
Iterative Methods for Solving Nonlinear Monotone Operator Equations with Applications in Compressive Sensing and Motion Control

Dissertation

zur Erlangung des Doktorgrades der Naturwissenschaften

(Dr. rer. nat)

der

Naturwissenschaftlichen Fakultät II, Chemie, Physik und Mathematik

der

MARTIN-LUTHER-UNIVERSITÄT HALLE-WITTENBERG

vorgelegt von

Herrn Abubakar Bakoji Muhammad

geboren in Gombe, Nigeria

Gutachter:

1. Frau Prof. Dr. Christiane Tammer (Martin-Luther-Universität Halle-Wittenberg)
2. Herr Prof. Dr. Axel Kröner (Martin-Luther-Universität Halle-Wittenberg)
3. Herr Prof. Dr. Jen-Chih Yao (China Medical University Taiwan)

Tag der Einreichung: 10.06.2024

Tag der Verteidigung: 28.10.2024

To my family

Acknowledgments

My profound gratitude goes out to my able supervisor, Prof. Dr. Christiane Tammer, for her unwavering support, guidance, insightful conversations, and innovative ideas. She is also very kind and patient, especially during difficult times. Her encouragement during the course of this program is truly exceptional, among the best one could find anywhere, and is most appreciated. It was a great joy and a unique experience to work with her. When I joined her research group, she accepted me warmly and helped me with both my research and my personal life. Our conversations were always quite fruitful because of her warmth and extensive knowledge of the subject, which gave me a lot of confidence when things got tough. Secondly, I would like to thank Dr. Rosalind Elster most sincerely for the joint work and careful reading of the thesis, which provided many valuable suggestions. I can't find the right words to express my gratitude for all the support you've provided. In addition, I would like to express my gratitude to all of my coauthors for their inspirational collaborative work, for dedicating time to mathematical discussions, and for their friendship.

In addition, I would like to thank my colleagues at the Institute of Mathematics, Uni-Halle, in particular all the current and former members of the working group on Optimization and Stochastics, including Dr. Christian G., Dr. Niklas H., Dr. Ernest Q., Dr. Bettina Z., Dr. Marcus H., Dr. Marcel M., Baharey K., Khalid I. U., and Mustapha I. Special thanks to Tuan An Vuh and his beloved wife, Tam Thanh Le, for their support during my early challenging years in Halle. I also want to thank our former and present administrative staff for their support. Also, to Mrs. Smykalla of the university's former International Office and Mrs. Klaube of the Dean's office, I say a very big thank you.

Furthermore, I am grateful to the Nigerian-German scholarship program, specifically PTDF Nigeria and DAAD Germany, for providing financial support for my PhD program. I also acknowledged the support from my home university, Gombe State University, Nigeria.

I would like to express my profound gratitude to my family for always being there for me. A special thanks to my beloved parents for their constant support, prayers, and unconditional love; without their sacrifices, I would not have made it this far. I am very much indebted to my dear wife, Mrs. Zainab Isah, and our beloved children, Al'ameen and Abdallah, for their patience, constant prayers, and emotional support during the hardest times. I am especially grateful to all my siblings and extended family members for their unwavering love and support. Special thanks to my second family in Halle, Maman Chinaza and her kind husband, Mr. Eric Ezeani. You guys are wonderful. Lastly, I would like to express my profound gratitude to all of my friends for their support, prayers, and trust in me. I will always remember your contributions, in one way or another.

Contents

1	Introduction	1
2	Mathematical Background	10
2.1	Fundamentals of Functional Analysis	10
2.1.1	Linear Spaces and Order Structure	10
2.1.2	Metric Spaces	16
2.1.3	Normed Spaces	17
2.1.4	Lipschitz Continuity	23
2.2	Properties of Operators and Functions	24
2.3	Reformulation of ℓ_1 -Norm Regularization in Compressive Sensing	35
2.4	Line Search	36
2.5	Overview of Solution Methods for Unconstrained Optimization Problems	38
2.5.1	Steepest Descent Method	39
2.5.2	Newton Methods	40
2.5.3	Quasi-Newton Methods	41
2.5.4	Conjugate Gradient Method	42
2.5.5	Spectral Gradient Method	45
2.5.6	Spectral-Conjugate Gradient Methods	45
3	Derivative-Free Algorithms for Nonlinear Systems of Equations	47
3.1	Algorithm 1: Modified Dai-Yuan Projection Methods (MDY)	48
3.2	Algorithm 2: HCDLS Method	52
3.3	Algorithm 3: Three-term Dai-Liao Projection (TDLP) Method	54
3.4	Convergence Analysis of Algorithms 1, 2 and 3	57
4	Two-Step Projection Methods	64
4.1	Inertial-Type Projection Method	64
4.1.1	Conjugate Gradient and Spectral Gradient Algorithms with Inertial-Step	65
4.1.2	Convergence Analysis of Algorithm 4 (CGAIS) and Algorithm 5 (SAIS)	68
4.2	Two-Step Hybrid Spectral Gradient Projection Method	72
4.2.1	Proposed Algorithm	73
4.2.2	Convergence Analysis of THSP Algorithm	78

5	Numerical Experiments	81
5.1	Experiments with the Derivative-Free Algorithms	81
5.1.1	Numerical comparisons of Algorithms 1, 2 and 3	82
5.1.2	Numerical Comparisons of Algorithm 3 (TDLP) with DPP and CGPM Methods	86
5.2	Numerical Experiments and Comparisons for Algorithms 4 and 5	87
5.2.1	Numerical Comparisons of Algorithm 4 (CGAIS) With CGWOI Method	88
5.2.2	Numerical Comparisons of Algorithm 5 (SAIS) With DAIS 1 and DAIS 2 Methods	90
6	Applications in Compressive Sensing and Motion Control Problems	92
6.1	Applications in Compressive Sensing	92
6.2	Applications in Motion Control Problems	99
7	Application for Solving Vector-Valued Approximation Problems	104
7.1	Formulation of the Problem	104
7.2	Solution Concepts in Vector Optimization	105
7.3	Weighted Sum Scalarization Approach	107
7.4	Formulation of the Multiobjective Approximation Problem	110
7.5	Application of Algorithm 3 (TDLP) for Solving Vector-Valued Approximation Problems	112
8	Conclusion and Outlook	115
8.1	Conclusion	115
8.2	Future Outlook	117
8.3	Summary of Contributions	117

List of Symbols and Abbreviations

$:=$	equal by definition
\mathbb{Z}	set of integers
\mathbb{N}	natural numbers, that is, $\mathbb{N} := \{1, 2, 3, \dots\}$
\mathbb{N}_0	$\mathbb{N} \cup \{0\}$
\mathbb{R}	real numbers
\mathbb{R}^n	n -dimensional Euclidean space
\mathbb{R}_+^n	nonnegative orthant of \mathbb{R}^n
\mathbb{R}^p	standard ordering cone in \mathbb{R}^p
\mathbb{R}_+	set of nonnegative real numbers
\mathbb{R}_{++}	positive real numbers
$\overline{\mathbb{R}}$	$\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$
$\ \cdot\ $	norm $\ \cdot\ : X \rightarrow \mathbb{R}$
$\langle \cdot, \cdot \rangle$	Euclidean scalar product or inner product
$\ \cdot\ _1$	Manhattan norm $\ \cdot\ _1 : \mathbb{R}^n \rightarrow \mathbb{R}$ (also known as ℓ_1 -norm or Lebesgue norm)
$\ \cdot\ _2$	Euclidean norm $\ \cdot\ _2 : \mathbb{R}^n \rightarrow \mathbb{R}$ (also known as ℓ_2 -norm)
$\ \cdot\ _\infty$	Maximum norm $\ \cdot\ _\infty : \mathbb{R}^n \rightarrow \mathbb{R}$ (also known as ℓ_∞ -norm or Chebyshev norm)
$\ \cdot\ _p$	p -Norm, $1 \leq p \leq \pm\infty$ (also known as ℓ_p -norm)
$E(f)$	Epigraph of $f : X \rightarrow \overline{\mathbb{R}}$
X, Y, Z, \dots	real linear spaces or topological linear spaces
K	cones (in X, Y, Z)
K -convex	cone-convex (function)
$\text{int}K$	interior of a cone K
K^*	dual cone to K
\geq_K, \leq_K	a partial ordering relation generated by K
X^*	the topological dual space of X
$f : X \rightarrow Y$	function with preimage space X and image space Y
$\ \cdot\ _X, \ \cdot\ _*$	norm in X , norm in X^*
$\text{dom}f$	domain of a vector-valued function $f : X \rightarrow Y$
$F : X \rightrightarrows Y$	Set-valued mapping
$\mathcal{D}(F)$	domain of a set-valued function $F : X \rightrightarrows Y$
$\mathcal{G}(F)$	graph of $F : X \rightrightarrows Y$

$(X, \langle \cdot, \cdot \rangle)$	inner product space
$(X, \ \cdot\ _X)$	norm space
$\partial f(x^0)$	Fenchel <i>subdifferential</i> of a function $f : X \rightarrow \overline{\mathbb{R}}$ at $x^0 \in X$
$N(x^0, E)$	normal cone N of the set E at the point x^0
$x_n \rightarrow x$	(strong) convergence of a sequence $\{x_n\}$ to x
$x_n \rightharpoonup x$	weak convergence of a sequence $\{x_n\}$ to x
$f[E]$	$f[E] := \cup_{x \in E} f(x)$ for $f : E \rightarrow Y$
$\text{Eff}(f[E], K)$	set of Pareto efficient element E w.r.t. objective function f and the cone K , i.e., $\text{Eff}(f[E], K) := \{y^0 \in E \mid f[E] \cap (y^0 - (K \setminus \{0\})) = \emptyset\}$
$\text{Eff}_W(f[E], K)$	set of weakly minimal elements of E w.r.t. objective function f and the cone K , i.e., $\text{Eff}_W(f[E], K) := \{y^0 \in E \mid f[E] \cap (y^0 - \text{int}K) = \emptyset\}$
$\text{Eff}_P(F[E], K)$	set of properly efficient elements of E with respect to the cone K , i.e., $\text{Eff}_P(F[E], K) := \{y^0 \in E \mid f[E] \cap (y^0 - (\bar{K} \setminus \{0\})) = \emptyset\}$
$\text{Min}(f[E], K)$	set of minimal solutions of E w.r.t. objective function f and the cone K , i.e., $\text{Min}(f[E], K) := \{x \in E \mid f(x) \in \text{Eff}(f[E], K)\}$
$\text{Min}_W(f[E], K)$	set of weakly minimal solutions of E w.r.t. objective function f and the cone K , i.e., $\text{Min}_W(f[E], K) := \{x \in E \mid f(x) \in \text{Eff}_W(f[E], K)\}$
CS	Compressive Sensing
MDY	Modified Dai-Yuan method
HCDLS	Hybrid CG method as a convex combination of Conjugate Descent and Liu–Storey
TDLP	Three term Dai–Liao Projection method
CGAIS	Conjugate Gradient Algorithm with Inertial–Step
CGWOI	Conjugate Gradient algorithm WithOut Inertial–Step
SAIS	Spectral gradient Algorithm with Inertial–Step
TSHP	Two-Step Hybrid Spectral Gradient Projection Method

List of Figures

1.1	The original Mars image (top left), the blurred image (top right), the restored image by MDY1 (bottom left) and by MDY2 (bottom right)	2
1.2	A 2-dimensional planar robot arm.	5
2.1	Unit balls for the norms: $\ \cdot\ _1$, $\ \cdot\ _2$ and $\ \cdot\ _\infty$ on \mathbb{R}^2	19
2.2	Representation of: convex, concave, non-convex and epigraph of the functional f (where $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$) [reprinted from [94, Figures: 2.1, 2.4, 2.5, 2.6].	22
3.1	Illustration of the metric projection of Algorithm 1 adapted from the Figure 1 in Abubakar et al. [1]	52
5.1	Performance profile based on number of: Iterations, Function Evaluations CPU Time.	85
5.2	Percentage Data Profile of MDY2, HCDLS and TDLP methods.	86
5.3	Performance profile based on number of: Iterations, Function Evaluations CPU Time	87
5.4	Performance profiles based on number of iterations (ITER) [reprinted from [125, Figure 1]].	89
5.5	Performance profiles based on number of function evaluations (FVAL) [reprinted from [125, Figure 2]].	89
5.6	Percentage Data Profile of CGAIS, CGWOI and HSS methods [reprinted from [125, Figure 3]].	90
5.7	Performance profiles based on number of iterations and function evaluations [reprinted from [125, Figures 4 and 5]].	91
6.1	Reconstruction of sparse signal. From top to bottom is the original signal, the measurement and the reconstructed signals by the: TDLP, HCDLS and MDY1.	97
6.2	Comparison results of the TDLP, HCDLS and MDY1 Algorithms. The x -axis represents the number of iterations (top and bottom left) and the CPU time in seconds (top and bottom right), while the y -axis represents the MSE (top left and right) and the function values (bottom left and right).	97
6.3	Original images (first column), blurred images (second column), restored images by: the TDLP Algorithm (third column), the HCDLS Algorithm (fourth column) and the MDY2 Algorithm (fifth column)	99

6.4	The robot's trajectories path and residual errors $\varepsilon(t_{k+1})$ along x and y axes of the motion control model for the Lissajous curve $q_{dk}^{(1)}$ of Algorithm 7 (MCGAIS) [reprinted from [125, Figure 6]].	102
6.5	The robot's trajectories path and residual errors $\varepsilon(t_{k+1})$ along x and y axes of the motion control model for the Lissajous curve $q_{dk}^{(2)}$ of Algorithm 7 (MCGAIS) [reprinted from [125, Figure 7]].	103
7.1	The set $\text{Eff}(F, K)$ and the set $\text{Eff}_w(F, K)$ of a set F w.r.t. $K = \mathbb{R}_+^2$ with $y^1, y^2 \notin \text{Eff}(F, K)$.	106
7.2	Representatives of the set of approximate solutions of Problem (P^m) generated for $\bar{\lambda} = [1, 1]$, $\hat{\lambda} = [4, 4]$ using Algorithm 8 (where Algorithm 3 (TDLP) is involved in <i>Step 3</i>) and the involved Problems $(P(t_{k+1}, \bar{\lambda}, \hat{\lambda}))$. The element $(f_1(x^0), f_2(x^0))$ (in red) is the approximate solution of (P^τ) with $\tau = 0.008\ A^T b\ _\infty$ generated in Section 6.1 using Algorithm 3.	113

List of Tables

5.1	The initial points used for Algorithm 1, 2, 3, 4 and 5	82
6.1	Twenty experimental results for sparse signal recovery.	95
6.2	Approximate solutions $\bar{x} := x_k \in \mathbb{R}^n$ for the sparse signal recovery experiments of Algorithm 1 (MDY1), Algorithm 2 (HCDLS) and Algorithm 3 (TDLP).	96
6.3	Numerical results of TDLP, HCDLS and MDY2 methods in image restorations.	98

Chapter 1

Introduction

Mathematics as an important tool for understanding diverse areas of applications in sciences and engineering as well as social sciences, is worth devoting ample time and energy to study it more. In recent years, different areas of interesting and fascinating real-world applications in science and engineering, medicine, social sciences, and finance have continued to emerge. It is therefore of utmost importance to develop algorithms to solve these mathematical problems. The algorithms developed in this thesis are very important for solving real world problems as described in the following.

Problem P1: Compressive Sensing

The study of Compressive Sensing (CS) usually comes up in electrical engineering, computer science, and applied mathematics. The term *compressive sensing* was coined based on the idea that data acquisition and compression can be performed simultaneously.

