha)f(y), \quad \forall x, y \in \mathbb{R}^n, \forall \alpha \in (0,1).$$

(iv) **concave**, **strictly concave**, or **strongly concave** if $-f$ is convex, strictly convex, or strongly convex, respectively.

Theorem 1.10 (Characterization of convexity via first-order derivatives)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable.*

(a) The following statements are equivalent:

(i) f is convex.

(ii) For all $x, y \in \mathbb{R}^n$,

$$f(x) - f(y) \geq \nabla f(y)^T(x - y).$$

(iii) For all $x, y \in \mathbb{R}^n$,

$$(\nabla f(x) - \nabla f(y))^T(x - y) \geq 0. \tag{1.8}$$

Equation (1.8) means that ∇f is a monotone operator.

(b) The following statements are equivalent:

(i) f is strictly convex.

(ii) For all $x, y \in \mathbb{R}^n$ such that $x \neq y$,

$$f(x) - f(y) > \nabla f(y)^T(x - y).$$

(iii) For all $x, y \in \mathbb{R}^n$ such that $x \neq y$,

$$(\nabla f(x) - \nabla f(y))^T(x - y) > 0. \tag{1.9}$$

Equation (1.9) means that ∇f is a strictly monotone operator.

(c) The following statements are equivalent:

(i) f is strongly convex.

(ii) There exists $\mu > 0$ such that for all $x, y \in \mathbb{R}^n$,

$$f(x) - f(y) \geq \nabla f(y)^T(x - y) + \frac{\mu}{2}\|x - y\|^2.$$

(iii) There exists $\mu > 0$ such that for all $x, y \in \mathbb{R}^n$,

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \mu \|x - y\|^2. \tag{1.10}$$

Equation (1.10) means that ∇f is a strongly monotone operator.

Proof. See [64] pages (8-9) □

Theorem 1.11 (Characterization of convexity via second-order derivatives)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable.*

(a) The following statements are equivalent:

(i) f is convex.

(ii) The Hessian matrix $\nabla^2 f(x)$ is positive semidefinite for all $x \in \mathbb{R}^n$ (i.e., it has only nonnegative eigenvalues).

(b) If the Hessian matrix $\nabla^2 f(x)$ is positive definite for all $x \in \mathbb{R}^n$, then f is strictly convex.

(c) The following statements are equivalent:

(i) f is strongly convex with parameter $\mu > 0$.

(ii) The smallest eigenvalue of the Hessian matrix $\nabla^2 f(x)$ satisfies

$$\lambda_{\min}\big(\nabla^2 f(x)\big) \geq \mu, \quad \forall x \in \mathbb{R}^n.$$

Proof. See [64] page 10 □

1.3 Unconstrained optimization problem

Now we are interested in some concepts and results related to the unconstrained optimization problem, which is typically formulated as:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth real-valued objective function, and there are no restrictions on the decision variables x

Definition 1.12. 1. $x^* \in \mathbb{R}^n$ is a global minimum of (UP) if and only if:

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

2. $x^* \in \mathbb{R}^n$ is a local minimum of (UP) if and only if there exists a neighborhood $V_\varepsilon(x^*)$ such that:
$$f(x^*) \leq f(x), \quad \forall x \in V_\varepsilon(x^*).$$

3. $x^* \in \mathbb{R}^n$ is a strict local minimum of (UP) if and only if there exists a neighborhood $V_\varepsilon(x^*)$ such that:
$$f(x^*) < f(x), \quad \forall x \in V_\varepsilon(x^*),\ x \neq x^*.$$

1.3.1 Existence and uniqueness results

Theorem 1.13. ***(Weierstrass).*** *If a subset $F \subset \mathbb{R}^n$ is compact and f is continuous on F, then f admits at least one minimum on F. That is,*

$$\exists x^* \in F : f(x^*) = \min_{x \in F} f(x) \iff \exists x^* \in F, \quad f(x^*) \leq f(x),\ \forall x \in F.$$

Proof. See[55] page 78. □

Theorem 1.14. ***(Existence).*** *If $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous and coercive function on $\mathbb{R}^n$, then problem* (UP) *admits at least one minimum $x^* \in \mathbb{R}^n$, i.e.,*

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

Proof. Let $f^* = \min f(x)$. We have $f^* < +\infty$ because f is coercive. Let $(x_k)_{k \in \mathbb{N}} \subset \mathbb{R}^n$ be a minimizing sequence, i.e.,
$$\lim_{k \to +\infty} f(x_k) = f^*.$$
We show that (x_k) is bounded. If instead $\lim_{k \to +\infty} \|x_k\| = +\infty$, then by the coercivity of f we would have
$$\lim_{k \to +\infty} f(x_k) = +\infty,$$
which contradicts the fact that
$$\lim_{k \to +\infty} f(x_k) = f^* < +\infty.$$
Hence (x_k) is bounded, and therefore it admits a convergent subsequence (convergence in $\mathbb{R}^n$) with limit x^*. By continuity of f, we obtain
$$\lim_{k \to +\infty} f(x_k) = f(x^*) = f^*.$$
Thus x^* is a solution to problem (UP). □

Theorem 1.15. ***(Sufficient condition for uniqueness).*** *If the function $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex, then the minimizer $x^* \in \mathbb{R}^n$ is unique satisfies:*

$$f(x^*) < f(x), \quad \forall x \in \mathbb{R}^n,\ x \neq x^*.$$

Proof. Suppose, by contradiction, that there exist two distinct minimizers $x_1, x_2 \in \mathbb{R}^n$, with $x_1 \neq x_2$, such that

$$f(x_1) = f(x_2) = m = \min_{x \in \mathbb{R}^n} f(x).$$

Consider their midpoint:

$$z = \tfrac{1}{2}(x_1 + x_2).$$

By the strict convexity of f, we have

$$f(z) < \tfrac{1}{2} f(x_1) + \tfrac{1}{2} f(x_2) = m.$$

This contradicts the assumption that m is the minimum value of f. Therefore, the minimizer of a strictly convex function is unique. □

Theorem 1.16. ***(Existence and Uniqueness).*** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function that is continuous, coercive, and strictly convex. Then there exists a unique $x^* \in \mathbb{R}^n$ such that*

$$f(x^*) = \min_{x \in \mathbb{R}^n} f(x),$$

i.e., the problem (UP) *admits a unique global minimizer x^*.*

1.3.2 Descent Direction

Definition 1.17. Let $f : \mathbb{R}^n \to \mathbb{R}$ and $x^* \in \mathbb{R}^n$. A vector $d \in \mathbb{R}^n$ is called a **descent direction** at x^* if there exists $\delta > 0$ such that

$$f(x^* + \alpha d) < f(x^*), \quad \forall \alpha \in (0, \delta).$$

Theorem 1.18. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable at $x^* \in \mathbb{R}^n$. If $d \in \mathbb{R}^n$ satisfies*

$$\nabla f(x^*)^T d < 0,$$

then d is a descent direction at x^*.

Proof. Since f is differentiable at x^*, we have

$$f(x^* + \alpha d) = f(x^*) + \alpha \nabla f(x^*)^T d + o(\alpha), \quad \text{as } \alpha \to 0.$$

Dividing by $\alpha > 0$, we obtain

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

Taking the limit as $\alpha \to 0^+$,

$$\lim_{\alpha \to 0^+} \frac{f(x^* + \alpha d) - f(x^*)}{\alpha} = \nabla f(x^*)^T d < 0.$$

Therefore, there exists $\delta > 0$ such that

$$\frac{f(x^* + \alpha d) - f(x^*)}{\alpha} < 0, \quad \forall \alpha \in (0, \delta).$$

Multiplying by $\alpha > 0$ gives

$$f(x^* + \alpha d) < f(x^*), \quad \forall \alpha \in (0, \delta).$$

Hence, d is a descent direction at x^*. □

1.3.3 General scheme of unconstrained optimization algorithms

Assume that d_k is a descent direction at point x_k, then we consider the next point as:

$$x_{k+1} = x_k + \alpha_k d_k, \quad \alpha_k \in (0, \delta) \quad \delta > 0.$$

By definition of a descent direction it results that :

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

A good choice of d_k and α_k thus allows the construction of a multitude of optimization algorithms.

Examples of descent directions

If $d_k = -\nabla f(x_k)$ and $\nabla f(x_k) \neq 0$, we obtain the **gradient method**.
If $d_k = -H(x_k)^{-1} \nabla f(x_k)$, where $H(x_k)$ is the Hessian matrix, and $H(x_k)$ is positive definite, then this is **Newton's method**.

Examples of step size α_k

The step size α_k is typically chosen optimally, meaning that α_k must satisfy:

$$f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k) : \quad \forall \alpha \in (0, \delta).$$

In other words, at each iteration we are reduced to studying a minimization problem of a single real variable. This is what we call line search.

Theorem 1.19. *(First-order necessary optimality condition).*
Suppose that x^* is a local minimizer of (UP) *and that f is differentiable at x^*. Then*

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

Proof. Let $d \in \mathbb{R}^n$ be arbitrary and define the curve

$$\gamma : (-\delta, \delta) \to \mathbb{R}^n, \quad \gamma(t) := x^* + td.$$

For sufficiently small $\delta > 0$, the curve $\gamma(t)$ lies in a neighborhood of local optimality of x^*, and hence the function $f \circ \gamma$ has a local minimum at $t = 0$.

Consequently,

$$\frac{f(\gamma(t)) - f(\gamma(0))}{t} = \frac{f(x^* + td) - f(x^*)}{t} \begin{cases} \geq 0, & t > 0, \\ \leq 0, & t < 0. \end{cases}$$

Since f is differentiable at x^*, the above difference quotient converges to $\nabla f(x^*)^T d$ as $t \to 0$. Therefore,

$$\nabla f(x^*)^T d = 0.$$

Because $d \in \mathbb{R}^n$ is arbitrary, we conclude that $\nabla f(x^*) = 0$.

□

Theorem 1.20. *(Second-order necessary optimality condition).*
Suppose that x^* is a local minimizer of (UP) *and that f is twice differentiable at x^*. Then the Hessian $\nabla^2 f(x^*)$ is positive semidefinite.*

Proof. Let $d \in \mathbb{R}^n$, and suppose f is twice differentiable at x^*. Then, for all $\alpha \neq 0$,

$$f(x^* + \alpha d) = f(x^*) + \tfrac{1}{2}\alpha^2 \, d^T \nabla^2 f(x^*) \, d + \alpha^2 \|d\|^2 \, \lambda(x^*, \alpha d), \tag{2.9}$$

where $\lambda(x^*, \alpha d) \to 0$ as $\alpha \to 0$. Hence,

$$\frac{f(x^* + \alpha d) - f(x^*)}{\alpha^2} = \tfrac{1}{2}\, d^T \nabla^2 f(x^*)\, d + \|d\|^2\, \lambda(x^*, \alpha d). \tag{2.10}$$

Since x^* is a local minimizer, there exists $\delta > 0$ such that

$$\frac{f(x^* + \alpha d) - f(x^*)}{\alpha^2} \geq 0, \qquad \forall\, \alpha \in (0, \delta]. \tag{2.11}$$

Passing to the limit as $\alpha \to 0$ with $\alpha \neq 0$, we obtain

$$d^T \nabla^2 f(x^*)\, d \geq 0, \qquad \forall\, d \in \mathbb{R}^n.$$

Therefore, the Hessian $\nabla^2 f(x^*)$ is positive semidefinite. □

1.3.4 Sufficient optimality conditions

Theorem 1.21. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable at point $x^* \in \mathbb{R}^n$. If $\nabla f(x^*) = 0$ and $H(x^*)$ is positive definite, then x^* is a strict local minimum of (f).*

Proof. From Taylor expansion:

$$f(x_k) = f(x^*) + \frac{1}{2}(x_k - x^*)^T H(x^*)(x_k - x^*) + \|x_k - x^*\|^2 \alpha(x^*, x_k - x^*),$$

If x^* is not a strict local minimum, then there exists a sequence $\{x_k\}$ with $x_k \to x^*$ and $f(x_k) \leq f(x^*)$. Substituting into the expansion leads to a contradiction with $H(x^*)$ being positive definite. □

1.3.5 Necessary and sufficient conditions (convex case)

Theorem 1.22. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and differentiable at $x^* \in \mathbb{R}^n$.*

1. *If x^* is a local minimum of problem (UP), then it is also a global minimum.*
2. *x^* is a local minimum of problem (UP) if and only if $\nabla f(x^*) = 0$.*

Proof. **(i) Local minimizer ⇒ global minimizer.**

Suppose that x^* is a local minimizer. Assume, by contradiction, that there exists $y \in \mathbb{R}^n$ such that $f(y) < f(x^*)$. By convexity of f, for any $t \in [0, 1]$, we have

$$f\big((1-t)x^* + ty\big) \leq (1-t)f(x^*) + tf(y).$$

Since $f(y) < f(x^*)$, the right-hand side is strictly smaller than $f(x^*)$ for all $t \in (0, 1]$. Hence, for sufficiently small $t > 0$, the point

$$z_t = (1 - t)x^* + ty$$

lies in a neighborhood of x^* and satisfies $f(z_t) < f(x^*)$, which contradicts the local minimality of x^*. Therefore, no such y exists, and x^* is a global minimizer.

(ii) Equivalence with $\nabla f(x^*) = 0$.

(a) If $\nabla f(x^*) = 0$, then x^* is a global minimizer.

Since f is convex and differentiable, the first-order condition for convex functions holds:

$$f(y) \geq f(x^*) + \nabla f(x^*)^T(y - x^*) \quad \forall y \in \mathbb{R}^n.$$

If $\nabla f(x^*) = 0$, this inequality reduces to $f(y) \geq f(x^*)$ for all $y \in \mathbb{R}^n$, which implies that x^* is a global (and hence local) minimizer.

(b) If x^* is a local minimizer, then $\nabla f(x^*) = 0$.

Since x^* is a local minimizer, the directional derivative of f at x^* in any direction $d \in \mathbb{R}^n$ must be nonnegative. The directional derivative is given by

$$f'(x^*; d) = \lim_{t \downarrow 0} \frac{f(x^* + td) - f(x^*)}{t} = \nabla f(x^*)^T d.$$

Thus,

$$\nabla f(x^*)^T d \geq 0 \quad \forall d \in \mathbb{R}^n.$$

By considering both d and $-d$, we conclude that $\nabla f(x^*)^T d = 0$ for all $d \in \mathbb{R}^n$. Hence, $\nabla f(x^*) = 0$.

□

2

Unconstrained optimization methods

In this chapter, we present the most well-known optimization algorithms for solving unconstrained optimization problems. However, we first present the most well-known techniques for calculating the step size α.

2.1 Line search

consider the following unconstrained minimization problem

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

Where $f : \mathbb{R}^n \to \mathbb{R}$: is a smooth cost function

Line search is the procedure for optimizing a univariate function and is a process for choosing a step α_k along a descent direction d_k. The choice of α_k greatly affects the speed and convergence of the algorithm. Let be the following one variable function

$$\psi_k(\alpha) = f(x_k + \alpha d_k).$$

Hence, we aim to find a value α_k, such that

$$\psi_k(\alpha) \leq \psi_k(0).$$

In one-dimensional optimization, there are two primary classes of methods.

2.1.1 Exact line search

The exact line search entails determining α_k in such a way that it minimizes the objective function in the direction d_k, i.e.,

$$\min_{\alpha \in \mathbb{R}_+^*} \psi_k(\alpha).$$

This is referred to us as Cauchy's rule, and the step determined by this rule is known as Cauchy's step or the optimal step. In certain instances, we may favor the smallest stationary point of ψ_k, which diminishes through this function.

$$\alpha_k = \inf\{\alpha > 0 : \psi_k(\alpha) \leq \psi_k(0)\}.$$

We discuss Curry's principle, and the step defined by this principle is referred to us as Curry's step, these two principles are sometimes denoted as an exact linear search.

Disadvantages of exact line search

For an arbitrary nonlinear function,

- It is not possible to define Cauchy's or Curry's move in this context.

- The process of determining these steps typically requires a significant amount of time.

- The calculation cannot be carried out with infinite precision.

- The additional efficiency produced by precise linear search does not compensate for the time lost in specifying such a step.

2.2 Uncertainty interval

In the majority of contemporary optimization algorithms, we abstain from conducting an exact line search, because determining α_k entails the computation of the function ψ_k a large number of times and this can be dissuasive from the point of view of calculation time.

In practice, we rather look for a value of α^* which ensures a sufficient decrease of f. This gives rise to the concept of a safety interval.

Definition 2.1. We designate $[\alpha_l, \alpha_r]$ as a safety interval if it organizes the values of α in the following order:

- If $\alpha < \alpha_l$ then α is considered too small,
- If $\alpha_r \geq \alpha \geq \alpha_l$ then α is satisfying,
- If $\alpha > \alpha_r$ then α is considered too large.

Algorithm 1: The safety interval algorithm

Step 0: (Initialization) Set $\alpha_{l,1} = \alpha_{r,1} = 0$, choose $\alpha_1 > 0$, and set $k = 1$.

Step 1:
- If α_k is suitable, then set $\alpha^* = \alpha_k$ and stop.
- If α_k is too small, then set $\alpha_{l,k+1} = \alpha_k$, $\alpha_{r,k} = \alpha_k r, k$, and proceed to Step 2.
- If α_k is too large, then set $\alpha_{r,k+1} = \alpha_k$, $\alpha_{l,k+1}] = \alpha_{l,k}$, and proceed to Step 2.

Step 2:
- If $\alpha_{r,k+1} = 0$ determine $\alpha_{k+1} \in]\alpha_{l,k+1}, +\infty[$.
- If $\alpha_{r,k+1} \neq 0$ determine $\alpha_{k+1} \in]\alpha_{l,k+1}, \alpha_{r,k+1}[$.

replace k with $k + 1$ and go to step 1..

Instead of computing α_k to minimize, we choose to impose conditions that are less restrictive, more easily confirmed, which nevertheless make it capable of aiding in the convergence of the algorithms. Specifically, there will no longer be a single step (or a few steps) verifying these conditions; instead, there will be an entire interval of steps (or multiple intervals), this will also simplify the search process. This is the approach we adopt with the Armijo, Goldstein, and Wolfe rules outlined in the following subsection.

2.2.1 Inexact line search

Finding a step is frequently referred to us as linear search since it amounts to examine the behavior of the function along the line given by the direction of descent.

Even though precise line searches result in an approximately optimal solution. However, it demands numerous observations in every iteration of the main algorithm. For this many mathematicians calculation (Armijo, Goldstein, Wolfe, Albaali, Lemaréchal, Fletcher ...) develop several rules.

The objective of this subsection is to introduce the principal tests. Initially, let us present the scheme of Armijo line search.

2.2.2 Armijo method

Armijo's rule places a restriction on the selection of α_k to ensure local minimization of f. A natural requirement is that f decreases as much as a portion $\rho \in]0,1[$ (due to the behavior predicted by the linear model of f in x_k). This results in the subsequent inequality occasionally referred to us as the Armijo condition:

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

where $g_k = \nabla f(x_k)$, or

$$\psi_k(\alpha_k) \leq \psi_k(0) + \rho \alpha_k \psi_k'(0).$$

The Armijo method can be implemented using the following algorithm.

Algorithm 2: Armijo line search

- **Step 0:** (Initialization) $\alpha_{l,1} = \alpha_{r,1} = 0$, $\alpha_1 > 0$, $\rho \in]0,1[$ set $k = 1$ and go to step 1.
- **Step 1:**
  - If $\psi_k(\alpha_k) \leq \psi_k(0) + \rho \alpha_k \psi_k'(0)$: STOP ($\alpha^* = \alpha_k$).
  - If $\psi_k(\alpha_k) > \psi_k(0) + \rho \alpha_k \psi_k'(0)$, then$\alpha_{r,k+1} = \alpha_k, \quad \alpha_{l,k+1} = \alpha_{l,k}$ and go to step 2.
- **Step 2:**
  - If$\alpha_{r,k+1} = 0$ determine $\alpha_{k+1} \in]\alpha_{l,k+1}, +\infty[$
  - If $\alpha_{r,k+1} \neq 0$ determine $\alpha_{k+1} \in]\alpha_{l,k+1}, \alpha_{r,k+1}[$ replace k with $k+1$ and go to step 1.

