



People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific
Research
Larbi Tébessi University - Tébessa
Faculty of Exact Sciences and Natural and Life
Sciences
Department of Mathematics



Master thesis in Mathematics
Option: Partial Derivative Equations and Applications

Topic:

**A novel hybrid conjugate gradient
algorithm constructed from multiple
methods**

Presented by:

Rayan Ghoul

Against the jury:

<i>Mr, Kamel Akrouf</i>	<i>Prof</i>	<i>Larbi Tébessi University</i>	<i>President</i>
<i>Mr, Tahar Bouali</i>	<i>Prof</i>	<i>Larbi Tébessi University</i>	<i>Examiner</i>
<i>Mr, Nouar Degaichi</i>	<i>MAA</i>	<i>Larbi Tébessi University</i>	<i>Supervisor</i>

Date of defense: 02/06/2026

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dedication

I dedicate this work to
to my parents
in gratitude for their unwavering love for me, their constant encouragement, their sacrifices and
their prayers Without their prayers, I would never have got this far.

To my three brothers
who believed in my abilities and encouraged me from a young age
To my brother Zakaria, for his moral support rather than material assistance
To my brother Ala E-dine, who was my first companion at university,
and to my little brother Youcef, whom I consider a source of security and tenderness

To my best friend Wiam.
Even though university separated us, she is still with me through everything.
To my dear husband
who has encouraged me ever since I met him, and has been my greatest source of motivation and
role model in my studies

To my dear friends,
Thank you for your sincere friendship, and for your constant support and encouragement.

And especially to my dear friend Chaima,
who has been by my side since the very beginning of this journey, and has been my greatest
source of support and comfort at every turn.

Finally, to all those who have helped me, directly or indirectly

Acknowledgements

First and foremost, all praise and thanks are due to Allah for granting me the strength and perseverance to complete this work.

I would like to express my deepest gratitude and sincere appreciation to my supervisor, Dr.Degaichi Nouar , for his invaluable guidance, continuous support, insightful advice, and constant encouragement throughout this research. His dedication and scientific rigor have been a great source of inspiration.

I am also profoundly grateful to Prof. Dr.Akrout Kamel and Prof.Dr. Tahar bouali for kindly accepting to evaluate and examine this thesis, and for their thoughtful time and expertise. My sincere thanks go to the University of Tebessa, particularly the Faculty of Exact Sciences, Nature and Life Sciences, and the Department of Mathematics, for providing a stimulating academic environment and the necessary facilities to carry out this work.

I wish to extend my appreciation to my academic training and to the completion of this thesis.

Special thanks to my fellow graduate students for their friendship and moral support.

Finally, I owe my deepest gratitude to my beloved family – my parents, siblings, and all my loved ones – for their unconditional love, endless prayers, sacrifices, and unwavering encouragement.

Without them, this achievement would not have been possible.

ملخص

يقدم هذا العمل دراسة لطريقة التدرج المترافق لحل أنظمة المعادلات الخطية الكبيرة والمتفرقة. تعتمد هذه الطريقة على بناء تسلسل من الاتجاهات المترافقة وتحديد أحجام الخطوات المثلى لتقليل الدالة التربيعية المرتبطة بها إلى أدنى حد ممكن بكفاءة. كما يتم فحص ومناقشة خصائصها العددية الرئيسية، بما في ذلك سلوك التقارب والكفاءة الحسابية.

الكلمات المفتاحية: التدرج المترافق، الأنظمة المتفرقة، المعادلات الخطية، التقارب، الطرق العددية.

Abstract / Résumé

Abstract

This work presents a study of the Conjugate Gradient method for solving large, sparse systems of linear equations. The method relies on constructing a sequence of conjugate directions and determining optimal step sizes to efficiently minimize the associated quadratic function. Its main numerical properties, including convergence behavior and computational efficiency, are examined and discussed.

Keywords: Conjugate Gradient, sparse systems, linear equations, convergence, numerical methods.

Résumé

Ce travail présente une étude de la méthode du gradient conjugué pour la résolution des systèmes linéaires creux de grande taille. Cette méthode repose sur la construction d'une suite de directions conjuguées ainsi que sur la détermination de pas optimaux, afin de minimiser efficacement la fonction quadratique associée. Les principales propriétés numériques, notamment la convergence et l'efficacité computationnelle, sont analysées et discutées.

Mots-clés : Gradient conjugué, systèmes creux, équations linéaires, convergence, méthodes numériques.

Contents

Introduction	7
1 Preliminaries	9
1.1 Scalar product	9
1.1.1 Vector norms	10
1.1.2 Orthogonality in space	11
1.2 Positive definite and semi-definite quadratic forms	12
1.3 Definite and semi-definite positive matrices	13
1.4 Convex functions	13
1.4.1 Properties of convex functions	13
1.5 Gradients and Hessians	14
1.6 The Hessian in Unconstrained Optimization	14
1.6.1 Critical Points	16
2 Optimization Problems	17
2.1 Linear Optimization	17
2.1.1 General Formulation	17
2.1.2 Standard Form	18
2.1.3 Geometric Interpretation	18
2.1.4 Simplex Method	19
2.2 Non-Linear Optimization	20
2.2.1 General Formulation	20
2.2.2 Constrained Optimization	20
2.2.3 Example	20
2.2.4 Unconstrained minimization problems and their algorithms	21
2.3 Descent Direction Methods	21
2.4 General Descent Methods	21
2.4.1 Principle of the Method	21
2.4.2 Convergence Tests	22
2.4.3 Convergence of a Descent Algorithm	22
2.4.4 Algorithm for the Gradient Descent Method	22

3	Numerical methods for solving unconstrained optimisation problems	23
3.1	Quadratic Minimization Equivalence	23
3.1.1	A-Conjugacy and Orthogonality	23
3.2	Gradient Method	24
3.2.1	The general principle of the method	24
3.2.2	Algorithm of gradient method	25
3.3	Conjugate Gradient method	25
3.3.1	Description of the Method	26
3.3.2	Computation of the Step Size α_k	26
3.3.3	Computation of β_k	27
3.3.4	Algorithm of conjugate gradient method	27
3.4	General Convergence Results for Nonlinear Conjugate Gradient Methods	27
3.4.1	The Fletcher-Reeves method	28
3.4.2	Polak-Ribière-Polyak Method	29
3.4.3	Conjugate Descent Method	31
3.4.4	Dai-Yuan Method	33
3.5	Comparison Between Nonlinear Conjugate Gradient Methods	34
3.5.1	Fletcher-Reeves Method (FR)	34
3.5.2	Polak-Ribière-Polyak Method (PRP)	34
3.5.3	Dai-Yuan Method (DY)	34
3.5.4	Conjugate Descent Method (CD)	35
3.5.5	General Comparison	35
4	Hybrid Conjugate Gradient Methods for Unconstrained Optimization	36
4.1	the fundamental principle of conjugate gradient hybrid methods	36
4.2	Objectives of Hybrid Methods	36
4.3	Linear and Convex Combinations	37
4.4	Hybrid Conjugate Gradient Methods Based on Convex Combinations	37
4.4.1	a hybrid of the PRP and HZ methods	37
4.4.2	Sufficient descent condition	39
4.4.3	Algorithm of hybrid PRP-HZ Conjugate Gradient Method (hPRPHZ)	41
4.4.4	Global Convergence	43
4.5	Numerical tests	45
4.5.1	numerical results	45
4.5.2	Performance profile	48
	Conclusion	51

List of Tables

3.1	A quick comparison of these methods	35
4.1	Numerical Results	45

List of Figures

2.1	Types of convex polyhedron	18
2.2	The graph	19
4.1	Figure 1	49
4.2	Figure 2	50

Introduction

Let us consider the nonlinear unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function and its gradient $g(x) = \nabla f(x)$ is available.

The nonlinear conjugate gradient (CG) method is highly useful for solving this kind of problem because of its simplicity and very low memory requirement[9]. The iterative formula of CG methods is given by:

$$x_{k+1} = x_k + s_k, \quad s_k = \alpha_k d_k, \quad k = 0, 1, \dots \quad (2)$$

where x_k is the k -th iterate, α_k is the step length obtained by performing a line search, and d_k is the search direction.

In practical computation, exact line search is time-consuming and computationally expensive. Therefore, inexact line search methods are commonly used [21]. A widely used inexact line search is the strong Wolfe line search, which determines the step length α_k in (2) such that:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k, \quad (3)$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq \sigma |g_k^T d_k|, \quad (4)$$

where the parameters δ and σ satisfy $0 < \delta < \sigma < 1$. The search direction d_k is generated by the rule:

$$d_k = \begin{cases} -g_k, & k = 0, \\ -g_k + \beta_{k-1} d_{k-1}, & k \geq 1, \end{cases} \quad (5)$$

where $g_k := \nabla f(x_k)$ is the gradient of f at x_k , and $\beta_k \in \mathbb{R}$ is a parameter that determines the different conjugate gradient methods.

One of the efficient methods, which possesses an approximate restart feature when jamming occurs, was proposed by Polak, Ribière, and Polyak [18] (PRP), with the following parameter:

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad (6)$$

where $\|\cdot\|$ denotes the Euclidean norm and $y_k = g_{k+1} - g_k$.

In spite of the numerical efficiency of the PRP method, Powell [20] constructed a counterexample showing that the method may cycle infinitely.

One of the conjugate gradient methods that is strong in theory was suggested by Hager and Zhang [15], with the following formula:

$$\beta_k^{HZ} = \frac{1}{d_k^T y_k} \left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right)^T g_{k+1}. \quad (7)$$

To achieve a method with good practical performance and strong convergence properties, we propose a hybridization of the PRP and HZ methods using a convex combination, in order to exploit the advantages of each. Under the strong Wolfe line search with parameter $\sigma \leq \frac{1}{2}$, Al-Baali [5] proved that the FR method satisfies the sufficient descent condition and is globally convergent for general objective functions. Dai and Yuan [12] showed that the DY method is descent and globally convergent under the Wolfe line search.

In contrast, the PRP and HS (Hestenes–Stiefel) methods are generally considered among the most efficient in practice, but their convergence properties are not always guaranteed.

Recently, Andrei[4] introduced a new hybrid conjugate gradient method (denoted as HYBRID), based on the HS and DY methods, for large-scale unconstrained optimization problems. In [17], Liu et al. discussed the global convergence of the LS (Liu–Storey) and DY methods with inexact line search for nonconvex problems. Djordjević [13] analyzed the global convergence of a convex combination of the FR (Fletcher–Reeves) and PRP methods with sufficient descent property.

The thesis is organized as follows. In Section 2, we obtain the parameter θ_k , discuss the sufficient descent property, and present the proposed algorithm. In Section 3, we establish the global convergence of the method. Numerical results are presented in Section 4. Finally, conclusions are given in the last section.

Chapter 1

Preliminaries

The purpose of this chapter is to review some basic concepts that will be used later on.

1.1 Scalar product

Definition 1. [1] Let E be an $\mathbb{R}$ -vector space. A scalar product on E is any application φ from $E \times E$ into $\mathbb{R}$ such that:

- φ is bilinear: for all $(x, y, z) \in E$ and all $\lambda \in \mathbb{R}$:

$$\varphi(\lambda x + y, z) = \lambda \varphi(x, z) + \varphi(y, z)$$

and

$$\varphi(z, \lambda x + y) = \lambda \varphi(z, x) + \varphi(z, y)$$

- Is symmetric: $\varphi(x, y) = \varphi(y, x)$ for all $x, y \in E$.
- Is definite: for all $x \in E$, if $\varphi(x, x) = 0$ then $x = 0$.
- Is positive: $\varphi(x, x) \geq 0$ for all $x \in E$.

Example 1. The canonical scalar product φ on $E = \mathbb{R}^n$ is defined by :

$$\varphi(x, y) = \sum_{i=1}^n x_i y_i, \quad \forall x, y \in \mathbb{R}^n$$

We also have a scalar product on $M_{n,1}(\mathbb{R})$ defined by :

$$\varphi(X, Y) = X^T Y = \sum_{i=1}^n X_i Y_i, \quad \forall X, Y \in M_{n,1}(\mathbb{R})$$

Notation 1. The scalar product of two vectors x and y will most often be denoted by $\langle x, y \rangle$. where :

$$\langle x, y \rangle = \varphi(x, y) \tag{1.1}$$

1.1.1 Vector norms

Definition 2. [2] A mapping $\|\cdot\| : E \rightarrow \mathbb{R}^+$ is said to be a norm on E if, for all $x, y \in E$ and all $\lambda \in \mathbb{R}$, the following conditions hold:

- For all $x \in E$, $\|x\| = 0$ if and only if $x = 0$.
- For all $x \in E$ and all $\lambda \in \mathbb{K}$, $\|\lambda x\| \leq |\lambda| \|x\|$.
- For all $x, y \in E$, $\|x + y\| \leq \|x\| + \|y\|$.