It has been noted in the literature that different types of signals are sparse. This suggests that a small number of non-zero coefficients can be optimally approximated on an acceptable basis. This is why many signals are compressible and compression algorithms like the Joint Photographic Experts Group (JPEG), Moving Picture Experts Group (MPEG), or MP3 work well in practice. JPEG, for example, takes advantage of the fact that images are typically sparse in the Discrete Cosine Basis (DCT) or wavelet basis and compresses them by merely storing the greatest DCT coefficients. When decompressing an image, all non-stored coefficients are simply set to zero. Figure 1.1 illustrates how natural images are sparse in the wavelet domain. The goal is usually to recreate these types of vectors using incomplete linear information. This results in an under-determined linear system. In general, an under-determined linear system can have an infinite number of solutions. However, by including more sparsity information, it is possible to identify the correct solution under certain conditions.

The CS are widely used in the medical sciences, biological engineering and other fields of science and engineering [21, 36]. The CS problem involves recovering a sparse signal, x , from the linear system

$$Ax = b,$$

where the vector $b \in \mathbb{R}^k$ represents the observations, while the sensing matrix is represented by the linear operator $A \in \mathbb{R}^{k \times n}$ ($k \ll n$). In an attempt to search for the sparsest signal among the solution set,

the above system of linear equations is converted into the following so-called " ℓ_0 -norm" optimization problem with constraints:

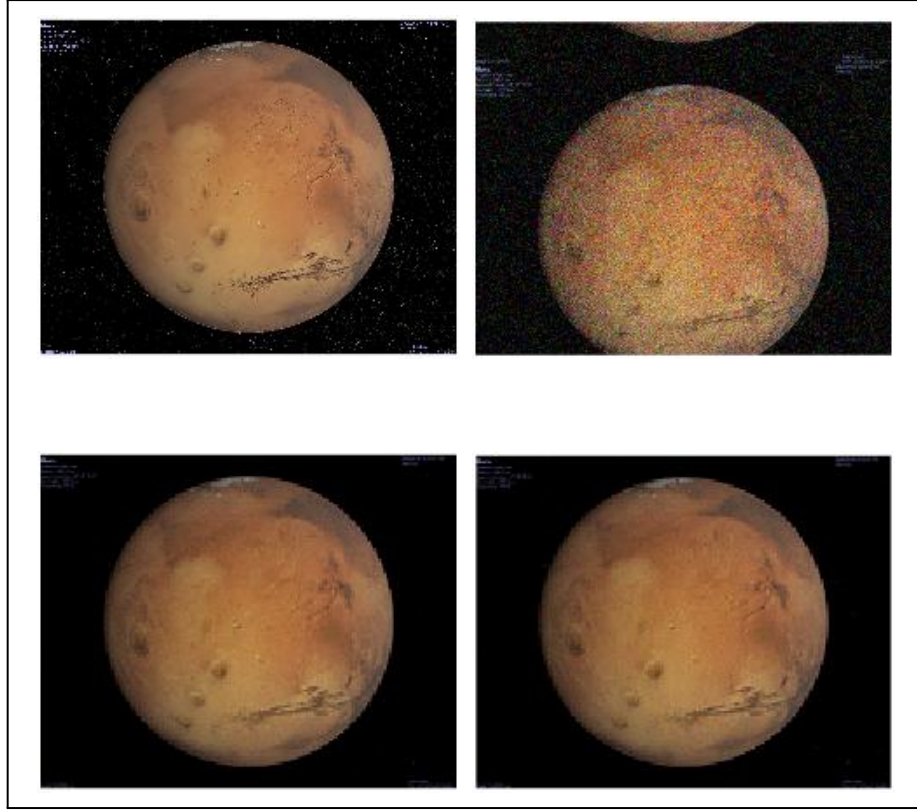


Figure 1.1: The original Mars image (top left), the blurred image (top right), the restored image by MDY1 (bottom left) and by MDY2 (bottom right)

$$\min_x \{\|x\|_0 : Ax = b\}, \quad (1.1)$$

where $\|x\|_0$ is given by the number of non-zero components of a vector $x \in \mathbb{R}^n$ and is called ℓ_0 -norm. Unfortunately, $\|\cdot\|_0$ does not satisfy the conditions of being a norm or a positively homogeneous set. However, due to the difficulty associated with $\|\cdot\|_0$, an alternative approach is to replace it with the ℓ_1 -norm (see [38]), that is,

$$\min_x \{\|x\|_1 : Ax = b\}, \quad (1.2)$$

where $\|x\|_1 := \sum_{i=1}^n |x_i|$. Under some mild assumptions, problem (1.2) has been shown to produce the desired results with some acceptable degrees of accuracy. When the measurements are affected by some noise, the constraint in (1.2) is usually relaxed to the following regularized least squares problem (in the literature [31, 64, 78] also named Lasso problem):

$$\min_x \tau \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2, \quad (1.3)$$

where τ is a positive regularization parameter, $\|\cdot\|_2$ represents the Euclidean norm of \mathbb{R}^n . We refer the reader to Chapter 2, Section 2.3 for the reformulation of ℓ_1 -norm regularization in compressive sensing.

Some algorithms for signal reconstruction have been developed over the years. However, researchers continue to propose algorithms designed to operate with a limited number of measurements. As a result, there is an interesting and important area of research dedicated to developing optimal explicit measurement matrices and all known "good" matrix constructions that involve randomness. This discovery has a lot of potential applications in signal and image processing.

Problem P2: Robotic Motion Control Problem

On the other hand, the study of robotics is an important aspect that takes engineering toward synthesis. This makes this field so interesting to investigate. This area of study focuses on the goal of synthesizing some difficult aspects of human function through the use of mechanisms, computers, actuators, and sensors.

There are basically two major reasons for the growth in the number of robots used in industries. One is cost effectiveness, that is, working with robots has become relatively cheaper compared to the cost of human labor. The second reason is the ability of the robot to effectively carry out tasks that might be dangerous or even impossible for human workers to perform.

In general, the study of the mechanics and control of manipulators is not a new science, but rather a collection of topics derived from "classical" sciences. Mechanical engineering provides strategies for studying machines in both static and dynamic conditions. Mathematics provides the necessary tools for characterizing spatial motions and other properties of manipulators. Control theory provides tools for creating and evaluating algorithms that achieve the desired motions or force applications. Electrical engineering skills are used to construct sensors and interfaces for industrial robots, while computer science provides a foundation for programming these devices to accomplish a particular task, see, for example, the book by John [97].

The location of objects in a three-dimensional space is of great interest in robotics research. The objects involved in the manipulator's operations, such as its linkages, parts and tools, are characterized by two attributes, namely, location and orientation. The mathematical representation and manipulation of these quantities is the point of interest in our study.

In what follows, we give a brief introduction to the following useful terminologies:

A Description of position and orientation

The positioning of objects in a three-dimensional space is an ongoing problem in robotics research. These objects include the manipulator's connections, the components and equipment it uses, and other items located in its surroundings. We can summarize these elements into two fundamental yet crucial characteristics: position and orientation. Of course, one area of urgent concern is how we express and manipulate these quantities mathematically.

Every time we describe a body's orientation and position in space, we firmly attach a coordinate system or frame to it. The frame's location and orientation relative to a reference coordinate system is then stated. We frequently consider converting or changing the description of a body's location and orientation from one frame to another since any frame might serve as a reference system for representing these features.

Forward kinematics of manipulators

Kinematics is the study of motion, regardless of the forces that cause it. The area of kinematics science focuses on the study of position, velocity, acceleration, and other higher-order derivatives. Position variables relate to time or other factors. Therefore, the study of manipulator kinematics encompasses all geometrical and time-based aspects of motion.

Virtually rigid links, connected by joints that allow neighboring links to move relative to one another, make up manipulators. These joints generally include position sensors, which allow for monitoring the relative positions of adjacent links. These displacements are known as joint angles in rotary or revolute joints. Some manipulators include sliding (or prismatic) joints. These joints translate the relative displacement between links, also referred to as the joint offset.

The manipulator's number of degrees of freedom refers to the number of independent position variables required to locate each component of the mechanism. This broad term applies to any mechanism. Despite having three movable components, a four-bar connection has just one degree of freedom. Industrial robots have the same number of joints as degrees of freedom because their manipulators are open kinematic chains with a single variable describing each joint position.

The end-effector is located at the free end of the manipulator's link chain. The robot's end-effector could be a gripper, an electromagnet, a welding torch, or another device, depending on its intended purpose. To define the position of the manipulator, we frequently establish the relationship between the base frame, which is coupled to the manipulator's stationary base, and the tool frame, which is attached to the end-effector.

Forward kinematics is a fundamental issue in mechanical manipulation studies. This is the static geometrical problem of determining the manipulator's end-effector's location and orientation. The forward kinematic problem specifically aims to determine the orientation and location of the tool frame with respect to the base frame given a set of joint angles. You can also conceptualize this as transforming the joint space description of the manipulator position into a Cartesian space description.

According to [176], the discrete-time kinematics equation for a two-jointed planar robot manipulator at the position level is given as

$$f(\boldsymbol{\theta}_k) = \mathbf{q}_k. \quad (1.4)$$

The vectors $\boldsymbol{\theta}_k \in \mathbb{R}^2$ and $\mathbf{q}_k \in \mathbb{R}^2$ represent the joint angle and the end effector location, respectively. The function $f(\cdot)$ is a kinematics mapping that maps the position and orientation of a robot's end effector, say, $\mathbf{q}_k = (x_e, y_e)^T$ with the following known structure

$$f(\boldsymbol{\theta}_k) = \begin{bmatrix} \ell_1 \cos(\theta_1) + \ell_2 \cos(\theta_1 + \theta_2) \\ \ell_1 \sin(\theta_1) + \ell_2 \sin(\theta_1 + \theta_2) \end{bmatrix}, \quad (1.5)$$

where ℓ_i ($i = 1, 2$) is the length of the i^{th} rod. In view of robotic control, we need to solve the following minimization problem

$$\min_{\mathbf{q}_k \in \mathbb{R}^2} f(\mathbf{q}_k), \quad \text{where } f(\mathbf{q}_k) = \frac{1}{2} \|\mathbf{q}_k - \mathbf{q}_{dk}\|^2, \quad (1.6)$$

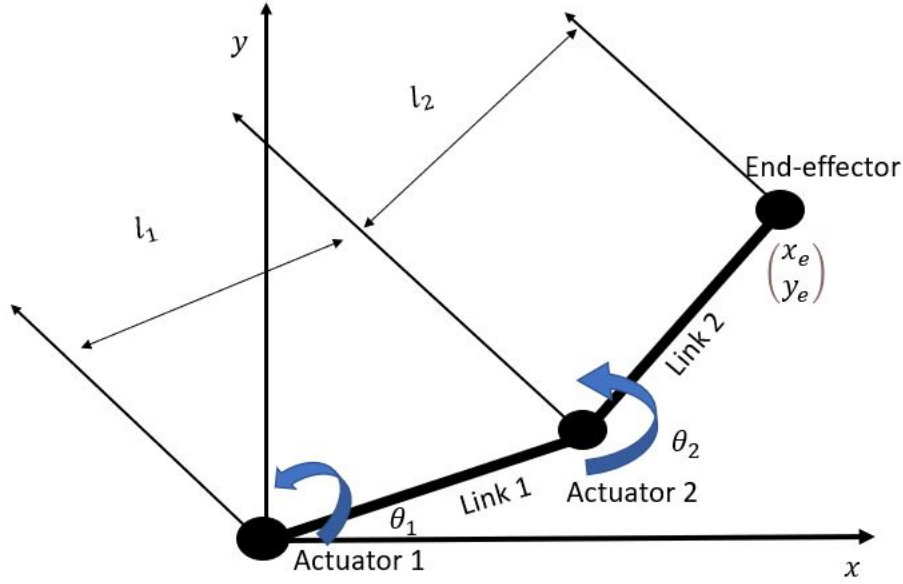


Figure 1.2: A 2-dimensional planar robot arm.

q_{dk} is the end-effector control track at each computational time intervals $t_k \in [0, t_f]$, where t_f is the end of task duration. The end-effector is controlled to track a Lissajous curve, expressed as

$$q_{dk} = \begin{bmatrix} \frac{3}{2} + \frac{1}{5} \sin\left(\frac{\pi t_k}{5}\right) \\ \frac{\sqrt{3}}{2} + \frac{1}{5} \sin\left(\frac{2\pi t_k}{5} + \left(\frac{\pi}{3}\right)\right) \end{bmatrix}, \quad (1.7)$$

see the geometrical representation of a 2-dimensional robot planar arm in Figure 1.2. For further information about robotic motion control, we refer the reader to Chapter 6, specifically, Section 6.2.

Like the specific Problems **P1** and **P2** mentioned above, there are numerous real-world problems that arise in engineering, biology, economics, mathematical finance, machine learning, statistics, physics, and other branches of science that usually lead to nonlinear optimization problems of the form

$$\min\{f(x) : x \in \mathbb{R}^n\}, \quad (1.8)$$

with the continuously differentiable cost function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ bounded from below. Nonlinear optimization problems are of significant interest from a theoretical as well as practical point of view. In general, one of the greatest difficulties associated with solution methods for nonlinear problems, of which (1.8) is a specific example, is the inability to always obtain exact solutions analytically, meaning that the existence of solutions or their well-posedness and compactness may not always be guaranteed. As a result, people turn to iterative techniques as a substitute for obtaining their approximate solutions.

An iterative method is a mathematical procedure that uses an initial estimate, say x_0 , of the solution, say x^* , to generate a sequence of approximations, say x_1, x_2, \dots , that converges in the vicinity of the solution x^* . The main objective of an iterative technique is to create methods that converge to the approximate solution from the given initial estimate as quickly as possible. Iterative methods are often useful for problems with a large number of variables in which analytical methods would be prohibitively expensive.

Iterative methods such as Newton methods, quasi-Newton methods, conjugate gradient methods, spectral gradient methods, and so on, have been widely employed to solve the general unconstrained optimization Problem (1.8). One of the efficient approaches usually employed to solve (1.8) is the use of *line search method*; see, for example, [57, 90, 129]. Given a suitable starting point $x_0 \in \mathbb{R}^n$, such methods utilize the following iterative procedure:

$$x_{k+1} := x_k + \alpha_k d_k, \quad k \geq 0. \quad (1.9)$$

The scalar $\alpha_k > 0$ represents the *step-size or step-length*, at iteration k , usually computed with the aid of suitable line search techniques. The vector $d_k \neq 0$ represents the *search direction*, determined using various methods (namely, Newton methods, quasi-Newton methods, spectral gradient methods, conjugate gradient methods and so on).

It is worth noting that under the differentiability assumption, a well-known necessary optimality condition for Problem (1.8) is:

$$x^* \text{ is a minimizer of } f \implies \nabla f(x^*) = 0.$$

This fact is called the first-order necessary optimality condition for a minimizer x^* of Problem (1.8). Now, we consider the following system of nonlinear equations with constraint $x \in \Psi \subseteq \mathbb{R}^n$ such that

$$F(x) = 0. \quad (1.10)$$

The gradient of f in Problem (1.8) is viewed as the function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ in Problem (1.10), that is, $F := \nabla f$. Note that the objective function in Problem (1.8) is a real-valued function, but its gradient is a vector-valued. As a result, algorithms for solving the general unconstrained optimization Problem (1.8) can be adapted to handle the nonlinear monotone operator Problem (1.10).

The main goal of the line-search is to produce step-lengths that will guarantee $\{\|F(x_k)\|\}$ is nonincreasing in every successive iterations. On the other hand, the search direction $d_k \in \mathbb{R}^n$ is a crucial part of the recursive formula (1.9) and can be computed by different methods. For instance, the classical choice for the search direction $d_k := -F(x_k)$ for all k is the most popular choice because it mimics the behaviour of the well-known *steepest descent method for solving general unconstrained optimization problem*; see [24, 33, 57, 90, 138] for further details.