Theorem 2.2. *Consider the function $\psi : \mathbb{R}_+^* \to \mathbb{R}$, which is defined as $\psi_k(\alpha) = f(x_k + \alpha d_k)$. Suppose that ψ_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 [20] page 308. □

Step length degradation is the main drawback of the Armijo method. Although the step sizes remain strictly positive, they may approach zero arbitrarily closely, which

can significantly slow down the progress of the method.
We now turn to the Goldstein–Price rule.

2.2.3 Goldstein-Price rule

Goldstein's conditions, as referenced in [24, 25], ensure that the step size α_k achieves a significant decrease while avoiding an excessively small value. The Goldstein rule is particularly useful when evaluating the function's gradient is impractical or computationally expensive.

By introducing a second inequality to Armijo's rule, the Goldstein-Price rule is obtained. Let ρ be a constant such that$\rho \in]0, 1/2]$. The two inequalities of the Goldstein-Price rule are defined as follows:

$$f(x_k) + (1-\rho)\alpha_k g_k^T d_k \leq f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k g_k^T d_k, \quad \rho \in]0, 1/2],$$

otherwise, says

$$\psi_k(0) + (1-\rho)\alpha_k \psi_k'(0) \leq \psi_k(\alpha_k) \leq \psi_k(0) + \rho\alpha_k \psi_k'(0).$$

Algorithm 3: Goldstein-Price line search

Step 0: (Initialization) Choose $\alpha_{l,1} = \alpha_{r,1} = 0$,, choose $\alpha_1 > 0$, $\rho \in]0, 1/2]$, set $k = 1$ and go to step 1.

Step 1: If $\psi_k(0) + (1-\rho)\psi_k'(0)\alpha_k \leq \psi_k(\alpha_k) \leq \psi_k(0) + \rho\psi_k'(0)\alpha$, **STOP** ($\alpha^* = \alpha_k$).
If $\psi_k(\alpha_k) > \psi_k(0) + \rho\psi_k'(0)\alpha_k$, then $\alpha_{r,k+1} = \alpha_k, \alpha_{l,k+1} = \alpha_{l,k}$ and go to step 2

Step 2: If $\psi_k(\alpha_k) < \psi_k(0) + (1-\rho)\psi_k'(0)\alpha_k$, then $\alpha_{r,k+1} = \alpha_{r,k}, \alpha_{l,k+1} = \alpha_k$ and go to step 2.

Step 3: If $\alpha_{r,k+1} = 0$ determine $\alpha_{k+1} \in [\alpha_{l,k+1}, +\infty)$
If $\alpha_{r,k+1} \neq 0$ determine $\alpha_{k+1} \in [\alpha_{l,k+1}, \alpha_{r,k+1}]$.

The following theorem ensures the existence of a step size satisfying the Goldstein-Price conditions under appropriate assumptions on f.

Theorem 2.3. *Let $\varphi : \mathbb{R}_+^* \to \mathbb{R}$ be defined by $\varphi_k(\alpha) = f(x_k + \alpha d_k)$. If φ_k is continuous and lower bounded, d_k is a descent direction at x_k, and $\rho \in]0, 1/2]$, then the set of steps satisfying the Goldstein-Price rule is non-empty.*

Proof. See [20], page 308. □

2.2.4 Wolfe line search

The commonly employed criteria for determining the step size in line search are the well-known standard Wolfe conditions [53, 54],

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

and

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \tag{2.2}$$

in other words,

$$\psi_k(\alpha_k) \leq \psi_k(0) + \rho \psi_k'(0) \alpha_k, \tag{2.3}$$

and

$$\psi_k'(\alpha_k) \geq \sigma \psi_k'(0). \tag{2.4}$$

where $0 < \rho < \sigma < 1$. The initial condition (2.1), referred to us as the Armijo condition, guarantees a significant reduction in the value of the objective function, while the second condition (2.2), known as the curvature condition, prevents the acceptance of excessively small step sizes.

2.2.5 Strong wolfe line search

It should be noted that the step-size is calculated by using (2.1) and (2.2) conditions may not be near enough to a minimum for the function $\psi_k(\alpha)$. In such circumstances, the strong Wolfe line search (SWL) can be chosen, this research combines (2.1) and the subsequent condition:

Algorithm 4: Weak Wolfe line search

Step 0: (Initialization) Choose $\alpha_{l,1} = \alpha_{r,1} = 0$, choose $\alpha_1 > 0$, $\rho \in]0, 1[$, $\sigma \in]\rho, 1[$, set $k = 1$ and go to step 1.

Step 1: If $\psi_k(\alpha_k) \leq \psi_k(0) + \rho\psi_k'(0)\alpha_k$ and $\psi_k'(\alpha_k) \geq \sigma\psi_k'(0)$: **STOP** ($\alpha^* = \alpha_k$).
If $\psi_k(\alpha_k) > \psi_k(0) + \rho\psi_k'(0)\alpha_k$, then $\alpha_{r,k+1} = \alpha_k, \alpha_{l,k+1} = \alpha_{l,k}$ and go to step 2
If $\psi_k'(\alpha_k) < \sigma\psi_k'(0)$, then $\alpha_{r,k+1} = \alpha_{r,k}, \alpha_{l,k+1} = \alpha_k$ and go to step 2

Step 2: If $\alpha_{r,k+1} = 0$ determine $\alpha_{k+1} \in]\alpha_{l,k+1}, +\infty[$.
If $\alpha_{r,k+1} \neq 0$ determine $\alpha_{k+1} \in]\alpha_{l,k+1}, \alpha_{r,k+1}[$.

$$\left|g_{k+1}^T d_k\right| \leq \sigma \left|g_k^T d_k\right| \quad \text{with} \quad \sigma \in]\rho, 1[, \tag{2.5}$$

in other words,

$$|\psi_k'(\alpha_k)| \leq -\sigma\psi_k'(0), \quad \sigma \in]\rho, 1[. \tag{2.6}$$

If σ approaches zero, then the step size that satisfies (2.1) and (2.5) tends to be the optimal step size.

Observe that if a step size α_k satisfies the strong Wolfe line search, then it satisfies the standard Wolfe conditions.

Algorithm 5: The strong Wolfe line search

Step 0: (Initialization) Choose $\alpha_{l,1} = \alpha_{r,1} = 0$, choose $\alpha_1 > 0$, $\rho \in]0, 1[$, $\sigma \in]\rho, 1[$, set $k = 1$ and go to step 1.

Step 1: If $\psi_k(\alpha_k) \leq \psi_k(0) + \rho\psi_k'(0)\alpha_k$ and $|\psi_k'(\alpha_k)| \leq -\sigma\psi_k'(0)$: **STOP** ($\alpha^* = \alpha_k$).
If $\psi_k(\alpha_k) > \psi_k(0) + \rho\psi_k'(0)\alpha_k$, then $\alpha_{r,k+1} = \alpha_k, \alpha_{l,k+1} = \alpha_{l,k}$ and go to step 2.
If $|\psi_k'(\alpha_k)| > -\sigma\psi_k'(0)$, then $\alpha_{r,k+1} = \alpha_{r,k}, \alpha_{l,k+1} = \alpha_k$ and go to step 2

Step 2: If $\alpha_{r,k+1} = 0$ determine $\alpha_{k+1} \in]\alpha_{l,k+1}, +\infty[$.
If $\alpha_{r,k+1} \neq 0$ determine $\alpha_{k+1} \in]\alpha_{l,k+1}, \alpha_{r,k+1}[$.

Theorem 2.4. *Let $\psi : \mathbb{R}_+ \to \mathbb{R}$ be a differentiable function bounded below, where*

$$\psi(\alpha) = f(x_k + \alpha d_k),$$

and suppose that d_k is a descent direction at x_k. Then, the set of step lengths satisfying the strong Wolfe line search conditions is nonempty.

Proof. See [20] pages 310-311. □

2.3 Algorithms for solving unconstrained optimization

Numerous algorithms have been devised, ranging from elementary first-order methods to advanced second-order schemes. This section provides a concise overview of these algorithms and their main characteristics.

2.3.1 Gradient method

Cauchy introduced the gradient method in 1847, establishing it as one of the simplest and most fundamental minimization techniques for unconstrained optimization [50]. Assume that the objective function f is continuously differentiable in a neighborhood of x_k and that

$$\nabla f(x_k) \neq 0.$$

Using the Taylor expansion of f around x_k, we obtain

$$f(x) = f(x_k) + (x - x_k)^T \nabla f(x_k) + o(\|x - x_k\|), \quad x \to x_k.$$

If

$$x - x_k = \alpha_k d_k,$$

then the search direction d_k is required to satisfy

$$d_k^T \nabla f(x_k) < 0, \quad \forall k \in \mathbb{N},$$

which defines a descent direction. Consequently, for sufficiently small step sizes $\alpha_k > 0$, we have

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

By the Cauchy–Schwarz inequality, it follows that

$$d_k^T \nabla f(x_k) \geq -\|d_k\| \, \|\nabla f(x_k)\|.$$

It is well known that choosing

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

minimizes the inner product $d_k^T \nabla f(x_k)$ over all unit vectors. Hence, the negative gradient direction represents the steepest descent direction.

Algorithm 6: Gradient algorithm

1. **Step 1:** Select $x_0 \in \mathbb{R}^n$ and let $\epsilon > 0$ be the termination criterion, $k = 0$.
2. **Step 2:** If $\|\nabla f(x_k)\|_2 < \epsilon$, terminate; otherwise, continue to Step 3.
3. **Step 3:** Compute the search direction $d_k = -\nabla f(x_k)$.
4. **Step 4:** Find step size $\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha d_k)$ and update $x_{k+1} = x_k + \alpha_k d_k$.
5. **Step 5:** Set $k = k + 1$ and go to Step 2.

2.3.2 Newton method

Suppose we want to solve the problem (UP). At $x = \bar{x}$, $f(x)$ can be approximated by:

$$f(x) \approx h(x) = f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T H(\bar{x})(x - \bar{x}),$$

which corresponds to the second-order Taylor approximation of f at the point $x = \bar{x}$. In this expression, $\nabla f(\bar{x})$ represents the gradient of f at $\bar{x}$, while $H(\bar{x})$ represents the Hessian matrix of f at $\bar{x}$.

Note that $h(x)$ is a quadratic function, which is minimized by solving $\nabla h(x) = 0$. Since the gradient of $h(x)$ is:

$$\nabla h(x) = \nabla f(\bar{x}) + H(\bar{x})(x - \bar{x}),$$

we are therefore motivated to solve:

$$\nabla f(\bar{x}) + H(\bar{x})(x - \bar{x}) = 0,$$

which yields:

$$x = \bar{x} - H(\bar{x})^{-1}\nabla f(\bar{x}).$$

The direction $-H(\bar{x})^{-1}\nabla f(\bar{x})$ is called the *Newton direction*, or the Newton step at $x = \bar{x}$.

This leads to the following algorithm for solving (UP):

Algorithm 7: Newton algorithm

1. **Step 1:** Select $x_0 \in \mathbb{R}^n$ and let $\epsilon > 0$ be the termination criterion, $k = 0$.
2. **Step 2:** If $\|\nabla f(x_k)\| < \epsilon$, terminate; otherwise, continue to Step 3.
3. **Step 3:** Compute the search direction $d_k = -H(x_k)^{-1}\nabla f(x_k)$.
4. **Step 4:** Update $x_{k+1} = x_k + d_k$.
5. **Step 5:** Set $k = k + 1$ and go to Step 2.

2.3.3 Quasi-Newton methods

The first quasi-Newton algorithm was developed by **W. C. Davidon** in the mid-1950s, marking a milestone in nonlinear optimization [12]. Davidon sought to solve a lengthy and computationally intensive optimization problem but faced significant challenges due to the limited computational capabilities of computers at that time when using conventional methods. To overcome these difficulties, he devised the quasi-Newton approach. Later, Fletcher and Powell [18] demonstrated that this algorithm was more efficient and reliable than existing methods. Subsequently, numerous variants emerged. Among the most well-known are **Broyden** method [5], the **SR1 formula** [6], the **DFP** method [18], and the **BFGS** method [23, 48].

In optimization, Newton's method leverages both first- and second-order derivatives, namely, the gradient $\nabla f(x)$ and the Hessian matrix $\nabla^2 f(x)$ to locate optimal points in multivariate problems. It applies to a twice continuously differentiable function $f(x)$ and aims to solve the nonlinear system $\nabla f(x) = \mathbf{0}$ to identify stationary

points. The iterative update is given by

$$x_{k+1} = x_k - H_k^{-1} \nabla f(x_k),$$

where k denotes the iteration index and $H_k = \nabla^2 f(x_k)$.

Convergence is typically declared when one of the following criteria is satisfied:

$$\|\nabla f(x_k)\| < \epsilon, \quad \text{or} \quad |f(x_{k+1}) - f(x_k)| < \epsilon,$$

where $\epsilon > 0$ is a prescribed tolerance.

Despite its rapid local convergence, Newton's method suffers from two major drawbacks:

1. The objective function must be twice continuously differentiable, and the Hessian matrix must be positive definite in the case of minimization.

2. The computation of the Hessian matrix and its inverse is computationally expensive, especially for large-scale optimization problems.

Quasi-Newton methods address these limitations while preserving the fundamental structure of Newton's method. The key innovation lies in approximating the Hessian matrix by a sequence of matrices B_k (or their inverses B_k^{-1}), thereby avoiding its explicit computation. The various quasi-Newton schemes differ primarily in the strategies used to update these approximations by exploiting curvature information obtained from gradient evaluations.

2.3.4 Theory and algorithm

To illustrate the basic idea behind quasi-Newton methods, we start by building a quadratic model of the objective function around the current iterate x_k:

$$m_k(\mathbf{p}) = f(x_k) + \nabla f(x_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T B_k \mathbf{p},$$

where B_k is an $n \times n$ symmetric positive definite matrix approximating the Hessian $\nabla^2 f(x_k)$. The minimizer of this quadratic model provides the search direction:

$$\mathbf{p}_k = -B_k^{-1} \nabla f(x_k).$$

The new iterate is then updated by

$$x_{k+1} = x_k + \alpha_k \mathbf{p}_k,$$

where $\alpha_k > 0$ is the step size, usually chosen to satisfy the Wolfe conditions. This iteration resembles Newton's method but uses the approximate Hessian B_k instead of the exact Hessian.

Secant Equation

To maintain curvature information in B_{k+1}, we require that the change in gradient matches the change predicted by the quadratic model:

$$B_{k+1}\mathbf{s}_k = \mathbf{y}_k, \tag{2.7}$$

where

$$\mathbf{s}_k = x_{k+1} - x_k = \alpha_k \mathbf{p}_k, \quad \mathbf{y}_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

To ensure B_{k+1} remains symmetric positive definite, it is sufficient that

$$\mathbf{s}_k^T \mathbf{y}_k > 0,$$

which is guaranteed under the Wolfe line search conditions. Among all symmetric matrices satisfying the secant equation (2.7), we typically choose the one closest to B_k in the sense of the weighted Frobenius norm:

$$B_{k+1} = \arg \min_{B=B^T,\, B\mathbf{s}_k=\mathbf{y}_k} \|B - B_k\|_F.$$

BFGS Method (Most Widely Used)

The BFGS update for the Hessian approximation $B_k \approx \nabla^2 f(x_k)$ is

$$B_{k+1} = \left(I - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) B_k \left(I - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}.$$

In practice, we maintain and update the inverse Hessian $H_k = B_k^{-1}$, which is used to compute the search direction directly:

$$H_{k+1} = \left(I - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) H_k \left(I - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}.$$

DFP Method (Alternative)

The DFP method updates the inverse Hessian approximation $H_k \approx (\nabla^2 f(x_k))^{-1}$ as follows:

$$H_{k+1} = H_k - \frac{H_k \mathbf{y}_k \mathbf{y}_k^T H_k}{\mathbf{y}_k^T H_k \mathbf{y}_k} + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}. \tag{2.8}$$

Note: In practice, BFGS is generally outperforms the DFP because it preserves positive definiteness under the simple condition $\mathbf{s}_k^T \mathbf{y}_k > 0$, while DFP may require stronger conditions. The search direction is always computed as

$$\mathbf{p}_k = -H_k \nabla f(x_k). \tag{2.9}$$

Algorithm 8: DFP algorithm

1. **Step 1:** Select $x_0 \in \mathbb{R}^n$ and let ϵ be the termination criterion, initial inverse Hessian approximation $D_0 = I$, and $k = 0$.
2. **Step 2:** Compute the search direction $d_k = -D_k \nabla f(x_k)$.
3. **Step 3:** Determine the step length λ_k via a line search satisfying the Wolfe conditions. Then set $\mathbf{s}_k = \lambda_k d_k$ and $x_{k+1} = x_k + \mathbf{s}_k$.
4. **Step 4:** If $\|\nabla f(x_{k+1})\| < \epsilon$, terminate; otherwise, proceed to Step 5.
5. **Step 5:** Compute $\mathbf{y}_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.
6. **Step 6:** Update $D_{k+1} = D_k + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} - \frac{D_k \mathbf{y}_k \mathbf{y}_k^T D_k}{\mathbf{y}_k^T D_k \mathbf{y}_k}$.
7. **Step 7:** Set $k = k + 1$ and return to Step 2.

Algorithm 9: BFGS algorithm

1. **Step 1:** Select $x_0 \in \mathbb{R}^n$ and let ϵ be the termination criterion, initial inverse Hessian approximation $H_0 = I$, and $k = 0$..
2. **Step 2:** Compute the search direction $d_k = -H_k \nabla f(x_k)$.
3. **Step 3:** Determine the step length λ_k via a line search satisfying the Wolfe conditions. Then set $\mathbf{s}_k = \lambda_k d_k$ and $x_{k+1} = x_k + \mathbf{s}_k$.
4. **Step 4:** If $\|\nabla f(x_{k+1})\| < \epsilon$, terminate; otherwise, proceed to Step 5.
5. **Step 5:** Compute $\mathbf{y}_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.
6. **Step 6:** Compute $\rho = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}$, then update $H_{k+1} = (I - \rho \mathbf{s}_k \mathbf{y}_k^T) H_k (I - \rho \mathbf{y}_k \mathbf{s}_k^T) + \rho \mathbf{s}_k \mathbf{s}_k^T$.
7. **Step 7:** Set $k = k + 1$ and return to Step 2.

3

Conjugate Gradient Methods

This chapter is dedicated to study the conjugate gradient methods in detail. Both of linear and nonlinear CG methods have been discussed, imparted with theoretical study. Starting from a point $x_0 \in \mathbb{R}^n$, the sequence of points $\{x_k\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ is generated by the following recursive scheme:

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.1}$$

where α_k is the step-size which ensures that $f(x_{k+1}) \leq f(x_k)$ following the direction d_k.