The pair $(E, \|\cdot\|)$ is then called a normed vector space.

Proposition 1. If $(E, \|\cdot\|)$ is a normed vector space, then for all $x, y \in E$, we have:

$$| \|x\| - \|y\| | \leq \|x - y\|$$

if $(E, \|\cdot\|)$ is a normed vector space, the mapping $d : E \times E \rightarrow \mathbb{R}^+$ defined by $d(x, y) = \|x - y\|$ is called the distance associated with the norm $\|\cdot\|$ on E . It satisfies the following properties:

$$d(x, y) = 0 \Leftrightarrow x = y, \quad \text{for all } (x, y) \in E^2$$

$$d(x, y) = d(y, x), \quad \text{for all } (x, y) \in E^2$$

$$d(x, z) \leq d(x, y) + d(y, z), \quad \text{for all } (x, y, z) \in E^3$$

If $\langle \cdot, \cdot \rangle$ is a scalar product on E , then the map $\|\cdot\|$ defined by :

$$\|x\| = \sqrt{\langle x, x \rangle}$$

is a norm on E .

Example 2. Examples of norms

we have $x \in X$ on $\mathbb{K}^n$ and $x = (x_1, x_2, \dots, x_n)$

$$\|x\|_1 = |x_1| + |x_2| + \dots + |x_n|$$

$$\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \dots + |x_n|^2}$$

$$\|x\|_\infty = \max(|x_1|, |x_2|, \dots, |x_n|)$$

then $\|x\|_1, \|x\|_2, \|x\|_\infty$ are three norms on $\mathbb{K}^n$.

On the space of bounded functions with values in $\mathbb{K}$, let X be a set and E the set of bounded functions from X to $\mathbb{K}$, for $f \in E$,

we set :

$$\|f\|_\infty = \sup_{x \in X} |f(x)|$$

then $\|\cdot\|_\infty$ is norm on E .

On the space of continuous functions with values in $\mathbb{k}$, let $I = [a, b]$ be a segment and $E = C(I, \mathbb{k})$ for $f \in E$, we set:

$$\|f\|_1 = \int_a^b |f(t)| dt$$

$$\|f\|_2 = \left(\int_a^b |f(t)|^2 dt \right)^{\frac{1}{2}}$$

then $\|\cdot\|_1, \|\cdot\|_2$ are two norms on E .

1.1.2 Orthogonality in space

Definition 3. A Hilbert space is a real (or complex) inner product space $(H, \langle \cdot, \cdot \rangle)$ which is complete with respect to the norm induced by the inner product, i.e.,

Two elements $f, g \in H$ are orthogonal, then we obviously have

$$\|f + g\|^2 = \|f\|^2 + \|g\|^2,$$

this formula is often known as the Pythagorean theorem.

Notation 2. If two elements $f, g \in H$ satisfy

$$\langle f, g \rangle = 0,$$

then they are said to be orthogonal, and we write

$$f \perp g.$$

Theorem 1. An element $f \in H$ is said to be orthogonal to the subset A of H (in symbols $f \perp A$), if $f \perp g$ for all $g \in A$. Two subsets A and B of H are said to be orthogonal (in symbols $A \perp B$) if $\langle f, g \rangle = 0$ for all $f \in A, g \in B$. If A is a subset of H , then the set $A^\perp = \{f \in H : f \perp A\}$ is called the orthogonal complement of A .

Proof. For proof, the reader may consult [30]. □

Proposition 2. (a) We have $\{0\}^\perp = H$ and $H^\perp = \{0\}$, i.e., 0 is the only element orthogonal to every element.

(b) For every subset A of H , the set $A^\perp$ is a closed subspace of H .

(c) If $A \subset B$, then $B^\perp \subset A^\perp$.

(d) We have

$$A^\perp = L(A)^\perp = \overline{L(A)}^\perp,$$

where $L(A)$ denotes the vector subspace generated by A , and $\overline{L(A)}$ denotes the closure of $L(A)$.

Proof. (a) For every $f \in H$ we have $\langle 0, f \rangle = 0$. If $f \neq 0$, then $\langle f, f \rangle \neq 0$, i.e., f is not orthogonal to H .

(b) If $f, g \in A^\perp$ and $a, b \in \mathbb{K}$, then for all $h \in A$ it follows that :

$$\langle af + bg, h \rangle = a^* \langle f, h \rangle + b^* \langle g, h \rangle = 0,$$

i.e., $af + bg \in A^\perp$. Therefore $A^\perp$ is a subspace. It remains to prove that $\overline{A^\perp} \subset A^\perp$. Let $f \in \overline{A^\perp}$, and let (f_n) be a sequence from $A^\perp$ such that

$f_n \rightarrow f$. Then we have for all $h \in A$ that

$$\langle f, h \rangle = \lim \langle f_n, h \rangle = 0,$$

consequently $f \in A^\perp$.

(c) If $f \in B^\perp$, then we have $\langle f, h \rangle = 0$ for all $h \in B$, therefore also for all $h \in A$, and thus $f \in A^\perp$.

(d) Since $A \subset L(A) \subset \overline{L(A)}$, from (c) it follows that :

$$\overline{L(A)}^\perp \subset L(A)^\perp \subset A^\perp.$$

It remains to prove that $A^\perp \subset \overline{L(A)}^\perp$. If $f \in A^\perp$, then we have $\langle f, h \rangle = 0$ for all $h \in A$, and therefore for all $h \in L(A)$, as well. If $h \in \overline{L(A)}$, then there exists a sequence (h_n) from $L(A)$ such that $h_n \rightarrow h$. Consequently,

we have

$$\langle f, h \rangle = \lim \langle f, h_n \rangle = 0$$

i.e., $f \in \overline{L(A)}^\perp$

□

1.2 Positive definite and semi-definite quadratic forms

Let $F(x) = x^T A x$ be a quadratic form with A symmetric.

Definition 4. The function F is said to be **positive definite** if

$$x^T A x > 0, \quad \forall x \in \mathbb{R}^n, x \neq 0.$$

It is said to be **positive semidefinite** (or **non-negative definite**) if

$$x^T A x \geq 0, \quad \forall x \in \mathbb{R}^n.$$

Definition 5. A quadratic form $F(x) = x^T A x$ is said to be **negative definite** if

$$x^T A x < 0, \quad \forall x \in \mathbb{R}^n, x \neq 0.$$

It is said to be **negative semidefinite** (or **non-positive definite**) if

$$x^T A x \leq 0, \quad \forall x \in \mathbb{R}^n.$$

Definition 6. A symmetric matrix A is said to be **positive definite** (respectively **non-negative definite**), denoted by $A > 0$ (respectively $A \geq 0$), if it is associated with a positive definite (respectively non-negative definite) quadratic form.

1.3 Definite and semi-definite positive matrices

Theorem 2 (Sylvester's Criterion for Positive Definiteness). As discussed in [32] a symmetric matrix A is said to be **positive definite** ($A > 0$) if and only if all its leading principal minors are positive:

$$A_1 > 0, \quad A_2 > 0, \quad \dots, \quad A_n > 0.$$

Definition 7 (Positive Semidefinite Matrix). A symmetric matrix A is said to be **positive semidefinite** ($A \geq 0$) if and only if all its principal minors are non-negative:

$$\det(A_{i_1, \dots, i_p}) \geq 0, \quad 1 \leq i_1 < \dots < i_p \leq n, \quad p = 1, 2, \dots, n.$$

Definition 8 (Negative Semidefinite Matrix). A symmetric matrix A is said to be **negative semidefinite** ($A \leq 0$) if and only if

$$(-1)^p \det(A_{i_1, \dots, i_p}) \geq 0, \quad 1 \leq i_1 < \dots < i_p \leq n.$$

Remark 1. Sylvester's criterion is valid only for symmetric matrices.

1.4 Convex functions

As discussed in [4] the case where the objective function is convex is very important because quadratic problems form the basis of many non-linear algorithms. We recall some definitions and properties.

Definition 9. (Convex set)

A set $S \subset \mathbb{R}^n$ is said to be convex if :

$$\forall (x, y) \in S \times S, \forall \lambda \in [0, 1] : (\lambda x + (1 - \lambda)y) \in S$$

Definition 10. (Convex function)

Let $S \subset \mathbb{R}^n$ be a non empty convex set. Function $f : S \rightarrow \mathbb{R}$ is said to be convex on S if for any $x_1, x_2 \in S$ and all $0 \leq \alpha \leq 1$, we have

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2)$$

If

$$f(\alpha x_1 + (1 - \alpha)x_2) < \alpha f(x_1) + (1 - \alpha)f(x_2)$$

for all $x_1 \neq x_2$, then f is called a strictly convex function on S . If there is a constant $c > 0$ such that for any $x_1, x_2 \in S$

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2) - \frac{1}{2}c\alpha(1 - \alpha)\|x_1 - x_2\|^2$$

then f is called a uniformly convex (or strongly convex) function on S .

1.4.1 Properties of convex functions

1-Continuity of convex functions

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function bounded above by a constant, and if in the neighborhood of a point $x_0 \in H$, f is continuous at x_0 .

2-Differentiability of convex functions

We next establish several key properties of differentiable convex functions.

Theorem 3. Let $S \subset \mathbb{R}^n$ be an open, non-empty, convex subset, let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be differentiable in S , then

1) f is convex in S if and only if for all $\hat{x} \in S$:

$$f(x) \geq f(\hat{x}) + \nabla f(\hat{x})^T(x - \hat{x}), \quad \forall x \in S$$

2) f is strictly convex in S if and only if for all $\hat{x} \in S$:

$$f(x) > f(\hat{x}) + \nabla f(\hat{x})^T(x - \hat{x}), \quad \forall x \in S$$

1.5 Gradients and Hessians

Gradients and Hessians are essential tools in multivariable calculus. They provide important information about the behavior of a function, such as its rate of change and curvature. We begin by defining the gradient.

Definition 11. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable function. The **gradient** of f at a point $x \in \mathbb{R}^n$ is defined by

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}. \quad (1.2)$$

Example 3. Consider the function: $f(x, y) = x^3 + 2xy - y^6$

$$\begin{aligned} \frac{\partial f}{\partial x} &= 3x^2 + 2y \\ \frac{\partial f}{\partial y} &= 2x - 6y^5 \end{aligned} \Rightarrow \nabla f(x, y) = \begin{pmatrix} 3x^2 + 2y \\ -6y^5 + 2x \end{pmatrix} \quad (1.3)$$

1.6 The Hessian in Unconstrained Optimization

Definition 12. Let f be a real-valued function of class C^2 , $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The Hessian of the function f is defined by :

$$Hf(x) = \frac{\partial^2 f}{\partial x_i^2} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix} \quad (1.4)$$

Notation 3. • If f is of class C^2 , then for all $i, j \in \{1, \dots, n\}$, we have

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}.$$

- The Hessian matrix of f is symmetric.

Theorem 4. Let $f : S \rightarrow \mathbb{R}$ be twice continuously differentiable on a convex set S . If the Hessian matrix $Hf(x)$ is positive definite for all $x \in S$, then f is strictly convex on S .

Proof. Let $x, y \in S$ with $x \neq y$, and define the function

$$\varphi(t) = f(x + t(y - x)), \quad t \in [0, 1].$$

Since S is convex, we have $x + t(y - x) \in S$ for all $t \in [0, 1]$, so φ is well-defined and twice differentiable.