The following are different approaches for determining the search direction d_k in (1.9) which lead to different iterative methods:

- Newton methods: $d_0 = -F(x_0)$ and $d_k = -J(x_k)^{-1}F(x_k)$, for $k \geq 1$, $J(x_k) = \nabla F(x_k)$ is the exact Jacobian matrix, see, for examples, [26, 57, 90, 99, 100, 101, 102, 121, 128] and the references therein,
- Quasi-Newton methods: $d_0 = -F(x_0)$ and $d_k = -A(x_k)^{-1}F(x_k)$, for $k \geq 1$, $A(x_k)$ is an approximation of the Jacobian matrix, see, for examples, [124, 134, 151] with further references therein,
- Spectral gradient methods: $d_0 = -F(x_0)$ and $d_k = -\lambda_k F(x_k)$, for $k \geq 1$, $\lambda_k > 0$ is the spectral gradient parameter, see for examples [29, 30, 125, 141],
- Conjugate gradient methods: $d_0 = -F(x_0)$ and $d_k = -F(x_k) + \beta_k d_{k-1}$, for $k \geq 1$, $\beta_k > 0$ is the conjugate gradient parameter, see [1, 46, 59, 61, 82, 115, 135, 137] and the references therein,

- Spectral conjugate gradient methods: $d_0 = -F(x_0)$ and $d_k = -\lambda_k F(x_k) + \beta_k d_{k-1}$, for $k \geq 1$, $\lambda_k > 0$ and $\beta_k > 0$ are the spectral and conjugate gradient parameters, see [14, 28, 113, 123],

For more details about the aforementioned methods for finding the search directions d_k , we refer the reader to Chapter 2, Section 2.5.

In this thesis, we consider Problem (1.10), where the point x is constrained within the set $\Psi \subseteq \mathbb{R}^n$. Specifically, we are looking for a vector $x \in \Psi \subseteq \mathbb{R}^n$ for which (1.10) holds. The feasible set Ψ is assumed to be nonempty, closed, and convex. Thus, we refer to Problem (1.10) as a system of nonlinear monotone operator equations which has numerous practical applications across different fields. Many mathematical problems arising from many applications, such as fixed point problems, differential equations, variational inequality problems, and so on, can be reformulated into Problem (1.10) (see [104, 116, 154, 155]). Furthermore, Problem (1.10) also appears as a sub-problem in generalized proximal algorithms that use Bregman's distances [89]. Optimization problems containing least square errors and ℓ_1 -norm regularization problems arising in compressive sensing can equally be translated into Problem (1.10) [75, 167]. These applications underline the importance of Problem (1.10) and also the importance of efficient algorithms with minimal computational costs for solving it.

The iterative technique (1.9) can be modified by incorporating two starting points, say x_0 and x_{-1} . The technique is known as *Inertial-type algorithm*, which defines its recursive formula as

$$w_k := x_k + \alpha_k(x_k - x_{k-1}), \quad k \geq 0, \quad \alpha > 0. \quad (1.11)$$

Inertial-type algorithms are iterative algorithms that use an inertial step (1.11). These methods are based on the heavy-ball approach pertains to the second-order-in-time dissipative dynamical system. In 1964, Polyak [136] began investigating inertial extrapolation as a speed-up approach for solving smooth convex minimization problems. Inertial-type methods are two-step iterative schemes in which the next iterate is determined by using the previous two iterates [25]. In order to speed up the iteration process, an inertial extrapolation term is required to accelerate the iterative sequence. Inertial-type methods speed up the iterative process to get the desired output. There is a growing interest in inertial-type algorithms for optimization, variational inequalities, and monotone inclusions, as evidenced in the references [17, 50, 143, 144, 146, 162]. Several studies have shown that iterative algorithms for solving the aforementioned nonlinear problems with an inertial step perform better numerically in terms of the number of iteration and time of execution compared to their counterparts without the inertial step. These two key advantages attract the researcher's interest in developing novel inertial-like methods.

Given a starting point, say x_0 , a classical iterative algorithms (such as Newton's method and its variants as well as quasi-Newton methods) use the formula (1.9) to update their sequence of iterates. The search direction d_k in (1.9) is updated based on x_k and its preceding point x_{k-1} as well as their images, that is $F(x_k)$ and $F(x_{k-1})$ (see, for example, [1, 73, 167, 168]). Incorporating the inertial step (1.11) into algorithms for solving variational inequalities and split feasibility problems improves their numerical performance. This brings up the following question: Does incorporating an inertial effect into a search direction improve the numerical performance of conjugate gradient-like algorithms? This question is answered in Chapter 5 (Sections 5.2.1 and 5.2.2, respectively).

The main new results in this thesis are listed below:

- Many algorithms have been presented and demonstrated to be efficient for solving convex constrained monotone operator equations of the type (1.10) in the literature. However, some of these iterative methods may not be well-defined and do not satisfy the descent condition, which is very critical to establishing global convergence results. As a result, we propose new derivative-free like iterative algorithms by modifying the search directions (or, in some cases, proposing new ones) and introducing new parameters, allowing us to get well-defined algorithms that satisfy the descent condition while also having global convergence.
- We establish global convergence of the new algorithms under weaker assumptions, as seen in the literature (for example, [1, 45, 46, 59, 114, 115, 123, 124, 175] and the references therein), particularly under pseudomonotonicity assumptions and other classical assumptions.
- In the numerical experiments, we demonstrate that the new algorithms outperform established methods in the literature [16, 178].
- We incorporate new inertial-like conjugate and spectral gradient algorithms that are well-defined and satisfy the descent conditions. The obtained numerical performance of the new algorithms are better than the state-of-the-art of similar existing methods; see, for example, [17, 125].
- Furthermore, we study several applications of our new methods, such as:
 - The proposed methods aim to minimize a non-smooth minimization problem that includes a least-squares data fitting term and a ℓ_1 -norm regularization term. Therefore, we tested our new algorithms for solving the ℓ_1 -norm regularization problem in compressive sensing to recover sparse signals and blurred images.
 - In addition, we apply the inertial-like conjugate and spectral gradient algorithms for solving motion control problems.
 - Finally, we apply our methods for solving vector-valued approximation problems using a suitable interactive procedure based on scalarization.

Structure of the Thesis

This thesis is structured as follows:

Chapter 2 provides the required mathematical background. For this reason, essential results from the fields of linear algebra, linear and non-linear functional analysis, properties of operators and functions as well as some basic notions from the field of multi-objective optimization will be reviewed. This chapter also provides a reformulation for ℓ_1 -norm regularization in compressive sensing. An overview of solution methods for unconstrained optimization problems will also be provided.

In Chapter 3, we establish the main results concerning the derivative-free algorithms for nonlinear systems of equations. Motivated by the results from the literature and Section 2.5, we introduce three main iterative algorithms for solving Problem (1.10), namely:

- Algorithm 1 (MDY): Modified Dai–Yuan [46] projection methods for solving nonlinear monotone operator Problem (1.10)
- Algorithm 2 (HCDLS): Hybrid conjugate gradient algorithm with spectral parameters as a convex combination of the Conjugate Descent (CD) proposed by Fletcher in [59] and the method by Liu–Storey (LS) derived in [115] for solving nonlinear monotone operator Problem (1.10)
- Algorithm 3 (TDLP): Three–term Dai–Liao [45] Projection method for solving nonlinear monotone operator Problem (1.10).

We demonstrate that the descent condition of Algorithms 1, 2 and 3 is satisfied. Additionally, the convergence analyses for the proposed new algorithms are provided.

Chapter 4 presents the second part of our main results, which involves two-step projection methods. This chapter proposes a two-step inertial-like projection approach based on Solodov and Svaiter’s hyperplane projection [150] and inertial-like algorithms. The algorithm uses two starting points, x_{-1} and x_0 , to update the sequence of iterates. In this chapter, the search direction d_k is updated using x_k and its preceding point x_{k-1} as well as their images $F(x_k)$ and $F(x_{k-1})$. This differs from the classical iterative algorithms given in Chapter 3 that employ $x_{k+1} := x_k + \alpha_k d_k$, $k \geq 0$, $\alpha_k > 0$ to update the sequence of iterates. A two-step hybrid algorithm based on the Barzilai and Borwein (BB) [22] spectral parameters is proposed to solve a nonlinear monotone operator equation (1.10). These BB spectral parameters can be considered as Jacobian approximations using scalar multiple identity matrices. Some important convergence results for these newly developed algorithms are also presented.

The computational performances of the theoretical results described in Chapters 3 and 4 have received a lot of attention in Chapter 5, which compares solutions to state-of-the-art large-scale convex constrained optimization problems. Plenty of Dolan and Morè [49] performance profiles and Morè and Wild [122] data profiles, to demonstrate the behaviour of the proposed algorithms are established.

In Chapter 6, practical applications of Algorithms 1, 2 and 3 (introduced in Chapter 3) for sparse signal reconstruction and image restoration in compressing sensing are described. The chapter applies Algorithms 4 and 5 from Chapter 4 to motion control problems with a two-joint planar robotic manipulator.

In Chapter 7, we study applications for solving vector-valued approximation problems. A vector-valued approximation problem is formulated. We investigate its solution concepts, specifically the so-called *Pareto Optimality*. Furthermore, we study the weighted (surrogate) sum scalarization approach and give the formulation of the multiobjective approximation problem. In addition, we study the application of Algorithm 3 for generating approximate solutions to multiobjective approximation problems.

Chapter 2

Mathematical Background

In this chapter, we introduce some preliminary concepts that will be used throughout the thesis. First, we review some well-known fundamental concepts of functional analysis such as linear spaces, monotone operators, normed spaces and Lipschitz continuity. Section 2.4 covers various concepts of line searches, while Section 2.5 provides an overview of methods for solving unconstrained optimization problems such as quasi-Newton methods, conjugate gradient methods, spectral gradient methods and spectral-conjugate gradient methods.

2.1 Fundamentals of Functional Analysis

Here, we begin by discussing some fundamental concepts such as linear spaces, monotone operators, metric spaces, normed spaces and Lipschitz continuity. Throughout this thesis, \mathbb{N} , \mathbb{Z} and \mathbb{R} represent the set of natural numbers, integers and real numbers, respectively. In addition, we denote the set of all nonnegative real numbers by \mathbb{R}_+ , which is defined as $\mathbb{R}_+ := \{x \in \mathbb{R} : x \geq 0\}$. Furthermore, we define $\mathbb{R}_+^n := \{y = (y_1, y_2, \dots, y_n) \in \mathbb{R}^n : y_i \geq 0, i = 1, 2, \dots, n\}$ and the n -dimensional Euclidean space is denoted by \mathbb{R}^n .

2.1.1 Linear Spaces and Order Structure

In this section, we provide the following definitions which are taken from [93, 159]. The properties of linear spaces are essential in deriving and analysing our proposed algorithms throughout Chapters 3 and 4 of this thesis. Furthermore, we will make use of the ordering cone in Chapter 7 for the formulation of the solution concepts of vector-valued approximation problems.

Definition 2.1.1. ([159]). A *(real) linear space* X is a nonempty set which is equipped with a mapping $+$: $X \times X \rightarrow X$, called **addition**, and with a mapping \cdot : $\mathbb{R} \times X \rightarrow X$, called **multiplication by scalars**, satisfying the following conditions:

- (i) $\forall x, y, z \in X : (x + y) + z = x + (y + z)$,
- (ii) $\forall x, y \in X : x + y = y + x$,
- (iii) $\exists 0 \in X \forall x \in X : x + 0 = x$,

$$(iv) \forall x \in X \exists x' \in X : x + x' = 0,$$

$$(v) \forall x, y \in X \forall \lambda \in \mathbb{R} : \lambda \cdot (x + y) = \lambda \cdot x + \lambda \cdot y,$$

$$(vi) \forall x \in X \forall \lambda, \mu \in \mathbb{R} : (\lambda + \mu) \cdot x = \lambda \cdot x + \mu \cdot x,$$

$$(vii) \forall x \in X \forall \lambda, \mu \in \mathbb{R} : \lambda \cdot (\mu \cdot x) = (\lambda \cdot \mu) \cdot x,$$

$$(viii) \forall x \in X : 1 \cdot x = x.$$

This space, denoted as $(X, +, \cdot)$, called a **(real) vector space** or a **set equipped with a linear structure**.

The elements of a vector space are called **vectors** or **points**. The vector 0 in condition (iii) is known as the **zero vector** of X and is usually denoted as 0_X . The vector x' in condition (iv) is the **additive inverse** of x , denoted by $-x$. We will write λx instead of $\lambda \cdot x$, for $\lambda \in \mathbb{R}$ and $x \in X$.

A linear space X is said to be **nontrivial** if $X \neq \{0\}$.

We will consider *real* linear spaces only and omit the word real later on.

Clearly, \mathbb{R}^ℓ with $\ell \in \mathbb{N}_{>}$ is a linear space when defining

$$\begin{aligned} x + y &:= (x_1 + y_1, \dots, x_\ell + y_\ell)^T \\ \lambda x &:= (\lambda x_1, \dots, \lambda x_\ell)^T \end{aligned}$$

for $x = (x_1, \dots, x_\ell)^T \in \mathbb{R}^\ell$, $y = (y_1, \dots, y_\ell)^T \in \mathbb{R}^\ell$ and $\lambda \in \mathbb{R}$.

In linear spaces F consisting of extended real-valued functions on some space X or of functions mapping X to another linear space Y , addition and multiplication by scalars are defined for all $f, g \in F$ and $\lambda \in \mathbb{R}$

$$\begin{aligned} (f + g)(x) &:= f(x) + g(x) && \text{for each } x \in X \\ (\lambda f)(x) &:= \lambda f(x) && \text{for each } x \in X \end{aligned}$$

Throughout this thesis, we will use the following notation. Assume that A, B are nonempty subsets of a linear space X , that Λ is a nonempty subset of \mathbb{R} , $x \in X$, and $\gamma \in \mathbb{R}$. Then

$$\begin{aligned} A + B &:= \{x + y \mid x \in A, y \in B\}, \quad A + \emptyset := \emptyset + A := \emptyset, \\ x + A &:= A + x := \{x\} + A, \\ \Lambda A &:= \Lambda \cdot A := \{\lambda x \mid \lambda \in \Lambda, x \in A\}, \quad \Lambda \cdot \emptyset := \emptyset, \\ \gamma A &:= \gamma \cdot A := \{\gamma\} \cdot A, \quad -A := (-1) \cdot A, \\ A - B &:= A + (-B) = \{x - y \mid x \in A, y \in B\}. \end{aligned}$$

Ordering Cones

Here, we will state some basic notions of cones of a vector space Y which we are going to use later in Chapter 7 when dealing with vector-valued approximation problems of this thesis. These concepts can be found, for example, in [68, 96, 98, 147]. The cones induce a class of binary relations, which are compatible with the linear structure of Y .

Definition 2.1.2. A nonempty set $K \subseteq Y$ is said to be a cone if $tx \in K$ for every $x \in K$ and every $t \geq 0$.

The cone K is called:

- (i) **convex** if $K + K \subseteq K$,
- (ii) **proper** if $K \neq \{0_Y\}$ and $K \neq Y$,
- (iii) **reproducing** if $K - K = Y$,
- (iv) **pointed** if $K \cap (-K) = \{0_Y\}$.

Obviously, if K is a cone, then $0_Y \in K$.

In what follows, we give some examples of a cone.

Example 2.1.3. ([159], Example 2.2.13).