3.1 Linear conjugate gradient methods

The linear conjugate gradient algorithm is dedicated for minimizing convex quadratic functions (or solving linear algebraic systems of equations with positive definite matrices). This algorithm was introduced by Hestenes and Stiefel in 1952 [40] for solving a system of linear equations or (equivalently minimising a quadratic cost function). Indeed, consider the following unconstrained quadratic problem

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

where the cost function f is a quadratic function defined as

$$f(x) = \frac{1}{2}x^T Q x - b^T x, \tag{3.2}$$

$Q \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix, and $b \in \mathbb{R}^n$ is a known vector. From (3.2), we have:

$$g_k = \nabla f(x) = Qx - b, \text{and} \quad \nabla^2 f(x) = Q.$$

Since the Hessian Q is symmetric and positive definite, from the optimality conditions, it follows that there is a unique minimizer x^* of (3.2), which is the solution of the following linear system:

$$Qx = b.$$

Theorem 3.1. *Consider the problem (P), where $Q \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix and $b \in \mathbb{R}^n$. For an iteration $k \in \mathbb{N}$, if α_k is an optimal step size following the descent direction d_k, then*

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}. \tag{3.3}$$

Proof. Consider the function φ

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

Then α_k is the optimal solution of the following one-dimensional problem.

$$\varphi(\alpha) = \min_{\alpha>0} \varphi(\alpha) \tag{3.4}$$

Assume Q is positive definite. Thus, $f(x) = \frac{1}{2}x^T Qx - b^T x$ is strictly convex on $\mathbb{R}^n$. Therefore, $\varphi(\alpha) = f(x_k + \alpha d_k)$ is strictly convex on $]0, +\infty[$, which is an open convex interval. Consequently, α_k is the optimal solution of (3.4) if and only if α_k satisfies $\varphi'(\alpha_k) = 0$. Now, we compute $\varphi'(\alpha)$:

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

Thus,

$$\varphi'(\alpha) = 0 \iff \alpha_k^T Q d_k = -g_k^T d_k$$

or equivalently,

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

□

Remark 3.2. The relation (3.3) does not verify in the case where Q is merely positive semi definite since f is not strictly convex.

3.1.1 Conjugate directions method

Definition 3.3. Let $Q \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix. The directions $d_0, d_1, \ldots$ are Q-conjugate if and only if

$$d_i^T Q d_j = 0 \quad \text{for} \quad 0 \leq i \neq j \leq k.$$

Theorem 3.4. *Let $Q \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix. If the directions $d_0, d_1, \ldots, d_k$, with $k \leq n-1$, are non-zero and Q-conjugate, then they are linearly independent.*

Proof. Suppose that

$$\alpha_0 d_0 + \alpha_1 d_1 + \ldots + \alpha_k d_k = 0. \tag{3.5}$$

Multiplying equation (3.5) by $d_j^T Q$, $0 \leq j \leq k$, we obtain

$$0 = \sum_{i=0}^{k} \alpha_i d_j^T Q d_i = \alpha_j d_j^T Q d_j,$$

Since

$$\alpha_j d_j^T Q d_j = 0 \implies \alpha_j = 0,$$

Since Q is positive definite, it follows that $d_j^T Q d_j > 0$. Therefore, the set of directions $\{d_0, d_1, \ldots, d_k\}$ is linearly independent. □

Conjugate direction algorithm

Below, in Algorithme (10), we sketch the main steps of conjugate direction algorithm.

Theorem 3.5. *The basic conjugate direction algorithm converges in n steps.*

Proof. First keep in mind that

$$\begin{aligned} x_n &= x_{n-1} + \alpha_{n-1} d_{n-1} \\ &\vdots \\ &= x_0 + \alpha_0 d_0 + \cdots + \alpha_{n-1} d_{n-1} \end{aligned}$$

Algorithm 10: Conjugate direction algorithm

Step 0: Choose a scalar $\epsilon > 0$ as the stopping criterion. Select an initial point $x_0 \in \mathbb{R}^n$ and conjugate directions $d_0, d_1, \ldots, d_{n-1}$. Set $k = 0$.

Step 1: Compute gradient $g_k = \nabla f(x_k) = Qx_k - b$. If $\|g_k\| < \epsilon$, **Stop**.

Step 2: Calculate the step size:

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

Step 3: Update solution:

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

Step 4: Set $k = k + 1$ and return to Step 1.

Now suppose $x^* - x_0 = \beta_0 d_0 + \cdots + \beta_{n-1} d_{n-1}$ (this is possible since $d_0, \ldots, d_{n-1}$ is a basis of $\mathbb{R}^n$.)

Now we will check that $\beta_k = \alpha_k$ for all $k = 0, \ldots, n-1$: multiplying $x^* - x_0$ by $(d_k)^T Q$, we obtain

$$d_k^T Q(x^* - x_0) = \beta_k d_k^T Q d_k.$$

We can solve for β_k:

$$\beta_k = \frac{{d_k}^T Q(x^* - x_0)}{{d_k}^T Q d_k},$$

$$= \frac{{d_k}^T Q\left[(x^* - x_k) + (x_k - x_0)\right]}{{d_k}^T Q d_k},$$

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

The 3rd equality above is due to $Q(x^* - x_k) = -(Qx_k - b) = -g_k$ and that $x_k - x_0 = \sum_{j=0}^{k-1} \alpha_j d_j$ which is a linear combination of $\{d_j : 0 \le j \le k-1\}$ that are all Q-conjugate with d_k. Therefore, $x^{(n)} - x_0 = x^* - x_0$, hence $x_n = x^*$.

□

Lemma 3.6. *For any $k = 0, 1, \ldots, n-1$, we have*

$$g_{k+1}^T d_j = 0, \quad \forall j \le k.$$

Proof. Multiplying both sides of $x_{k+1} = x_k + \alpha_k d_k$ by Q gives:

$$Qx_{k+1} = Qx_k + \alpha_k Q d_k,$$

which yields

$$g_{k+1} = g_k + \alpha_k Q d_k,$$

since$g_k = Qx_k - b$ for all k

We will use this fact along with induction to prove the claim. It is easy to show

$$g_1^T d_0 = 0.$$

Suppose the claim holds for k, i.e.,

$$g_k^T d_i = 0, \quad \text{for } i = 0, 1, \ldots, k-1.$$

Then

$$g_{k+1}^T d_i = g_k^T d_i + \alpha_k d_k^T Q d_i.$$

If $i \leq k-1$, then $g_{k+1}^T d_i = 0$ due to the induction hypothesis and Q-conjugacy.

If $i = k$,

$$g_{k+1}^T d_k = g_k^T d_k + \alpha_k d_k^T Q d_k = 0.$$

due to the definition of α_k. Therefore, the claim also holds for $k+1$. □

3.1.2 Conjugate gradient method

For solving Problem (P), in the conjugate direction method, the directions $d_0, d_1, \ldots, d_{n-1}$ are chosen in advance. Whereas, in the conjugate gradient method, we start from a point $x_0 \in \mathbb{R}^n$, with $g_0 = Qx_0 - b$.

The sequence of iterators $\{x_k\}_{k\in\mathbb{N}}$ are produced following schema (3.1), where the directions d_k, $k = 1, \ldots, n$, are computed as follows:

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

The parameter β_k is determined so that the direction d_k is conjugate with the previous directions d_i, $i = 0, \ldots, k-1$. In other terms, we must have:

$$d_i^T Q d_k = 0 \quad \text{for} \quad i = 0, \ldots, n-1.$$

The term conjugate gradient is derived from two components: gradient and conjugacy.

i- The gradient is used because d_k is computed using the gradient at the point x_k.

ii- The method is called conjugate because the directions $d_1, d_2, \ldots, d_{k-1}$ are conjugate.

Below, in Algorithm (11), we sketch the main steps of a conjugate gradient algorithm for minimising a quadratic cost function.

Algorithm 11: Conjugate gradient algorithm (quadratic case)

Step 0: Choose a scalar $\epsilon > 0$ sufficiently small to stop the algorithm. Select a point $x_0 \in \mathbb{R}^n$ and set $k = 0$. Compute $f(x_0)$, $g_0 = Qx_0 - b$ and $d_0 = -g_0$.

Step 1: If $\|g_k\| \leq \epsilon$, then stop; otherwise:
Calculate the step size

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

Step 2: Update the solution

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

Step 3: Compute the new gradient

$$g_{k+1} = Qx_{k+1} - b.$$

Step 4: Compute the conjugate coefficient

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

Step 5: Update the conjugate direction

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

Step 6: Increment $k = k + 1$ and return to Step 1.

Now, we prove that directions produced by a quadratic CG algorithm are Q-conjugate.

Proposition 3.7. *The directions $d_0, d_1, \ldots, d_{n-1}$ generated by the CG algorithm are Q-conjugate.*

Proof. We prove this by induction.

First,

$$d_1^T Q d_0 = (-g_1 + \beta_0 d_0)^T Q d_0 = 0,$$

due to the definition of β_0.

Now assume $d_0, \ldots, d_k$ are Q-conjugate. We need to show that d_{k+1} is Q-conjugate with all of them.

We need the following facts:

- $d_k = -g_k + \beta_{k-1} d_{k-1}$, for all k, by definition of d_k in CG.
- $g_{k+1} = g_k + \alpha_k Q d_k$, for all k, by $x_{k+1} = x_k + \alpha_k d_k$ in CG.
- $g_{k+1}^T d_j = 0$, for all $j \leq k$, .
- $g_{k+1}^T g_j = 0$, for all $j \leq k$, since $g_{k+1}^T g_j = g_{k+1}^T (d_j - \beta_{j-1} d_j) = 0$.

Now we come back to $d_{k+1}^T Q d_j = (-g_{k+1} + \beta_k d_k)^T Q d_j$.

If $j < k$, then

$$d_{k+1}^T Q d_j = -g_{k+1}^T Q d_j = -g_{k+1}^T \frac{g_{j+1} - g_j}{\alpha_j} = 0.$$

If $j = k$, then

$$d_{k+1}^T Q d_j = d_{k+1}^T Q d_k = (-g_{k+1} + \beta_k d_k)^T Q d_k = 0.$$

due to definition of β_k.

Therefore, $d_{k+1}^T Q d_j = 0$ for all $j \leq k$. □

Next, we establish that a CG algorithm applied for (P) converges in n steps for quadratic $f(x)$.

Theorem 3.8. *Algorithm* (11) *converges to the optimal minimizer* x^* *in* n *steps.*

Proof. The proof is similar to the proof of Theorem 3.5. □

3.2 Nonlinear conjugate gradient method

As mentioned above, the conjugate gradient method was firstly developed by Hestenes and Steifel in 1952 [40] for minimizing strictly convex quadratic functions; later, it is extended to handle nonlinear problems. In 1964, Fletcher and Reeves are firstly proposed a variant to handle nonlinear problems [19], later in 1969 by Polak, Ribière Ployak [44, 45]. Other variants were studied later, such as the conjugate descent method

proposed by Fletcher in 1987. These methods generate a sequence $\{x_k\}_{k\in\mathbb{N}}$ following schema (3.1), where α_k is the step size determined by a exact or inexact line search techniques. The descent search direction d_k is typically computed by the following iterative formula

$$d_0 = -g_0; \;\; d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \in \mathbb{N}^*,$$

where $\beta_k \in \mathbb{R}$ is the conjugate parameter that characterizes the various conjugate gradient variants. The most famous classical conjugate gradient methods include HS (Hestenes and Stiefel) [40], FR (Fletcher and Reeves) [19], PRP (Polak-Ribière-Polyak) [44, 45], CD (conjugate descent) [19], LS (Liu and Storey) [46], and DY (Dai and Yuan) [15], where their parameters β_k are given respectively as follows

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

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

where $y_k = g_{k+1} - g_k$ and $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^n$. The DY, CD and FR have better theoretical convergence properties, but practically, they are less effective. Conversely, the LS, HS and PRP methods are more efficient in practice, but they may not always be convergent. Moreover, if f is a strongly convex quadratic, then theoretically, all the above parameters are equivalent with an exact line search. For non-quadratic cost functions, each choice for the update parameter leads to different performance.

Below, in Algorithm (12), we sketch the main steps of a conjugate gradient algorithm.

3.2.1 Zoutendijk conditions for global convergence

In each CG iteration, the step size α_k is chosen to improve the value of f. Since $\alpha > 0$, the direction d_k should satisfy the *descent condition*

$$g_k^T d_k < 0, \text{for all } k \in \mathbb{N}. \tag{3.6}$$

If there exists a constant $c > 0$ such that

$$g_k^T d_k < -c\|g_k\|^2 \text{for all } k \in \mathbb{N} \tag{3.7}$$

Algorithm 12: The nonlinear conjugate gradient algorithm

- **Step 0:** Select $x_0 \in \mathbb{R}^n$ and let ϵ be the termination criterion, compute $f(x_0)$ and $g(x_0)$. Let $d_0 = -g_0$.;
- **Step 1:** If $\|g_k\|_\infty \leq \epsilon$, then stop. Otherwise, go to the next step.;
- **Step 2:** Determine the step-length α_k using a line search process.;

  Set $x_{k+1} = x_k + \alpha_k d_k$ and $k = k + 1$;
- **Step 3:** Compute β_k;
- **Step 4:** $d_k = -g_k + \beta_k d_{k-1}$ and go to **Step 1**.;

then the search directions satisfy the *sufficient descent condition*.

The determination of the step size α_k is crucial for global convergence. Usually, it is determined using inexact line searches, which guarantee taking steps that should be neither too long nor too short, such as the weak Wolfe line search

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \alpha_k \delta d_k^t g_k, \tag{3.8}$$

$$g(x_k + \alpha_k d_k)^t d_k \geq \sigma d_k^t g_k, \tag{3.9}$$

or the strong Wolfe line search

$$\begin{aligned} f(x_k + \alpha_k d_k) - f(x_k) &\leq \alpha_k \delta d_k^t g_k, \\ \left| g(x_k + \alpha_k d_k)^t d_k \right| &\leq -\sigma d_k^t g_k, \end{aligned} \tag{3.10}$$

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

Ideally, we would like to determine the line search in a CG algorithm when the standard Wolfe conditions are satisfied. For some CG algorithms, however, stronger versions of the Wolfe conditions are needed to ensure convergence and to enhance stability. Either of the following assumptions are often utilized in convergence analysis for CG algorithms:

Lipschitz assumption: In some neighborhood $\mathcal{N}$ of the level set

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

is bounded .

The gradient $\nabla f(x)$ is Lipschitz continuous, that is there exists a constant $L > 0$ such

that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \qquad \text{for all } x, y \in \mathcal{N}.$$

Boundedness assumption: The level set $\mathcal{L}$ is bounded, that is there exists a constant $B > 0$ such that

$$\|\nabla f(x)\| \leq B \qquad \text{for all } x \in \mathcal{L}.$$

The following theorem, often called the Zoutendijk condition, is used to prove the global convergence of nonlinear CG method; it is given by Zoutendijk [69] and Wolfe [53, 54].

Theorem 3.9. *Consider any iterative method of the form* (3.1) *where* d_k *satisfies the descent condition* (3.6) *and* α_k *satisfies the standard Wolfe conditions. If the Lipschitz assumption holds, then*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{3.11}$$

Proof. See[38] page 30 . □

Theorem 3.10. *Global convergence proofs for CG methods are often based on the Zoutendijk condition combined with analysis showing that :*

(i) the sufficient descent condition

$$g_k^T d_k \leq -c\|g_k\|^2 \quad \textit{holds,}$$

(ii) there exists a constant M *such that*

$$\|d_k\|^2 \leq M.$$

Combining (i), (ii), and (3.11)*, we have*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.12}$$

Throughout this survey, the statement that a CG method converges globally means either $g_k = 0$ *for some* k *or* (3.12) *holds.*

Proof. Suppose the stepsizes α_k satisfy the Wolfe conditions. Then the *Zoutendijk condition* holds:

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

Using (i) and (ii), we have

$$\frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \frac{(c\|g_k\|^2)^2}{M} = \frac{c^2}{M}\|g_k\|^4.$$

Hence,

$$\sum_{k=0}^{\infty} \|g_k\|^4 < \infty.$$

If $\|g_k\|$ did not satisfy

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

then there would exist some $\epsilon > 0$ such that $\|g_k\| \geq \epsilon$ for infinitely many k. This would make the series above diverge, which is a contradiction.

Thus, the CG method converges globally in the sense that either $g_k = 0$ for some finite k or

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

□

Another result related to the Zoutendijk condition, found in [14] [26]), is the following: assuming the search directions are descent,

Theorem 3.11. *Consider any conjugate gradient method of the form* (3.1)-(3.2) *where* d_k *is a descent condition direction and* α_k *satisfies the strong Wolfe conditions*(3.10) (3.8) . *If the Lipschitz assumption holds, then either*

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

or

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

Proof. See [38] pages 105-106. □

4

Two new conjugate gradient algorithms for nonlinear unconstrained optimization

This chapter presents our main contribution by developing two new conjugate gradient methods for solving nonlinear unconstrained optimization problems by introducing new conjugate gradient parameters, named hPB(hybrid PRP-BRB) [34] and MZ (Maiza-Ziadi)[33]. The hPB is a hybrid algorithm by making a convex combination between PRP (Polak-Ribière-Polyak)[45] and BRB (Rahali-Belloufi-Benzine) [41] conjugate gradient parameters, whereas the MZ formula is inspired by the hSM parameter[49].

In this study, we consider the following nonlinear optimization problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Numerous practical problems in real-life applications can be expressed as unconstrained optimization problems that involve differentiable cost functions [68, 66]. Spotting a solution for these problems becomes difficult when their dimensions are high. Conjugate gradient methods (CG) are extensively employed to deal with these situations thanks to their low memory requirements and simplicity [37]; they are also widely employed in numerous applications, in particular in image processing [30, 9], designing adaptive filters and learning algorithms [56], neural networks [47], machine learning and signal processing [63, 57], molecular physics [67] and statistical modeling [35]. Starting from a point $x_0 \in \mathbb{R}^n$, the

sequence of points $\{x_k\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ is generated by the following recursive scheme:

$$x_{k+1} = x_k + \alpha_k d_k, \quad k \in \mathbb{N}, \tag{4.1}$$

where d_k is the descent direction and α_k is the step size that ensures that $f(x_{k+1}) < f(x_k)$. The determination of the step size α_k is crucial for global convergence. The descent search direction d_k is typically computed by the following iterative formula

$$d_0 = -g_0; \;\; d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \in \mathbb{N}^*,$$

where $\beta_k \in \mathbb{R}$ is the conjugate parameter

Over the last decade, this research topic has attracted the attention of several researchers and numerous studies have been carried out to develop new efficient conjugate gradient formulas with good numerical performances and global convergent properties. We cite, for example, the RMIL method proposed by Rivaie et al. [43], the BRB method proposed by Rahali et al. [41], the WYL method proposed by Wei and al. [61], the NHS and NPRP methods proposed by Zhang [62], the hSM method proposed by Sulaiman-Mohammed [49], and the MHS method proposed by Yao et al. [61], where their parameters β_k are given respectively as follows:

$$\beta_k^{RMIL} = \frac{g_{k+1}^t y_k}{\|d_k\|^2}, \quad \beta_k^{BRB} = \frac{\|g_{k+1}\|^2}{\|d_k\|^2},$$

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

$$\beta_k^{hSM} = \frac{g_{k+1}^T(g_{k+1} + g_k)}{\|d_k\|^2}, \quad \beta_k^{MHS} = \frac{g_{k+1}^T\left(g_{k+1} - \frac{\|g_{k+1}\|}{g_k} g_k\right)}{y_k^T g_k}.$$

$$\beta_k^{WYL} = \frac{g_{k+1}^t(g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k)}{\|g_k\|^2}$$

For more recent conjugate gradient formulas, the reader can see [35, 37].

In order to achieve effective performance and good convergence properties, numerous combinations of CG methods have been proposed. We gather some famous hybrid conjugate coefficients based on a convex combination in Table 4.1 below.

Inspired by the previous works, we proposed two new conjugate gradient parameters, β_k^{MZ} [33] and β_k^{hPB} [34]. The MZ parameter is obtained by combining the RMIL

Authors	Formula	References
Liu and Li	$\beta^{hLSDY} = \lambda\beta^{DY} + (1-\lambda)\beta^{LS}$	[31]
Andrei	$\beta^{hHSDY} = \lambda\beta^{DY} + (1-\lambda)\beta^{HS}$	[2]
Zheng et al.	$\beta^{hHSCD} = \lambda\beta^{CD} + (1-\lambda)\beta^{HS}$	[62]
Djordjevic	$\beta^{hLSCD} = \lambda\beta^{CD} + (1-\lambda)\beta^{LS}$	[17]

Table 4.1: Hybrid conjugate gradient methods with convex combinations

[43] and PRP [45] formulas, whereas the hPB parameter is expressed as a convex combination of the Polak-Ribière-Polyak (PRP) and Rahali-Belloufi-Benzine (BRB) [41] formulas.