Compute the first derivative:

$$\varphi'(t) = \nabla f(x + t(y - x)) \cdot (y - x).$$

Compute the second derivative:

$$\varphi''(t) = (y - x)^T H(x + t(y - x))(y - x).$$

Since $H(x)$ is positive definite for all $x \in S$ and $y - x \neq 0$, it follows that

$$\varphi''(t) > 0 \quad \text{for all } t \in [0, 1].$$

Thus, φ is strictly convex on $[0, 1]$. Hence, for all $t \in (0, 1)$,

$$\varphi(t) < (1 - t)\varphi(0) + t\varphi(1).$$

Substituting back,

$$f(x + t(y - x)) < (1 - t)f(x) + tf(y).$$

Therefore, f is strictly convex on S . □

Example 4. A function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, defined by :

$$f(x_1, x_2, x_3) = 2x_1^2x_2 + 2x_2^2x_3 + 5x_1^2x_2x_3 - 3x_1 - 2x_2 - x_3$$

We have :

$$\begin{aligned} \nabla f(x_1, x_2, x_3) &= \begin{pmatrix} 4x_1x_2 + 10x_1x_2x_3 - 3 \\ 2x_1^2 + 4x_2x_3 + 5x_1^2x_3 - 2 \\ 2x_2^2 + 5x_1^2x_2 - 1 \end{pmatrix} \\ Hf(x_1, x_2, x_3) &= \begin{pmatrix} 4x_2 + 10x_2x_3 & 4x_1 + 10x_1x_3 & 10x_1x_2 \\ 4x_1 + 10x_1x_3 & 4x_3 & 4x_2 + 5x_1^2 \\ 10x_1x_2 & 4x_2 + 5x_1^2 & 0 \end{pmatrix} \end{aligned}$$

1.6.1 Critical Points

Stationary points

Definition 13. Let f be a differentiable function. A point x_0 is called a stationary point if

$$f'(x_0) = 0.$$

At such a point, the tangent line to the graph of the function is horizontal.

A stationary point can be:

- A local minimum, if the function attains its smallest value within a certain neighborhood at that point.
- A local maximum, if the function attains its largest value within a neighborhood.
- An inflection point, if the direction of change of the function does not change, but the concavity of the graph changes around this point.

Example 5. Let the function : $f(x) = x^3 - 3x^2 + 2$

First, let's calculate the derivative : $f'(x) = 3x^2 - 6x$

Next, we solve the equation $f'(x) = 0 \Rightarrow 3x^2 - 6x = 0$

$$\Rightarrow x = 0 \text{ ou } x = 2$$

These are the stationary points.

Next, we check the sign of the derivative in the intervals defined by these points: :

For $x < 0$: we put $x = -1 \Rightarrow f'(-1) = 9 > 0 \rightarrow$ the function is increasing.

For $0 < x < 2$: we put $x = 1 \Rightarrow f'(1) = -3 < 0 \rightarrow$ the function is decreasing.

For $x > 2$: we put $x = 3 \Rightarrow f'(3) = 9 > 0 \rightarrow$ the function is increasing again.

From this, we conclude that $x = 0$ there is a local maximum, and at $x = 2$ there is a local minimum.

Chapter 2

Optimization Problems

Optimization problems arise in many areas of mathematics, engineering, economics, and computer science. The main objective of optimization is to determine the best possible solution among a set of feasible solutions while satisfying a number of constraints.

In general, an optimization problem consists of minimizing or maximizing an objective function subject to equality and/or inequality constraints. Consider the problem can be written as:

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

where:

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable
- $x \in \mathbb{R}^n$ is the vector of decision variables.

Optimization problems are generally classified into several categories depending on the nature of the objective function and the constraints. The most common classes are linear optimization and nonlinear optimization problems.

2.1 Linear Optimization

Definition 14. A linear optimization (or linear programming) problem is the problem of minimizing a linear objective function subject to linear equality and inequality constraints.

2.1.1 General Formulation

A general linear programming problem is written as:

$$\min c^T x$$

subject to:

$$Ax \leq b,$$

where:

- $x \in \mathbb{R}^n$ is the vector of decision variables,
- $c \in \mathbb{R}^n$ is the constant vector,
- $A \in M_{n,1}(\mathbb{R})$ is the constraint matrix,
- $b \in \mathbb{R}^m$ is the constraint vector.

2.1.2 Standard Form

A linear programming problem is said to be in standard form if it is expressed as:

$$\min c^T x \quad \text{subject to} \quad Ax = b, x \geq 0.$$

Any linear program can be transformed into this form by introducing slack or surplus variables.

2.1.3 Geometric Interpretation

The feasible region of a linear programming problem is a convex polyhedron. A fundamental result in linear optimization states that, whenever an optimal solution exists, it is attained at an extreme point (vertex) of the feasible set.

This property is central to the development of efficient solution algorithms.

Definition 15. A polyhedron P is said to be convex if, for every pair of points $x, y \in P$, the line segment joining them is entirely contained in P , that is,

$$\{\lambda x + (1 - \lambda)y : 0 \leq \lambda \leq 1\} \subseteq P.$$

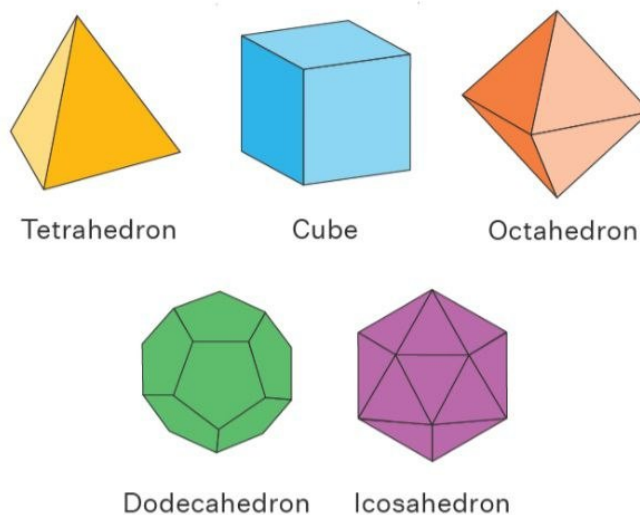


Figure 2.1: Types of convex polyhedron

2.1.4 Simplex Method

The simplex method is an iterative algorithm for solving linear programming problems. It proceeds by moving from one vertex of the feasible polyhedron to an adjacent one, ensuring a monotonic improvement of the objective function until optimality is reached.

Example

Consider the linear programming problem:

$$\min z = 3x_1 + 2x_2$$

subject to:

$$x_1 + x_2 \leq 4, \quad 2x_1 + x_2 \leq 5, \quad x_1, x_2 \geq 0.$$

The feasible set is a convex polygon whose vertices are:

$$(0, 0), (0, 4), (1, 3), (2.5, 0).$$

Evaluating the objective function at these points yields:

$$z(0, 0) = 0, \quad z(0, 4) = 8, \quad z(1, 3) = 9, \quad z(2.5, 0) = 7.5.$$

Hence, an optimal solution is given by:

$$x^* = (0, 0), \quad z_{\min} = 0.$$

We can explain this using the following graph :

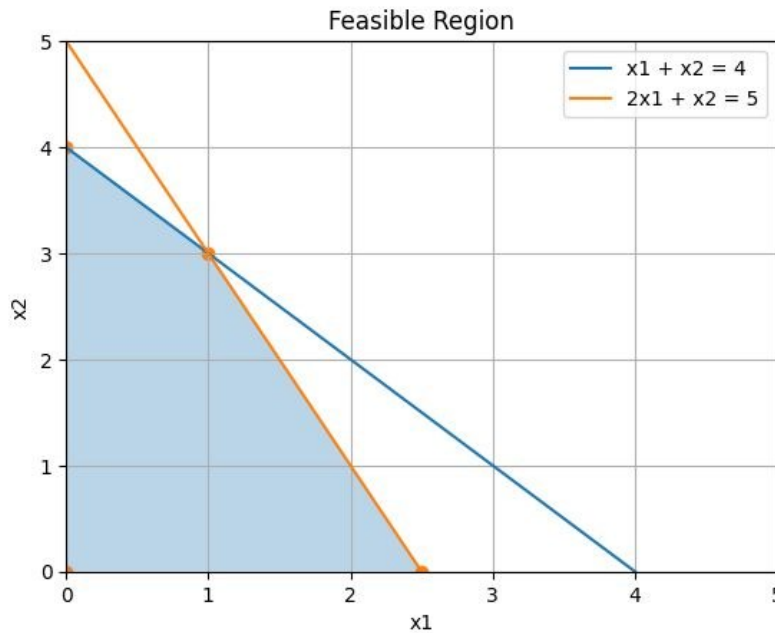


Figure 2.2: The graph

2.2 Non-Linear Optimization

Definition 16. A nonlinear optimization problem is an optimization problem in which either the objective function or at least one constraint is nonlinear.

2.2.1 General Formulation

A general nonlinear optimization problem is formulated as:

$$\min f(x)$$

subject to:

$$g_i(x) \leq 0, \quad i = 1, \dots, m,$$

$$h_j(x) = 0, \quad j = 1, \dots, p,$$

where f , g_i , and h_j are generally nonlinear functions.

2.2.2 Constrained Optimization

Constrained optimization problems involve additional conditions imposed on the admissible solutions.

A classical approach for handling equality constraints is the method of Lagrange multipliers. The associated Lagrangian function is defined by:

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x).$$

2.2.3 Example

Consider the unconstrained optimization problem:

$$2x + 4 = 0 \quad \text{which implies} \quad x^* = -2.$$

We compute the first derivative:

$$f'(x) = 2x + 4.$$

The stationary point satisfies:

$$2x + 4 = 0 \quad \Rightarrow \quad x^* = -2.$$

The second derivative is:

$$f''(x) = 2 > 0,$$

Hence x^* is a strict local (and global) minimizer.

The optimal value is:

$$f(-2) = 0.$$

2.2.4 Unconstrained minimization problems and their algorithms

Optimality conditions for unconstrained non-linear optimization problems

Theorem 5. (Necessary conditions) Let x^* be a local minimum of f on $\mathbb{R}^n$, such that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable, then we have:

1. $\nabla f(x^*) = 0$. (First-order optimality condition)
2. $\nabla^2 f(x^*)$ is a positive semidefinite matrix (second-order necessary condition).

Remark 2. A local minimum of a differentiable function is a stationary point. The converse is false.

Theorem 6. (Sufficient conditions) Let $f \in C^2(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ such that:

1. $\nabla f(x^*) = 0$.
 2. $\nabla^2 f(x^*)$ is positive definite.
- Then x^* is a local minimum of f .

We can summarize the situation cited before as :

Theorem 7. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex and differentiable function. Then x^* is a global minimum of f if and only if:

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

2.3 Descent Direction Methods

Definition 17. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $x \in \mathbb{R}^n$. A vector $d \in \mathbb{R}^n$ is a descent direction for f at x if:

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

Definition 18. Any vector $d_k \in \mathbb{R}^n$ that satisfies:

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

is called a descent direction.

Remark 3. If it is not possible to find a vector d_k such that $d_k^T \nabla f(x_k) < 0$, then $d_k^T \nabla f(x_k) = 0$, and in that case x_k will generally be the desired solution.

2.4 General Descent Methods

2.4.1 Principle of the Method

The transition from x_k to x_{k+1} is carried out in two steps:

1. At the point x_k , a search direction d_k is chosen.
2. A step size $\alpha_k > 0$ is determined such that:

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) < f(x_k), \quad \forall k \in \mathbb{N}.$$

Thus:

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

2.4.2 Convergence Tests

A stopping criterion is:

$$\|\nabla f(x_k)\| < \varepsilon.$$

2.4.3 Convergence of a Descent Algorithm

An algorithm is globally convergent if the sequence (x_k) converges to a point satisfying a necessary optimality condition, independently of x_0 .

2.4.4 Algorithm for the Gradient Descent Method

Algorithm 1. Step 0: Initialization $x_0 \in \mathbb{R}^n$

Step 1: If $\|\nabla f(x_k)\| \approx 0$, STOP

Step 2: $d_k = -\nabla f(x_k)$

Step 3: Choose $\alpha_k > 0$

Step 4: $x_{k+1} = x_k + \alpha_k d_k$

Step 5: $k \leftarrow k + 1$

Remark 4. $f(x_{k+1}) \leq f(x_k)$ is not sufficient for convergence.

- If f is bounded below, then $f(x_k)$ converges, but not necessarily to $f(x^*)$.

Chapter 3

Numerical methods for solving unconstrained optimisation problems

This chapter introduces an important class of algorithms for solving unconstrained optimisation problems. The central concept is that of the descent direction. After describing how a gradient descent algorithm works, we provide some examples of such algorithms and set out criteria for estimating the quality of the gradient descent near a solution: the admissibility criterion and the unit step.