1. The **nonnegative orthant** of the p -dimensional Euclidean space is given by

$$\mathbb{R}_+^p := \{(x_1, \dots, x_p)^T \in \mathbb{R}^p \mid \forall i \in \{1, \dots, p\} : x_i \geq 0\}.$$

This is a nontrivial pointed convex reproducing cone, which is the usual ordering cone. The interior of the Euclidean space \mathbb{R}_+^p is defined as

$$\text{int } \mathbb{R}_+^p := \{(x_1, \dots, x_p)^T \in \mathbb{R}^p \mid \forall i \in \{1, \dots, p\} : x_i > 0\}.$$

2. ([93]). For $X = \mathbb{R}^p$, the ordering cone of the component-wise partial ordering on \mathbb{R}^p is given by

$$K := \{x \in \mathbb{R}^p \mid \forall i \in \{1, \dots, p\} : x_i \geq 0\} = \mathbb{R}_+^p.$$

It is also called the natural ordering cone. Other ordering cones in \mathbb{R}^p are for instance

$$\{x \in \mathbb{R}^p \mid x_i \geq 0 \text{ for all } i \in \{1, \dots, p\} \text{ and } x_i = 0 \text{ for all } i \in \{m+1, \dots, p\} \text{ for some } 1 \leq m < p$$

or $\{0_{\mathbb{R}^p}\}$ and \mathbb{R}_p itself. \mathbb{R}_+ , \mathbb{R}_- , $\{0\}$ and \mathbb{R} are the only ordering cones in \mathbb{R} .

3. Another nontrivial pointed convex reproducing cone in \mathbb{R}^p , $p > 1$, is given by

$$C := \left(\bigcup_{i=1}^p C^i \right) \cup \{0\},$$

where $C^i := \{(x_1, \dots, x_p)^T \in \mathbb{R}^p \mid x_i > 0, x_j = 0 \text{ for all } j < i\}$.

4. For the space of continuous functional $C[a, b]$,

$$C[a, b]_+ := \{x \in C[a, b] \mid \forall t \in [a, b] : x(t) \geq 0\}$$

is a nontrivial pointed convex reproducing cone with a nonempty interior with respect to the topology generated by the supremum norm $\|\cdot\|_\infty$.

5. Let $C^1[a, b]$ be the real vector space formed for all continuously differentiable real-valued functions defined on the interval $[a, b] \subset \mathbb{R}$ and equipped with the norm

$$\|f\|_1 := \left(\int_a^b (f(t))^2 dt + \int_a^b (f'(t))^2 dt \right)^{\frac{1}{2}}$$

for any $f \in C^1[a, b]$. The natural ordering cone

$$C^1[a, b]_+ := \{f \in C^1[a, b] \mid f \geq 0\}$$

has nonempty interior.

Pre-order Structure

Next, we consider order relations with respect to (w.r.t.) a given convex cone K in a linear space Y between two vectors and two nonempty sets. The order structure (in particular partial order) plays an important role for introducing solution concepts in vector-valued approximation problems (see, for example, Chapter 7 of this thesis). In that chapter, the sets of maximal or minimal elements are denoted by $\text{Eff}_{\text{Max}}(M_0, \mathfrak{R})$ and $\text{Eff}_{\text{Min}}(M_0, \mathfrak{R})$, respectively. Furthermore, these sets are known as the (maximal- or minimal-) **efficient points** of M_0 with respect to \mathfrak{R} . We may use "Max" or "Min" instead of "maximum" or "minimal" for clarity.

Definition 2.1.4. Assuming M is a nonempty set, $M \times M$ refers to the set of ordered pairs of elements of M , defined as

$$M \times M := \{(x, y) \mid x, y \in M\}.$$

If \mathfrak{R} is a nonempty subset of $M \times M$, it is referred to be a **binary relation** on M . The notation $x\mathfrak{R}y$ represents $(x, y) \in \mathfrak{R}$. The pair (M, \mathfrak{R}) is known as a set M with binary relation \mathfrak{R} . Two elements $x, y \in M$ are said to be **comparable** if $x\mathfrak{R}y$ or $y\mathfrak{R}x$ holds.

The **binary relation** \mathfrak{R} is called:

- (i) **reflexive** if $x\mathfrak{R}x$ for every $x \in M$;
- (ii) **transitive** if for all $x, y, z \in M$: $x\mathfrak{R}y$ and $y\mathfrak{R}z$ imply that $x\mathfrak{R}z$;
- (iii) **symmetric** if for all $x, y \in M$: $x\mathfrak{R}y$ imply that $y\mathfrak{R}x$;
- (iv) **antisymmetric** if for all $x, y \in M$: $x\mathfrak{R}y$ and $y\mathfrak{R}x$ imply that $x = y$;
- (v) **complete** if any two elements of M are comparable.
- (vi) a **preorder** if \mathfrak{R} is reflexive and transitive;
- (vii) a **partial order** if \mathfrak{R} is reflexive, transitive and antisymmetric;
- (viii) an **equivalent relation** if \mathfrak{R} is reflexive, symmetric and transitive.

We will now provide the following examples of order relation.

Example 2.1.5.

1. ([68]). Let X be a nonempty set and $M := P(X)$ denote the class of subsets of X . The binary relation $R := \{(A, B) \in M \times M \mid A \subset B\}$ is a partial order on M . If X contains at least two elements, then \mathfrak{R} is not a linear order.

2. ([68]). Let \mathbb{N} be the set of nonnegative integers and

$$\mathfrak{R}_{\mathbb{N}} := \{(n_1, n_2) \in \mathbb{N} \times \mathbb{N} \mid \exists p \in \mathbb{N} : n_2 = n_1 + p\}.$$

Then \mathbb{N} is well-ordered by $\mathfrak{R}_{\mathbb{N}}$. Of course, $\mathfrak{R}_{\mathbb{N}}$ defines the usual order relation on \mathbb{N} and $n_1 \mathfrak{R}_{\mathbb{N}} n_2$ will always be denoted by $n_1 \leq n_2$ or equivalently, $n_2 \geq n_1$.

3. The **nonnegative orthant** of the p -dimensional Euclidean space is given by

$$\mathbb{R}_+^p := \{(x_1, \dots, x_p)^T \in \mathbb{R}^p \mid \forall i \in \{1, \dots, p\} : x_i \geq 0\}.$$

This is a nontrivial pointed convex reproducing cone, which is the usual ordering cone. The interior of the Euclidean space \mathbb{R}_+^p is defined as

$$\text{int } \mathbb{R}_+^p := \{(x_1, \dots, x_p)^T \in \mathbb{R}^p \mid \forall i \in \{1, \dots, p\} : x_i > 0\}.$$

For $K := \mathbb{R}_+^p$, the relation \mathfrak{R}_K is the relation " \leq ". A base of K is given by

$\left\{ (x_1, \dots, x_p)^T \in \mathbb{R}_+^p \mid \sum_{i=1}^n x_i = 1 \right\}$. For $K := \mathbf{I} \in \mathbb{R}^p$, the norm $\|\cdot\|_{K,k}$ becomes the maximum norm $\|\cdot\|_{\infty}$.

4. ([159]). The usual partial order defined on \mathbb{R}^p is

$$\mathfrak{R}_p := \{(x, y) \in \mathbb{R}^p \times \mathbb{R}^p \mid \forall i \in \{1, \dots, p\} : x_i \leq y_i\},$$

where $x = (x_1, \dots, x_p)^T$ and $y = (y_1, \dots, y_p)^T$. It is called a linear order if and only if $n = 1$. On \mathbb{R}^p , we will denote R_p by \leq and R_p^{-1} by \geq . Furthermore, we consider on \mathbb{R}^p the asymmetric transitive relation

$$\mathfrak{R}_{<} := \{(x, y) \in \mathbb{R}^p \times \mathbb{R}^p \mid \forall i \in \{1, \dots, p\} : x_i < y_i\},$$

we denote $R_{<}$ by $<$ and $R_{<}^{-1}$ by $>$.

Definition 2.1.6. Let \mathfrak{R} be a binary relation on a nonempty set M and assume $M_0 \subseteq M$. An element $\bar{x} \in M_0$ is called a **maximal** or a **minimal** element of M_0 w.r.t. \mathfrak{R} if for every $x \in M_0$:

$$\bar{x} \mathfrak{R} x \implies x \mathfrak{R} \bar{x} \quad \text{or}$$

$$x \mathfrak{R} \bar{x} \implies \bar{x} \mathfrak{R} x, \text{ respectively.}$$

$\text{Max}(M_0, \mathfrak{R})$ represents the set of all maximal elements of M_0 , while $\text{Min}(M_0, \mathfrak{R})$ represents the set of all minimal elements of M_0 .

If \mathfrak{R} is a partial order on M , a subset $M_0 \subseteq M$ may have none, one or several minimal (maximal) elements.

Definition 2.1.7. Let \mathfrak{R} be a binary relation on a nonempty set M with M_0 a subset of M . M_0 is said to be **bounded below** (or **above**) with regard to \mathfrak{R} if there is some $a \in M$ such that $a\mathfrak{R}x$ ($x\mathfrak{R}a$, correspondingly) for all $x \in M_0$. In this case, the element a is referred to as the **lower** (or **upper**) bound of M_0 .

If \mathfrak{R} is a partial order, an element $a \in M$ is called the **infimum** (or **supremum**) of M if a is a lower (or upper) bound of M_0 and for any lower (or upper) bound a' of M_0 we obtain $a'\mathfrak{R}a$ ($a\mathfrak{R}a'$, respectively).

Definition 2.1.8. Let Y be a vector space and \mathfrak{R} be a binary relation on Y . We say that \mathfrak{R} is compatible with the linear structure of Y if the following properties are met:

- (i) for all $\lambda \geq 0$, $x, y \in Y : x\mathfrak{R}y \implies \lambda x\mathfrak{R}\lambda y$,
- (ii) for all $x, y, z \in Y : y\mathfrak{R}z \implies (x+y)\mathfrak{R}(x+z)$.

Definition 2.1.9. Assume Y is a vector space and K is a proper, convex cone in Y . A nonempty set B of K is considered a base for K if each nonzero element $y \in K$ has a unique representation of the form $y = \lambda b$ where $\lambda > 0$ and $b \in B$.

In the vector space Y , we consider an ordering relation \geq_K generated by a proper, convex cone $K \subseteq Y$. This order relation is defined as

$$x \geq_K y \quad \text{if and only if} \quad x - y \in K \quad \text{for all} \quad x, y \in Y. \quad (2.1)$$

If there is no confusion, we will often use the notation \leq_K as an ordering relation on Y , that is $x \leq_K y \Leftrightarrow y \geq_K x$.

We state some features of \geq_K in the following proposition.

Proposition 2.1.10. [68] Let Y be a vector space and K a convex cone in Y . Then ordering relation \geq_K defined by (2.1) has the following properties:

- (i) $x \geq_K x$ for all $x \in Y$ (reflexive),
- (ii) $x \geq_K y, y \geq_K z$ implies $x \geq_K z$ for all $x, y, z \in Y$ (transitive),
- (iii) $x \geq_K y$ implies $x + z \geq_K y + z$ for all $x, y, z \in Y$,
- (iv) $x \geq_K y$ implies $\lambda x \geq_K \lambda y$ for all $\lambda \geq 0$ and $x, y \in Y$.
- (v) If K is pointed, then \geq_K is antisymmetric. Furthermore, \geq_K is considered a partial order.

The theorem below provides a significant characterization of a partial order in a real linear space.

Theorem 2.1.11. [93] Let Y be a real linear space.

- (i) If \leq is a partial order on Y , then the set

$$K := \{x \in Y \mid 0_Y \leq x\}$$

is a convex cone. If, in addition, \leq is antisymmetric, then K is pointed.

(ii) If K is a convex cone on Y , then the binary relation

$$\leq_K := \{(x, y) \in Y \times Y \mid y - x \in K\}$$

is referred to as a partial order on Y . If, in addition, K is pointed, then \leq_K is antisymmetric.

Next, we define metric and normed spaces. These are of extremely important in our work for two reasons. First, normed spaces are distance functions, which are fundamental tools in the definition of the subdifferential of not necessarily convex functions. Secondly, in the later course of this thesis (especially in Chapter 7), we intend to investigate some vector-valued approximation problems in which the distance functions are induced by norms.

2.1.2 Metric Spaces

First, let us recall the following definition of a *topological space*.

Definition 2.1.12. [93] Let X be a nonempty set. A topology τ on X is defined to be a subset of X satisfying the following axioms:

- (i) the empty set \emptyset and the entire set X belong to τ .
- (ii) every union of sets of τ belongs to τ .
- (iii) every finite intersection of sets of τ belongs to τ .

A *topological space* is defined as a set X with a particular topology τ .

A topological space is defined as an ordered pair (X, τ) that consists of a set X and a topology τ on X , but we often omit the specific mention of τ to avoid confusion.

If X is a topological space with topology τ , a subset U of X is considered an *open set* of X if U belongs to the collection τ . Using this terminology, one can say that a topological space is a set X together with a collection of subsets of X , called *open sets* such that \emptyset and X are both open, and such that arbitrary unions and finite intersections of open sets are open.

Example 2.1.13. (i) If X is any set, the collection of all subsets of X is a topology on X , referred to as the **discrete topology**. The **indiscrete topology**, often known as the **trivial topology**, is a topology on X that includes only X and \emptyset .

(ii) Assume X is a set. Define τ_c as the collection of all subsets U of X such that $X \setminus U$ is either countable or whole of X . Then τ_c is a topology on X .

An important class of topological spaces is the so-called *metric space*, given in the following definition.

Definition 2.1.14. [93] Let X be a nonempty set. A metric d on X is defined as a function $d : X \times X \rightarrow \mathbb{R}$ that satisfies the following conditions for all $x, y, z \in X$

- (i) $d(x, y) = 0 \iff x = y$ (**definiteness**)
- (ii) $d(x, y) = d(y, x)$ (**symmetry**),

(iii) $d(x, z) \leq d(x, y) + d(y, z)$ (**triangle inequality**).

The pair (X, d) is called a **metric space**.

A very important class of metric space is the class of *normed space*.

2.1.3 Normed Spaces

Definition 2.1.15. [93] Let X be a linear space. A norm on X is a function $\|\cdot\| : X \rightarrow \mathbb{R}_+$ that holds the following properties for all $x, y \in X$ and all $\lambda \in \mathbb{R}$:

(i) $\|x\| = 0 \iff x = 0_X$ (**definiteness**),

(ii) $\|\lambda x\| = |\lambda| \|x\|$ (**positive homogeneity**),

(iii) $\|x + y\| \leq \|x\| + \|y\|$ (**triangle inequality**),

where 0_X denotes the origin in the linear space X . The pair $(X, \|\cdot\|)$ is called a **normed space**.

Observe that if $(X, \|\cdot\|)$ is a normed space, then the norm $\|\cdot\|$ always induces a metric on X given by

$$\text{for all } x, y \in X : d(x, y) := \|x - y\|.$$

Assuming that the pair $(X, \|\cdot\|)$ is complete (every *Cauchy sequence* $\{x_n\} \in X$ converges to $x \in X$), then the space is called *Banach space*. The Euclidean space $(\mathbb{R}^n, \|\cdot\|)$ with respect to the norm $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ provides one well-known example of a Banach space. The set $C(X, \mathbb{R})$ of all continuous functions on a metric space X is a real Banach space. Other examples of Banach spaces include ℓ^∞ , ℓ^p and $C[0, 1]$. Notice that, in the case that X is a normed space, we assume that the topology τ of X is generated by the metric induced by the norm $\|\cdot\|$.