To achieve good convergence properties, we adopt a new inexact line search technique that fits in with the suggested conjugate gradient parameters where the step size α_k satisfies the following conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \alpha_k \delta d_k^t g_k \frac{\|g_k\|^2}{\|d_k\|^2}, \tag{4.2}$$

$$\left|g(x_k + \alpha_k d_k)^t d_k\right| \leq -\sigma d_k^t g_k \frac{\|g_k\|^2}{\|d_k\|^2}, \tag{4.3}$$

where $\sigma \in \left(0, \frac{\mu-1}{\mu^2(\mu^2+1.2)}\right]$, with $\mu > 1$ and $0 < \delta < \sigma$. Since $\mu > 1$, it results that $0 < \delta < \sigma < 1$.

The new line search technique is a modification of the strong Wolfe one by adding the term $\frac{\|g_k\|^2}{\|d_k\|^2}$ (to the right side of the strong Wolfe inequalities). The new term depends on the gradient values and the descent direction at each point x_k. We have adopted this line search to fit in with the proposed CG parameters and ensure global convergence with good numerical performance.

This modification permits getting a step size that is neither too long nor too short. Also, as shown below, the proposed methods exhibit good convergence properties and satisfactory performance. The proposed approaches ensures global convergence under mild assumptions and the algorithms are successfully applied to a broad set of test functions (with varied analytical expressions and structures), ranging from the simplest to the most complex, as well as in image processing.

4.1 The proposed conjugate gradient algorithms

4.1.1 The MZ algorithm

Inspired by the RMIL [43] and PRP [45] formulas,

$$\beta_k^{RMIL} = \frac{g_{k+1}^t(g_{k+1} - g_k)}{\|d_k\|^2},$$
$$\beta_k^{PRP} = \frac{g_{k+1}^t y_k}{\|g_k\|^2},$$

we have designed a new parameter β_k^{MZ} (where 'MZ' relates to the authors for easy reference only), which is computed as follows:

$$\beta_k^{MZ} = \frac{g_{k+1}^t \left(g_{k+1} + \frac{\|d_k\|^2}{\|g_k\|^2} y_k \right)}{\|d_k\|^2}. \tag{4.4}$$

Indeed,

$$\begin{aligned}\beta_k^{MZ} &= \frac{g_{k+1}^t g_{k+1} + \|d_k\|^2 \frac{g_{k+1}^t y_k}{\|g_k\|^2}}{\|d_k\|^2} \\ &= \frac{g_{k+1}^t \left(g_{k+1} + \frac{\|d_k\|^2}{\|g_k\|^2} y_k \right)}{\|d_k\|^2}.\end{aligned}$$

The main steps of the MZ method are sketched in Algorithm 13 below.

Algorithm 13: The MZ algorithm

Input: Scalar $\mu > 1$, parameters δ, σ with $0 < \delta < \sigma < \frac{\mu-1}{\mu^2(\mu^2+1.2)}$, and stopping tolerance $\epsilon > 0$

Output: Optimal solution x_k

Initialization:
$k \leftarrow 0$
Select initial point $x_0 \in \mathbb{R}^n$
Compute $g_0 = \nabla f(x_0)$
Set $d_0 = -g_0$
while $\|g_k\| \geq \epsilon$ **do**
  Compute step length α_k using line search conditions (4.2) and (4.3)
  Update $x_{k+1} = x_k + \alpha_k d_k$
  Compute $g_{k+1} = \nabla f(x_{k+1})$
  if $|g_k^T g_{k+1}| \geq 0.2\|g_{k+1}\|^2$ **then**
    $d_{k+1} = -g_{k+1}$ // Powell restart
  else
    Compute $y_k = g_{k+1} - g_k$
    Compute $d_{k+1} = -g_{k+1} + \beta_k^{MZ} d_k$
  $k \leftarrow k + 1$

4.1.2 The hPB algorithm

The main idea for this approach is to make a convex combination of BRB and PRP methods in order to combine a good practical performance and powerful global convergence properties of both of BRB [41] and PRP [45] methods, where their parameters are defined as follows

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

The new hybrid conjugate gradient parameter, β^{hPB} , is computed as follows:

$$\beta_k^{hPB} = \begin{cases} \beta_k^{PRP} & \text{if } \beta_k^{PRP} > 0, \\ \lambda_k\, \beta_k^{BRB} + (1-\lambda_k)\beta_k^{PRP} & \text{if } \lambda_k\beta_k^{BRB} + (1-\lambda_k)\beta_k^{PRP} > 0 \text{ and } \beta_k^{PRP} \leq 0 \\ \beta_k^{BRB} & otherwise. \end{cases} \tag{4.5}$$

The new hybrid conjugate coefficient β_k^{hPB} is chosen positive, where the hybridization parameter λ_k is defined in such a way that the descent direction d_{k+1} fulfills the conjugacy condition, that is $d_{k+1}^t y_k = 0$. Indeed, in the case where

$$\beta_k^{hPB} = \lambda_k\, \beta_k^{BRB} + (1-\lambda_k)\beta_k^{PRP},$$

it follows that

$$\begin{aligned} d_{k+1} &= -g_{k+1} + \beta_k^{hPB} d_k, && (4.6)\\ &= -g_{k+1} + (\lambda_k\, \beta_k^{PRP} + (1-\lambda_k)\beta_k^{BRB}) d_k, \\ &= -g_{k+1} + \left(\lambda_k\, \frac{g_{k+1}^t y_k}{\|g_k\|^2} + (1-\lambda_k)\frac{\|g_{k+1}\|^2}{\|d_k\|^2}\right) d_k, && (4.7) \end{aligned}$$

by multiplying both sides of (4.7) by y_k^t from the left, it follows that

$$y_k^t d_{k+1} = -y_k^t g_{k+1} + \left(\lambda_k\, \frac{g_{k+1}^t y_k}{\|g_k\|^2} + (1-\lambda_k)\frac{\|g_{k+1}\|^2}{\|d_k\|^2}\right) y_k^t d_k$$

using the conjugacy condition, it results that

$$\lambda_k = \frac{\vartheta_k - \psi_k}{\phi_k}, \tag{4.8}$$

where $\psi_k = \|g_{k+1}\|^2 \|g_k\|^2 y_k^t d_k$, $\vartheta_k = y_k^t g_{k+1} \|g_k\|^2 \|d_k\|^2$ and $\phi_k = y_k^t g_{k+1} y_k^t d_k \|d_k\|^2 - \psi_k$.

For each iteration k, we set $\lambda_k = \max\left\{0, \min\left\{\frac{\vartheta_k - \psi_k}{\phi_k}, 1\right\}\right\}$, and as the search progresses, if for an iteration $k \in \mathbb{N}$ we have $\phi_k = 0$, we set $\beta_k^{hPB} = \beta_k^{BRB}$. The main steps of the hPB method are sketched in Algorithm 14 below.

Algorithm 14: The hPb algorithm

Input: Scalar $\mu > 1$, parameters δ, σ with $0 < \delta < \sigma < \frac{\mu-1}{\mu^2(\mu^2+1.2)}$, and stopping tolerance $\epsilon > 0$

Output: Optimal solution x_k

Initialization:
$k \leftarrow 0$
Select initial point $x_0 \in \mathbb{R}^n$
Compute $g_0 = \nabla f(x_0)$
Set $d_0 = -g_0$
while $\|g_k\| \geq \epsilon$ **do**
  Compute step length α_k using line search conditions (4.2) and (4.3)
  Update $x_{k+1} = x_k + \alpha_k d_k$
  Compute $g_{k+1} = \nabla f(x_{k+1})$
  if $|g_k^T g_{k+1}| \geq 0.2\|g_{k+1}\|^2$ **then**
    $d_{k+1} = -g_{k+1}$ // Powell restart
  else
    Compute $y_k = g_{k+1} - g_k$
    if $\beta_k^{PRP} > 0$ **then**
      $\beta_k^{hPB} \leftarrow \beta_k^{PRP}$
    else
      Compute $\phi_k = y_k^T g_{k+1} y_k^T d_k \|d_k\|^2 - \|g_{k+1}\|^2 \|g_k\|^2 y_k^T d_k$
      if $\phi_k = 0$ **then**
        $\beta_k^{hPB} \leftarrow \beta_k^{BRB}$
      else
        Compute $\lambda_k = \max\left\{0, \min\left\{\frac{\vartheta_k - \psi_k}{\phi_k}, 1\right\}\right\}$
        if $\lambda_k \beta_k^{BRB} + (1-\lambda_k)\beta_k^{PRP} > 0$ **then**
          $\beta_k^{hPB} \leftarrow \lambda_k \beta_k^{BRB} + (1-\lambda_k)\beta_k^{PRP}$
        else
          $\beta_k^{hPB} \leftarrow \beta_k^{BRB}$
    Compute $d_{k+1} = -g_{k+1} + \beta_k^{hPB} d_k$
  $k \leftarrow k + 1$

4.2 The sufficient descent property of the proposed approaches

It is well known that the sufficient descent property is crucial for global convergence to hold. To establish it, we first prove the following result that will be required below.

Theorem 4.1. *Let* $\{g_k\}_{k\in\mathbb{N}}$, $\{\beta_k\}_{k\in\mathbb{N}}$ *and* $\{d_k\}_{k\in\mathbb{N}}$ *be the sequences generated by the MZ (resp. hPB) algorithm under conditions* (4.2) *and* (4.3). *Then*

$$\frac{\|g_k\|}{\|d_k\|} \leqslant \mu, \ \forall k \in \mathbb{N}. \tag{4.9}$$

Proof. It is clear that, in both variants (MZ or hPB), in the case where Powell's restart criterion holds (i.e., $|g_k^t g_{k+1}| \geq 0.2 \|g_{k+1}\|^2$), the descent direction d_k is defined as $d_k = -g_k$ and the relation (4.9) holds.

Now, if the Powell condition does not hold, we prove the above relation by induction. For $k = 0$, the condition (4.9) holds since $d_0 = -g_0$. Assume the relation (4.9) holds for $k \geq 1$, and let us prove it for $k+1$. Since

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{4.10}$$

by multiplying both sides of (4.10) by g_{k+1}^t, we obtain

$$d_{k+1}^t g_{k+1} = -\|g_{k+1}\|^2 + \beta_k d_k^t g_{k+1}. \tag{4.11}$$

On the other hand, for both variants, we have

$$|\beta_k| \leq \frac{\|g_{k+1}\|^2}{\|d_k\|^2} + 1.2\frac{\|g_{k+1}\|^2}{\|g_k\|^2}. \tag{4.12}$$

Indeed, in case where $\beta_k = \beta_k^{MZ}$, then

$$\begin{aligned}
\left|\beta_k^{MZ}\right| &\leq \frac{\|g_{k+1}\|^2}{\|d_k\|^2} + \frac{\left|\|g_{k+1}\|^2 - g_{k+1}^t g_k\right|}{\|g_k\|^2} \\
&\leq \frac{\|g_{k+1}\|^2}{\|d_k\|^2} + \frac{\|g_{k+1}\|^2 + \left|g_{k+1}^t g_k\right|}{\|g_k\|^2} \\
&\leq \frac{\|g_{k+1}\|^2}{\|d_k\|^2} + 1.2\frac{\|g_{k+1}\|^2}{\|g_k\|^2}.
\end{aligned}$$

Also, in case where $\beta_k = \beta_k^{MZ}$, then

$$\begin{aligned}\left|\beta_k^{hPB}\right| &\leq \left|\beta_k^{BRB}\right| + \left|\beta_k^{PRP}\right| \\ &\leq \frac{\|g_{k+1}\|^2}{\|d_k\|^2} + \frac{\|g_{k+1}\|^2 + \left|g_{k+1}^t g_k\right|}{\|g_k\|^2} \\ &\leq \frac{\|g_{k+1}\|^2}{\|d_k\|^2} + 1.2\frac{\|g_{k+1}\|^2}{\|g_k\|^2}.\end{aligned}$$

Hence, from (4.11) and condition (4.3) it follows that

$$\begin{aligned}\|g_{k+1}\|^2 &\leq \left|d_{k+1}^t g_{k+1}\right| + |\beta_k| \left|d_k^t g_{k+1}\right| \\ &\leq \left|d_{k+1}^t g_{k+1}\right| + \sigma |\beta_k| \left|d_k^t g_k\right| \frac{\|g_k\|^2}{\|d_k\|^2} \\ &\leq \|d_{k+1}\| \|g_{k+1}\| + \sigma |\beta_k| \frac{\|g_k\|^3}{\|d_k\|} \\ &\leq \|d_{k+1}\| \|g_{k+1}\| + \sigma \left(1.2\frac{\|g_k\|}{\|d_k\|} + \frac{\|g_k\|^3}{\|d_k\|^3}\right) \|g_{k+1}\|^2 . \qquad (4.13)\end{aligned}$$

Furthermore, since $\sigma \leq \frac{\mu-1}{\mu^2(\mu^2+1.2)}$, we get

$$\begin{aligned}1 - \sigma \left(\frac{\|g_k\|^3}{\|d_k\|^3} + 1.2\frac{\|g_k\|}{\|d_k\|}\right) &\geq 1 - \sigma(\mu^3 + 1.2\mu), \\ &\geq 1 - \frac{\mu - 1}{\mu} = \frac{1}{\mu} > 0, \qquad (4.14)\end{aligned}$$

by dividing both sides of (4.13) by $\|g_{k+1}\| \, . \, \|d_{k+1}\|$, then from (4.14) it results that

$$\frac{\|g_{k+1}\|}{\|d_{k+1}\|} \leq \left(1 - \sigma \left(\frac{\|g_k\|^3}{\|d_k\|^3} + 1.2\frac{\|g_k\|}{\|d_k\|}\right)\right)^{-1} \leq \mu.$$

Thus, the proof is complete. □

Theorem 4.2. *The sequences $\{g_k\}_{k\in\mathbb{N}}$, $\{\beta_k\}_{k\in\mathbb{N}}$ and $\{d_k\}_{k\in\mathbb{N}}$ generated by the MZ (resp. hPB) Algorithm under the new line search conditions (4.2) and (4.3) with $\sigma \in \left(0, \frac{\mu-1}{\mu^2(\mu^2+1.2)}\right]$ and $\mu > 1$ satisfy*

$$g_k^t d_k \leqslant -\xi \|g_k\|^2 , \quad \forall k \in \mathbb{N}. \qquad (4.15)$$

where $\xi > 0$.

Proof. It is evident, in the case where the restart criterion of Powell holds (i.e., $|g_k^t g_{k+1}| \geq 0.2\|g_{k+1}\|^2$), the descent direction d_k, in both variants, is then given by $d_k = -g_k$ and the relation (4.15) holds.

Now, in the case where the Powell condition doesn't hold, we prove the above relation by induction. Indeed, for $k = 0$, the search direction is $d_0 = -g_0$, which implies that $d_0^t g_0 = -\|g_0\|^2$, and relation (4.15) holds.
Suppose that (4.15) is true for $k \geq 1$. For $k+1$, by multiplying two sides of (4.10) by g_{k+1}^t, we get

$$\begin{aligned}
d_{k+1}^t g_{k+1} &= -\|g_{k+1}\|^2 + \beta_k d_k^t g_{k+1} \\
&\leq -\|g_{k+1}\|^2 + |\beta| \left| d_k^t g_{k+1} \right| \\
&\leq -\|g_{k+1}\|^2 + \sigma \left(\frac{1.2\|g_{k+1}\|^2}{\|g_k\|^2} + \frac{\|g_{k+1}\|^2}{\|d_k\|^2} \right) \|g_k\| \|d_k\| \frac{\|g_k\|^2}{\|d_k\|^2}, \text{(using relations (4.3) and (4.12))} \\
&\leq -\|g_{k+1}\|^2 + \sigma \left(1.2\frac{\|g_k\|}{\|d_k\|} + \frac{\|g_k\|^3}{\|d_k\|^3} \right) \|g_{k+1}\|^2 \\
&\leq -\|g_{k+1}\|^2 \left(1 - \sigma(\mu^3 + 1.2\mu)\right) \leq -\frac{1}{\mu}\|g_{k+1}\|^2, \text{(using relations (4.9) and (4.14))}.
\end{aligned}$$

Then the proof is complete for $\xi = \frac{1}{\mu}$. □

4.3 The convergence analysis of the proposed approaches

Before analysing the convergence of the proposed approaches, we first show that they are well-defined. In the following theorem, we prove the existence of a step size α ($0 < \alpha < \infty$) that meets the conditions (4.2) and (4.3), where $0 < \delta < \sigma$, with $\sigma \in \left(0, \frac{\mu-1}{\mu^2(\mu^2+1.2)}\right]$ and $\mu > 1$.

Theorem 4.3. *Let f be a twice continuously differentiable function that is bounded below. If d_k is a descent direction, then there exists a strictly positive real constant α that meets the conditions* (4.2) *and* (4.3).

Proof. let's define the following function

$$h(\alpha) = f(x_k + \alpha d_k) - f(x_k) - \delta\alpha g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2}.$$

Using a standard Taylor development, from relation (4.9), it follows that

$$\begin{aligned} h(\alpha) &= f(x_k + \alpha d_k) - f(x_k) - \delta\alpha g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2} \\ &= \big(f(x_k) + \alpha g_k^t d_k + o(\alpha)\big) - f(x_k) - \delta\alpha g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2}, \\ &= \alpha g_k^t d_k - \delta\alpha g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2} + o(\alpha) \\ &\leq \alpha\left(1 - \delta\mu^2\right) g_k^t d_k + o(\alpha) \\ &\leq \alpha\left(1 - \frac{\mu - 1}{\mu^2 + 1.2}\right) g_k^t d_k + o(\alpha) < 0 \end{aligned}$$

Furthermore, since the function f is lower-bounded, it results that $\lim_{\alpha\to+\infty} h(\alpha) = +\infty$ and $h(0) = 0$. Hence, the function $h(.)$ changes its sign, then there exists a constant $\tau > 0$ such that $h(\tau) = 0$. It is clear that $h(\alpha)$ has a negative sign over the interval $[0, \tau]$, and its global minimum can not occur at the endpoints since $h(0) = h(\tau) = 0$. Therefore, there exists $\alpha^* \in (0, \tau)$, such that $h(\alpha^*) < 0$ and $h'(\alpha^*) = 0$. Hence

$$h(\alpha^*) = f(x_k + \alpha^* d_k) - f(x_k) - \delta\alpha^* g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2} < 0,$$

so that

$$f(x_k + \alpha^* d_k) < f(x) + \delta\alpha^* g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2},$$

and the first condition (4.2) holds. On the other hand, we have

$$h'(\alpha) = g_{k+1}^t d_k - \delta g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2}.$$

Since $h'(\alpha^*) = 0$, then

$$\sigma g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2} \leq \delta g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2} = g_{k+1}^t d_k < 0,$$

therefore,

$$\left|g_{k+1}^t d_k\right| \leq -\sigma g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2}.$$

Thus, the second condition (4.3) is also satisfied. □

4.3.1 The global convergence of the proposed approaches

The global convergence property is crucial for any conjugate gradient method. To establish it, we make the following assumptions for both approaches:

Assumption 4.3.1. *The level set* $\Omega = \{x \in \mathbb{R}^n | \ f(x) \leq f(x_0)\}$ *is bounded for any starting point* x_0.