3.1 Quadratic Minimization Equivalence

The system $Ax = b$ is equivalent to the unconstrained quadratic optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x^T Ax - b^T x. \quad (3.1)$$

The gradient of f is:

$$\nabla f(x) = Ax - b. \quad (3.2)$$

At the minimizer x^* , we have:

$$\nabla f(x^*) = 0 \Rightarrow Ax^* = b. \quad (3.3)$$

The Hessian matrix is:

$$Hf(x) = A, \quad (3.4)$$

which is positive definite, ensuring that f is strictly convex and has a unique global minimum.

3.1.1 A-Conjugacy and Orthogonality

Definition (A-conjugacy). Two vectors $u, v \in \mathbb{R}^n$ are said to be A -conjugate if

$$\langle u, v \rangle_A = u^T Av = 0. \quad (3.5)$$

This defines an inner product induced by A :

$$\langle u, v \rangle_A = u^T A v. \quad (3.6)$$

The A -conjugacy relation is symmetric: if u is A -conjugate to v , then v is A -conjugate to u .

3.2 Gradient Method

The idea of this method, which dates back to Cauchy (1847), is very natural, but not always efficient. It consists in observing that, at any point in $\mathbb{R}^n$, the direction opposite to the gradient corresponds locally to the direction of the steepest decrease of the function f .

3.2.1 The general principle of the method

Let us apply the previous algorithm to the following quadratic function:

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

where A is a real symmetric positive definite matrix of size $n \times n$, and $b \in \mathbb{R}^n$ is a constant vector.

The classical conjugate gradient method was originally developed for quadratic optimization problems. Later, several nonlinear extensions were proposed in order to solve general unconstrained optimization problems.

Let

$$g_k = \nabla f(x_k) = A x_k - b,$$

and define

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

Then, for all $\alpha \geq 0$, we have

$$\begin{aligned} \phi(\alpha_k) &= f(x_k + \alpha_k d_k) = f(x_k - \alpha_k g_k) \\ &= \frac{1}{2} (x_k - \alpha_k g_k)^T A (x_k - \alpha_k g_k) - (x_k - \alpha_k g_k)^T b \\ &= f(x_k) - \alpha_k g_k^T g_k + \frac{\alpha_k^2}{2} g_k^T A g_k. \end{aligned}$$

The function ϕ admits a unique minimum for

$$\alpha_k = \frac{g_k^T g_k}{g_k^T A g_k}.$$

Hence,

$$x_{k+1} = x_k - \frac{\|g_k\|^2}{g_k^T A g_k} g_k.$$

Note that

$$g_k^T A g_k > 0,$$

since A is positive definite and $g_k \neq 0$.

3.2.2 Algorithm of gradient method

Initialization:

$$k = 0, \quad d_0 = -g_0.$$

While $\|g_k\| > \varepsilon$ **do**

$$d_k = -g_k,$$

$$\alpha_k = \frac{g_k^T g_k}{g_k^T A g_k},$$

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

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

$$k = k + 1.$$

End While

3.3 Conjugate Gradient method

The conjugate gradient method is a descent method with optimal step size that allows the minimization of a quadratic function from $\mathbb{R}^n$ into $\mathbb{R}$ in at most n iterations.

Consider the following unconstrained quadratic optimization problem:

$$\begin{cases} \min f(x) = \frac{1}{2} x^T A x + b^T x + c, \\ x \in \mathbb{R}^n, \end{cases}$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}$.

Definition 19. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix, and let

$$\{d_1, d_2, \dots, d_k\}$$

be a set of nonzero directions. These directions are said to be A -conjugate if

$$d_i^T A d_j = 0, \quad \forall i \neq j.$$

Note that if A is the identity matrix, then the conjugate directions are orthogonal.

1. If $d_1, d_2, \dots, d_k$ are A -conjugate directions, then they are linearly independent.
2. The vector space generated by a set of n conjugate directions has dimension n .

3.3.1 Description of the Method

The method consists in minimizing f starting from an initial point x_0 , along n directions

$$d_0, d_1, \dots, d_{n-1},$$

which are mutually conjugate with respect to A .

1. The first fundamental idea of the conjugate gradient algorithm consists in choosing each descent direction to be conjugate to the previous descent direction with respect to A .
2. The second fundamental idea consists in seeking d_k in the form of a linear combination of d_{k-1} and the gradient at the point x_k , namely:

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

where β_k is chosen such that the successive directions are conjugate with respect to Q .

3. The sequence is then constructed as follows:

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

with

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

3.3.2 Computation of the Step Size α_k

Since α_k minimizes f along the direction d_k , we have

$$\forall k, \quad d_k^T \nabla f(x_{k+1}) = 0.$$

Thus,

$$d_k^T (A(x_k + \alpha_k d_k) + b) = 0,$$

which gives

$$\alpha_k = -\frac{d_k^T (Ax_k + b)}{d_k^T A d_k} = -\frac{d_k^T \nabla f(x_k)}{d_k^T A d_k}.$$

Since A is positive definite and the directions d_k are mutually conjugate, we have

$$\forall k, \quad d_k^T A d_k \neq 0.$$

3.3.3 Computation of β_k

For

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

the conjugacy condition requires that

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

Hence,

$$(-g_{k+1} + \beta_k d_k)^T A d_k = 0,$$

which leads to

$$-g_{k+1}^T A d_k + \beta_k d_k^T A d_k = 0.$$

Since $d_k \neq 0$, this equation always admits a solution, and we obtain

$$\beta_k = -\frac{g_{k+1}^T A d_k}{d_k^T A d_k}.$$

3.3.4 Algorithm of conjugate gradient method

Initialization:

Choose x_0 and set $k = 0$, $d_0 = -\nabla f(x_0)$.

While $\|\nabla f(x_k)\| > \varepsilon$ **do**

$$\alpha_k = -\frac{d_k^T \nabla f(x_k)}{d_k^T A d_k},$$

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

$$\beta_k = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2},$$

$$d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k,$$

$$k = k + 1.$$

End While

3.4 General Convergence Results for Nonlinear Conjugate Gradient Methods

Our problem is to minimize a function f of n real variables:

$$\min\{f(x) : x \in \mathbb{R}^n\} \tag{2.6}$$

where f is continuously differentiable and g denotes its gradient. Let g_k be the gradient of f at the point x_k .

We recall that different conjugate gradient methods generate sequences $\{x_k\}$ of the form:

$$x_{k+1} = x_k + \alpha_k d_k \quad (2.7)$$

where the search direction is defined by the recurrence relation:

$$d_k = \begin{cases} -g_0, & k = 0, \\ -g_k + \beta_k d_{k-1}, & k \geq 1. \end{cases} \quad (2.8)$$

The coefficient β_k determines the specific conjugate gradient method (Fletcher–Reeves, Polak–Ribière–Polyak, etc.). The step size $\alpha_k \in \mathbb{R}^+$ is determined by a line search procedure.

Notation (4). The set

$$L = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_1)\}$$

is bounded, where $x_0 \in \mathbb{R}^n$ is the initial point.

- (ii) On a neighborhood V of L , the objective function f is continuously differentiable and its gradient is Lipschitz continuous, i.e.,

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in V.$$

3.4.1 The Fletcher-Reeves method

The Fletcher-Reeves method is a direct extension of the linear conjugate gradient method to the case of arbitrary functions. When applied to a quadratic function, it is identical to the linear conjugate gradient method. For the Fletcher-Reeves method, the β_k variant is :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \quad (3.7)$$

Algorithme de la méthode de Fletcher-Reeves [25] This method is very interesting, on the

one hand because it requires storing very little information (essentially three vectors of dimension n); on the other hand, due to its convergence speed, which is significantly faster than that of classical gradient algorithms.

Algorithm 2. Step 0: Initialization

Given an initial point x_0 , compute

$$g_0 = \nabla f(x_0)$$

Set

$$d_0 = -g_0$$

Set $k = 0$ and go to Step 1.

Step 1: Stopping test

If

$$\|g_k\| = 0$$

then STOP ($x^* = x_k$).

Otherwise, go to Step 2.

Step 2: Iteration

Compute

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

where

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

Update the direction:

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

where

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

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

Convergence of the Fletcher–Reeves Method The first convergence result for the nonlinear conjugate gradient method (Fletcher–Reeves version) with inexact line search (strong Wolfe conditions with $\sigma < \frac{1}{2}$) was established by Al-Baali.

Touati Ahmed and Story generalized this result to the case $0 \leq \beta_k \leq \beta_k^{FR}$.

Gilbert and Nocedal further extended this result to the case $|\beta_k| \leq \beta_k^{FR}$.

3.4.2 Polak–Ribière–Polyak Method

This method was introduced by Polak, Ribière, and Polyak.

For the Polak–Ribière–Polyak method, the parameter β_k is defined as:

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2} \quad (3.8)$$

Algorithme de la méthode de Polak-Ribière-Polyak

Algorithm 3. Polak–Ribière–Polyak (PRP) Method

Step 0: Initialization

Let x_0 be the initial point, $g_0 = \nabla f(x_0)$, and set $d_0 = -g_0$.

Set $k = 0$ and go to Step 1.

Step 1: Stopping test

If $g_k = 0$, STOP ($x^* = x_k$).

Otherwise, go to Step 2.

Step 2: Iteration

Compute:

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

where

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

and update the search direction:

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

with

$$\beta_{k+1}^{PRP} = \frac{g_{k+1}^\top (g_{k+1} - g_k)}{\|g_k\|^2} = \frac{g_{k+1}^\top y_k}{\|g_k\|^2}.$$

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

Convergence of the PRP Method The convergence of this method is guaranteed for a strongly convex function with line search. However, if f is not convex, the method may fail to converge.

To overcome this drawback, Powell modified the choice of β_k^{PRP} in order to ensure convergence under exact line search.

Gilbert and Nocedal proposed a new method that converges in the sense:

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

with either exact or inexact line search.

The following result is due to Polak and Ribière.

Theorem 8. If f is strongly convex, continuously differentiable, and its gradient is Lipschitz continuous, then the Polak–Ribière method with exact line search generates a sequence $\{x_k\}$ that converges to the unique point x^* which minimizes f .

Proof. From [26], we first show that:

$$\cos \theta_k = \frac{-d_k^\top g_k}{\|g_k\| \|d_k\|}$$

is uniformly positive.

$$\begin{aligned} d_{k-1}^\top y_{k-1} &= d_{k-1}^\top (g_k - g_{k-1}) \\ &= d_{k-1}^\top g_k - d_{k-1}^\top g_{k-1} \\ &= d_{k-1}^\top g_{k-1} = -(-g_{k-1} + \beta_k^{PRP} d_{k-2})^\top g_{k-1} \\ &= \|g_{k-1}\|^2. \end{aligned}$$

The strong convexity of f implies that

$$d_{k-1}^\top y_{k-1} = \frac{1}{\alpha_{k-1}} (x_k - x_{k-1})^\top y_{k-1} \geq \frac{\eta}{\alpha_{k-1}} \|x_k - x_{k-1}\|^2,$$

where $\eta > 0$ is the strong convexity constant of f .

Using the Lipschitz constant L of g , we obtain

$$|\beta_k^{PRP}| = \frac{|g_k^\top y_{k-1}|}{\|g_{k-1}\|^2} = \frac{|g_k^\top y_{k-1}|}{d_{k-1}^\top y_{k-1}} \leq \frac{\alpha_{k-1}}{\eta} \frac{L \|g_k\| \|x_k - x_{k-1}\|}{\|x_k - x_{k-1}\|^2} = \frac{L}{\eta} \frac{\|g_k\|}{\|d_{k-1}\|}.$$

Hence, we can bound $\|d_k\|$ as follows:

$$\begin{aligned}\|d_k\| &\leq \|g_k\| + |\beta_k^{PRP}| \|d_{k-1}\| \\ &\leq \|g_k\| + \frac{L}{\eta} \frac{\|g_k\|}{\|d_{k-1}\|} \|d_{k-1}\| \\ &\leq \left(1 + \frac{L}{\eta}\right) \|g_k\|.\end{aligned}$$

Moreover,

$$d_k^\top g_k = \|g_k\|^2 \leq -\left(1 + \frac{L}{\eta}\right) \|g_k\| \|d_k\|.$$

Thus,

$$\cos \theta_k = \frac{-d_k^\top g_k}{\|g_k\| \|d_k\|} \geq \left(1 + \frac{L}{\eta}\right)^{-1}.$$

□

From the exact line search, the Zoutendijk condition holds. Since f is strongly convex, f is bounded below and therefore $\{f(x_k)\}$ is bounded from below.

It follows that $g_k \rightarrow 0$. Moreover, $\{x_k\}$ is bounded (since f is strongly convex), hence it admits convergent subsequences.