We call a norm $\|\cdot\| : X \rightarrow \mathbb{R}$ *strictly convex* if for any $x, y \in X$, $x \neq y$, with $\|x\| = \|y\| = 1$, it follows that

$$\| \lambda x + (1 - \lambda)y \| < 1 \text{ for } \lambda \in]0, 1[.$$

A normed space $(X, \|\cdot\|)$ with the underlying strictly convex norm $\|\cdot\| : X \rightarrow \mathbb{R}$ is called strictly convex. In addition, the normed space $(X, \|\cdot\|)$ is considered *reflexive* if the *canonical embedding* of X into its *bidual space* $(X^*)^*$ (where X^* is the *dual space* of X), namely $J : X \rightarrow (X^*)^*$, defined for any $x \in X$, by

$$J(x)(x^*) = x^*(x), \quad x^* \in X^*,$$

is surjective. Every reflexive normed space is a Banach space, while each finite-dimensional Banach space is reflexive.

An important class of strictly convex normed spaces are the *inner product spaces*.

Definition 2.1.16. [93] Let X be a real linear space. A function $\langle \cdot, \cdot \rangle : X \times X \rightarrow \mathbb{R}$ is called an *inner product* on X if for all $x, y, z \in X$ and for all $\lambda \in \mathbb{R}$, the function $\langle \cdot, \cdot \rangle$ satisfies the following conditions:

(i) $\langle x, x \rangle > 0$ for $x \neq 0_X$ (**positivity**),

- (ii) $\langle x, y \rangle = \langle y, x \rangle$ (*symmetry*),
- (iii) $\langle \lambda x, y \rangle = \lambda \langle x, y \rangle$ (*positive homogeneity*),
- (iv) $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$ (*additivity*).

The pair $(X, \langle \cdot, \cdot \rangle)$ is known as *inner product space* (or *pre-Hilbert space*). Assuming that $(X, \langle \cdot, \cdot \rangle)$ is complete, the space is called *Hilbert space*.

Notice that each inner product space is a normed space with the underlying norm

$$\| \cdot \| := \sqrt{\langle \cdot, \cdot \rangle}.$$

Furthermore, each Hilbert space is both an inner product space and a reflexive normed space. In contrast, an inner product space, often known as a normed space, does not always correspond to a Hilbert space. However, in a finite-dimensional case, each inner product space is a Hilbert space. Hence, the space $(\mathbb{R}^n, \langle \cdot, \cdot \rangle)$ with respect to an inner product space that is defined by

$$\langle x, y \rangle := \sum_{i=1}^n x_i y_i,$$

for all $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \mathbb{R}^n$, is a Hilbert space. In a normed space $(X, \| \cdot \|)$, the *open* and *closed* balls with respect to the center x and the radius $r > 0$ are denoted by $B_X(x, r)$ and $B_X[x, r]$ respectively, and are defined as

$$B_X(x, r) := \{y \in X \mid \|x - y\| < r\},$$

and

$$B_X[x, r] := \{y \in X \mid \|x - y\| \leq r\}.$$

In addition, B_X stands for the unit closed ball of X .

We will talk about some well-known norms in the finite-dimensional space \mathbb{R}^n in the next example. These norms will be very helpful in our study, especially in Section 2.3 and when we deal with vector-valued approximation problems in Chapter 7 of this thesis.

Example 2.1.17.

1. The **Manhattan norm**, often known as the *city block norm* or the *rectangular norm*, it is defined for all $x \in \mathbb{R}^n$ by

$$\|x\|_1 := |x_1| + \dots + |x_n| =: \sum_{i=1}^n |x_i|.$$

In vector-valued approximation problems (see Chapter 7), the Manhattan norm is extremely essential. Apart from its use in vector-valued approximation problems, it is also used in the city networks, that is, in location theory. The Manhattan norm is also employed in machine engineering and the branch of robotics (see Section 6.2).

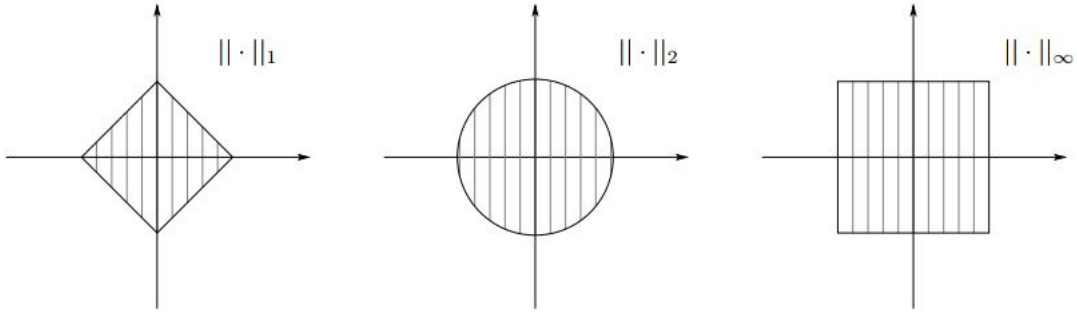


Figure 2.1: Unit balls for the norms: $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ on \mathbb{R}^2 .

2. The **Euclidean norm** in \mathbb{R}^n represents the length of the vector $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ in the form

$$\|x\|_2 := \sqrt{x_1^2 + \dots + x_n^2} =: \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}$$

The Euclidean norm is the most well-known norm in \mathbb{R}^n . It allows movement in all directions and is employed in the reformulation of ℓ_1 -norm regularization problems in compressive sensing, as shown in Section 2.3 and Chapter 6 of this thesis.

3. The **Maximum norm** is defined for all $x \in \mathbb{R}^n$ as:

$$\|x\|_\infty := \max\{|x_i| \mid i = 1, \dots, n\}.$$

In Figure 2.1, we illustrate the closed unit balls for the special case $n = 2$ of the norms given in the following example

$$B_{\|\cdot\|_i}(0_{\mathbb{R}^2}, 1) := \{x \in \mathbb{R}^2 \mid \|x\|_i \leq 1\}, \quad i \in \{1, 2, \infty\}.$$

All of these norms measure the length of a vector in some sense, and they are equivalent in the sense that each one is bounded above and below by a multiple of the other. To be precise, we obtain for all $x \in \mathbb{R}^n$, the following relationships exist among the three norms defined above:

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1,$$

and

$$\|x\|_1 \leq \sqrt{n}\|x\|_2 \leq n\|x\|_\infty.$$

For $1 \leq p \leq \infty$, the p -norm is defined for all $x \in \mathbb{R}^n$ as

$$\|x\|_p := \begin{cases} (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}, & \text{for } 1 \leq p < \infty, \\ \max\{|x_1|, \dots, |x_n|\}, & \text{for } p = \infty. \end{cases} \quad (2.2)$$

Obviously, for $p = 1$ yields the Manhattan norm while $p = 2$, yields the Euclidean norm. The limit $p \rightarrow \infty$ in (2.2) implies the maximum norm as shown in Example 2.1.17. Figure 2.1 shows examples of unit balls with a special p -norm.

Example 2.1.18. ([159]).

1. \mathbb{R}^n is a Banach space when equipped with one of the norms $\|\cdot\|_p$, $1 \leq p < +\infty$, and also when equipped with the norm $\|\cdot\|_\infty$, where

$$\|x\|_p := \sqrt[p]{\sum_{i=1}^n |x_i|^p} \text{ and } \|x\|_\infty := \max_{i=1, \dots, n} |x_i| \text{ for all } x = (x_1, \dots, x_n)^T \in \mathbb{R}^n.$$

$\|x\|_\infty$ is called the **Chebyshev norm** or **maximum norm** on \mathbb{R}^n .

$\|x\|_2$ is said to be the **Euclidean norm**. The related norm topology on \mathbb{R}^n is the so-called **Euclidean topology** τ_0 . Equipped with this topology, \mathbb{R}^n is called the **Euclidean space**. In \mathbb{R} , the open sets in this topology are the empty set and all unions of open intervals.

2. Consider the space of real sequences $x = (x_i)_{i \in \mathbb{N}}$ with the norms $\|\cdot\|_p$, $1 \leq p \leq +\infty$, defined by

$$\|x\|_p := \sqrt[p]{\sum_{i \in \mathbb{N}} |x_i|^p} \text{ if } 1 \leq p < \infty, \text{ and } \|x\|_\infty := \sup_{i \in \mathbb{N}} |x_i|.$$

Each space ℓ^p , $1 \leq p \leq +\infty$, of real sequences x with $\|x\|_p < +\infty$ is a Banach space when equipped with $\|\cdot\|_p$. The subspace c_0 of all sequences in ℓ^∞ converging to zero is a Banach space as well. This space contains each space ℓ^p , $1 \leq p < +\infty$.

3. The space $C[a, b]$ of all real-valued functions that are continuous on the interval $[a, b]$ equipped with **Chebyshev** or **supremum norm** $\|\cdot\|_\infty$ defined by

$$\|x\|_\infty := \sup_{t \in [a, b]} |x(t)|,$$

is a Banach space.

The above norms have been found to be useful in many practical problems not limited to location problems which can be considered as special approximation problems, compressive sensing, machine engineering and many other areas of applications. For more details, we refer the reader to [68, 109, 167, 168] and the references therein. For applications and more details of these norms, we refer the reader to Section 2.3, Chapter 6 and 7 of this thesis.

An important class of sequences in normed spaces are those that satisfy Cauchy's property.

Definition 2.1.19. Suppose $(X, \|\cdot\|_X)$ is a normed space and $\{x_k\}_{k \geq 1}$ is a sequence in X . If $\|x_n - x_m\| \rightarrow 0$, the sequence $\{x_k\}_{k \geq 1}$ is Cauchy.

The following results concerning projection operator can be found in [172]. The projection and its corresponding properties are very important for deriving the algorithms proposed in this thesis (see Chapters 3 and 4).

Theorem 2.1.20. Let Ψ be a nonempty, closed, and convex subset of a Euclidean norm of \mathbb{R}^n . Then, for any $x \in \mathbb{R}^n$, its projection onto Ψ denoted by $P_\Psi(x)$, is defined as

$$P_\Psi(x) := \arg \min \{\|x - y\|_2 : y \in \Psi\}. \quad (2.3)$$

The operator $P_\Psi(x) : \mathbb{R}^n \rightarrow \Psi$ is known as a metric projection with the following properties:

- (i) P_Ψ is non-expansive, which means $\|P_\Psi(x) - P_\Psi(y)\|_2 \leq \|x - y\|_2$ for all $x, y \in \mathbb{R}^n$.
- (ii) It holds $\|P_\Psi(x) - y\|_2 \leq \|x - y\|_2$ for all $y \in \Psi$.
- (iii) It holds $\langle P_\Psi(x) - P_\Psi(y), x - y \rangle \geq \|x - y\|_2^2$ for all $x, y \in \mathbb{R}^n$.
- (iv) It holds $\|P_\Psi(x)\|_2 \leq \|x\|_2$ for all $x \in \mathbb{R}^n$.

Next, we define the *convex map* using the following terms:

Definition 2.1.21. [93] Assume X and Y are real linear spaces. A map $T : X \rightarrow Y$ is called *linear*, if for all $x, y \in X$ and all $\lambda, \mu \in \mathbb{R}$,

$$T(\lambda x + \mu y) = \lambda T(x) + \mu T(y).$$

The set of continuous (bounded) linear maps between two real normed spaces $(X, \|\cdot\|_X)$ and $(Y, \|\cdot\|_Y)$ is a linear space as well and it is denoted by $B(X, Y)$. The norm $\|\cdot\| : B(X, Y) \rightarrow \mathbb{R}$ defined by

$$\|T\| = \sup_{x \neq 0_X} \frac{\|T(x)\|_Y}{\|x\|_X} \text{ for all } T \in B(X, Y)$$

$(B(X, Y), \|\cdot\|)$ is also a normed space. Furthermore, the class of linear maps is contained within the class of convex maps.

Convex sets and convex functions are very important in optimization theory. The useful properties of convex sets along with the differential ability of convex functions make the search for a minimum much easier. However, not all models in the applications deal with convexity, but when they do, it is much easier to ensure the existence of solutions and set algorithms that provide optimal solutions to the problem.

Definition 2.1.22. Let S be a subset of a real linear space X . S is called *convex* if $\lambda x + (1 - \lambda)y \in S$ whenever $x, y \in S$ and $\lambda \in [0, 1]$.

We call the set $[x, y] := \{\lambda x + (1 - \lambda)y \in S\}$ the **line segment** connecting the points x and y . Geometrically, a set S is convex if and only if the line segment of each two points of S is completely included in S .

Example 2.1.23.

- (i) The whole space X , the empty set and any singleton set are all convex sets.
- (ii) Hyperplanes and halfspaces in \mathbb{R}^n are convex sets.
- (iii) The intersection of any collection of convex sets is convex.
- (iv) If A and B are convex sets, then their sum (also called the **Minkowski sum**)

$$A + B := \{a + b | a \in A, b \in B\}$$

is convex. Note that if $A = \{a\}$ then the sum $\{a\} + B$ is usually written $a + B$. The set $a + B$ is a convex set whenever B is convex. Also the set $-A$ is convex, whenever A is convex.

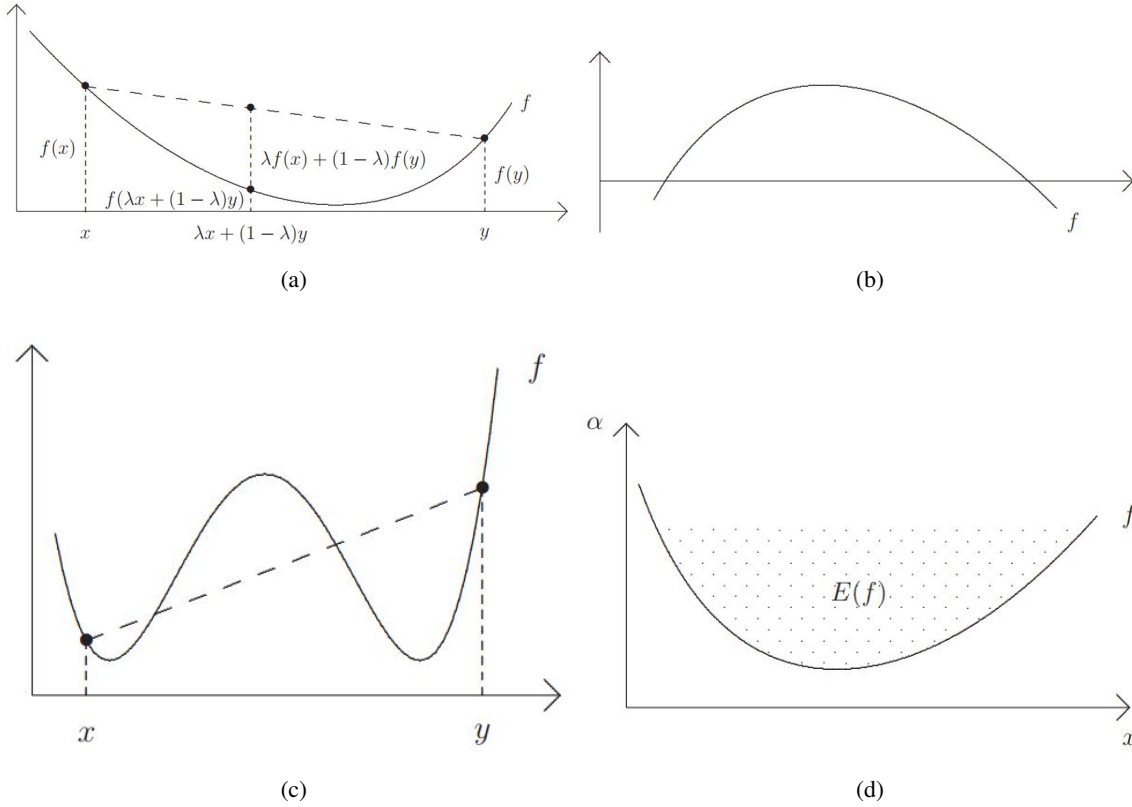


Figure 2.2: Representation of: convex, concave, non-convex and epigraph of the functional f (where $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$) [reprinted from [94, Figures: 2.1, 2.4, 2.5, 2.6].