Assumption 4.3.2. f *is continuously differentiable, whose gradient function* g *satisfies the Lipschitz condition in some closed neighborhood* $\mathcal{N}$ *of* Ω, *i.e.,* $\exists L > 0$ *such that*

$$\|g(x) - g(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathcal{N}. \tag{4.16}$$

The above assumptions imply the existence of a real number $\Gamma \geq 0$ such that

$$\|g_k\| \leq \Gamma \quad \forall x \in \Omega. \tag{4.17}$$

To establish the global convergence of both algorithms (see Theorem 4.7 below), we need to prove the following results, which will be needed below.

Lemma 4.4. *Let* $\{g_k\}_{k\in\mathbb{N}}$, $\{\alpha_k\}_{k\in\mathbb{N}}$ *and* $\{d_k\}_{k\in\mathbb{N}}$ *be the sequences generated by the MZ (resp. hPB) algorithm; then for some* $\varpi > 0$ *we have*

$$g_k^t d_k \geq -\varpi \|g_k\|^2, \ \forall k \in \mathbb{N}. \tag{4.18}$$

Proof. By multiplying Equation (4.10) by g_{k+1}, we get

$$\left|d_{k+1}^t g_{k+1}\right| \leq \|g_{k+1}\|^2 + |\beta_k| \left|d_k^t g_{k+1}\right|.$$

Then, from relation (4.9) it follows that,

$$\begin{aligned} \left|d_{k+1}^t g_{k+1}\right| &\leq \|g_{k+1}\|^2 + \sigma \left(1.2\frac{\|g_k\|}{\|d_k\|} + \frac{\|g_k\|^3}{\|d_k\|^3}\right) \|g_{k+1}\|^2, \\ &\leq \left(1 + \sigma\left(1.2\mu + \mu^3\right)\right) \|g_{k+1}\|^2, \end{aligned}$$

which means that,

$$-\varpi \|g_{k+1}\|^2 \leq d_{k+1}^t g_{k+1} \leq \varpi \|g_{k+1}\|^2$$

where $\varpi = 1 + \sigma\left(1.2\mu + \mu^3\right)$ and the proof is complete. □

Lemma 4.5. *Under the above assumptions, the sequence of step size $\{\alpha_k\}_{k\in\mathbb{N}}$ generated by the MZ (resp. hPB) algorithm under the new line search conditions (4.2) and (4.3) with $\sigma \in \left(0, \frac{\mu-1}{\mu^2(\mu^2+1.2)}\right]$ and $\mu > 1$, satisfies*

$$\alpha_k \geq \frac{\sigma\|g_k\|^2/\|d_k\|^2 - 1}{L\|d_k\|^2} d_k^t g_k, \ \forall k \in \mathbb{N}.$$

Proof. From Theorem 4.1, it follows that

$$\sigma\frac{\|g_k\|^2}{\|d_k\|^2} < \frac{\mu-1}{\mu^2+1.2} < 1,$$

hence, from condition (4.3) it results that

$$\left(\sigma\frac{\|g_k\|^2}{\|d_k\|^2} - 1\right) d_k^t g_k < d_k^t(g_{k+1} - g_k) \leq L\alpha_k\|d_k\|^2,$$

which completes the proof. □

Lemma 4.6. *Under Assumption 4.3.1, the sequences $\{g_k\}_{k\in\mathbb{N}}$ and $\{d_k\}_{k\in\mathbb{N}}$ produced by the MZ (resp. hPB) algorithm satisfy*

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

Proof. Using the condition (4.2) and relation (4.18), we have

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\geq -\delta\alpha_k g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2} \\ &\geq \frac{\delta\alpha_k}{\varpi}\frac{(g_k^t d_k)^2}{\|d_k\|^2}. \end{aligned}$$

Then we have,

$$\frac{\delta\alpha_k}{\varpi}\frac{(g_k^t d_k)^2}{\|d_k\|^2} \leq f(x_k) - f(x_{k+1}).$$

Let $m = \min\{\alpha_k : k \in \mathbb{N}\}$, by summing this inequality from $k = 0$ to infinity, we get

$$\frac{\delta m}{\varpi}\sum_{k\geq 0}\frac{(g_k^t d_k)^2}{\|d_k\|^2} \leq \sum_{k\geq 0}\frac{\delta\alpha_k}{\varpi}\frac{(g_k^t d_k)^2}{\|d_k\|^2} < +\infty.$$

and the inequality (4.19) holds. □

Theorem 4.7. *Under the above assumptions, the MZ (resp. hPB) algorithm converges globally in the sense that*

$$\lim_{k\to+\infty}\inf\|g_k\| = 0. \tag{4.20}$$

Proof. Assume that the assertion (4.20) does not hold. Then, there exists a strictly positive value $r > 0$ such that

$$\|g_k\| > r, \ \forall k \in \mathbb{N}. \tag{4.21}$$

Let $m = \min\{\alpha_k : k \in \mathbb{N}\}$ and $D = \max\{\|x - y\| : x, y \in \Omega\}$. From relation (4.1), it results that

$$\|d_k\|^2 = \frac{\|x_{k+1} - x_k\|^2}{\alpha_k^2} \leq \frac{D^2}{m^2}.$$

On the other hand, from (4.12), (4.21), (4.17) and (4.9) we get

$$\begin{aligned} d_{k+1} &\leq \|g_{k+1}\| + |\beta_k| \, \|d_k\| \\ &\leq \Gamma + \Big(\frac{1.2\Gamma^2}{r^2} + \frac{\mu^2\Gamma^2}{r^2}\Big)\frac{D}{m} = M, \end{aligned}$$

hence,

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

But, from (4.21), (4.15) and (4.19), it results that

$$\xi^2 r^4 \sum_{k\geq 0} \frac{1}{\|d_k\|^2} \leq \sum_{k\geq 0} \frac{\xi^2 \|g_k\|^4}{\|d_k\|^2} \leq \sum_{k\geq 0} \frac{(g_k^t d_k)^2}{\|d_k\|^2} < +\infty,$$

therefore

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

which contradicts the claim (4.21), so the assertion (4.20) is true. □

4.4 Numerical implementations

4.4.1 Application on some typical test functions

Here, we present a series of computational performances concerning the MZ and hPB algorithms, applied to 84 test functions with 310 test problems, as listed in Table 5.1 and taken from CUTEr [22] and [3], with dimensions ranging from 2 to 100,000, as well as on restoring four images with different noise levels. All the codes are written and implemented in Matlab. In all experiments, the both algorithms are implemented with the setting parameters $\mu = 1.6$, $\delta = 10^{-4}$, and $\sigma = 10^{-3}$.

To demonstrate the effectiveness of the suggested approaches, we compared them with some recent similar algorithms. Indeed, we have compared the MZ algorithm,

Table 4.2: List of test functions.

Function	Dimension n	Function	Dimension n
Extended Maratos	500, 700, 1000,2000,4000, 9000, 9500, 60,000, 10,000, 15,000	Arwhead	50, 60, 80, 100, 150, 200, 800, 1000, 5000
COSINE	10, 100 , 500, 1000	SINE	10, 100 , 500, 1000
ENGVAL1	100, 600, 800, 1000, 1500, 1600,1800, 10,000	Diagonal 1	2,10, 100, 1000, 10,000, 100,000
Generalized Tridiagonal 1	2, 10, 20,300, 500, 700, 10,000	FLETCHCR	2, 4
Extended White and Holst	1000, 2000, 3000, 4000, 5000, 6000	Diagonal 2	2, 4, 10, 800, 1000, 80,000
Diagonal 7	500, 700, 1000, 1500, 2000, 7000, 8000	Extended Rosenbrock	10, 20,100 ,1200, 3000, 4000, 5000
Quadratic QF2	100, 200,1000,5000, 7000, 9000, 10,000, 50,000	Diagonal 8	100, 200, 400, 500, 1000, 1500, 2000
Extended Freudenstein and Roth	10, 100, 1000, 4000, 9000, 10,000, 20,000, 50,000, 60,000, 80,000	Diagonal 3	2, 4, 6, 10, 50, 100, 200, 400, 700
DENSCHNF	10, 100, 10,000, 25,000, 30,000 50,000, 70,000, 80,000, 90,000	Extended Himmelblau	4, 6,10, 9000, 10,000
Perturbed Quadratic	2	POWER	2
QUARTC	2	Raydan 1	2
Generalized Rosenbrock	2, 50, 800, 1000	Perturbed quadratic diagonal	2
DENSCHNB	10, 90, 100, 2000, 3000, 4000, 5000, 6000, 7000, 9000	Raydan 2	1000, 4000, 50,000,80,000
HIMMELBG	2000, 3000, 6000, 30000, 50,000, 80,000	LIARWHD	10,50, 4000,5000, 5500, 10,000, 20,000, 80,000
Extended quadratic exponential EP1	40,000, 50,000, 60,000, 70,000	Diagonal 5	100, 200, 700, 1000, 1500, 2000, 2200, 2500
Extended BD1	4, 800, 900, 2000, 3000, 5000 20,000, 40,000, 60,000, 70,000, 80,000	Extended quadratic penalty QP1	4, 6, 8, 10, 50, 100, 700, 1000, 1500
Hager	2, 4,10,50, 80, 150, 300	NONSCOMP	2, 4, 1000, 5000, 70,000
HIMMELLH	10, 50, 300, 500, 10,000, 50,000, 60,000, 80,000, 10000	Quadratic QF1	5000, 6000, 8000, 9000, 20,000, 50,000, 70,000, 80,000
Extended quadratic penalty QP2	40, 60, 200	Diagonal 4	20,000, 30,000, 40,000, 50,000, 60,000, 70,000
DIXON3DQ	2, 4, 600	Extended PSC1	2, 4, 10, 100, 1000,10,000
Almost Perturbed Quadratic	2, 4, 6	Diagonal 9	10, 30,50, 60, 70, 90
Extended Tridiagonal 1	6, 10, 20,80, 90, 100, 150, 300, 500, 700, 1000, 5000, 6000	NONDIA	10, 100,500,1000,5000,100,000
EIGENALS	2550	BDQRTIC	5000
EIGENBLS	2550	EIGENCLS	2652
BROYDN7D	5000	BRYBND	5000
CHENHARK	5000	CHAINWOO	4000
EXTROSNB	1000	ENGVAL1	5000
ENGVAL2	3	ERRINROS	50
EXTROSNB	1000	FLETCBV2	5000
FLETCBV3	5000	FLETCHBV	5000
ROSENBR	2	S308	2
SPMSRTLS	4999	DIXMAANE	3000
DIXMAANF	3000	DIXMAANG	3000
DIXMAANH	3000	JENSMP	2
VAREIGVL	50	DIXMAANI	3000
KOWOSB	4	VIBRBEAM	8
DIXMAANJ	3000	LIARWHD	5000
WATSON	12	DIXMAANK	3000
LOGHAIRY	2	WOODS	4000
HILBERTA	2	TOINTGOR	50
HIMMELBB	2	TOINTPSP	50
HIMMELBF	4	TOINTQOR	50

with five CG methods which are: HRM [27] with $\theta = 0.4$, NPRP [62], NHS [62], NMFR, MP-CG [32] and CG-DESCENT [28] methods. The methods HRM, NPRP, NHS, and NMFR are implemented using the strong Wolfe conditions, whereas MP-CG and CG-DESCENT are implemented using the weak Wolfe conditions by setting $\delta = 10^{-4}$ and $\sigma = 10^{-3}$ (the other parameters are set as default as taken in [32]). Also, in order to illustrate the fact that the hPB method performs better using the proposed inexact line search, we compare it with hPB-SW (hPB method using strong Wolfe conditions with $\delta = 10^{-4}$ and $\sigma = 10^{-3}$). Furthermore, we compared it with five CG variants which are: HRM [27] with $\theta = 0.4$, PRP$^+$ [21], PRP [44], BRB [41] and NHS [62].

For these comparisons, the same starting point is assigned for each test problem, and each implementation is considered successful if a point x_k where $\|g(x_k)\|_\infty \leq 10^{-6}$ is reached within 2000 iterations, with CPU time less than 500 seconds; alternatively, the implementation is assigned as a failure.

Throughout the numerical results, in Figures 4.1–4.8, we compare the performance of of the implemented algorithms using the logarithmic performance profile of Dolan and Moré [16], relative to the number of iterations, function evaluations, gradient evaluations, and CPU-time. For a solver s, we define the ratio

$$r_{\mathrm{P},s} = \frac{N_{\mathrm{P},s}}{\min\{N_{\mathrm{P},s} : s \in S\}},$$

where $N_{\mathrm{P},s}$ denotes either the number of iterations, the number of function (gradient) evaluations, or the CPU time required by the solver s to solve a problem P. If a solver s does not solve the problem P, the ratio $r_{\mathrm{P},s}$ is assigned a large number. The logarithmic performance profile for each solver s is defined as follows:

$$\rho_s(\tau) = \frac{\text{number of problems where } \log_2(r_{\mathrm{P},s}) \leq \tau}{\text{total number of problems}},$$

For each method, we plot the fraction $\rho_s(\tau)$ of problems for which the method has a number of iterations (resp. number of function (gradient) evaluations and CPU time) that is within a factor τ. The curve that is shaped on the top corresponds to the code that solves the majority of the test problems within the given factor τ; for more details, see [16].

From the numerical comparison, figures 4.1–4.8 show that both MZ and hPB outperform the other methods. In particular, figures 4.1–4.4 indicate that MZ is faster in

approximately 36 of the test problems and successfully solves about 97 of them, followed by the CG-DESCENT method with 96. Similarly, figures 4.5–4.8 demonstrate that hPB is faster in about 48 of the test problems and successfully solves around 96, followed by hPB-sw and BRB, which solve 95 and 93, respectively. These results confirm the competitiveness and fast convergence of the proposed algorithms in the majority of test problems.

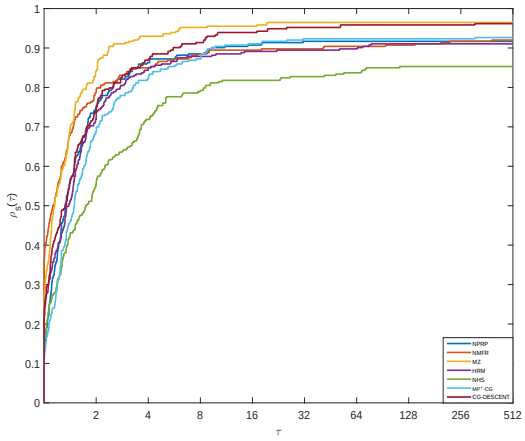

Figure 4.1: Performance profiles plot based on CPU time.

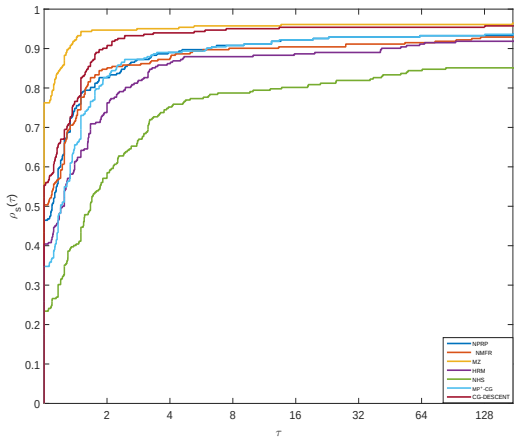

Figure 4.2: Performance profiles plot based on the number of iterations.

4.4.2 Application for restoring images

In optimization fields, image restoration problems are considered among the most difficult ones; they aim to restore the original image from one that has been corrupted by

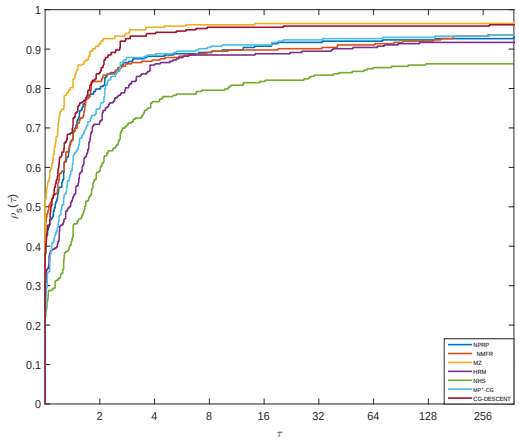

Figure 4.3: Performance profiles plot based on the number of function evaluations.

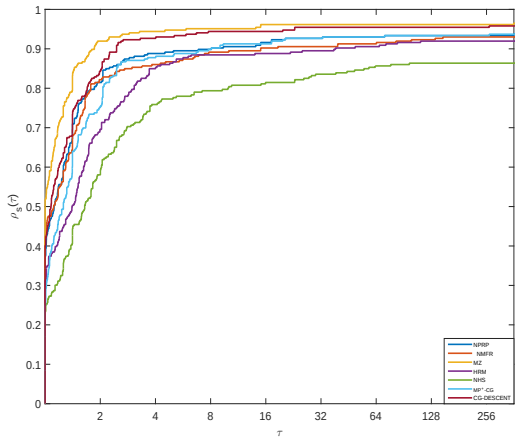

Figure 4.4: Performance profiles plot based on the number of gradient evaluations.

impulse noise. For this comparison, four test images of size 512×512 are chosen to evaluate the effectiveness of the proposed algorithm against the same variants used in the previous comparisons. Indeed, the Man.png, Boat.png, Lena.jpg, and Bridge.bmp are chosen to examine the performance of the MZ variant, whereas we have selected Man.png, Boat.png, Hill.jpg and Bridge.bmp for comparing the performence of the hPB method.

The image quality is assessed by two factors: the peak signal-to-noise ratio (PSNR) and its relative error (Err),

$$PSNR = 10 \log_{10} \frac{M \times N \times 255^2}{\sum_{i,j}(x^r_{i,j} - x^*_{i,j})^2}, \qquad Err = \frac{\|x^r - x^*\|}{\|x^*\|}$$

where M and N are the sizes of the image, $x^r_{i,j}$ represents the pixel values of the re-

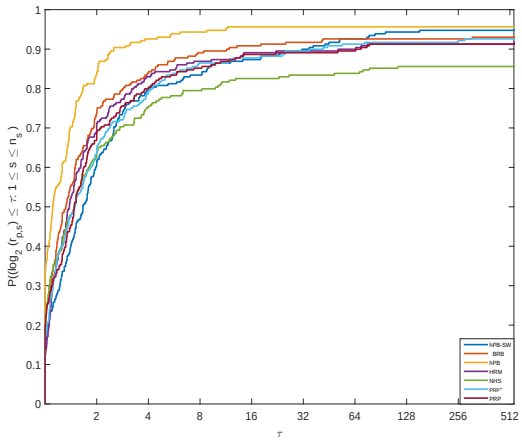

Figure 4.5: Performance profiles plot based on CPU time.

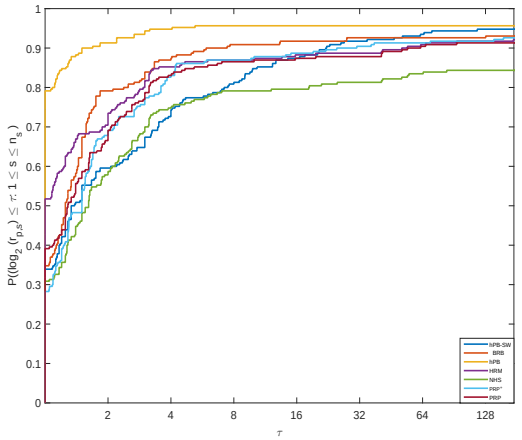

Figure 4.6: Performance profiles plot based on the number of iterations.

stored image, and $x^*_{i,j}$ denotes the original pixel values. The setting parameters of the proponent algorithms are set similarly to the previous test, and each computation will stop if any of the following criteria are fulfilled:

$$Iter > 300 \text{ or } \frac{|f(x_{k+1}) - f(x_k)|}{|f(x_k)|} < 10^{-4}.$$

Figures 4.9–4.11, and figures 4.12–4.14, show the restored images by impulse 30 %, 50%, and 70 % of noise. The performance of each algorithm is measured by the restored image quality, the elapsed time, and the number of iterations. The numerical outcomes are reported in tables 4.3–4.5, and tables 4.6–4.8. The algorithm with a high PSNR and minimal error with less CPU time is considered the best.