The limit of any such subsequence must be the unique minimizer x^* of f (since $g_k \rightarrow 0$). Therefore, the whole sequence $\{x_k\}$ converges to x^* .

Remark 5. Gilbert and Nocedal ensured convergence using either exact or inexact line search by hybridizing the FR and PRP methods. Therefore, this new method consists in choosing the parameter β_k as follows:

$$\beta_k = \begin{cases} -\beta_k^{FR} & \text{if } \beta_k^{PRP} < -\beta_k^{FR}, \\ \beta_k^{PRP} & \text{if } |\beta_k^{PRP}| \leq \beta_k^{FR}, \\ \beta_k^{FR} & \text{if } \beta_k^{PRP} > \beta_k^{FR}. \end{cases}$$

3.4.3 Conjugate Descent Method

This method was proposed in 1987 by Fletcher and Reeves.

For the conjugate descent method, the parameter β_k is defined as

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^\top g_{k-1}}. \quad (3.9)$$

Algorithm of the Conjugate Descent Method Step 0: Initialization

Let x_0 be the starting point, $g_0 = \nabla f(x_0)$, and set $d_0 = -g_0$.
Set $k = 0$ and go to Step 1.

Step 1: Stopping Test

If $g_k = 0$, then STOP and $x^* = x_k$.
Otherwise, go to Step 2.

Step 2: Update

Define

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

where

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

Compute

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

where

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

Set $k \leftarrow k + 1$ and return to Step 1.

Convergence of the conjugate gradient method

Yuan ([52, 1993]) demonstrated the convergence of this method in the sense specified in the statement:

$$\liminf_{k \rightarrow \infty} \|\nabla g_k\| = 0$$

A step that meets the following conditions

Wolfe Conditions:

We recall the weak Wolfe conditions:

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

$$\nabla f(x_{k+1})^T d_k \geq \sigma \nabla f(x_k)^T d_k$$

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

Strong Wolfe conditions:

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

$$|\nabla f(x_{k+1})^T d_k| \leq \sigma |\nabla f(x_k)^T d_k|$$

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

Relaxed Wolfe conditions:

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

$$\sigma_1 \nabla f(x_k)^T d_k \leq \nabla f(x_{k+1})^T d_k \leq \sigma_2 \nabla f(x_k)^T d_k$$

where $0 < \rho < \sigma_1 < 1$ and $\sigma_2 > 0$.

If $\sigma_1 < \frac{1}{2}$ and $\sigma_2 = 0$, Dai and Yuan proved this result under the assumptions $\sigma_1 < 1$ and $\sigma_2 = 0$.

3.4.4 Dai–Yuan Method

This method was introduced by Dai and Yuan.

For the conjugate gradient method, the parameter β_k is defined as:

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^\top (g_k - g_{k-1})} = \frac{\|g_k\|^2}{d_{k-1}^\top y_{k-1}}. \quad (3.10)$$

It is worth noting that this variant has the same numerator and denominator as those of the Fletcher–Reeves and Hestenes–Stiefel methods, respectively.

This method possesses several properties; for instance, it satisfies the descent property at each iteration.

Dai–Yuan Method Algorithm [24] Step 0: Initialization

Let x_0 be the starting point, $g_0 = \nabla f(x_0)$, and set $d_0 = -g_0$.
Set $k = 0$ and go to Step 1.

Step 1: Stopping Test

If $g_k = 0$, then STOP and $x^* = x_k$.
Otherwise, go to Step 2.

Step 2: Update

Define

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

where

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

Compute

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

where

$$\beta_{k+1}^{DY} = \frac{\|g_{k+1}\|^2}{d_k^\top (g_{k+1} - g_k)} = \frac{\|g_{k+1}\|^2}{d_k^\top y_k}, \quad \text{with } y_k = g_{k+1} - g_k.$$

Set $k \leftarrow k + 1$ and return to Step 1.

Convergence of the Dai–Yuan Method Dai and Yuan proved the convergence of the Dai–Yuan method in the sense of (3.15) when the step size α_k is determined by the weak Wolfe conditions..

Theorem 9. Suppose that Hypothesis 2.1.1 is satisfied. The sequence $\{x_k\}$ generated by the algorithm converges in the sense that

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

3.5 Comparison Between Nonlinear Conjugate Gradient Methods

Nonlinear conjugate gradient methods are among the most efficient techniques for solving large-scale unconstrained optimization problems. Several variants have been proposed in the literature, each characterized by a different choice of the parameter β_k . In this section, we compare some classical methods, namely the Fletcher–Reeves (FR), Polak–Ribière–Polyak (PRP), Dai–Yuan (DY), and Conjugate Descent (CD) methods.

3.5.1 Fletcher–Reeves Method (FR)

The Fletcher–Reeves method defines the parameter β_k by

$$\beta_k^{FR} = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}.$$

The FR method is one of the earliest nonlinear conjugate gradient methods and possesses strong theoretical convergence properties under standard assumptions and exact line search. It is generally stable and easy to implement. However, in practice, it may suffer from slow convergence and loss of efficiency when the objective function is highly nonlinear.

3.5.2 Polak–Ribière–Polyak Method (PRP)

The PRP method uses

$$\beta_k^{PRP} = \frac{\nabla f(x_k)^T (\nabla f(x_k) - \nabla f(x_{k-1}))}{\|\nabla f(x_{k-1})\|^2}.$$

The PRP method is widely appreciated for its practical efficiency and fast convergence behavior. It often performs better than the FR method in numerical experiments, especially for difficult nonlinear problems. However, the parameter β_k^{PRP} may become negative, which can destroy the descent property. To overcome this issue, modified versions such as PRP^+ are commonly used.

3.5.3 Dai–Yuan Method (DY)

The Dai–Yuan method defines

$$\beta_k^{DY} = \frac{\|\nabla f(x_k)\|^2}{d_{k-1}^T (\nabla f(x_k) - \nabla f(x_{k-1}))}.$$

The DY method has strong global convergence properties under weak assumptions and is known for its robustness. Unlike some other methods, it preserves the descent condition even under inexact line search conditions satisfying the Wolfe criteria. Moreover, the DY method is often more stable than PRP for nonconvex optimization problems.

3.5.4 Conjugate Descent Method (CD)

The Conjugate Descent method uses

$$\beta_k^{CD} = -\frac{\|\nabla f(x_k)\|^2}{d_{k-1}^T \nabla f(x_{k-1})}.$$

The CD method guarantees sufficient descent under appropriate line search conditions and is recognized for its numerical stability. Although it may be less efficient than PRP in some applications, it generally exhibits good robustness and reliable convergence behavior.

3.5.5 General Comparison

The following table summarizes the main characteristics of these methods.

Table 3.1: A quick comparison of these methods

Method	Convergence	Robustness	Efficiency	Descent Property
FR	Strong	Moderate	Moderate	Guaranteed
PRP	Good	Moderate	High	Not always guaranteed
DY	Very strong	High	High	Guaranteed
CD	Strong	High	Moderate	Guaranteed

Regarding line search strategies, the FR method usually requires exact or strong Wolfe line search conditions to ensure convergence. The PRP method is highly sensitive to the choice of line search and may fail without suitable safeguards. In contrast, the DY and CD methods are more compatible with inexact line search techniques and generally preserve good convergence properties under standard Wolfe conditions.

Overall, the PRP method is often preferred for its numerical efficiency, while the DY method is particularly attractive because of its robustness and strong theoretical convergence properties. The FR and CD methods remain important due to their simplicity and stability.

Chapter 4

Hybrid Conjugate Gradient Methods for Unconstrained Optimization

4.1 the fundamental principle of conjugate gradient hybrid methods

Hybrid methods of gradient descent are based on the fundamental idea of combining two or more operations from different methods into a single algorithm, with the aim of capitalising on the advantages of each method whilst avoiding their shortcomings

Problem Setting

We consider the problem of minimizing a nonlinear function:

$$\min_{x \in \mathbb{R}^n} f(x) \quad (4.1)$$

Conjugate gradient methods are based on constructing a sequence of search directions d_k defined by the following recurrence relation:

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad (4.2)$$

where:

- $g_k = \nabla f(x_k)$ is the gradient,
- β_k is a parameter that determines the specific type of the method.

4.2 Objectives of Hybrid Methods

Hybrid conjugate gradient methods are designed to achieve several complementary goals:

- **Enhancing numerical stability**, by incorporating robust schemes such as the HZ method, which improves the reliability of the iterations.
- **Accelerating convergence**, through the use of more efficient updating strategies such as the PRP method, which often exhibits faster practical performance.
- **Preserving the conjugacy property**, by an appropriate adaptive selection of the parameter θ_k , ensuring consistency of the search directions.
- **Adapting to the problem structure**, where θ_k is updated dynamically during the iterative process according to the behavior of the objective function.

4.3 Linear and Convex Combinations

In general, a linear combination of two vectors or scalars x and y is defined as:

$$z = \alpha x + \beta y. \quad (4.3)$$

In the case of a **convex combination**, we impose the additional constraints:

$$\alpha + \beta = 1, \quad \alpha \geq 0, \quad \beta \geq 0. \quad (4.4)$$

Thus, it can be equivalently written as:

$$z = (1 - \lambda)x + \lambda y, \quad \lambda \in [0, 1]. \quad (4.5)$$

This guarantees that the resulting value z lies within the interval defined by x and y .

In order to combine the strengths of two update parameters $\beta_k^{(1)}$ and $\beta_k^{(2)}$, a hybrid coefficient β_k^{hyb} is defined as a convex combination of these parameters, where the superscript *hyb* denotes the corresponding hybrid scheme.

$$\beta_k^{hyb} = (1 - \theta_k)\beta_k^{(1)} + \theta_k\beta_k^{(2)}, \quad \theta_k \in [0, 1]. \quad (4.6)$$

where:

- $\beta_k^{(1)}$ and $\beta_k^{(2)}$ are two parameters corresponding to different methods,
- θ_k is a hybrid parameter that controls the balance between them.

4.4 Hybrid Conjugate Gradient Methods Based on Convex Combinations

4.4.1 a hybrid of the PRP and HZ methods

In order to combine the strengths of the PRP and HZ methods, the hybrid coefficient β_k^{hPRPHZ} is defined as a convex combination of the two update parameters, where *hPRPHZ* denotes the hybrid PRP–HZ scheme. The hybrid coefficient is defined by:

$$\beta_k^{hPRPHZ} = (1 - \theta_k)\beta_k^{HZ} + \theta_k\beta_k^{PRP}, \quad (4.7)$$

where the individual parameters are given by:

$$\beta_k^{HZ} = \frac{1}{d_k^T y_k} \left(y_k - \frac{2d_k \|y_k\|^2}{d_k^T y_k} \right)^T g_{k+1}, \quad (4.8)$$

$$\beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}. \quad (4.9)$$

where $\theta_k \in [0, 1]$ is referred to as the hybridization parameter. Note that if $\theta_k = 0$, then $\beta_k^{hPRPHZ} = \beta_k^{HZ}$, whereas if $\theta_k = 1$, then $\beta_k^{hPRPHZ} = \beta_k^{PRP}$. On the other hand, if $0 < \theta_k < 1$, the parameter θ_k is chosen in such a way that, at each iteration, the conjugacy condition $d_{k+1}^T y_k = 0$ is satisfied independently of the line search. Clearly, this choice ensures that the generated search directions retain the conjugacy property.

$$d_{k+1} = -g_{k+1} + [(1 - \theta_k)\beta_k^{HZ} + \theta_k\beta_k^{PRP}] d_k \quad (4.10)$$

$$d_{k+1} = -g_{k+1} + (1 - \theta_k) \frac{1}{d_k^T y_k} \left(y_k^T g_{k+1} - \frac{2\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1} \right) d_k + \theta_k \frac{g_{k+1}^T y_k}{\|g_k\|^2} d_k \quad (4.11)$$

Multiplying both sides of the above equation by y_k yields:

$$-g_{k+1}^T y_k + (1 - \theta_k) \left(y_k^T g_{k+1} - \frac{2\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1} \right) + \theta_k \frac{g_{k+1}^T y_k}{\|g_k\|^2} d_k^T y_k = 0. \quad (4.12)$$

$$\left[- \left(y_k^T g_{k+1} - \frac{2\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1} \right) + \frac{g_{k+1}^T y_k}{\|g_k\|^2} d_k^T y_k \right] \theta_k = g_{k+1}^T y_k - y_k^T g_{k+1} + \frac{2\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1} \quad (4.13)$$

$$\theta_k = \frac{\frac{2\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}}{\frac{g_{k+1}^T y_k}{\|g_k\|^2} d_k^T y_k - y_k^T g_{k+1} + \frac{2\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}}. \quad (4.14)$$