Definition 2.1.24. [93] Let X be a real linear space and $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$ be a functional. The function f is called convex if for all $x, y \in X$ and for all $\lambda \in [0, 1]$:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad (2.4)$$

(see Figure 2.2 (a) and (c)).

A map $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$ is considered concave if $-f$ is convex (see Figure 2.2(b)).

If f is a linear map, then both f and $-f$ are convex maps.

Definition 2.1.25. Let X be a real linear space and $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$. A function f is called quasi-convex if the level set

$$L_{\leq}(r) := \{x \in X \mid f(x) \leq r\}$$

is convex for all $r \in \mathbb{R}$.

Since some part of this thesis deals with multiobjective optimization, we speak about generalized types of convex vector-valued functions using an ordering cone in Chapter 7.

Example 2.1.26.

- (i) Some functions are neither convex nor concave. Linear functions are the only ones that may be both convex and concave simultaneously.

(ii) Norms are convex functions (see Section 2.1.3).

For the function $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$, the set

$$\text{dom } f := \{x \in X \mid f(x) < +\infty\}$$

is known as the *domain* of f and the *effective domain* when f is extended real-valued.

Definition 2.1.27. [74] Let X and Y be real linear spaces and $F : X \rightarrow Y$ be a map. Then, the graph of F denoted by $\mathcal{G}(F)$, is defined by

$$\mathcal{G}(F) := \{(x, y) \in X \times Y \mid y = F(x)\}. \quad (2.5)$$

Convex sets and convex functions have a strong relationship, as demonstrated in the following definition.

Definition 2.1.28. [93] Assume X is a real linear space and $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$. The set is known as the graph of a function F while the set

$$E(f) := \{(x, r) \in X \times \mathbb{R} \mid (f(x) \leq r)\} \quad (2.6)$$

is known as the *epigraph* of f (see Figure 2.2(d)).

We can easily prove that f is a convex function if and only if $E(f)$ is a convex set (for the proof, see for example [69]).

2.1.4 Lipschitz Continuity

Here, we give the notion of Lipschitz continuity of a function over a nonempty, closed and convex subset $X_0 \subseteq X$. The concept of Lipschitz continuity belongs to the most important assumptions in the proof of the global convergence of our proposed algorithms as we shall see in Chapters 3 and 4 of this thesis. For more information on the concept of Lipschitz continuity of extended real-valued functions, we refer the reader to the book by Tammer and Weidner [159].

Definition 2.1.29. Let X be a normed space, $F : X \rightarrow \mathbb{R} \cup \{-\infty, +\infty\}$, $X_0 \subseteq X$ and $X_0 \neq \emptyset$. A function F is said to be **Lipschitz continuous** on X_0 if F is finite-valued on X_0 and there exists a constant $L \in \mathbb{R}_+$ such that for all $x, y \in X_0$

$$|F(x) - F(y)| \leq L\|x - y\|. \quad (2.7)$$

In this case, L is called a **Lipschitz constant** of F on X_0 . F is called **Lipschitz continuous** if F is Lipschitz continuous on X . Furthermore, F is called **Lipschitz function** if there is a constant $L \in \mathbb{R}_+$ such that (2.7) holds for all $x, y \in X_0$.

Note that Lipschitz continuity *at a point* depends only on the behaviour of the function near that point. For F to be Lipschitz continuous at x , the inequality (2.7) must hold for all y sufficiently near x , but it is not necessary that (2.7) holds if y is not near x . Also, F may be Lipschitz continuous at other points, but different values of L may be required for (2.7) to hold near those points. For example, $F(x) = \frac{1}{x}$ for $x > 0$ is Lipschitz continuous at each $x > 0$, but there is no single L for which (2.7) holds for all $x > 0$.

If F is Lipschitz continuous at x , then it is continuous at x . If F is a real-valued function defined on $X_0 \subseteq X$ that is differentiable at $x \in X_0$ then, F is Lipschitz continuous at x . In fact, this is more generally true: a function $F : X_0 \subseteq X \rightarrow \mathbb{R} \cup \{-\infty, +\infty\}$ is Lipschitz continuous at $x \in X_0$ if it is differentiable at x . In summary, we have

$$\text{differentiable at } x \implies \text{Lipschitz continuous at } x \implies \text{continuous at } x.$$

The converse implications of the above statement does not hold. Specifically, for $F(x) = \sqrt{|x|}$ is continuous at $x = 0$ but not Lipschitz continuous at $x = 0$ because its derivative is unbounded as x approaches zero. Also, the function $F(x) = |x|$ is Lipschitz continuous at $x = 0$ but not differentiable at $x = 0$. In summary, we have

$$\text{differentiable at } x \not\Leftarrow \text{Lipschitz continuous at } x \not\Leftarrow \text{continuous at } x.$$

F is said to be **locally Lipschitz continuous** on X_0 if for each $x \in X_0$, there exists a neighborhood U of x such that F is Lipschitz continuous on $U \cap X_0$. F is **locally Lipschitz continuous** if F is a finite-valued functional that is locally Lipschitz continuous on X .

2.2 Properties of Operators and Functions

In the following sections, we will collect some relevant results from the field of monotone operators and differentiability properties of functions. It is one of the most significant assumptions in the proofs of the thesis's convergence results. This can be seen clearly in Chapters 3, 4 and 7. These results can be found in [173].

In linear and convex optimization, solving the **dual** problem for a given original problem may be simpler and faster. When the dual problem is formulated, and a solution to it exists, then we get more information and possibly a solution to the original problem.

Dual Spaces

For a linear space X , the algebraic dual space is

$$X' := \{\acute{x} : X \rightarrow \mathbb{R} \mid \acute{x} \text{ is linear}\}.$$

For a linear topological space X the topological dual space is

$$X^* := \{x^* : X \rightarrow \mathbb{R} \mid x^* \text{ is linear and continuous}\}.$$

A linear functional $\acute{x} : X \rightarrow \mathbb{R}$ considered bounded if there exists a number $M \geq 0$ such that $|\acute{x}(x)| \leq M\|x\|$ for all $x \in X$. Linearly continuous and linearly bounded functionals are equivalent.

The norm for $x^* \in X^*$ is defined as

$$\|x^*\|_* := \sup_{x \neq 0} \frac{|x^*(x)|}{\|x\|}.$$

It is easy to verify that $\|\cdot\|_*$ is a norm on the space X^* . We refer to $\|\cdot\|_*$ as the dual norm of $\|\cdot\|$. Furthermore, it holds that

$$\|x^*\|_* = \sup_{x \neq 0} \frac{|x^*(x)|}{\|x\|} = \sup_{\|x\|=1} |x^*(x)| = \sup_{\|x\| \leq 1} |x^*(x)| = \sup_{\|x\| \leq 1, x \neq 0} \frac{|x^*(x)|}{\|x\|}.$$

The following theorem describes the set X^* , which includes all linear continuous functionals on X .

Theorem 2.2.1. ([69, Theorem 3.1]). *Assume $(X, \|\cdot\|)$ is a normed space and X^* represents the set of all linearly bounded functional on X . The set X^* is both a linear space and a normed space with the norm $\|x^*\|_*$. The generalized Schwarz's inequality states that*

$$|x^*(x)| \leq \|x^*\|_* \|x\| \quad \text{for all } x \in X, x^* \in X^*. \quad (2.8)$$

The dual space $(X^*, \|\cdot\|_*)$ is always complete and therefore a Banach space. The dual space X^* has a double dual space, represented by X^{**} . The double dual space has a Banach norm of $\|\cdot\|_{**}$. The normed space $(X, \|\cdot\|)$ is considered reflexive when $X = X^{**}$ and the inclusion $X \subset X^{**}$ is always true (see [69, Theorem 3.1]). The spaces \mathbb{R}^n , \mathbb{C}^n and all Hilbert spaces are reflexive, where \mathbb{C} represents the set of complex numbers.

Dual Cone

We now give some more properties of cones. Let X be a normed space and $K \subset X$ be a cone. The set

$$K^* := \{x^* \in X^* \mid \forall x \in K : x^*(x) \geq 0\} \quad (2.9)$$

is known as the **dual cone** of K .

Example 2.2.2.

1. We consider the standard ordering cone in \mathbb{R}^p :

$$\mathbb{R}_+^p = \{y \in \mathbb{R}^p \mid \forall i = 1, \dots, p : y_i \geq 0\}. \quad (2.10)$$

Then

$$(\mathbb{R}_+^p)^* = \{z^* \in (\mathbb{R}^p)^* = \mathbb{R}^p \mid \forall y \in \mathbb{R}_+^p : (z^*)^T y \geq 0\} = \mathbb{R}_+^p.$$

In particular, $(\mathbb{R}_+^2)^* = \mathbb{R}_+^2$.

2. For $K = \mathbb{R}^p$, $K^* = \{z^* \in (\mathbb{R}^p)^* = \mathbb{R}^p \mid \forall y \in \mathbb{R}^p : (z^*)^T y \geq 0\} = \{0\}$. Conversely, if $K = \{0\}$, then $K^* = \{z^* \in (\mathbb{R}^p)^* = \mathbb{R}^p \mid (z^*)^T 0 \geq 0\} = \mathbb{R}^p$.

The previous examples show that $(K^*)^* = K$.

Orthogonality

Geometrical features such as orthogonality and projection can be well described in Hilbert spaces using the structure of the inner product. The property of orthogonality is defined as below.

Definition 2.2.3. Let X be a Hilbert space. For $x, y \in X$ we say that x is **orthogonal** to y , denoted by $x \perp y$, if $\langle x, y \rangle = 0$. For the sets $A, B \subset X$, we say that $A \perp B$ if $\langle x, y \rangle = 0$ for all $x \in A$ and all $y \in B$. Additionally, we define the **orthogonal complement** of a subset $A \subset X$ as

$$A^\perp := \{x \in X \mid \forall y \in A : x \perp y\}. \quad (2.11)$$

For each subset $A \subset X$, we can show that A^\perp is a closed linear subspace of the Hilbert space X . Additionally, it is clear that $A \cap A^\perp = \{0\}$.

Theorem 2.2.4. (Complementary subspaces [83]). If A is a complete subspace of the inner product space X , then every $x \in X$ can be uniquely represented as:

$$x = u + v : \quad u \in A, v \in A^\perp.$$

This implies that the orthogonal decomposition $A \oplus A^\perp = X$. The subspaces A and A^\perp are considered complementary subspaces.

Theorem 2.2.4 applies in the particular case where X is a Hilbert space and A is a closed subspace. The following two theorems establish the existence and uniqueness of optimal solutions to a general optimization problem in inner product spaces.

Theorem 2.2.5. ([83, Theorem 21.1]). Assume $(X, \langle \cdot, \cdot \rangle)$ is an inner product space. If $S \neq \emptyset$ is a convex and complete subset of X , then for $x \in X$, the problem $\|x - y\| \rightarrow \min_{y \in S}$ has a unique solution in S , that is, there exists a unique element $y^0 \in S$ with $\|x - y^0\| \leq \|x - y\|$ for every $y \in S$.

In particular, the assumptions of Theorem 2.2.5 are also fulfilled when X is a complete space (a Hilbert space) and S is a nonempty, convex and closed subset of X .

Theorem 2.2.6. ([83, Theorem 21.2]). Let S be a linear subspace of the inner product space $(X, \langle \cdot, \cdot \rangle)$. If the problem $\|x - y\| \rightarrow \min_{y \in S}$ for some element $x \in X$ has ever a solution $y^0 \in S$, then y^0 is the only solution in S and $x - y^0 \perp S$.

Definition 2.2.7. Let $(X, \|\cdot\|_X)$ be a normed space with $f : X \rightarrow X^*$. Then f is called:

- (i) **continuous** at the point $x \in X$ if $x_n \rightarrow x$ implies $f(x_n) \rightarrow f(x)$. f is called **continuous** if it is continuous at each point in X .
- (ii) **hemicontinuous** if the real function $t \mapsto \langle f(x + ty), z \rangle$ is continuous on $[0, 1]$ for all $x, y, z \in X$.

Definition 2.2.8. Let $(X, \|\cdot\|_X)$ be a normed space with $f : X \rightarrow X^*$. Then f is called:

- (i) **monotone** if $\langle f(x) - f(y), x - y \rangle \geq 0$ for all $x, y \in X$,
- (ii) **strictly monotone** if $\langle f(x) - f(y), x - y \rangle > 0$ for all $x, y \in X$ with $x \neq y$,
- (iii) **strongly monotone** if there exists a constant $c > 0$ such that $\langle f(x) - f(y), x - y \rangle \geq c\|x - y\|_X$ for all $x, y \in X$,

(iv) *pseudomonotone* if $x \rightharpoonup X$ and $\limsup_{n \rightarrow +\infty} \langle f(x_n), x_n - x \rangle \leq 0$ implies for all $y \in X$

$$\langle f(x_n), x - y \rangle \leq \liminf_{n \rightarrow +\infty} \langle f(x_n), x_n - y \rangle.$$

Definition 2.2.9. Let $(X, \|\cdot\|_X)$ be a normed space. A set-valued operator $F : X \rightrightarrows X^*$ is called:

(i) *monotone* if it holds

$$\langle x^* - y^*, x - y \rangle \geq 0, \text{ for every } (x, x^*), (y, y^*) \in \mathcal{G}(F),$$

(ii) *maximal monotone* if F is monotone, it follows from $(x, x^*) \in X \times X^*$ and

$$\langle x^* - y^*, x - y \rangle \geq 0, \text{ for every } (y, y^*) \in \mathcal{G}(F)$$

that $(x, x^*) \in \mathcal{G}(F)$.

Definition 2.2.10. Let X be a real Banach space. A set-valued operator $F : X \rightrightarrows X^*$ is called *semi-monotone* if $\mathcal{D}(F) = X \times X$ and the following conditions are satisfied:

(i) For any $u \in X$, $F(u, \cdot) : X \rightrightarrows X^*$, is maximal monotone with $\mathcal{D}(F(u, \cdot)) = X$.

(ii) Let $x \in X$ and $\{u_n\} \subseteq X$ be a sequence such that $u_n \rightharpoonup u$. Then, for every $w \in F(u, x)$, there exists a sequence $\{w_n\} \in X^*$ such that $\{w_n\} \in F(u_n, x)$ and $w_n \rightharpoonup w$.

Separation Theorems for Convex Sets

There are many well-known fundamental theorems that play a role in the background of our results, such as Zorn's Lemma and Hahn-Banach-Theorem.

The following separation theorem is an important result from functional analysis that is based on the Hahn-Banach-Theorem. The separation theorem is useful for characterizing vector-valued approximation solutions (see Section 7.3, in particular Theorem 7.3.7).

Theorem 2.2.11. (Separation Theorem [69, Theorem 5.11]). Let X be a real normed space. Let $B, C \subset X$ be nonempty convex sets with $\text{int } B \neq \emptyset$ and $\text{int } B \cap C = \emptyset$. Then there exists a continuous linear functional $x^* \in X^*$ and a real number α such that

$$\forall s \in B, \forall t \in C : x^*(s) \leq \alpha \leq x^*(t). \quad (2.12)$$

If the sets B and C are open, then the separation in (2.12) is strict and made by a continuous linear functional $x^* \in X^* \setminus \{0_{X^*}\}$ and a real number α such that

$$\forall s \in B, \forall t \in C : x^*(s) < \alpha < x^*(t). \quad (2.13)$$

Theorem 2.2.12. (Separation Theorem [93, Theorem 3.18]). Let X be a real locally convex space and B be a nonempty, closed convex subset of X . Then $x \in X \setminus B$ if and only if there exists a continuous linear functional, i.e., $x^* \in X^* \setminus \{0\}$ and a real number α such that

$$\forall s \in B : x^*(x) < \alpha \leq x^*(s). \quad (2.14)$$

Differentiability Properties of Functions

In this section, we will review the concepts of directional derivatives in order to define the subdifferential of a convex function, especially of the norm. The subdifferential is our primary tool for determining the optimality conditions in Section 7.2.