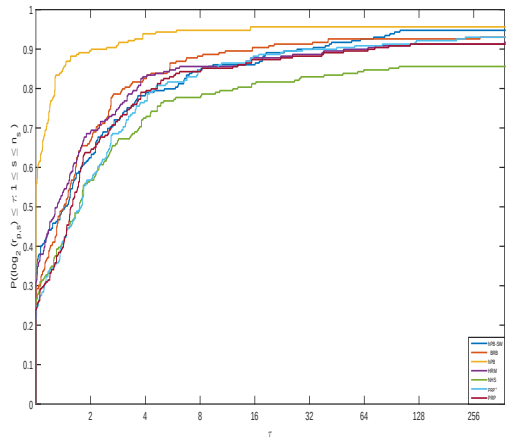

Figure 4.7: Performance profiles plot based on the number of function evaluations.

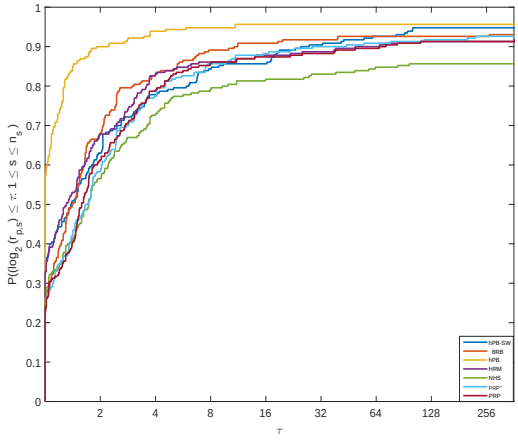

Figure 4.8: Performance profiles plot based on the number of gradient evaluations.

Upon examining the results in tables 4.3–4.5 and figures 4.9–4.11, it becomes evident that the MZ algorithm delivers good performance. In fact, as illustrated in Table 4.3 and Figure 4.9, we can observe that the NHS, NPRP and MP-CG methods failed to restore images with 30% noise, whereas the other methods were successful with a PSNR greater than 25, and the MZ method has the highest PSNR value for the Man and Bridge images. On the other hand, the visual outcomes of Figure 4.10 with 50% noise show that the NHS, NPRP, and MP-CG methods also failed to remove the noise with a PSNR less than 25, while the HRM, MZ, NMFR, and CG-DESCENT methods succeeded in restoring all the images and the bold values in Table 4.4 indicate the superiority of the MZ method for restoring three images. For 70% noise, as shown in

Table 4.3: Numerical results for image restoration problems with 30% salt-and-pepper.

Methods \ Images		Man	Boat	Lena	Bridge
NHS	Iter	26	26	25	23
	CPU	15.7869	14.9272	14.5193	14.4291
	PSNR	15.7231	17.5183	17.3303	16.2491
	Err	0.3405	0.2462	0.2993	0.3110
NMFR	Iter	18	16	15	35
	CPU	13.5430	12.8837	13.1202	16.5120
	PSNR	31.5222	33.6216	37.6712	28.4505
	Err	0.0552	0.0385	0.0287	0.0763
HRM	Iter	13	14	16	19
	CPU	14.4333	16.1088	15.9510	18.3902
	PSNR	31.4316	33.5530	**37.7937**	28.5705
	Err	0.0558	0.0388	**0.0283**	0.0752
NPRP	Iter	6	6	6	6
	CPU	12.4714	12.5510	**11.1241**	12.1049
	PSNR	15.7089	17.5182	17.3155	16.2439
	Err	0.3410	0.2461	0.2998	0.3111
MZ	Iter	15	27	18	20
	CPU	16.3077	13.2510	12.3024	17.5385
	PSNR	**31.5635**	33.1524	37.6352	**28.5895**
	Err	**0.0458**	0.0423	0.0296	**0.07401**
MP-CG	Iter	**4**	**4**	**4**	**4**
	CPU	**12.0037**	**11.4063**	11.5515	**11.8034**
	PSNR	15.7392	17.5272	17.3172	16.2386
	Err	0.3398	0.2459	0.2997	0.3113
CG-DESCENT	Iter	17	17	17	16
	CPU	14.0641	13.9686	14.3062	16.8352
	PSNR	31.5597	**33.6639**	**37.7533**	28.5640
	Err	0.0549	**0.0383**	**0.0285**	0.0753

Figure 4.11, and Table 4.5, the NHS, NPRP, and MP-CG methods failed to restore the original images; the HRM method succeeded in removing the noise from the Boat and Bridge images, and the NMFR method succeeded in restoring the images of Man, Boat, and Lena, while the MZ and CG-DESCENT methods succeeded in removing the noise from all images, where the highest PSNR value corresponds to the MZ method.

Upon examining the results in tables 4.6 - 4.8 and figures 4.12 - 4.14, it becomes evident that the hPB algorithm delivers good performance. In fact, as illustrated in Table 4.6 and from Figure 4.12, we can observe that the NHS method failed to restore images with 30% of noise, whereas the other methods were successful with a PSNR greater than 25, and the hPB method has the highest PSNR value for the Man and Boat images. On the other hand, the visual outcomes of Figure 4.13 with 50% of noise show that the NHS method also failed to remove the noise with a PSNR less than 25, whereas the PRP^+ method failed to restore the image of Man, while the HRM, hPB, BRB, PRP and hPB-sw methods succeeded for restoring all the images and the bold values in Table 4.7 indicate the superiority of the hPB method for restoring the images

Table 4.4: Numerical results for image restoration problems with 50% salt-and-pepper.

Methods \ Images		Man	Boat	Lena	Bridge
NHS	Iter	31	34	33	30
	CPU	24.3502	24.8263	24.8593	23.0874
	PSNR	13.4363	15.2070	14.7535	14.1568
	Err	0.4430	0.3212	0.4027	0.3957
NMFR	Iter	22	17	17	18
	CPU	20.1598	**18.0440**	18.1020	17.7847
	PSNR	29.1302	31.0974	35.0066	26.5072
	Err	0.072728	0.051554	0.039116	0.095465
HRM	Iter	19	14	26	14
	CPU	31.6329	22.9351	33.0992	21.4267
	PSNR	29.0482	30.0439	35.0633	26.2812
	Err	0.073418	0.058202	**0.038862**	0.097982
NPRP	Iter	**6**	**6**	**6**	**5**
	CPU	**16.0684**	18.4926	**16.6391**	**16.2163**
	PSNR	13.4675	15.1712	14.7757	14.1672
	Err	0.441405	0.322532	0.401702	0.395223
MZ	Iter	19	22	22	21
	CPU	27.0201	22.20145	24.2571	24.3507
	PSNR	**29.1725**	**31.1735**	**35.1438**	26.6304
	Err	**0.0723**	**0.0523**	**0.0393**	0.0943
MP-CG	Iter	**6**	**6**	**6**	**5**
	CPU	18.9429	19.0158	18.4746	16.5455
	PSNR	13.4606	15.2050	14.7792	14.1677
	Err	0.4417	0.3212	0.4015	0.3952
CG-DESCENT	Iter	22	19	20	23
	CPU	26.4330	23.1224	21.7718	29.0420
	PSNR	29.1528	31.1018	35.0045	**26.7484**
	Err	0.0725	0.0525	0.0395	**0.0935**

of Man, Boat and Hill. For 70% of noise, as shown in Figure 4.14 and Table 4.8, the NHS method failed to restore the original images; the HRM, PRP$^+$, BRB and hPB-sw methods succeeded to remove the noise from two images, while the hPB and PRP methods succeeded in restoring the images of Man, Boat and Hill, where the highest PSNR value corresponds to the hPB method.

On the whole, the numerical and visual outcomes of removing $30\%, 50\%$, and 70% of noise show a satisfactory performance of the proposed algorithms. Notably, the bold values in tables 4.3–4.5, and tables 4.6 - 4.8, highlight the efficiency of the proposed algorithms and take a short time to restore the majority of the test images.

Table 4.5: Numerical results for image restoration problems with 70% salt-and-pepper.

Methods \ Images		Man	Boat	Lena	Bridge
NHS	Iter	34	37	34	31
	CPU	29.7937	30.6846	27.8669	22.3260
	PSNR	11.3935	12.9901	12.5870	12.1899
	Err	0.5604	0.4146	0.5168	0.4963
NMFR	Iter	24	23	20	25
	CPU	24.9760	23.0873	**13.6592**	25.6359
	PSNR	26.2295	28.1877	28.5882	24.4006
	Err	0.101563	0.072068	0.075126	0.121667
HRM	Iter	10	26	20	25
	CPU	28.8301	49.0126	40.3861	48.7749
	PSNR	22.3136	28.1401	30.0993	24.2470
	Err	0.159416	0.072465	0.068821	0.123838
NPRP	Iter	**6**	**6**	**6**	**6**
	CPU	**22.5660**	**20.0892**	20.4119	**19.4860**
	PSNR	11.3863	13.0485	12.5869	12.1776
	Err	0.560914	0.411816	0.516821	0.496963
MZ	Iter	33	30	29	37
	CPU	27.2350	27.0357	26.4757	30.5668
	PNSR	**26.5342**	**28.2927**	**32.9605**	**25.5351**
	Err	**0.1003**	**0.071202**	**0.0424**	**0.1020**
MP-CG	Iter	**6**	**6**	**6**	7
	CPU	22.9555	24.2128	23.7843	26.3187
	PNSR	11.4052	13.0353	12.6075	12.1781
	Err	0.5596	0.4124	0.5156	0.4969
CG-DESCENT	Iter	23	23	26	18
	CPU	32.5424	34.5871	34.7645	29.9523
	PNSR	26.2393	28.2483	31.7190	24.3003
	Err	0.1014	0.0715	0.0571	0.1230

Table 4.6: The numerical outcomes of restoring images with 30% salt-and-pepper.

Methods \ Images		Man	Boat	Hill	Bridge
HRM	Iter	14	**14**	18	19
	CPU	14.4333	16.1088	19.6281	18.3902
	PSNR	31.4316	33.5530	34.8439	28.5705
	Err	0.055800	0.038858	0.0376	0.075280
NHS	Iter	26	26	24	23
	CPU	15.7869	14.9272	14.8145	14.4291
	PSNR	15.7231	17.5183	17.0234	16.2491
	Err	0.3405	0.2462	0.2931	0.3110
hPB	Iter	13	17	17	14
	CPU	16.5572	**11.2373**	16.3538	16.4848
	PSNR	**31.5644**	**33.7000**	**34.98430**	28.4354
	Err	**0.054953**	**0.038206**	**0.0370**	0.076460
PRP^+	Iter	**9**	**14**	17	**13**
	CPU	**11.7255**	17.2765	17.4224	**12.5381**
	PSNR	29.8526	32.7850	34.9240	28.3708
	Err	0.066923	0.042450	0.0373	0.077031
PRP	Iter	11	11	16	14
	CPU	16.0896	14.6281	16.4851	15.5467
	PSNR	31.3374	33.1674	34.7344	28.5131
	Err	0.0564	0.0406	0.0381	0.0757
BRB	Iter	13	15	**15**	14
	CPU	17.8876	19.6420	**16.0520**	17.0935
	PSNR	31.4483	32.7617	34.8471	28.5141
	Err	0.055693	0.042564	0.0376	0.075770
hPB-SW	Iter	13	15	14	15
	CPU	18.3206	17.9993	16.6398	19.4814
	PSNR	28.5141	31.4483	34.8471	**32.7617**
	Err	0.07577	0.055693	0.0376	**0.042564**

Figure 4.9: The restored images with 30% salt-and-pepper by NMFR, HRM, NHS, NMFR, NPRP, MZ, MP-CG, and CG-DECENT.

Figure 4.10: The restored images with 50% salt-and-pepper by NMFR, HRM, NHS, NMFR, NPRP, MZ, MP-CG, and CG-DECENT.

Figure 4.11: The restored images with 70% salt-and-pepper by NMFR, HRM, NHS, NMFR, NPRP, MZ, MP-CG, and CG-DECENT.

Figure 4.12: The restored images with 30% salt-and-pepper by HRM, NHS, hPB, PRP$^+$, PRP, BRB and hPB-SW.

Figure 4.13: The restored images with 50% salt-and-pepper by HRM, NHS, hPB, PRP$^+$, PRP, BRB and hPB-SW.

Figure 4.14: The restored images with 70% salt-and-pepper by HRM, NHS, hPB, PRP$^+$, PRP, BRB and hPB-SW.

Table 4.7: The numerical outcomes of restoring images with 50% salt-and-pepper.

Methods \ Images		Man	Boat	Hill	Bridge
HRM	Iter	19	**14**	21	**14**
	CPU	31.6329	22.9351	29.2036	21.4267
	PSNR	29.0482	30.0439	32.5694	26.2812
	Err	0.073418	0.058202	0.0489	0.097982
NHS	Iter	31	34	30	30
	CPU	24.3502	24.8263	**21.1211**	23.0874
	PSNR	13.4363	15.2070	14.7059	14.1568
	Err	0.4430	0.3212	0.3828	0.3957
hPB	Iter	19	24	**15**	18
	CPU	26.5356	22.2738	22.325	23.2843
	PSNR	**29.1751**	**31.1843**	**32.6828**	26.7130
	Err	**0.072353**	**0.051041**	**0.0483**	0.093230
PRP^+	Iter	**5**	17	20	16
	CPU	**16.5329**	22.4427	29.8420	**21.0229**
	PSNR	15.3421	30.9165	32.4165	26.5201
	Err	0.355716	0.052639	0.0498	0.095324
PRP	Iter	20	17	21	24
	CPU	36.9955	24.7400	30.9204	38.2190
	PSNR	28.9228	30.8359	32.5124	26.4954
	Err	0.0744	0.0531	0.0492	0.0955
BRB	Iter	24	17	**15**	22
	CPU	29.8683	**21.7964**	22.5881	27.3311
	PSNR	29.1394	31.1089	30.7900	26.6435
	Err	0.072651	0.051485	0.0600	0.093978
hPB-sw	Iter	19	24	**15**	22
	CPU	24.5807	22.4704	22.6654	19.7673
	PSNR	26.6435	29.1394	30.7900	**31.1089**
	Err	0.093978	0.072651	0.0600	**0.051485**

Table 4.8: The numerical outcomes of restoring images with 70% salt-and-pepper.

Methods \ Images		Man	Boat	Hill	Bridge
HRM	Iter	10	26	19	25
	CPU	28.8301	49.0126	44.8694	48.7749
	PSNR	22.3136	28.1401	27.8223	24.2470
	Err	0.159416	0.072465	0.0845	0.123838
NHS	Iter	34	37	34	31
	CPU	29.7937	30.6846	**29.0856**	**22.3260**
	PSNR	11.3935	12.9901	12.5343	**12.1899**
	Err	0.5604	0.4146	0.4915	0.4963
hPB	Iter	40	34	33	37
	CUP	30.8349	28.3926	31.1979	30.1883
	PNSR	**26.3222**	**28.2927**	**29.8827**	24.5192
	Err	**0.100484**	**0.071202**	**0.0667**	0.120017
PRP^+	Iter	29	24	16	21
	CPU	50.3015	35.5985	29.1123	35.9517
	PSNR	26.1898	28.2480	29.3224	24.1335
	Err	0.102028	0.071570	0.0711	0.125468
PRP	Iter	20	15	**14**	15
	CPU	34.7941	28.4139	32.8371	41.2126
	PSNR	26.1118	27.5034	25.6558	22.0517
	Err	0.1029	0.0779	0.1085	0.1594
BRB	Iter	**5**	21	30	**20**
	CPU	**20.0740**	27.3770	38.0558	31.7151
	PSNR	13.1600	28.2074	29.7744	24.1351
	Err	0.457310	0.071905	0.0675	0.125444
hPB-sw	Iter	20	**5**	30	21
	CPU	32.7535	**20.4136**	38.0359	29.5325
	PSNR	24.1351	13.16	29.7744	**28.2074**
	Err	0.125444	0.45731	0.0675	**0.071905**

5

An improved PRP conjugate gradient method for unconstrained optimization

In this chapter, we study the convergence properties of the PRP conjugate gradient method under conditions (4.2) and (4.3) that are proposed to compute a step size α_k for the MZ and hPB conjugate gradient methods. However, in the present approach, we select the following parameters $\sigma \in \left(0, \frac{\mu-1}{1.2\mu^2}\right]$ with $1 < \mu < 2$, and $0 < \delta < \sigma$ to assure a sufficient descent property. The generated descent direction and the convergence properties of the PRP method are studied under the new line search conditions, where the global convergence is proven under mild assumptions. The proposed approach is evaluated on various test functions, and a comparison with similar recent PRP variants is carried out.

We consider in this work the following unconstrained optimization problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable with gradient $g(x) = \nabla f(x)$. Conjugate gradient methods are considered among the most efficient methods for solving large-scale optimization problems. Among them, the PRP variant has attracted the attention of many researchers thanks to its straightforward iterative process and low memory requirements.

Although the PRP method is considered classical, it remains competitive and robust for solving large-scale complex problems. Despite its practical success, global

convergence is uncertain for non-convex minimization, and numerous studies have discussed this drawback. Powell [39] proved the divergence of the PRP method for non-convex minimization by giving a counterexample. Moreover, Dai [13] proved that the PRP method may generate ascent directions during the searching process. To overcome these difficulties, Gilbert and Nocedal [21] proposed a modified β_k parameter by setting $\beta_k = \max\{0, \beta_k^{PRP}\}$ and the proposed modified method converges globally with good theoretical properties under Wolfe line search conditions.

In the last decade, many researchers have paid attention to study and enhance the performance of the PRP method. Zhou and Li [65] have proved the global convergence of the PRP method using a modified Armijo line search technique, and the proposed method exhibits good performance. Also, Yuan and his collaborators have analyzed the convergence of the PRP method under several inexact line search techniques; see [60, 58]. Recently, Yuan et al. [59] have studied the convergence of the PRP method using a step size α_k that meets the following conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \alpha_k \delta d_k^t g_k + \delta \frac{\alpha_k^2}{2} \|d_k\|^2 \delta_1,$$
$$g(x_k + \alpha_k d_k)^t d_k \geq \sigma d_k^t g_k + \delta \alpha_k \|d_k\|^2 \delta_1,$$

with $\delta \in (0, \frac{1}{2})$, $\sigma \in (\delta, 1)$ and δ_1 is a sufficiently small scalar. Their proposed modification is competitive, and the generated search directions satisfy the sufficient descent condition.

Building on the above discussion, in order to combine good practical performance and powerful convergence properties, we have analysed the convergence properties of the PRP method under the new conditions (4.2) and (4.3) cited in the previous chapter, but where $\sigma \in \left(0, \frac{\mu-1}{1.2\mu^2}\right]$, $1 < \mu < 2$, and $0 < \delta < \sigma$. The proposed inexact line search technique fits in with the PRP conjugate gradient parameter. The proposed approach, named PRP-NLS (PRP with New Line Search), converges globally under mild assumptions. The algorithm is successfully applied on a broad set of test functions (with varied analytical expressions and structures) that ranges from the simplest to the hardest.

5.1 The inexact line search conditions and the corresponding algorithm

In the proposed PRP-NLS algorithm, we have adopted the following conditions to compute the step size α_k

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \alpha_k \delta d_k^t g_k \frac{\|g_k\|^2}{\|d_k\|^2}, \tag{5.1}$$

$$\left|g(x_k + \alpha_k d_k)^t d_k\right| \leq -\sigma d_k^t g_k \frac{\|g_k\|^2}{\|d_k\|^2}, \tag{5.2}$$

where $\sigma \in \left(0, \frac{\mu-1}{1.2\mu^2}\right]$, with $1 < \mu < 2$ and $0 < \delta < \sigma$. Since $1 < \mu < 2$, it follows that $0 < \delta < \sigma < 1$. The main steps of the PRP-NLS method are sketched in Algorithm 15 below.