It is possible that the value of θ_k , computed from (3.14), lies outside the interval $[0, 1]$. Therefore, it is projected onto this interval as follows:

$$\theta_k = \begin{cases} 0, & \text{if } \theta_k^* \leq 0, \\ 1, & \text{if } \theta_k^* \geq 1, \\ \theta_k^*, & \text{otherwise,} \end{cases} \quad (4.15)$$

where

$$\theta_k^* = \frac{\frac{2\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}}{\frac{g_{k+1}^T y_k}{\|g_k\|^2} d_k^T y_k - y_k^T g_{k+1} + \frac{2\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1}}. \quad (4.16)$$

Theorem 10. [13] If relations (3.7) and (3.11) hold, then

$$d_{k+1}^{hPRPHZ} = (1 - \theta_k) d_{k+1}^{HZ} + \theta_k d_{k+1}^{PRP}. \quad (4.17)$$

Proof. We have

$$d_{k+1}^{hPRPHZ} = -g_{k+1} + \beta_k^{hPRPHZ} d_k. \quad (4.18)$$

By adding and subtracting $\theta_k g_{k+1}$, we obtain

$$d_{k+1}^{hPRPHZ} = -g_{k+1} + \beta_k^{hPRPHZ} d_k + \theta_k g_{k+1} - \theta_k g_{k+1} \quad (4.19)$$

$$= (1 - \theta_k)(-g_{k+1}) + \theta_k(-g_{k+1}) + \beta_k^{hPRPHZ} d_k. \quad (4.20)$$

Using

$$\beta_k^{hPRPHZ} = (1 - \theta_k)\beta_k^{HZ} + \theta_k\beta_k^{PRP}, \quad (4.21)$$

we get

$$d_{k+1}^{hPRPHZ} = (1 - \theta_k)(-g_{k+1} + \beta_k^{HZ} d_k) + \theta_k(-g_{k+1} + \beta_k^{PRP} d_k). \quad (4.22)$$

Hence,

$$d_{k+1}^{hPRPHZ} = (1 - \theta_k)d_{k+1}^{HZ} + \theta_k d_{k+1}^{PRP}. \quad (4.23)$$

This completes the proof. $\square$

Assumption 1. The level set

$$S = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\} \quad (4.24)$$

is bounded. That is, there exists a constant $B > 0$ such that

$$\|x\| \leq B, \quad \forall x \in S. \quad (4.25)$$

Assumption 2.[8] In a neighborhood $\mathcal{N}$ of S , the function f is continuously differentiable and its gradient $\nabla f(x)$ is Lipschitz continuous. That is, there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{N}. \quad (4.26)$$

Under these assumptions, there exists a constant $\Gamma \geq 0$ such that

$$\|\nabla f(x)\| \leq \Gamma, \quad \forall x \in S. \quad (4.27)$$

4.4.2 Sufficient descent condition

According to Theorem 8, we have the following:

First, if $\theta_k = 0$, then

$$d_{k+1}^{hPRPHZ} = d_{k+1}^{HZ}. \quad (4.28)$$

In this case, the sufficient descent condition for the hybrid method holds whenever it holds for the HZ method.

It is shown by Hager and Zhang in [15] that the direction d_{k+1}^{HZ} satisfies the sufficient descent condition for all k . The details are given as follows:

Theorem 11. [15] Assume that $d_k^T y_k \neq 0$ and

$$d_{k+1} = -g_{k+1} + \tau d_k, \quad d_0 = -g_0, \quad (4.29)$$

where $\tau \in [\beta_k^{HZ}, \max\{\beta_k^{HZ}, 0\}]$. Then,

$$g_{k+1}^T d_{k+1}^{HZ} \leq -\frac{7}{8} \|g_{k+1}\|^2. \quad (4.30)$$

Proof. According to [23], two cases. We start from the definition of the direction:

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

- Step 1: Compute the directional derivative

Multiply by g_{k+1}^T :

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

Distribute:

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

Since $g_{k+1}^T g_{k+1} = \|g_{k+1}\|^2$, we get:

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

—

- Step 2: We have the Cauchy-Schwarz inequality

$$U^T V \leq \frac{1}{2} (\|U\|^2 + \|V\|^2).$$

$$g_{k+1}^T d_k \leq \|g_{k+1}\| \|d_k\| \leq \frac{1}{2} (\|g_{k+1}\|^2 + \|d_k\|^2).$$

Let us substitute in the following sentence

$$\begin{aligned} g_{k+1}^T d_k &\leq -\|g_{k+1}\|^2 + \frac{1}{2} \beta_k (\|g_{k+1}\|^2 + \|d_k\|^2) \\ &\leq \left(\frac{1}{2} \beta_k - 1\right) \|g_{k+1}\|^2 + \frac{1}{2} \beta_k \|d_k\|^2 \end{aligned}$$

We have:

$$\beta_k > 0.$$

Then

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

From the Wolfe curvature condition, we have:

$$\beta_k \leq \frac{1}{4}$$

$$g_{k+1}^T d_k \leq \left(\frac{1}{8} - 1\right) \|g_{k+1}\|^2$$

Therefore:

$$g_{k+1}^T d_k \leq -\frac{7}{8} \|g_{k+1}\|^2$$

□

Theorem 12. [11] Assume that (3.24), (3.25) hold. Let η be a nonnegative constant such that

$$\|g_k\|^2 \geq \eta \|s_k\|^2, \quad \eta \geq L. \quad (4.31)$$

Then d_{k+1}^{PRP} satisfies the sufficient descent condition for all k .

Proof. Using the Cauchy–Bunyakovsky–Schwarz inequality, we have:

$$g_{k+1}^T d_{k+1}^{PRP} = -\|g_{k+1}\|^2 + \beta_k^{PRP} g_{k+1}^T s_k \quad (4.32)$$

$$= -\|g_{k+1}\|^2 + \frac{g_{k+1}^T y_k}{\|g_k\|^2} (g_{k+1}^T s_k) \quad (4.33)$$

$$\leq -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2 \|y_k\| \|s_k\|}{\|g_k\|^2}. \quad (4.34)$$

From (3.26), we have $y_k \leq L \|s_k\|$, hence:

$$g_{k+1}^T d_{k+1}^{PRP} \leq -\|g_{k+1}\|^2 + \frac{\|g_{k+1}\|^2 L \|s_k\|^2}{\|g_k\|^2}. \quad (4.35)$$

Using (3.38), we obtain:

$$g_{k+1}^T d_{k+1}^{PRP} \leq -\left(1 - \frac{L}{\eta}\right) \|g_{k+1}\|^2 \quad (4.36)$$

Finally [13], for $0 < \theta_k < 1$, there exist constants λ_1, λ_2 such that $0 < \lambda_1 \leq \theta_k \leq \lambda_2 < 1$, we get:

$$g_{k+1}^T d_{k+1}^{hPRPHZ} \leq \eta_1 g_{k+1}^T d_{k+1}^{PRP} + (1 - \eta_2) g_{k+1}^T d_{k+1}^{HZ}. \quad (4.37)$$

Therefore, there exists a constant $k > 0$ such that:

$$g_{k+1}^T d_{k+1}^{hPRPHZ} \leq -k \|g_{k+1}\|^2. \quad (4.38)$$

□

4.4.3 Algorithm of hybrid PRP-HZ Conjugate Gradient Method (hPRPHZ)

This algorithm is of the gradient descent type; its objective is to minimize the function $f(x)$ in an intelligent and efficient manner. It is a hybrid method that combines the PRP and HZ methods.

Algorithm 4 (hPRPHZ). Initialization: Choose an initial point $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$ Compute $g_0 = \nabla f(x_0)$ Set $d_0 = -g_0$, $\alpha_0 = \frac{1}{\|g_0\|^2}$, and $k = 0$

Step 1: If $\|g_k\| < \varepsilon$ then stop, else go to Step 2.

Step 2: Compute α_k using the strong Wolfe line search conditions

Step 3: Update

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

Compute

$$g_{k+1} = \nabla f(x_{k+1}), \quad y_k = g_{k+1} - g_k$$

Step 4: If

$$\frac{g_{k+1}^T y_k}{\|g_k\|^2} d_k^T y_k - y_k^T g_{k+1} + 2 \frac{\|y_k\|^2}{d_k^T y_k} d_k^T g_{k+1} = 0$$

then $\theta_k = 0$ else, compute θ_k as in (3,16)

Step 5: Compute β_k as in (3,17)

Step 6: Compute

$$d = -g_{k+1} + \beta_k^{hPRPHZ} d_k$$

If the restart criterion of Powell condition :

$$|g_{k+1}^T g_k| \geq 0.2 \|g_{k+1}\|^2$$

is satisfied, then $d_{k+1} = -g_{k+1}$, else define $d_{k+1} = d$.

Step 7: Compute the initial guess

$$\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|}$$

Step 8: Set $k = k + 1$ and go to Step 1.

Numerical Example of the hPRPHZ Algorithm

Consider the function:

$$f(x, y) = x^2 + y^2$$

Initialization:

Let

$$x_0 = (2, 1), \quad \varepsilon > 0$$

Compute:

$$g_0 = \nabla f(x_0) = (4, 2)$$

$$d_0 = -g_0 = (-4, -2)$$

$$\alpha_0 = \frac{1}{\|g_0\|^2} = \frac{1}{20} = 0.05$$

Iteration $k = 0$:

Step 1:

$$\|g_0\| = \sqrt{20} \neq 0 \Rightarrow \text{Continue}$$

Step 2: Take the initial step size:

$$\alpha_0 = 0.05$$

Step 3: Compute the next iterate:

$$x_1 = x_0 + \alpha_0 d_0$$

$$x_1 = (2, 1) + 0.05(-4, -2)$$

$$x_1 = (2 - 0.2, 1 - 0.1)$$

$$x_1 = (1.8, 0.9)$$

Compute the new gradient:

$$g_1 = \nabla f(x_1) = (3.6, 1.8)$$

Compute:

$$\begin{aligned} y_0 &= g_1 - g_0 = (-0.4, -0.2) \\ s_0 &= x_1 - x_0 = (-0.2, -0.1) \end{aligned}$$

—
Result at $k = 0$:

$$x_1 = (1.8, 0.9), \quad g_1 = (3.6, 1.8)$$

—
Remark:

The iteration successfully moves from x_0 toward the minimizer $(0, 0)$ following the descent direction.

4.4.4 Global Convergence

The following lemma presents the Zoutendijk condition[23], and a detailed proof can be found in [16].

Lemma 1. [17] Suppose that Assumption 1 and Assumption 2 hold. If d_k is a descent direction and the step size α_k satisfies

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \quad \sigma < 1, \quad (4.39)$$

then

$$\alpha_k \geq \frac{1 - \sigma}{L} \frac{|g_k^T d_k|}{\|d_k\|^2}. \quad (4.40)$$

Proof. From (3.46), using the Cauchy–Bunyakovsky–Schwarz inequality and condition (3.26), we obtain:

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

Since d_k is a descent direction and $\sigma < 1$, it follows that:

$$\alpha_k \geq \frac{1 - \sigma}{L} \frac{|g_k^T d_k|}{\|d_k\|^2} \quad (4.41)$$

Since d_k is a descent direction and $\sigma < 1$, the assertion (3.47) holds. Obviously, from the strong Wolfe conditions and (3.45), the step length α_k satisfies (3.47). According to Assumptions (1) and (2) together with (3.47), it is easy to obtain that $g_k^T d_k \neq 0$ for all $k \geq 0$. Thus, $\alpha_k = 0$ does not satisfy (4). This indicates that the step size generated by the hPRPHZ method is strictly positive, i.e., there exists a constant $\lambda > 0$ such that

$$\alpha_k \geq \lambda, \quad \forall k \geq 0. \quad (4.42)$$

□

Lemma 2. Suppose that Assumptions (1) and (2) hold. Consider the common iteration (2), where d_k is a descent direction and α_k satisfies the Wolfe line search conditions (3). Then the Zoutendijk condition holds:

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \quad (4.43)$$

holds.

The following theorem gives the global convergence of the hPRPHZ method.