Next, we define the directional derivative, which is an extension of the well-known derivative in real space. For examples of literature, see [69, Definition 3.34], [93, Definition 2.12] and the well-known book [145].

Definition 2.2.13. (*Gâteaux Derivative*): Consider X a linear space, S a nonempty subset of X , Y a normed space, and $f : S \rightarrow Y$ a mapping. For $x^0 \in S$, $h \in X$, the mapping f is called *Gâteaux differentiable* at x^0 in the direction h if there exists an $\varepsilon > 0$ with $[x^0 - \varepsilon h, x^0 + \varepsilon h] \subset S$ and if the limit

$$f'(x^0, h) := \lim_{t \rightarrow 0} \frac{f(x^0 + th) - f(x^0)}{t} \quad (2.15)$$

exist. $f'(x^0, h)$ is the **Gâteaux derivative** of f at x^0 in the direction h . If this limit exists for all $h \in X$, f is called *Gâteaux differentiable* at x^0 and $f'(x^0, \cdot)$ is called *Gâteaux derivative* of f at x^0 .

The following definition does not include the entire interval $[x^0 - \varepsilon h, x^0 + \varepsilon h]$. Later on, we discover that this scenario may be sufficient for our research at times.

Definition 2.2.14. (*Right-Hand side Gâteaux Derivative*). Consider the assumptions outlined in Definition 2.2.13. For $x^0 \in S$, $h \in X$, if $\varepsilon > 0$ exists with only $[x^0, x^0 + \varepsilon h] \subset S$ and the limit

$$f'_+(x^0, h) := \lim_{t \rightarrow +0} \frac{f(x^0 + th) - f(x^0)}{t} \quad (2.16)$$

exists, then $f'(x^0, h)$ is called *directionally differentiable* at x^0 in the direction h and $f'_+(x^0, h)$ is called **right-hand side direction derivative** (or *direction derivative*) of f at x^0 in the direction h .

Similarly, if $[x^0 - \varepsilon h, x^0] \subset S$ and $t \rightarrow -0$ in the limit, we refer to the *left-hand side direction derivative* as $f'_-(x^0, h)$.

We illustrate some properties of Gâteaux derivative and the direction derivative (defined in 2.2.13 and 2.2.14, respectively) and the relationship between them. It is easy to show that the following statements are true:

- $f'(x^0, \cdot)$ is positively homogeneous but not always linear. (A mapping $A : X \rightarrow Y$ is considered positively homogeneous when $A(\alpha x) = \alpha A(x)$ for all $x \in X$ and $\alpha \in \mathbb{R}_+$).
- Consider the assumptions outlined in Definition 2.2.13. The function f is Gâteaux differentiable at $x^0 \in S$ in the direction h if and only if f is right-hand side and left-hand side directionally differentiable at x^0 in direction h and $f'_+(x^0, h) = f'_-(x^0, h)$. The equation is $f'(x^0, h) = f'_+(x^0, h) = f'_-(x^0, h)$.
- If the function f is left-hand side differentiable at x^0 in the direction h , then it is also right-hand side differentiable in the direction $-h$. The equation $f'_-(x^0, h) = f'_+(x^0, -h)$ is always valid.

Other generalized definitions of derivatives can be found in the literature. The Fréchet derivative, an extension of the directional derivative in Banach spaces, is a stronger condition than the Gâteaux derivative.

We now present some differential properties of a function under convexity assumptions. Let X be a linear space and $S \subset X$ be convex. If $f : S \rightarrow \mathbb{R} \cup \{+\infty\}$ is convex, then f is right hand-side and left-hand side Gâteaux differentiable at every algebraic interior point x^0 of S with $f(x^0) \in \mathbb{R}$, in every direction $h \in \mathbb{R}$, and the Gâteaux derivative mapping $f'(x^0, \cdot) : X \rightarrow \mathbb{R}$ is linear (see [69, Theorem 3.32]). Furthermore, it preserves the monotonicity of the difference quotient. The following theorem highlights an important inequality for determining optimality conditions under convexity assumptions.

Theorem 2.2.15. ([69, Theorem 3.33]). *Let X be a linear space and $S \subset X$ a convex set with $S = \text{core}(S)$ (S consists only of algebraic interior points). If the function $f : X \rightarrow \mathbb{R}$ is Gâteaux differentiable at all the points in S , then the following propositions are equivalent:*

1. f is convex.
2. $f'(x^0, \cdot)$ is linear for all $x \in S$ and the following subgradient inequality holds for all $x, x^0 \in S$:

$$f'(x^0, x - x^0) \leq f(x) - f(x^0). \quad (2.17)$$

In the example below, we compute the Gâteaux derivative of certain convex functions, specifically the norm. The norm as a distance function is the primary tool for studying our vector-valued approximation problems in Chapter 7. The differentiability properties of the norm are particularly evident when establishing the optimality conditions in Section 7.2.

Example 2.2.16. ([69]). *Let $(X, \langle \cdot, \cdot \rangle)$ be a Hilbert space with the norm $\|x\| := \sqrt{\langle x, x \rangle}$. Consider the function $f(x) := \|x - x^0\|^2$ for a given $x^0 \in X$.*

We compute the Gâteaux derivative $f'(x, h)$ of the function f for $x, h \in X$. To this end, we compute the following quotient for $t \in \mathbb{R}$:

$$\begin{aligned} \frac{f(x+th) - f(x)}{t} &= \frac{\langle x+th - x^0, x+th - x^0 \rangle - \langle x - x^0, x - x^0 \rangle}{t} \\ &= \frac{\langle x - x^0, x+th - x^0 \rangle + \langle th, x+th - x^0 \rangle - \langle x - x^0, x - x^0 \rangle}{t} \\ &= \frac{\langle x - x^0, x - x^0 \rangle + \langle x - x^0, th \rangle + \langle th, x - x^0 \rangle + \langle th, th \rangle - \langle x - x^0, x - x^0 \rangle}{t} \\ &= \frac{2\langle x - x^0, th \rangle + \langle th, th \rangle}{t} = \frac{2t\langle x - x^0, h \rangle + t^2\langle h, h \rangle}{t} \\ &= 2\langle x - x^0, h \rangle + t\langle h, h \rangle. \end{aligned}$$

Computing the limit yields

$$f'(x, h) = \lim_{t \rightarrow 0} \frac{f(x^0 + th) - f(x^0)}{t} = 2\langle x - x^0, h \rangle.$$

Subdifferentials of Convex and Nonconvex Functions

The subdifferential of convex functions play an important role in nonlinear optimization. In general, distance functions are not differentiable. The search for a more general concept of differentiability leads to the subdifferential of a function, which turns to be an important tool to formulate optimality conditions, particularly in location theory or vector-valued approximation problems (see Chapter 7). Next, we give the following definitions.

Definition 2.2.17. (*The Subdifferential [93]*). Assume X is a real Banach space, $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$, $x, x^0 \in X$ at x^0 in the direction x . If $f'_+(x^0, \cdot)$ given in (2.16) exists, the set

$$\partial_G f(x^0) := \{x^* \in X^* \mid \forall x \in X : x^*(x) \leq f'_+(x^0, x)\} \quad (2.18)$$

is called a **subdifferential** of f at x^0 and the elements of $\partial_G f(x^0)$ are called **subgradients**.

In the following definitions, we consider the normal cone and the subdifferential in the sense of convex analysis (**Fenchel subdifferential**) of convex functions defined as follows:

Definition 2.2.18. (*Fenchel Subdifferential [171, Section 2.4]*). Let X be a real Banach space and $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$ a proper convex function, the **subdifferential** or **Fenchel subdifferential** of f at $x^0 \in \text{dom } f$ is defined by

$$\partial f(x^0) := \{x^* \in X^* \mid \forall x \in X : x^*(x - x^0) \leq f(x) - f(x^0)\}, \quad (2.19)$$

for $x^0 \notin \text{dom } f$, one puts $\partial f(x^0) = \emptyset$. If $\partial f(x^0)$ is nonempty, f is said to be subdifferentiable at x^0 .

Note:

1. The elements of the subdifferential are functionals from the dual space which we call the subgradients, that means the subdifferential $\partial(\cdot)$ is a set-valued operator.
2. If f is convex, then $\partial_G f(x^0) = \partial f(x^0)$ for $x^0 \in X$.

Definition 2.2.19. (*Normal Cone*). Let E be a nonempty convex subset of a real Banach space X . The normal cone N of the set E at the point x^0 is defined by

$$N(x^0, E) := \{x^* \in X^* \mid x \in E : x^*(x - x^0) \leq 0\}.$$

Observe that if x^0 is an interior point of the set E , then the normal cone of E at x^0 is $N(x^0, E) = \{0\}$.

It follows directly from Definition 2.2.18 and 2.2.19 that the normal cone to a set E at a given point can also be equivalently defined by the subdifferential of the indicator function associated with this set at that point,

$$N(x^0, E) = \partial \delta_E(x^0), \quad (2.20)$$

where δ_E is the indicator function of E .

The following theorem demonstrates, under which conditions the subdifferential of a convex function exists.

Theorem 2.2.20. (*[69, Theorem 5.12]*). Let X be a real Banach space, $x^0 \in X$ and $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$ be a convex functional. If $f(x^0) < +\infty$ and f is continuous at x^0 , then

$$\partial f(x^0) \neq \emptyset.$$

Calculus Rules for Subdifferentials

To apply optimality conditions in some applications and algorithms, we must compute the subdifferential of the sum of functions. As a result, we recall the following theorem about the sum rule for convex functions:

Theorem 2.2.21. (The Subdifferential Sum Rule, [69, Theorem 5.13]). For $n \geq 2$, let $f_1, \dots, f_n : X \rightarrow \mathbb{R} \cup \{+\infty\}$ be convex functionals on a Banach space X . If the values $f_1(x^0), \dots, f_n(x^0) < +\infty$ exist for an element x^0 and f_1, \dots, f_{n-1} are continuous at x^0 , then it holds for all $x \in X$ that

$$\partial \left(\sum_{i=1}^n f_i(x) \right) = \sum_{i=1}^n \partial f_i(x). \quad (2.21)$$

The right side of equation (2.21) represents the Minkowski sum of the sets of subdifferentials $\partial f_i(x)$ of the functions f_i .

The Subdifferential for Norm Functions

The subdifferential for norm functions plays an important role in the computation of the p -norms, including $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$. These are norm-induced distance functions (or norm-squared). For the calculation of the subdifferentials of the norms, we resort to the following well-known result, which can be found in the book by Göpfert, Riedrich and Tammer [69, Theorem 5.15].

Theorem 2.2.22. Let X be a Banach space. Then the norm $\|\cdot\|_X$ is subdifferentiable and the following holds:

$$\begin{aligned} \forall x \in X \setminus \{0\} : \partial \|\cdot\|_X(x) &:= \{x^* \in X^* \mid \langle x^*, x \rangle = \|x\| \text{ and } \|x^*\|_* = 1\}, \\ \text{at } x = 0 \in X : \partial \|\cdot\|_X(x) &:= \{x^* \in X^* \mid \|x^*\|_* \leq 1\}. \end{aligned}$$

Proof. a) It follows from Hahn–Banach Theorem that for all $x_0 (\neq 0) \in X$, a linear continuous functional $x_0^* \in X^*$ exists, where

$$x_0^*(x_0) = \|x_0\|_X \text{ and } \|x_0^*\|_* = 1. \quad (2.22)$$

This results in

$$x_0^*(x - x_0) \leq \|x_0^*\|_* \|x\|_X - x_0^*(x_0) = \|x\|_X - \|x_0\|_X, \quad (2.23)$$

That is, $x_0^* \in \partial \|\cdot\|_X(x_0)$.

b) Conversely, for $x_0^* \in \partial \|\cdot\|_X(x_0)$, it holds from (2.23) that

$$\forall x \in X, \quad \|x_0\|_X - \|x\|_X \leq x_0^*(x_0 - x). \quad (2.24)$$

Now, if we set $x = \lambda x_0$ ($\lambda \in \mathbb{R}$, $\lambda \geq 0$), we can conclude from (2.24) that

$$(1 - \lambda)(x_0^*(x_0) - \|x_0\|_X) \geq 0. \quad (2.25)$$

If we take $\lambda > 1$ and $\lambda < 1$, we get

$$x_0^*(x_0) = \|x_0\|_X.$$

Using (2.24), we can conclude that $x_0^*(x) \leq \|x\|_X$ for any $x \in X$ and $\|x_0^*\|_* = 1$.

c) For $x_0 = 0$, the statement follows directly from the Definition 2.2.17 of the subdifferential. Indeed, for $x_0 = 0$, we obtain

$$\partial\|x_0\| = \{x^* \in X^* \mid x^*(x) \leq \|x\| \ (x \in X)\}.$$

□

We now use the result of Theorem 2.2.22 to compute the subdifferentials of the p -norms ($p = 1, 2$ and $p = \infty$) of Example 2.1.17 in the following theorems, along with their proofs for the reader's convenience which are essential for numerical computations as shown in the following results.

Theorem 2.2.23. ([94]). For $p = 1$, we compute the subdifferential of the ℓ_1 -Norm (Manhattan Norm) for $f : \mathbb{R}^n \rightarrow \mathbb{R}$ where $f(x) = \|x\|_1 = \sum_{i=1}^n |x_i|$. For the Fenchel subdifferential it holds that

$$\partial\|\cdot\|_1(x) = \left\{ x^* \in \mathbb{R}^n \mid x_i^* \in \begin{cases} \{1\}, & \text{for } x_i > 0 \\ [-1, 1], & \text{for } x_i = 0, \ i = 1, \dots, n \\ \{-1\}, & \text{for } x_i < 0 \end{cases} \right\}.$$

Proof. For $x = 0$, using Theorem 2.2.22 and $(\|\cdot\|_1)_* = \|\cdot\|_\infty$

$$\begin{aligned} \partial\|\cdot\|_1(0) &= \{x^* \in X^* \mid (\|x^*\|_1)_* \leq 1\} \\ &= \{x^* \in \mathbb{R}^n \mid \|x^*\|_\infty \leq 1\} \\ &= \{x^* \in \mathbb{R}^n \mid x_i^* \in [-1, 1]\} \\ &= \{x^* \in \mathbb{R}^n \mid x_i^* \in [-1, 1], \text{ for } x_i = 0, \ i = 1, \dots, n\}. \end{aligned}$$

Let $x \in \mathbb{R}^n \setminus \{0\}$, then using Theorem 2.2.22 and $(\|\cdot\|_1)_* = \|\cdot\|_\infty$ we can conclude that,

$$\begin{aligned} \partial\|\cdot\|_1(0) &= \{x^* \in X^* \mid \langle x^*, x \rangle = \|x\|_1, (\|x^*\|_1)_* = 1\} \\ &= \left\{ x^* \in \mathbb{R}^n \mid \sum_{i=1}^n x_i^* x_i = \sum_{i=1}^n |x_i|, \|x^*\|_\infty = 1 \right\} \\ &= \{x^* \in \mathbb{R}^n \mid x_i^* x_i = |x_i|, \text{ for all } i = 1, \dots, n, \|x^*\|_\infty = 1\}. \end{aligned}$$

The last equation follows for all $i = 1, \dots, n$, $\|x^*\|_\infty = 1$, implying that $|x_i^*| \leq 1$ and therefore $x_i^* x_i \leq |x_i|$ holds. For $x_i < 0$, the theorem applies to $x_i^* = -1$, but for $x_i > 0$, $x_i^* = 1$. Since at least one component of x differs from 0, it follows that $\|x^*\|_\infty = 1$ is satisfied. It also follows that for $x_i = 0$, $x_i^* \in [-1, 1]$ and hence the assertion of the theorem holds. □

Theorem 2.2.24. ([94]). For $p = 2$, we compute the subdifferential of the ℓ_2 -Norm (Euclidean Norm) for $f : \mathbb{R}^n \rightarrow \mathbb{R}$ where $f(x) = \|x\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}$. For the Fenchel subdifferential, it holds that

$$\text{For all } x \in \mathbb{R}^n \setminus \{0\}, \quad \partial\|\cdot\|_2(x) = \left\{ \frac{x}{\|x\|_2} \right\}.$$

$$\text{Similarly, } \partial\|\cdot\|_2(0) = \{x^* \in \mathbb{R}^n \mid \|x^*\|_2 \leq 1\}.$$

Proof. According to Theorem 2.2.22, if $x = 0$, the statement $(\|\cdot\|_2)_* = \|\cdot\|_2$ holds.