Algorithm 15: The PRP-NLS algorithm

Step 0: (Initialization) Choose a scalar $1 < \mu < 2$ and the parameters δ and σ such that $0 < \delta < \sigma < \frac{\mu-1}{1.2\mu^2}$. Choose a scalar $\epsilon > 0$ sufficiently small to stop the algorithm. Compute $f(x_0)$, $g_0 = \nabla f(x_0)$ and set $d_0 = -g_0$. Select a point $x_0 \in \mathbb{R}^n$ and set $k = 0$.

Step 1: If $\|g_k\| \leq \epsilon$, then stop; otherwise go to the next step.

Step 2: Compute the steplegth α_k using the new line search technique (5.1) and (5.2).

Step 3: Put $x_{k+1} = x_k + \alpha_k d_k$ and set $g_{k+1} = \nabla f(x_{k+1})$.

Step 4: **Search direction computation:** if the restart criterion of Powell $|g_{k+1}^T g_k| \geq 0.2 \|g_{k+1}\|^2$ holds, then set $d_{k+1} = -g_{k+1}$; otherwise set $d_{k+1} = -g_{k+1} + \beta_k^{PRP} d_k$.

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

5.2 The sufficient descent property of PRP-NLS algorithm

Before proving the intended property, we first prove the following result that will be needed below.

Theorem 5.1. *Let $\{g_k\}_{k\in\mathbb{N}}$, $\{\beta_k\}_{k\in\mathbb{N}}$ and $\{d_k\}_{k\in\mathbb{N}}$ be the sequences generated by the PRP-NLS Algorithm under the conditions (5.1) and (5.2). Then*

$$\frac{\|g_k\|}{\|d_k\|} \leqslant \mu, \ \forall k \in \mathbb{N}. \tag{5.3}$$

Proof. It clear that, if the Powell condition holds, then $|g_k^t g_{k+1}| \geq 0.2\|g_{k+1}\|^2$, then the search direction d_k is given by $d_k = -g_k$, and the relation (5.3) holds. Now, if Powell condition doesn't hold. We prove the above relation by recurrence. For $k = 0$, the condition (5.3) holds, since $d_0 = -g_0$. Assume the relation (5.3) satisfied for $k \geq 1$, and let prove it for $k+1$. For $k+1$, we have

$$d_{k+1} = -g_{k+1} + \beta_k^{PRP} d_k, \tag{5.4}$$

Multiplying both sides of (5.4) by g_{k+1}^t, we get

$$d_{k+1}^t g_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{PRP} d_k^t g_{k+1}. \tag{5.5}$$

On the other hand, we have

$$\begin{aligned} \left|\beta_k^{PRP}\right| &= \frac{\left|\|g_{k+1}\|^2 - g_{k+1}^t g_k\right|}{\|g_k\|^2} \\ &\leq \frac{\|g_{k+1}\|^2 + \left|g_{k+1}^t g_k\right|}{\|g_k\|^2} \\ &\leq 1.2\frac{\|g_{k+1}\|^2}{\|g_k\|^2}. \end{aligned} \tag{5.6}$$

Hence, from (5.5) and condition (5.2), it follows that

$$\begin{aligned} \|g_{k+1}\|^2 &\leq \left|d_{k+1}^t g_{k+1}\right| + \left|\beta_k^{PRP}\right| \left|d_k^t g_{k+1}\right| \\ &\leq \left|d_{k+1}^t g_{k+1}\right| - \sigma \left|\beta_k^{PRP}\right| d_k^t g_k \frac{\|g_k\|^2}{\|d_k\|^2} \\ &\leq \left|d_{k+1}^t g_{k+1}\right| + \sigma \left|\beta_k^{PRP}\right| \left|d_k^t g_k\right| \frac{\|g_k\|^2}{\|d_k\|^2} \\ &\leq \|d_{k+1}\| \|g_{k+1}\| + \sigma \left|\beta_k^{PRP}\right| \frac{\|g_k\|^3}{\|d_k\|} \\ &\leq \|d_{k+1}\| \|g_{k+1}\| + 1.2\sigma \frac{\|g_k\|}{\|d_k\|} \|g_{k+1}\|^2. \end{aligned} \tag{5.7}$$

Furthermore, since we have

$$\left(1 - 1.2\sigma\frac{\|g_k\|}{\|d_k\|}\right) \geq (1 - 1.2\sigma\mu),$$
$$\geq (1 - \frac{\mu - 1}{\mu}) = \frac{1}{\mu} > 0, \tag{5.8}$$

by dividing both sides of (5.7) by $\|g_{k+1}\| \,.\, \|d_{k+1}\|$, then from (5.8) it results that

$$\frac{\|g_{k+1}\|}{\|d_{k+1}\|} \leq \frac{1}{\left(1 - 1.2\sigma\frac{\|g_k\|}{\|d_k\|}\right)} = \mu.$$

Thus, the proof is complete □

Now, we are in the position to prove the sufficient descent condition.

Theorem 5.2. *The sequences* $\{g_k\}_{k\in\mathbb{N}}$, $\{\beta_k\}_{k\in\mathbb{N}}$ *and* $\{d_k\}_{k\in\mathbb{N}}$ *generated by the PRP-NLS Algorithm under the new line search conditions* (5.1) *and* (5.2) *with* $\sigma \in \left(0, \frac{\mu-1}{1.2\mu^2}\right]$ *and* $1 < \mu < 2$ *satisfy*

$$g_k^t d_k \leqslant -\xi \|g_k\|^2, \quad \forall k \in \mathbb{N} \tag{5.9}$$

where $\xi > 0$.

Proof. It is evident, in the case where the restart criterion of Powell holds i.e, $|g_k^t g_{k+1}| \geq 0.2\|g_{k+1}\|^2$, then $d_k = -g_k$, and the relation (5.9) holds.

Now in case where the Powell condition doesn't hold, we proof the above relation by induction. Indeed, for $k = 0$, the search direction $d_0 = -g_0$, this implies that $d_0^t g_0 = -\|g_0\|^2$, and relation (5.9) holds.
Suppose that (5.9) is true for $k \geq 1$. For $k+1$, by multiplying both sides of (5.4) by g_{k+1}^t, we get

$$\begin{aligned}
d_{k+1}^t g_{k+1} &= -\|g_{k+1}\|^2 + \beta_k^{PRP} d_k^t g_{k+1} \\
&\leq -\|g_{k+1}\|^2 + \left|\beta^{PRP}\right| \left|d_k^t g_{k+1}\right| \\
&\leq -\|g_{k+1}\|^2 + 1.2\sigma\frac{\|g_{k+1}\|^2}{\|g_k\|^2}\|g_k\|\|d_k\|\frac{\|g_k\|^2}{\|d_k\|^2}, \text{(using (5.2) and (5.7))} \\
&\leq -\|g_{k+1}\|^2 + 1.2\sigma\frac{\|g_k\|}{\|d_k\|}\|g_{k+1}\|^2 \\
&\leq -\|g_{k+1}\|^2 (1 - 1.2\sigma\mu)) \leq -\frac{1}{\mu}\|g_{k+1}\|^2, \text{(using relations (5.3) and (5.8)).}
\end{aligned}$$

Then the proof complete for $\xi = \frac{1}{\mu}$. □

5.3 The convergence analysis of PRP-NLS algorithm

Before analysing the convergence of the proposed approach, we first show that it is well-defined. In the next theorem, we prove the existence of a step size α ($0 < \alpha < \infty$) that meets the conditions (5.1) and (5.2), where $0 < \delta < \sigma$, with $\sigma \in \left(0, \frac{\mu-1}{1.2\mu^2}\right]$ and $1 < \mu < 2$.

Theorem 5.3. *Let f be a twice continuously differentiable function that is bounded below. If $g_k^t d_k < 0$, then there exists a strictly positive real value α that meets the conditions (5.1) and (5.2).*

Proof. The proof is similar the proof of Theorem 4.3. □

5.3.1 The global convergence of PRP-NLS algorithm

The global convergence property is crucial for any conjugate gradient method. To establish it, we assume that:

Assumption 5.3.1. *The level set $\Omega = \{x \in \mathbb{R}^n | \ f(x) \leq f(x_0)\}$ is bounded for any starting point x_0.*

Assumption 5.3.2. *f is continuously differentiable whose gradient function g satisfies the Lipschitz condition in some closed neighbourhood $\mathcal{N}$ of Ω, i.e. $\exists L > 0$ such that*

$$\|g(x) - g(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathcal{N}. \tag{5.10}$$

The above assumptions imply the existence of a real number $\Gamma \geq 0$ such that

$$\|g_k\| \leq \Gamma \quad \forall x \in \Omega. \tag{5.11}$$

To establish that the PRP-NLS algorithm convergence globally (see Theorem 5.7 below),we need to prove the following results, which will be needed below.

Lemma 5.4. *Consider the sequences $\{g_k\}_{k\in\mathbb{N}}$, $\{\alpha_k\}_{k\in\mathbb{N}}$ and $\{d_k\}_{k\in\mathbb{N}}$ generated by the PRP-NLS algorithm; then for some $\varpi > 0$ we have*

$$g_k^t d_k \geq -\varpi \|g_k\|^2 \tag{5.12}$$

Proof. By multiplying Equation (5.4) by g_{k+1} we get

$$\left|d_{k+1}^t g_{k+1}\right| \leq \|g_{k+1}\|^2 + \left|\beta_k^{PRP}\right| \left|d_k^t g_{k+1}\right|.$$

Then, from relation (5.3) it follows that,

$$\begin{aligned}\left|d_{k+1}^t g_{k+1}\right| &\leq \|g_{k+1}\|^2 + 1.2\sigma \frac{\|g_k\|}{\|d_k\|} \|g_{k+1}\|^2, \\ &\leq (1 + 1.2\sigma\mu) \|g_{k+1}\|^2,\end{aligned}$$

which means that,

$$-\varpi \|g_{k+1}\|^2 \leq d_{k+1}^t g_{k+1} \leq \varpi \|g_{k+1}\|^2$$

where $\varpi = (1 + 1.2\sigma\mu)$, and the proof is complete. □

Lemma 5.5. *Under the above assumptions, the sequence of step size $\{\alpha_k\}_{k\in\mathbb{N}}$ generated by the PRP-NLS algorithm under the new line search conditions (5.1) and (5.2) with $\sigma \in \left(0, \frac{\mu-1}{1.2\mu^2}\right]$ and $1 < \mu < 2$, satisfies*

$$\alpha_k \geq \frac{\sigma\|g_k\|^2/\|d_k\|^2 - 1}{L\|d_k\|^2} d_k^t g_k > 0, \ \forall k \in \mathbb{N}.$$

Proof. From Theorem 5.1, it results that

$$\sigma \frac{\|g_k\|^2}{\|d_k\|^2} < \frac{\mu - 1}{1.2} < 1,$$

hence, from condition (5.2) it results that

$$0 < \left(\sigma \frac{\|g_k\|^2}{\|d_k\|^2} - 1\right) d_k^t g_k < d_k^t (g_{k+1} - g_k) \leq L\alpha_k \|d_k\|^2,$$

which completes the proof. □

Lemma 5.6. *Under Assumption 5.3.1, the sequences $\{g_k\}_{k\in\mathbb{N}}$ and $\{d_k\}_{k\in\mathbb{N}}$ produced by the PRP-NLS algorithm satisfy*

$$\sum_{k\geq 0} \frac{(g_k^t d_k)^2}{\|d_k\|^2} < +\infty. \tag{5.13}$$

Proof. Using the condition (5.1) and relation (5.12), we have

$$\begin{aligned}f(x_k) - f(x_k + \alpha_k d_k) &\geq -\delta\alpha_k g_k^t d_k \frac{\|g_k\|^2}{\|d_k\|^2} \\ &\geq \frac{\delta\alpha_k}{\varpi} \frac{(g_k^t d_k)^2}{\|d_k\|^2}.\end{aligned}$$

Then we have,

$$\frac{\delta\alpha_k}{\varpi}\frac{(g_k^t d_k)^2}{\|d_k\|^2} \leq f(x_k) - f(x_{k+1}).$$

Let $m = \min\{\alpha_k : k \in \mathbb{N}\}$. Under Assumption 5.3.1, by summing this inequality from $k = 0$ to infinity, we get

$$\frac{\delta m}{\varpi}\sum_{k\geq 0}\frac{(g_k^t d_k)^2}{\|d_k\|^2} \leq \sum_{k\geq 0}\frac{\delta\alpha_k}{\varpi}\frac{(g_k^t d_k)^2}{\|d_k\|^2} < +\infty.$$

and the inequality (5.13) holds.

□

Now, we are in a position to address the global convergence of the PRP-NLS algorithm.

Theorem 5.7. *Under the above assumptions, the PRP-NLS algorithm converges globally in the sense that,*

$$\lim_{k\to+\infty}\inf\|g_k\| = 0. \tag{5.14}$$

Proof. Assume that the assertion (5.14) doesn't hold. Then, there exists a strictly positive value $r > 0$ such that

$$\|g_k\| > r, \ \forall k \in \mathbb{N}. \tag{5.15}$$

Let $m = \min\{\alpha_k : k \in \mathbb{N}\}$ and $D = \max\{\|x - y\| : x, y \in \Omega\}$. From relation (4.1), it results that

$$\|d_k\|^2 = \frac{\|x_{k+1} - x_k\|^2}{\alpha_k^2} \leq \frac{D^2}{m^2}.$$

On the other hand, from (5.6), (5.15)and (5.11) we get

$$\begin{aligned} d_{k+1} &\leq \|g_{k+1}\| + \left|\beta_k^{PRP}\right| \|d_k\| \\ &\leq \Gamma + \left(\frac{1.2\Gamma^2}{r^2}\right)\frac{D}{m} = M, \end{aligned}$$

hence,

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

But, from (5.15), (5.9) and (5.13) it results that

$$\xi^2 r^4 \sum_{k\geq 0}\frac{1}{\|d_k\|^2} \leq \sum_{k\geq 0}\frac{\xi^2\|g_k\|^4}{\|d_k\|^2} \leq \sum_{k\geq 0}\frac{(g_k^t d_k)^2}{\|d_k\|^2} < +\infty,$$

therefore

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

which contradicts the claim (5.15), so the assertion (5.14) is true.

□

5.4 Computational experiments

Here, we present a series of computational performances concerning the PRP-NLS algorithm, applied on 270 test functions taken from CUTEr [3] and [2], as listed in Table 5.1, with dimensions ranging from 2 to 100000, as well as on restoring four images with different noise levels. All the codes are written and implemented in Matlab. In all experiments, the PRP-NLS algorithm is implemented with the setting parameters $\mu = 1.6$, $\delta = 10^{-4}$ and $\sigma = 10^{-3}$.

To demonstrate the effectiveness of the suggested approach, we compared it with PRP [44](with $\theta = 0.4$), PRP$^+$ [21] (with $\delta = 10^{-4}$ and $\sigma = 10^{-3}$) and PRPMWWP [59] (with $\delta = 0.1\delta^* = 0.05\delta_1 = 10^{-16}$ and $\sigma = 0.9$). For this comparison, the same starting point is assigned for each test problem, and each implementation is considered successful if a point x_k where $\|g(x_k)\|_\infty \leq 10^{-6}$ is reached within 2000 iterations, with CPU time less than 500 seconds; alternatively, the implementation is assigned as a failure.

Using the logarithmic performance profile of Dolan and Moré [16], Figures 5.1-5.4 illustrate that the PRP-NLS method outperforms the others; notably, it is faster for approximately 65% of the test problems and successfully solves around 96% of them, followed by the NPRP method with 94% of test problems. The PRPMWWP method has solved around 93% of test problems, whereas the PRP method has solved about 88% the test problems. These outcomes prove the competitiveness and rapid convergence of the PRP-NLS algorithm in the majority of the test problems.

Function	Dimension n	Function	Dimension n
Extended Maratos	500, 700, 1000,2000,4000, 9000, 9500, 60000, 10000, 15000	Arwhead	50, 60, 80, 100, 150, 200, 800, 1000, 5000
COSINE	10, 100 , 500, 1000	SINE	10, 100 , 500, 1000
ENGVAL1	100, 600, 800, 1000, 1500, 1600,1800, 10000	Diagonal 1	2,10, 100, 1000, 10000, 100000
Generalized Tridiagonal 1	2, 10, 20,300, 500, 700, 10000	FLETCHCR	2, 4
Extended White and Holst	1000, 2000, 3000, 4000, 5000, 6000	Diagonal 2	2, 4, 10, 800, 1000, 80000
Diagonal 7	500, 700, 1000, 1500, 2000, 7000, 8000	Extended Rosenbrock	10, 20,100 ,1200, 3000, 4000, 5000
Quadratic QF2	100, 200,1000,5000, 7000, 9000, 10000, 50000	Diagonal 8	100, 200, 400, 500, 1000, 1500, 2000
Extended Freudenstein and Roth	10, 100, 1000, 4000, 9000, 10000, 20000, 50000, 60000, 80000	Diagonal 3	2, 4, 6, 10, 50, 100, 200, 400, 700
DENSCHNF	10, 100, 10000, 25000, 30000 50000, 70000, 80000, 90000	Extended Himmelblau	4, 6,10, 9000, 10000
Perturbed Quadratic	2	POWER	2
QUARTC	2	Raydan 1	2
Generalized Rosenbrock	2, 50, 800, 1000	Perturbed quadratic diagonal	2
DENSCHNB	10, 90, 100, 2000, 3000, 4000, 5000, 6000, 7000, 9000	Raydan 2	1000, 4000, 50000,80000
HIMMELBG	2000, 3000, 6000, 30000, 50000, 80000	LIARWHD	10,50, 4000,5000, 5500, 10000, 20000, 80000
Extended quadratic exponential EP1	40000, 50000, 60000, 70000	Diagonal 5	100, 200, 700, 1000, 1500, 2000, 2200, 2500
Extended BD1	4, 800, 900, 2000, 3000, 5000 20000, 40000, 60000, 70000, 80000	Extended quadratic penalty QP1	4, 6, 8, 10, 50, 100, 700, 1000, 1500
Hager	2, 4,10,50, 80, 150, 300	NONSCOMP	2, 4, 1000, 5000, 70000
HIMMELLH	10, 50, 300, 500, 10000,50000, 60000, 80000, 10000	Quadratic QF1	5000, 6000, 8000, 9000, 20000, 50000, 70000, 80000
Extended quadratic penalty QP2	40, 60, 200	Diagonal 4	20000, 30000, 40000, 50000, 60000, 70000
DIXON3DQ	2, 4, 600	Extended PSC1	2, 4, 10, 100, 1000,10000
Almost Perturbed Quadratic	2, 4, 6	Diagonal 9	10, 30,50, 60, 70, 90
Extended Tridiagonal 1	6, 10, 20,80, 90, 100, 150, 300, 500, 700, 1000, 5000, 6000	NONDIA	10, 100,500,1000,5000,100000

Table 5.1: List of test functions.

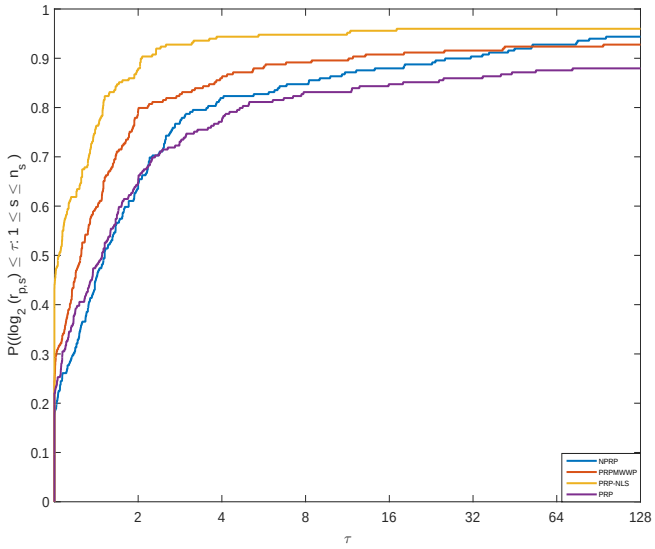

Figure 5.1: Performance profiles plot based on CPU time.