Theorem 13. Suppose that Assumptions (1) and (2) hold. Let $\{x_k\}$ be generated by the hPRPHZ algorithm. Then

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (4.44)$$

Proof. Suppose, by contradiction, that (3.51) is false. Then there exists a constant $c > 0$ such that

$$\|g_k\|^2 \geq c, \quad \forall k \text{ sufficiently large.} \quad (4.45)$$

From (3.26), we obtain

$$\|y_k\| = \|g_{k+1} - g_k\| \leq LD, \quad (4.46)$$

where $D = \max\{\|x - y\| : x, y \in N\}$ is the diameter of the level set N .

By using (4) and (3.45), we have

$$d_k^T y_k \geq \sigma d_k^T g_k - d_k^T g_k \quad (4.47)$$

$$\geq (1 - \sigma)k\|g_k\|^2 \quad (4.48)$$

$$\geq (1 - \sigma)kc. \quad (4.49)$$

□

From (3.11), (3.27), (3.52), (3.53), and (3.54), we obtain

$$|\beta_k^{hPRPHZ}| \leq |\beta_k^{HZ}| + |\beta_k^{PRP}| \quad (4.50)$$

$$\leq \frac{1}{d_k^T y_k} \left(\|y_k\| \|g_{k+1}\| + \frac{2\|y_k\|^2 \|d_k\| \|g_{k+1}\|}{d_k^T y_k} \right) + \frac{\|g_{k+1}\| \|y_k\|}{\|g_k\|^2} \quad (4.51)$$

$$\leq \frac{LD\Gamma}{(1 - \sigma)kc} \left(1 + \frac{2LD^2}{\lambda(1 - \sigma)kc} \right) + \frac{\Gamma LD}{c} = Q. \quad (4.52)$$

Also, from (3.21) and (3.49), we have

$$\|d_{k+1}\| \leq \|g_{k+1}\| + |\beta_k^{hPRPHZ}| \|d_k\| \quad (4.53)$$

$$= \|g_{k+1}\| + |\beta_k^{hPRPHZ}| \|s_k\| \frac{1}{\alpha_k} \quad (4.54)$$

$$\leq \Gamma + \frac{QD}{\lambda} = W. \quad (4.55)$$

Hence,

$$\|d_{k+1}\| \leq W$$

which implies

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

Therefore,

$$\sum_{k \geq 0} \frac{(d_k^T g_k)^2}{\|d_k\|^2} = +\infty,$$

which contradicts Lemma (2). Thus, therefore the claim (3.51) is proved.

4.5 Numerical tests

In this section, we present some numerical experiments. We tested the PRP and HZ methods on problems from the CUTE library [10] and compared their performance with that of the hPRPHZ method. For the numerical tests, the parameters in Wolfe's strong line search procedures were chosen to be $\sigma = 0.9$ and $\delta = 0.0001$. We stop the iteration if the condition $\|g(x_k)\| \leq 10^{-6}$ is satisfied. In this paper, all codes were written in MATLAB and run on a PC equipped with an Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz processor, 4GB RAM memory, and the Windows 10 Professional operating system.

4.5.1 numerical results

Convenient for comparison, all tests are performed under a variant of the generalized Wolfe line search as follows. Table 1 lists the numerical results. The meaning of each column is given below:

- **problem**: the name of the test problem,
- **n**: the dimension of the test problem,
- **iter**: the number of iterations,
- **time**: the CPU time in seconds.

Table 4.1: Numerical Results

Problems	n	hPRPHZ		PRP		HZ	
		time	iter	time	iter	time	iter
EXTROSNB	50	0.13	5819	0.19	5294	0.24	7808
CHNROSNB	50	0.03	273	0.02	285	0.01	500
TOINTGOR	50	0.88	151	0.01	155	0.01	250
ERRINROS	50	0.02	1444	0.09	2416	NaN	NaN
DECONVU	61	0.02	462	0.06	460	0.07	581
EIGENCLS	90	0.25	360	0.07	350	0.11	743
MANCINO	100	0.35	12	0.60	11	1.15	27
GENROSE	100	0.02	347	0.02	392	0.03	626
NONCVXU2	100	0.06	396	0.03	414	0.05	801
SENSORS	100	1.07	44	0.97	45	1.36	66
TESTQUAD	100	0.01	321	0.01	303	0.01	925
EIGENALS	110	0.04	389	0.08	359	0.16	806
FMINSRF2	121	0.03	115	0.01	124	0.01	250
CURLY30	200	0.48	2819	0.36	3066	NaN	NaN
CURLY20	200	0.71	2951	0.30	2835	NaN	NaN
SPARSINE	200	0.06	445	0.08	445	0.13	917
HILBERTB	200	0.21	6	0.22	6	0.25	13
HILBERTA	200	0.71	50	0.37	25	0.38	38
CURLY10	200	0.21	3100	0.20	3182	NaN	NaN

Problems	n	hPRPHZ		PRP		HZ	
		time	iter	time	iter	time	iter
PENALTY2	200	0.18	365	0.14	417	NaN	NaN
ARGLINA	300	0.23	2	0.25	2	0.26	5
ARGLINC	300	0.08	19	0.27	25	NaN	NaN
ARGLINB	300	0.13	23	0.20	17	NaN	NaN
BROWNAL	400	0.07	13	0.20	7	0.27	37
EIGENALS	420	3.97	1344	2.49	1306	4.74	2998
EIGENBLS	420	3.55	4978	10.11	5440	14.93	9714
EIGENBLS	420	0.10	8726	NaN	2428	0.90	NaN
EIGENCLS	462	4.25	1802	4.10	1883	5.99	3312
FLETGBV2	500	0.001	NaN	NaN	NaN	3.52	NaN
FLETGBV2	500	0.26	480	0.22	482	0.36	962
COSINE	500	0.02	23	0.00	22	0.01	26
GENHUMPS	500	1.01	2258	2.15	2531	2.70	4147
GENROSE	500	0.29	1309	0.49	1624	0.46	2278
SINQUAD	500	0.08	111	0.04	93	NaN	NaN
DIXON3DQ	500	0.24	500	0.06	500	0.08	1003
NONCVXUN	500	0.10	15084	0.05	15797	0.90	NaN
EIGENBLS	930	0.10	15122	0.05	15401	0.90	NaN
EIGENALS	930	0.10	1116	0.05	1116	0.90	2231
GENROSE	1000	0.10	2261	0.50	2261	0.90	4720
CHAINWOO	1000	0.10	4516	0.50	4483	0.90	8793
TESTQUAD	1000	0.10	1344	0.05	1306	0.90	2998
NONDQUAR	1000	0.38	3147	1.45	4900	1.63	8128
BDQRTIC	1000	0.46	1025	0.49	798	NaN	NaN
DIXON3DQ	1000	0.31	1002	0.27	1002	0.33	2005
FLETGBV2	1000	1.40	1055	1.00	1044	1.29	1886
NONCVXU2	1000	1.56	2055	1.92	2015	3.64	3919
CURLY30	1000	8.86	15122	8.87	15401	NaN	NaN
CURLY20	1000	10.91	15084	6.96	15797	NaN	NaN
SPMSRTLS	1000	0.24	151	0.15	151	0.20	281
CURLY10	1000	4.27	14406	4.06	13659	NaN	NaN
SPARSINE	1000	2.57	4516	4.32	4483	6.59	8793
FLETCHCR	1000	6.03	7479	4.93	9139	3.57	8986
GENHUMPS	1000	2.24	3555	5.84	3435	7.55	5807
FREUROTH	1000	0.07	187	0.16	137	NaN	NaN
BRYBND	1000	0.06	52	0.06	35	0.08	73
POWER	1000	0.06	117	0.06	222	0.04	236
COSINE	1000	0.03	24	0.02	24	0.03	29
ARWHEAD	1000	0.01	16	0.03	19	NaN	NaN
EG2	1000	0.01	6	0.01	6	NaN	NaN
BROYDN7D	1000	5.39	498	0.27	371	NaN	NaN
CURLY30	1000	0.10	783	0.05	783	0.90	1565
CURLY20	1000	0.10	303	NaN	306	0.90	620

Problems	n	hPRPHZ		PRP		HZ	
		time	iter	time	iter	time	iter
FMINSURF	1024	0.12	238	0.24	300	0.28	455
FMINSRF2	1024	0.14	282	0.26	355	0.29	517
DIXMAANA	1500	0.01	10	0.03	9	0.03	22
DIXMAANB	1500	0.01	10	0.03	10	0.04	24
CRAGGLVY	2000	0.33	132	0.37	142	NaN	NaN
DIXMAANB	3000	0.04	10	0.06	10	0.07	23
DIXMAANA	3000	0.21	10	0.05	9	0.07	20
SPMSRTL	4999	0.001	2232	5.47	2183	6.45	4093
SPARSINE	5000	0.13	370	1.57	544	1.12	719
GENHUMPS	5000	0.10	1802	0.05	1883	0.90	3312
TESTQUAD	5000	0.10	3555	0.50	3435	0.90	5807
ENGVAL1	5000	NaN	1	NaN	NaN	3.52	NaN
FLETCHCR	5000	0.10	492	0.05	669	0.90	949
GENROSE	5000	0.10	4978	0.05	5440	0.90	9714
ARWHEAD	5000	0.21	7298	36.84	6398	NaN	NaN
COSINE	5000	59.19	37	0.20	35	NaN	NaN
SINQUAD	5000	0.10	5099	0.50	5058	0.90	10058
POWER	5000	0.65	259	0.61	408	0.40	514
NONDQUAR	5000	4.24	5099	7.50	5058	9.47	10058
TRIDIA	5000	1.09	783	1.07	783	1.31	1565
CRAGGLVY	5000	0.74	143	0.99	138	NaN	NaN
FREUROTH	5000	0.40	96	0.59	76	NaN	NaN
BRYBND	5000	0.45	39	0.32	40	0.38	66
VARDIM	5000	0.20	44	0.13	47	NaN	NaN
LIARWHD	5000	0.26	32	0.30	48	0.25	46
DQRTIC	5000	0.09	34	0.10	34	0.10	66
BDQRTIC	5000	1.35	8726	7.64	2428	NaN	NaN
QUARTC	5000	0.09	34	0.09	34	0.10	66
FMINSURF	5625	1.00	492	3.47	669	3.39	949
FMINSRF2	5625	3.05	227	1.31	214	1.89	430
DIXMAANE	6000	1.22	303	1.26	306	2.13	620
DIXMAANJ	6000	23.80	296	1.18	275	2.17	557
DIXMAANK	6000	1.81	264	1.11	248	1.80	587
DIXMAANL	6000	0.97	245	1.32	215	3.01	702
DIXMAANF	6000	1.04	230	1.12	230	1.62	437
DIXMAANG	6000	1.34	227	1.08	227	1.45	420
DIXMAANH	6000	0.99	224	1.16	224	2.64	825
DIXMAANI	6000	9.43	2661	9.06	2261	13.98	4720
DIXMAANC	6000	0.22	11	0.24	11	0.26	23
DIXMAANI	6000	0.10	296	NaN	275	0.90	557
DIXMAAND	6000	0.25	13	0.13	12	0.16	25
LIARWHD	10000	0.57	41	0.46	39	0.48	46
TRIDIA	10000	7.32	1116	3.19	1116	3.89	2231

Problems	n	hPRPHZ		PRP		HZ	
		time	iter	time	iter	time	iter
SCHMVETT	10000	2.38	60	1.50	64	2.59	105
MOREBV	10000	1.19	97	0.89	97	1.28	201
WOODS	10000	0.84	257	1.17	230	2.14	487
SPARSQR	10000	0.35	23	0.38	23	1.13	131
VARDIM	10000	0.27	62	0.29	57	NaN	NaN
ENGVAL1	10000	0.28	35	0.41	34	NaN	NaN
POWELLSG	10000	0.25	77	0.23	49	0.72	362
TQUARTIC	10000	0.19	61	0.65	52	0.58	38
NONDIA	10000	0.26	16	0.23	10	0.31	26
TOINTGSS	10000	0.03	5	0.21	5	0.38	20
FMINSRF2	10000	0.28	26	0.12	23	0.13	27
NONDQUAR	10000	1.32	234	2.42	225	3.52	440
POWER	10000	43.75	142	0.71	62	NaN	NaN
COSINE	10000	3.66	8476	31.52	4721	53.25	8965
FMINSURF	10000	0.64	8771	2.14	5022	2.41	6779
SPMSRTL	10000	0.001	NaN	NaN	NaN	0.28	NaN
FREUROTH	10000	0.001	NaN	NaN	NaN	1.89	NaN
BDQRTIC	10000	0.001	1	NaN	NaN	0.28	NaN
VAREIGVL	10000	0.001	1	NaN	NaN	1.89	NaN
BRYBND	10000	0.10	34677	0.50	456454	0.90	40000
SINQUAD	10000	0.10	14406	0.05	13659	0.90	NaN
TESTQUAD	10000	0.10	7479	0.05	9139	0.90	8986
FMINSURF	15625	0.36	108	0.47	62	NaN	NaN
FMINSRF2	15625	1.13	28	0.26	23	0.28	28

Notes on the table

The numerical results clearly demonstrate that the proposed hybrid method (hPRPHZ) outperforms the classical PRP and HZ methods in terms of robustness and efficiency.

In most test problems, the hPRPHZ method achieves competitive or superior computational time while maintaining a reasonable number of iterations. Moreover, unlike the HZ method, which frequently fails (NaN values), the hybrid method shows strong stability across all test cases.

Although the PRP method sometimes requires fewer iterations, it does not consistently lead to lower computational time. This confirms that iteration count alone is not sufficient to evaluate performance.

Overall, the hybrid hPRPHZ method provides a good balance between convergence speed, numerical stability, and robustness, making it a reliable choice for solving large-scale unconstrained optimization problems.

4.5.2 Performance profile

Figures 1 and 2 show the performance of these methods with respect to the number of iterations (Iter) and CPU time, evaluated using the performance profiles of DOLAN and MORÉ [7].

Benchmark results are generated by running a solver on a set P of problems and recording information of interest such as the number of iterations (**Iter**) and CPU time (**Tcpu**). Let S be the set of solvers under comparison. Assume that S consists of n_s solvers and P consists of n_p problems.

For each problem $p \in P$ and solver $s \in S$, let $t_{p,s}$ denote the computing time (or the number of iterations) required by solver s to solve problem p . The comparison between different solvers is based on the following performance ratio:

$$r_{p,s} = \frac{t_{p,s}}{\min \{t_{p,s} : s \in S\}}.$$

Assume that a sufficiently large parameter $r_M \geq r_{p,s}$ for all p, s is chosen, and

$$r_{p,s} = r_M$$

if and only if solver s does not solve problem p . Define

$$\rho_s(\tau) = \frac{1}{n_p} \text{size} \{p \in P : \log(r_{p,s}) \leq \tau\}.$$

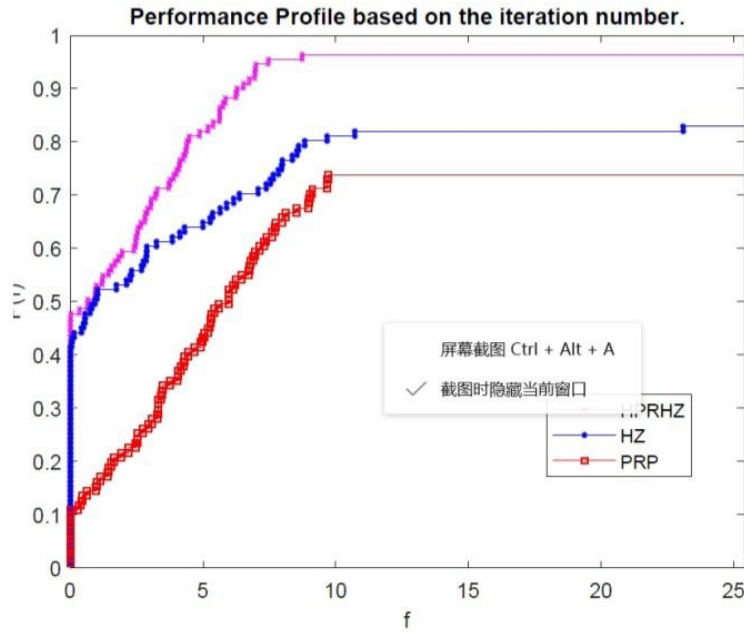


Figure 4.1: Figure 1

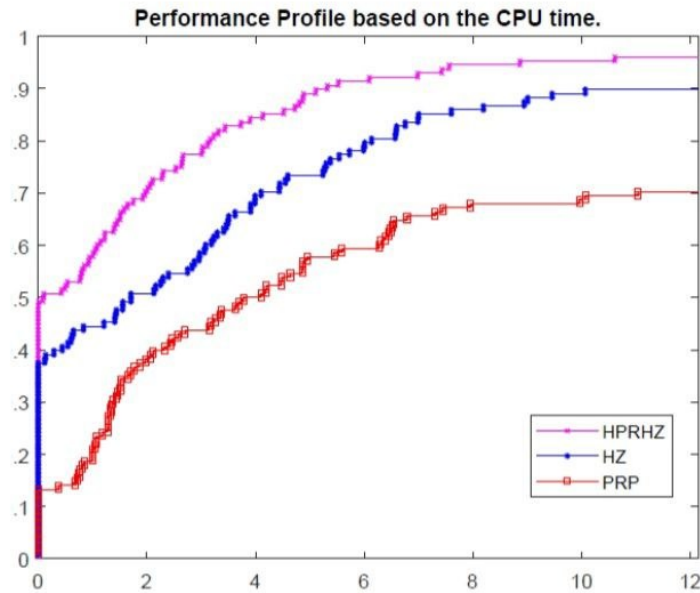


Figure 4.2: Figure 2

where $\text{size}(A)$ denotes the number of elements in the set A . Then, $\rho_s(\tau)$ represents the probability for solver $s \in S$ that the performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}^n$. The function ρ_s is the cumulative distribution function associated with the performance ratio.

The value $\rho_s(1)$ represents the probability that solver s outperforms all the other solvers. In other words, for each method, we plot the fraction P of problems for which the method is within a factor of the best performance.

The left side of the figure shows the percentage of test problems for which a method is the fastest, while the right side indicates the percentage of test problems successfully solved by each method. The top curve corresponds to the method that solved the largest number of problems within a factor of the best computing time.

Based on the performance profile theory described above, the performance profiles presented in Figures 1–2 are generated according to the data reported in Table 1.

From these figures, it can be observed that the proposed hPRPHZ method outperforms the other conjugate gradient methods on the considered test problems.

Conclusion

In this thesis, we studied nonlinear conjugate gradient methods for unconstrained optimization problems, with particular emphasis on hybrid conjugate gradient techniques. After presenting the necessary mathematical preliminaries related to vector spaces, convexity, gradients, Hessians, and optimality conditions, we reviewed several classical conjugate gradient methods such as Fletcher–Reeves (FR), Polak–Ribière–Polyak (PRP), Hestenes–Stiefel (HS), Dai–Yuan (DY), and Hager–Zhang (HZ).

The main contribution of this work consists in the study of a hybrid conjugate gradient method based on a convex combination of the PRP and HZ parameters. The proposed hybrid strategy was designed to combine the practical efficiency of the PRP method with the strong theoretical convergence properties and numerical stability of the HZ method. An adaptive parameter θ_k was introduced in order to preserve the conjugacy condition and improve the overall performance of the algorithm.

Furthermore, sufficient descent conditions and global convergence properties of the proposed method were analyzed under standard assumptions and strong Wolfe line search conditions. Numerical experiments and performance profiles demonstrated that the hybrid PRP–HZ method is competitive and effective for solving large-scale unconstrained optimization problems.

Overall, this study confirms that hybrid conjugate gradient methods represent a promising direction in nonlinear optimization, as they provide a good balance between robustness, convergence behavior, and computational efficiency. Future work may focus on extending these techniques to constrained optimization problems, developing new adaptive hybridization parameters, or applying these methods to practical problems arising in engineering, machine learning, and scientific computing.

Bibliography

- [1] D. Thomine, *Notes de cours d'Algèbre M302*, Université Paris-Saclay, L3 Math (2023). pp. 1–2.
- [2] bibmath.net, *Résumé de cours : espaces vectoriels normés*, February 22, 2026.
- [3] J. Weidmann, *Linear Operators in Hilbert Spaces*, Graduate Texts in Mathematics, Springer. p. 29.
- [4] N. Andrei, *Convex functions*, AMO – Advanced Modeling and Optimization **9** (2007), no. 2.
- [5] M. Al-Baali, *Descent property and global convergence of the Fletcher–Reeves method with inexact line search*, IMA Journal of Numerical Analysis **5** (1985), 121–124.
- [6] B. Naima, *Optimisation sans contraintes : aspect théorique et algorithmique*, Université de Batna 2.
- [7] N. Andrei, *Another hybrid conjugate gradient algorithm for unconstrained optimization*, Numerical Algorithms **47** (2008), 143–156.
- [8] ———, *New hybrid conjugate gradient algorithms for unconstrained optimization*, Encyclopedia of Optimization, 2009.
- [9] B. Balam and M. Narayanan and P. Rajendrakumar, *Optimal design of multi-parametric nonlinear systems using a genetic algorithm approach*, Nonlinear Dynamics **67** (2012), 2759–2777.
- [10] I. Bongartz and A. R. Conn and N. Gould and P. L. Toint, *CUTE: constrained and unconstrained testing environments*, ACM Trans. Math. Software **21** (1995), 123–160.
- [11] N. Chenna, *Comments on “New Hybrid Conjugate Gradient Method as a Convex Combination of FR and PRP Methods”*, Filomat **33** (2019), 3083–3100.
- [12] Y.-H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM Journal on Optimization **10** (1999), 177–182.
- [13] S. S. Djordjević, *New hybrid conjugate gradient method as a convex combination of FR and PRP methods*, Filomat **30** (2016), 3083–3100.
- [14] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming **91** (2002), 201–213.
- [15] W. W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent*, SIAM Journal on Optimization **16** (2005), 170–192.
- [16] X. Z. Jiang and G.-D. Ma and J.-B. Jian, *A new globally convergent conjugate gradient method*, Chinese Journal of Engineering Mathematics **28** (2011), 779–786.
- [17] J. Liu and S. Li, *New hybrid conjugate gradient method for unconstrained optimization*, Applied Mathematics and Computation **245** (2014), 36–43.
- [18] E. Polak and G. Ribiere, *Note sur la convergence des méthodes de directions conjuguées*, RAIRO **3** (1969), 35–43.

- [19] B. T. Polyak, *The conjugate gradient method in extremal problems*, USSR Computational Mathematics and Mathematical Physics **9** (1969), 94–112.
- [20] M. J. D. Powell, *Nonconvex minimization calculations and the conjugate gradient method*, Numerical Analysis, 1984, pp. 122–141.
- [21] Z. Wei and G. Li and L. Qi, *New nonlinear conjugate gradient formulas*, Applied Mathematics and Computation **179** (2006), 407–430.
- [22] G. Yuan and Z. Wei and Q. Zhao, *A modified PRP conjugate gradient algorithm*, IIE Transactions **46** (2014), 397–413.
- [23] G. Zoutendijk, *Nonlinear programming, computational methods*, Integer and Nonlinear Programming, 1970, pp. 37–86.
- [24] T. Bouli, *Convergence des méthodes du gradient conjugué avec recherche linéaire non monotone*, Université Badji Mokhtar Annaba, 2014.
- [25] L. Zahia-Debailia and A. Laouamri, *Convergence globale d’une nouvelle méthode à direction conjuguée*, Université Larbi Tebessi, Tébessa.
- [26] M. J. D. Powell, *Convergence properties of algorithms for nonlinear optimization*, SIAM Review **28** (1986), 487–500.
- [27] Dcode, *Point stationnaire d’une fonction*, March 2026. <https://www.dcode.fr/point-stationnaire-fonction>.
- [28] S. Radjef, *Optimisation : cours et exercices*, Université des Sciences et de la Technologie Mohamed Boudiaf, Oran. pp. 9–12.
- [29] Jahida, *Résultats de convergence des méthodes du gradient conjugué non linéaire*, Université de Tébessa, 2014.
- [30] DataCamp, *Hessian Matrix Tutorial*, March 6, 2026. <https://www.datacamp.com/fr/tutorial/hessian-matrix>.
- [31] HQurve, *Stationary Points and Optimization*, April 2026. <https://hqurve-site.gitlab.io/math2270/development/stationary-points-and-optimization.htm>.
- [32] S. Radjef (Ep. Douar), *Optimisation : cours et exercices*, USTO Oran, 2019. pp. 10–12.
- [33] Jackie Nicholas, *The first derivative and stationary points* (2004). pp. 1–4.