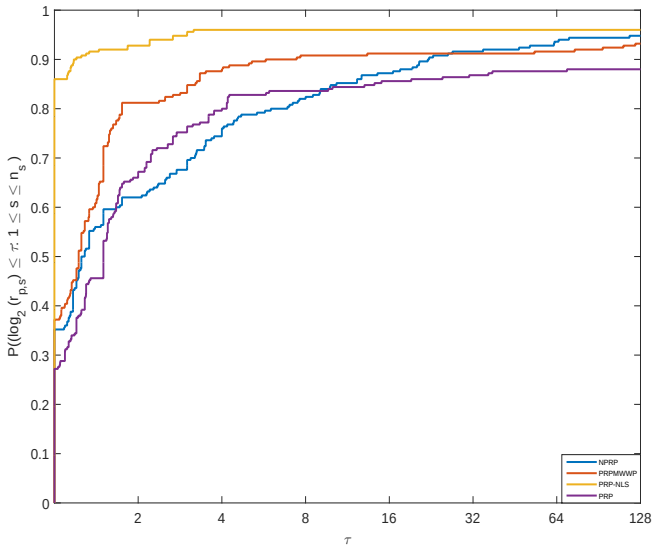

Figure 5.2: Performance profiles plot based on the number of iterations.

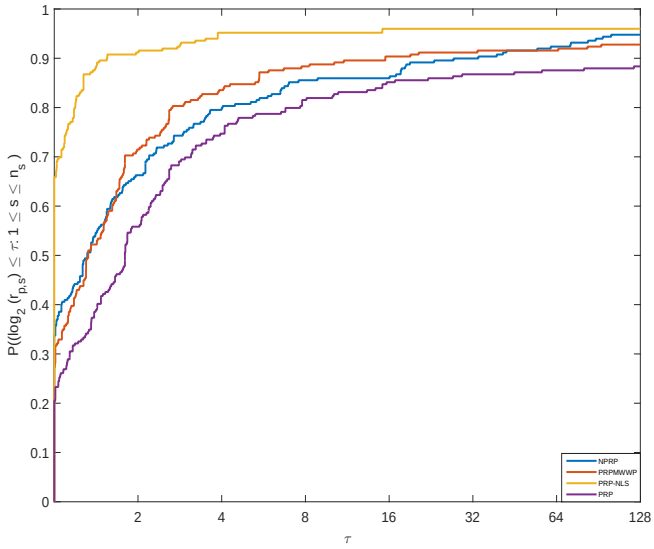

Figure 5.3: Performance profiles plot based on the number of function evaluations.

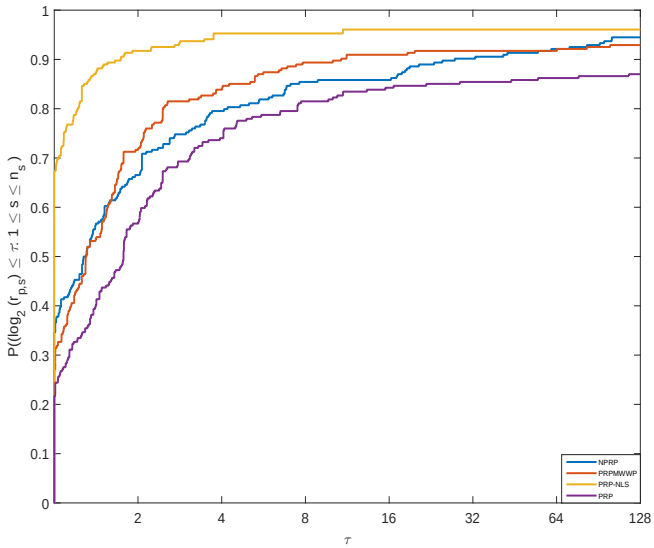

Figure 5.4: Performance profiles plot based on the number of gradient evaluations.

Conclusion and perspectives

This thesis focuses on the development and analysis of novel nonlinear conjugate gradient methods for solving unconstrained optimization problems. Motivated by the central role of optimization in various scientific and engineering fields, we explored innovative strategies to enhance classical methods through advanced line search techniques and hybrid approaches.

In the first part of this work, we introduced an improved conjugate gradient algorithm incorporating a newly adapted line search strategy. Theoretical analysis established its descent property and global convergence under standard assumptions. Numerical experiments demonstrated its superior performance compared to classical techniques, particularly on large-scale test problems and practical applications such as image restoration.

This research led to three main contributions:

1. A novel conjugate gradient parameter for nonlinear unconstrained optimization problems, combined with an adapted line search technique, published in the Electronic Research Journal [33].

2. A hybrid approach combining the PRP and BRB (Rahali–Belloufi–Benzine) methods, published in the International Journal of Applied Mathematics and Computer Science [34].

3. An enhanced PRP (Polak–Ribière–Polyak) method, incorporating an appropriately adjusted step length to improve convergence.

Beyond algorithm development, this thesis highlights the crucial role of line search strategies in nonlinear optimization, showing that their careful design and integration can significantly improve algorithmic performance. Applications to image restoration illustrate the practical relevance of the proposed methods.

Perspectives and future work

Several directions for future research naturally arise from this work, including:

- Exploring alternative hybridization strategies and adaptive parameter-tuning mechanisms.
- Investigating preconditioning techniques to accelerate convergence for large-scale problems.
- Applying the developed algorithms to large-scale machine learning and data-driven problems.

Overall, these perspectives suggest that further advancements in nonlinear conjugate gradient methods could continue to balance convergence guarantees with computational efficiency, while addressing increasingly complex and practical applications.

Bibliography

[1] M. Al-Baali, "Convergence of nonlinear conjugate gradient methods with inexact line searches," *Comput. Optim. Appl.*, vol. 7, pp. 87–108, 1997.

[2] N. Andrei, "Another nonlinear conjugate gradient algorithm for unconstrained optimization," *Optim. Methods Softw.*, vol. 24, no. 1, pp. 89–104, 2009.

[3] N. Andrei, "An unconstrained optimization test functions collection," *Adv. Model. Optim.*, vol. 10, pp. 147–161, 2008.

[4] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific J. Math.*, vol. 16, no. 1, pp. 1–3, 1966.

[5] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Math. Comput.*, vol. 19, no. 92, pp. 577–593, 1965.

[6] C. G. Broyden, "Quasi-Newton methods and their application to function minimization," *Math. Comput.*, vol. 21, no. 99, pp. 368–381, 1967.

[7] H. Cartan, "Calcul Différentiel," *Hermann, Paris*, 1967, Theorem 5.6.3.

[8] R. L. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *Compt. Rend. Acad. Sci.*, vol. 25, pp. 536–538, 1847.

[9] Y. Chen, H. Huang, K. Huang, M. Roohi, and C. Tang, "A selective chaos-driven encryption technique for protecting medical images," *Physica Scripta*, vol. 100, 0152a3, 2025.

[10] G. Collomb, W. Härdle, and S. Hassani, "Anoteon prediction via estimation of the conditional mode function," *J. Stat. Plann. Inference*, vol. 15, pp. 227–236, 1987.

[11] J. W. Daniel, "The conjugate gradient method for linear and nonlinear operator equations," *SIAM J. Numer. Anal.*, vol. 4, no. 1, pp. 10–26, 1967.

[12] W. C. Davidon, "Variable metric method for minimization," *SIAM J. Optim.*, vol. 1, pp. 1–17, 1959.

[13] Y. H. Dai, "Analysis of conjugate gradient methods," Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Science, 1997.

[14] Y. H. Dai, J. Y. Han, G. H. Liu, D. F. Sun, H. X. Yin, and Y. Yuan, "Convergence properties of nonlinear conjugate gradient methods," *SIAM J. Optim.*, vol. 10, no. 2, pp. 345–358, 1999.

[15] Y. H. Dai and Y. Yuan, "An efficient hybrid conjugate gradient method for unconstrained optimization," *Ann. Oper. Res.*, vol. 103, pp. 33–47, 2001.

[16] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Math. Program.*, vol. 91, pp. 201–213, 2002.

[17] S. S. Djordjevic, "New hybrid conjugate gradient method as a convex combination of LS and CD methods," *Filomat*, vol. 31, no. 6, pp. 1813–1825, 2017.

[18] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Comput. J.*, vol. 6, no. 2, pp. 163–168, 1963.

[19] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley, 1997.

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

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

[22] N. I. M. Gould, D. Orban, and Ph. L. Toint, "CUTEr: a constrained and unconstrained testing environment," *ACM Trans. Math. Softw.*, vol. 29, no. 4, pp. 373–394, 2003.

[23] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Math. Comput.*, vol. 24, no. 109, pp. 23–26, 1970.

[24] A. A. Goldstein and J. F. Price, "An effective algorithm for minimization," *Num. Math.*, vol. 10, pp. 184–189, 1969.

[25] A. A. Goldstein, "On steepest descent," *SIAM J. Control*, vol. 3, pp. 147–151, 1965.

[26] J. Y. Han, G. H. Liu, and H. X. Yin, "Convergence properties of conjugate gradient methods with strong Wolfe linesearch," *Syst. Sci. Math. Sci.*, vol. 11, pp. 112–116, 1998.

[27] M. Hamoda, M. Mamat, M. Rivaie, and Z. Salleh, "A conjugate gradient method with strong Wolfe-Powell line search for unconstrained optimization," *Appl. Math. Sci.*, vol. 10, no. 13–16, pp. 689–704, 2016.

[28] W. W. Hager and H. Zhang, "Algorithm 851: CG DESCENT, a conjugate gradient method with guaranteed descent," *ACM Trans. Math. Softw.*, vol. 32, no. 1, pp. 113–137, 2006.

[29] H. Heuser, "Lehrbuch der Analysis. Teil 2, 8. Auflage," *Teubner, Leipzig*, 2002, Satz 168.1.

[30] H. M. Khudhur and I. H. Halil, "Noise removal from images using the proposed three-term conjugate gradient algorithm," *Comput. Res. Model.*, vol. 16, no. 4, pp. 841–853, 2024.

[31] J. K. Liu and S. J. Li, "New hybrid conjugate gradient method for unconstrained optimization," *Appl. Math. Comput.*, vol. 245, pp. 36–43, 2014.

[32] I. E. Livieris and P. Pintelas, "Globally convergent modified Perry conjugate gradient method," *Appl. Math. Comput.*, vol. 218, pp. 9197–9207, 2012.

[33] A. Maiza, R. Ziadi, M. A. Saleh, A. Z. Almaymuni, "An improved conjugate gradient algorithm by adapting a new line search technique," *Electron. Res. Arch.*, vol. 33, no. 4, pp. 2148–2171, 2025.

[34] A. Maiza, R. Ziadi, M. A. Saleh, A. Z. Almaymuni, "A Combined Conjugate Gradient Method Based on a New Inexact Line Search Technique for Unconstrained Minimization and Image Restoration Problems," *Int. J. Appl. Math. Comput. Sci.*, vol. 33, no. 2, 2025.

[35] A. E. Mehamdia, Y. Chaib, and T. Bechouat, "Two improved nonlinear conjugate gradient methods with application in conditional model regression function," *J. Ind. Manag. Optim.*, vol. 21, no. 2, pp. 658–675, 2025.

[36] O. O. O. Yousif and M. A. Saleh, "Another modified version of RMIL conjugate gradient method," *Appl. Numer. Math.*, vol. 202, pp. 120–126, 2024.

[37] O. O. O. Yousif, R. Ziadi, M. A. Saleh, and A. Z. Almaymuni, "Another updated parameter for the Hestenes-Stiefel conjugate gradient method," *Int. J. Anal. Appl.*, vol. 23, 2025.

[38] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[39] J. M. Powell, "Nonconvex minimization calculations and the conjugate gradient method," in *Numerical Analysis: Proceedings of the 10th Biennial Conference held at Dundee, Scotland, June 28–July 1, 1983.* Springer Berlin Heidelberg, 1984, pp. 122–141.

[40] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Natl. Bur. Stand.*, vol. 49, no. 1, pp. 409–436, 1952.

[41] N. Rahali, M. Belloufi, and R. Benzine, "A new conjugate gradient method for acceleration of gradient descent algorithms," *Moroccan J. Pure Appl. Anal.*, vol. 7, no. 1, pp. 1–11, 2021.

[42] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, 1970.

[43] M. Rivaie, M. Mamat, L. W. June, and I. Mohd, "A new class of nonlinear conjugate gradient coefficients with global convergence properties," *Appl. Math. Comput.*, vol. 218, pp. 11323–11332, 2012.

[44] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Comput. Math. Math. Phys.*, vol. 9, no. 4, pp. 94–112, 1969.

[45] E. Polak and G. Ribière, "Note sur la convergence de méthodes des directions conjuguées," *Rev. Fr. Inform. Rech. Opér.*, vol. 16, pp. 35–43, 1969.

[46] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: theory," *J. Optim. Theory Appl.*, vol. 69, pp. 129–137, 1991.

[47] M. A. Saleh, "Enhancing Deep Learning Optimizers for Detecting Malware Using Line Search Method under Strong Wolfe Conditions," in *Proc. 2023 3rd Int. Conf. Comput. Inf. Technol. (ICCIT)*, 2023, pp. 222–226.

[48] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Math. Comput.*, vol. 24, no. 111, pp. 647–656, 1970.

[49] I. M. Sulaiman, P. Kaelo, R. Khalid, and M. K. M. Nawawi, "A descent generalized RMIL spectral gradient algorithm for optimization problems," *Int. J. Appl. Math. Comput. Sci.*, vol. 34, pp. 225–233, 2024.

[50] P. S. Stanimirović, B. Ivanov, M. Haifeng, and D. Mosić, "A Survey of gradient methods for solving nonlinear optimization," *Electron. Res. Arch.*, vol. 4, pp. 1573, 2020.

[51] D. Touati-Ahmed and C. Storey, "Efficient hybrid conjugate gradient technique," *J. Optim. Theory Appl.*, vol. 64, pp. 379–397, 1990.

[52] W. Wolfe, "Convergence conditions for ascent methods," *SIAM J. Optim.*, vol. 6, no. 4, pp. 1260–1273, 1979.

[53] P. Wolfe, "Convergence conditions for ascent methods," *SIAM Rev.*, vol. 11, pp. 226–235, 1969.

[54] P. Wolfe, "Convergence conditions for ascent methods. II: Some corrections," *SIAM Rev.*, vol. 13, pp. 185–188, 1971.

[55] J. Lebl, "Basic Analysis I: Introduction to Real Analysis, Volume I," *Open textbook (PDF)*, ver. 6.2, May 23, 2025.

[56] K. Xiong and S. Wang, "The online random Fourier features conjugate gradient algorithm," *IEEE Signal Process. Lett.*, vol. 26, no. 5, pp. 740–744, 2019.

[57] K. Xiong, H. H. Iu, and S. Wang, "Kernel Correntropy Conjugate Gradient Algorithms Based on Half-Quadratic Optimization," *IEEE Trans. Cyber.*, vol. 55, no. 11, pp. 5497–5510, 2020.

[58] G. Yuan, Z. Wei, Y. Yang, "The global convergence of the Polak-Ribière-Polyak conjugate gradient algorithm under inexact line search for nonconvex functions," *J. Comput. Appl. Math.*, vol. 362, pp. 262–275, 2019.

[59] G. Yuan, L. Guan, Z. Wang, "The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems," *Appl. Math. Comput.*, vol. 152, pp. 1–11, 2020.

[60] G. Yu, L. Guan, Z. Wei, "Globally convergent Polak-Ribière-Polyak conjugate gradient methods under a modified Wolfe line search," *Appl. Math. Comput.*, vol. 215, pp. 3082–3090, 2009.

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

[62] L. Zhang, "An improved Wei-Yao-Liu nonlinear conjugate gradient method for optimization computation," *Appl. Math. Comput.*, vol. 215, no. 6, pp. 2269–2274, 2009.

[63] M. Zhang, X. Wang, X. Chen, and A. Zhang, "The kernel conjugate gradient algorithms," *IEEE Trans. Signal Process.*, vol. 66, pp. 4377–4387, 2018.

[64] X. Zhang, "Notes for Optimization Algorithms," *Spring 2023, Department of Mathematics, Purdue University.*

[65] W. Zhou, "On the convergence properties of the unmodified PRP method with a non-descent line search," *Opt. Meth. Soft.*, vol. 29, no. 3, pp. 484–496, 2014.

[66] R. Ziadi and A. Bencherif-Madani, "A mixed algorithm for smooth global optimization," *J. Math. Model.*, vol. 11, pp. 207–228, 2024.

[67] R. Ziadi, R. Ellaia, and A. Bencherif-Madani, "Global optimization through a stochastic perturbation of the Polak-Ribière conjugate gradient method," *J. Comput. Appl. Math.*, vol. 317, pp. 672–684, 2017.

[68] R. Ziadi and A. Bencherif-Madani, "A perturbed quasi-Newton algorithm for bound-constrained global optimization," *J. Comput. Math.*, vol. 43, pp. 143–173, 2025.

[69] G. Zoutendijk, "Nonlinear Programming, Computational Methods," in *Integer and Nonlinear Programming*, J. Abadie (ed.), North-Holland, Amsterdam, 1970, pp. 37–86.

ملخص:
هذه الأطروحة تتناول دراسة نظرية وعددية لبعض الخوارزميات لحل مسائل الأمثلة غير الخطية، وخاصة طرق التدرج المترافق. في هذا السياق، اقترحنا تحسين خوارزمية بولاك - ريبيير- بولياك من خلال تعديل خطوة الانتقال. علاوة على ذلك، تم اقتراح خوارزميتين جديدين لتجويد دوال تفاضلية غير خطية.
التجارب العددية تظهر أن الخوارزميات المقترحة قدمت نتائج مرضية لحل مسائل ذات الأبعاد الكبيرة، مما يعزز الأداء العام لهذه الخوارزميات في التطبيقات العملية.

كلمات مفتاحية : الأمثلة غير المقيدة، الامثلة غير الخطية، طرق التدرج المترافق، البحث الخطي، التقارب الشامل.

Abstract :

This thesis presents a theoretical and numerical study of various methods for solving nonlinear optimization problems, with a particular focus on conjugate gradient methods. In this context, we propose an enhancement of the Polak–Ribière–Polyak method by introducing a new steplength. Furthermore, two novel approaches have been developed for the unconstrained minimization of nonlinear and differentiable cost functions. The numerical results demonstrate that the proposed methods offer significant improvements in solving large-scale problems, thereby enhancing the overall performance of these algorithms in practical applications.

Key-words: Unconstrained optimization, Nonlinear optimization, Conjugate gradient methods, Line search, Global convergence.

Résumé :

Cette thèse porte sur l'étude théorique et numérique de certaines méthodes de résolution de problèmes d'optimisation non linéaire, en particulier les méthodes de gradient conjugué. Dans ce cadre, nous avons proposé une amélioration de la méthode de Polak-Ribière-Polyak en introduisant un nouveau pas de déplacement. Par ailleurs, deux nouvelles approches ont été développées pour la minimisation sans contraintes de fonctions coût non linéaires et différentiables. Les résultats numériques obtenus montrent que les méthodes proposées offrent des améliorations significatives pour la résolution de problèmes de grande dimension, renforçant ainsi la performance globale de ces algorithmes dans des applications pratiques.

Mots-clés : Optimisation sans contraintes, Optimisation non linéaire, Méthodes du gradient conjugué, Recherche linéaire, Convergence globale.