Let $x \in \mathbb{R}^n \setminus \{0\}$, then it follows from Theorem 2.2.22 that

$$\begin{aligned} \partial\|\cdot\|_2(x) &= \{x^* \in X^* \mid \langle x^*, x \rangle = \|x\|_2, \|x^*\|_2 = 1\} \\ &\subseteq \{x^* \in \mathbb{R}^n \mid \langle x^*, x \rangle \langle x^*, x \rangle = \|x\|_2^2, \|x^*\|_2^2 = 1\} \\ &= \{x^* \in \mathbb{R}^n \mid \langle x^*, x \rangle^2 = \langle x, x \rangle, \langle x^*, x^* \rangle = 1\} \\ &= \{x^* \in \mathbb{R}^n \mid \langle x^*, x \rangle^2 = \langle x, x \rangle \langle x^*, x^* \rangle, \langle x^*, x^* \rangle = 1\}. \end{aligned}$$

The Cauchy–Schwarz inequality, for any $x, y \in \mathbb{R}^n$, it always holds that $\langle x, y \rangle^2 \leq \langle x, x \rangle \langle y, y \rangle$. This inequality is only valid when x and y are linearly dependent. So we obtain $x^* = r \cdot x$ with $r \in \mathbb{R}$. Furthermore, $\|r \cdot x\|_2 = \|x^*\|_2 = 1$. In particular, $\|x\|_2 \neq 0$ and hence $r = \frac{1}{\|x\|_2}$. For the remaining two possibilities, we obtain for $x_1^* = \frac{x}{\|x\|_2}$ and $x_2^* = -\frac{x}{\|x\|_2}$, one may verify by substituting that $x_1^* \in \partial\|\cdot\|_2(x)$ and $x_2^* \notin \partial\|\cdot\|_2(x)$ holds. \square

Theorem 2.2.25. ([94]). For $p = \infty$, we compute the subdifferential of the ℓ_∞ -Norm (Maximum Norm) for

$$J := \left\{ j \mid j \in \{1, \dots, n\}, |x_j| = \max_{i \in \{1, \dots, n\}} |x_i| \right\}, \text{ for any } x \in \mathbb{R}^n.$$

The ℓ_∞ -Norm is defined for $f : \mathbb{R}^n \rightarrow \mathbb{R}$ as $f(x) = \|x\|_\infty = \max_{i \in \{1, \dots, n\}} |x_i|$. The Fenchel subdifferential have the following property

$$\begin{aligned} \partial\|\cdot\|_\infty(x) &= \{x^* \in \mathbb{R}^n \mid \sum_{j \in J} |x_j^*| = 1, \text{sgn}(x_j^*) = \text{sgn}(x_j) \forall j \in J, \\ &\quad x_k^* = 0 \forall k \in \{1, \dots, n\} \setminus J \text{ for } x \in \mathbb{R}^n \setminus \{0\}, \\ \partial\|\cdot\|_\infty(0) &= \{x^* \in \mathbb{R}^n \mid \|x^*\|_1 \leq 1\}. \end{aligned}$$

Proof. The subdifferential at $x = 0$ follows $(\|\cdot\|_\infty)_* = \|\cdot\|_1$, as stated in Theorem 2.2.22. Now for $x \neq 0$, the first statement of Theorem 2.2.22 provides

$$\begin{aligned} \partial\|\cdot\|_\infty(x) &= \{x^* \in X^* \mid \langle x^*, x \rangle = \|x\|_\infty, \|x^*\|_1 = 1\} \\ &= \left\{ x^* \in \mathbb{R}^n \mid \sum_{i=1}^n x_i^* x_i = \max_{i \in \{1, \dots, n\}} |x_i|, \sum_{i=1}^n |x_i^*| = 1 \right\}. \end{aligned}$$

For x_k , with $k \in \{1, \dots, n\} \setminus J$, $\sum_{i=1}^n x_i^* x_i \leq x_k^* x_k + (1 - |x_k^*|) \max_{i \in \{1, \dots, n\}} |x_i| \leq \max_{i \in \{1, \dots, n\}} |x_i|$ is holds for $x_k^* \neq 0$. This yields $x_k^* = 0$ for all such k .

Now from the definition of J in the statement of the theorem, we obtain

$$\begin{aligned}
\partial \|\cdot\|_\infty(x) &= \left\{ x^* \in \mathbb{R}^n \mid \sum_{i=1}^n x_i^* x_i = \max_{i \in \{1, \dots, n\}} |x_i|, \sum_{i=1}^n |x_i^*| = 1 \right\} \\
&= \left\{ x^* \in \mathbb{R}^n \mid x_k^* = 0 \forall k \in \{1, \dots, n\} \setminus J, \sum_{j \in J} x_j^* x_j = \max_{i \in \{1, \dots, n\}} |x_i|, \sum_{j \in J} |x_j^*| = 1 \right\} \\
&= \left\{ x^* \in \mathbb{R}^n \mid x_k^* = 0 \forall k \in \{1, \dots, n\} \setminus J, \sum_{j \in J} x_j^* \operatorname{sgn}(x_j) = 1, \sum_{j \in J} |x_j^*| = 1 \right\} \\
&= \left\{ x^* \in \mathbb{R}^n \mid x_k^* = 0 \forall k \in \{1, \dots, n\} \setminus J, x_j^* \operatorname{sgn}(x_j) = |x_j^*| \forall j \in J, \sum_{j \in J} |x_j^*| = 1 \right\} \\
&= \left\{ x^* \in \mathbb{R}^n \mid x_k^* = 0 \forall k \in \{1, \dots, n\} \setminus J, \operatorname{sgn}(x_j) = \operatorname{sgn}(x_j^*) \forall j \in J, \sum_{j \in J} |x_j^*| = 1 \right\},
\end{aligned}$$

hence, the assertion holds. \square

The structure of the subdifferential is important for deriving algorithms, especially the proximal algorithms.

Optimality Condition

One of the primary goal of optimization theory is to find the best solution and determine whether or not it exists. Here, we introduce different formulations of necessary and sufficient optimality conditions using the right-hand side direction derivative (defined in 2.2.13), the Gâteaux-derivative, and the subdifferential of convex functions. As mentioned above, convexity assumptions yield the existence of Gâteaux-derivative (see also [69, Theorem 3.32]).

The following necessary and sufficient optimality condition is given by a variational inequality.

Theorem 2.2.26. ([69]). *Consider X a linear space, S a convex set and $f : S \rightarrow \mathbb{R}$ a convex function. Then, for $x, x^0 \in S$ it holds:*

1. x^0 is a minimal solution to the nonlinear optimization problem $\min_{x \in S} f(x)$, if and only if for all $x \in S$:

$$f'_+(x^0, x - x^0) \geq 0.$$

2. If S is a linear subspace and f is Gâteaux-differentiable, then x^0 is a minimal solution of the nonlinear optimization problem $\min_{x \in S} f(x)$, if and only if for all $x \in S$,

$$f'(x^0, x) = 0.$$

Remark 2.2.27. *The necessary and sufficient optimality condition for a minimal solution to the nonlinear optimization problem given in Theorem 2.2.26(2) is the same as the first-order necessary conditions for a local minimizer used for nonlinear monotone operator problem stated in Theorem 2.5.3 (see Section 2.5).*

The following theorem provides a necessary and sufficient optimality condition for minimal solutions to $\min_{x \in X} f(x)$. We employ this type of optimality criterion later in our results in Chapter 7, specifically in Section 7.4.

Theorem 2.2.28. ([69, Theorem 5.14]). *Assume X is a Banach space and $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$ is convex with $f(x) < +\infty$. It holds that $x^0 \in X$ is a minimal solution of the nonlinear optimization problem $\min_{x \in X} f(x)$ if and only if*

$$0 \in \partial f(x^0). \quad (2.26)$$

The proof follow easily from (2.19).

Lemma 2.2.29. (Minty Lemma). *Assume Ψ is a nonempty, closed and convex subset of a normed space $(X, \|\cdot\|_X)$ and $F : X \rightarrow X^*$ is monotone and hemicontinuous. Then $x \in \Psi$ satisfies*

$$\langle F(x), y - x \rangle \geq 0, \text{ for every } y \in \Psi, \quad (2.27)$$

if and only if it satisfies

$$\langle F(y), y - x \rangle \geq 0, \text{ for every } y \in \Psi. \quad (2.28)$$

Theorem 2.2.30. (Hartmann-Stampacchia Theorem). *Assume Ψ is a nonempty, closed and convex subset of a normed space $(X, \|\cdot\|_X)$ and $F : X \rightarrow X^*$ is monotone and hemicontinuous. If in addition either the set Ψ is bounded or F is coercive, that is, there exists $x_0 \in \Psi$ such that*

$$\lim_{\substack{\|x\|_X \rightarrow +\infty \\ x \in \Psi}} \frac{\langle F(x) - F(x_0), x - x_0 \rangle}{\|x - x_0\|_X} = +\infty.$$

The variational inequality (2.27) has a solution.

Remark 2.2.31. *If F is also strictly (or strongly) monotone, it is clear that the solution to variational inequality (2.27) is unique, assuming it exists.*

2.3 Reformulation of ℓ_1 -Norm Regularization in Compressive Sensing

The following result is deduced from the Lasso Problem (1.3) as described in Chapter 1 and is given as Remark 2.3.1 below.

Remark 2.3.1. *The difficulty of solving (1.3) is that the solution solely depends on the choice of the regularization parameter τ . In order to get rid of this difficulty, reformulation of (1.3) without a regularization parameter τ is essential. The corresponding formulation takes the following form of a vector optimization problem with two objective functions $f_1 := \|x\|_1$, $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}^1$, $f_2 := \|Ax - b\|_2^2$ and $f_2 : \mathbb{R}^n \rightarrow \mathbb{R}^1$ such that*

$$\min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} (f_1(x), f_2(x)), \quad (2.29)$$

where the function $f : \mathbb{R}^n \rightarrow \mathbb{R}^2$ is vector-valued and f_1 and f_2 are real valued functions. We will study vector optimization problems of type (2.29) in Chapter 7.

Iterative methods for solving Problem (1.3) have been presented in several papers, (see [25, 29, 58, 75]). Due to the fact that the proposed algorithms in this thesis are derivative-free, they can be applied to handle nonsmooth problems that are in the form of (1.10) effectively. Therefore, we solve the nonsmooth Problem (1.3) in a different way by converting it into the form of (1.10). Fortunately, Hu, Wang and Xiao [167] translated problem (1.3) into a nonlinear system of equations based on the work of Figueiredo, Nowak and Wright [58] as follows:

Let x be any vector in \mathbb{R}^n . Then x can be split into positive and negative parts, that is,

$$x = u - v, \quad u \geq 0, \quad v \geq 0,$$

where $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$ and $u_i = (x_i)_+$, $v_i = (-x_i)_+$, for all $i = 1, 2, \dots, n$ with $(\cdot)_+ = \max\{0, \cdot\}$. The ℓ_1 -norm of a vector x can be represented as $\|x\|_1 = e_n^T u + e_n^T v$, where $e_n = (1, 1, \dots, 1)^T \in \mathbb{R}^n$. Hence, the ℓ_1 -norm problem (1.3) is transformed as

$$\min_{u,v} \frac{1}{2} \|b - A(u - v)\|_2^2 + \tau e_n^T u + \tau e_n^T v, \quad \text{such that } \tau \geq 0, u \geq 0, v \geq 0. \quad (2.30)$$

However, from [58], problem (2.30) can also be rewritten as a quadratic programming problem with box constraints

$$\min_z \frac{1}{2} z^T D z + c^T z, \quad \text{such that } z \geq 0, \quad (2.31)$$

$$\text{where } z = \begin{bmatrix} u \\ v \end{bmatrix}, \quad c = \tau e_{2n} + \begin{bmatrix} -x \\ x \end{bmatrix}, \quad x = A^T b \quad \text{and} \quad D = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}.$$

Clearly, D is a positive semi-definite matrix, which implies that problem (2.31) is a convex quadratic problem.

Hu, Wang and Xiao [167] translated problem (2.31) into a linear variable inequality problem which is equivalent to a linear complementary problem. Furthermore, they pointed out that z is a solution of the linear complementary problem if and only if it is a solution of the nonlinear equation:

$$F(z) = \min\{z, Dz + c\} = 0, \quad (2.32)$$

where $F(\cdot)$ is said to be Lipschitz continuous and monotone, see [131, 167]. Hence, solving problem (1.3) is equivalent to solve Problem (1.10). Therefore, our proposed algorithms (see the algorithms proposed in Chapters 3 and 4) can be applied to solve problem (1.3) effectively (see, for example Chapter 6, in particular Section 6.1 for the application).

In the next section, let us discuss the main procedures for step-size determination in the frame of line search strategy for unconstrained optimization. After that an overview on solution methods for unconstrained optimization methods will be presented.

2.4 Line Search

Iterative techniques usually employ line search to determine step length. The goal of this strategy is to discover the ideal step length that minimizes or maximizes a specified objective function along a specific

search direction. Line search involves iteratively adjusting the step length to find a good solution. At each iteration, the objective function is evaluated at the new step length, and the step length is modified in accordance with the function value and specific parameters.

The Armijo rule (see, for example [12]), is a widely used criterion in unconstrained optimization problems for determining the step length. It entails multiplying the step length by a factor of less than one until an adequate reduction in the objective function is attained. The Wolfe condition [166] are a widely used criterion that requires examining the curvature and sufficient decrease conditions to guarantee that the step length is in the direction of the steepest descent and is sufficiently small.

Line search for finding step length is a strong technique that may be applied to a variety of optimization problems. It is especially beneficial for problem solving in optimization algorithms such as the popular Newton method, quasi-Newton method, conjugate gradient methods, and so on (1.10). The algorithm selects a direction d_k and the current iterate x_k and searches for a new iterate with a lesser function value. Starting with an initial guess, say x_0 , the iterations are generated via the following updating formula

$$x_{k+1} := x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \quad (2.33)$$

where $d_k \in \mathbb{R}^n$ is the **search direction** along which the values of f are reduced and $\alpha_k \in \mathbb{R}$ is the **step-size or step-length** determined by a line search procedure. To ensure the algorithm's effectiveness, the search direction d_k at iteration k must be a descent direction, with

$$d_k^T g(x_k) < 0, \quad (2.34)$$

In (2.34), $g(x_k) = \nabla f(x_k) = F(x_k)$ represents the gradient of f at a point x_k . To ensure **global convergence**, the iterative algorithm (2.33) requires a reasonable step-length α_k in the direction d_k to fulfill the **sufficient descent** condition, which states that

$$F(x_k)^T d_k \leq -t \|F(x_k)\|^2, \quad \text{where } t \text{ is a positive constant,} \quad (2.35)$$

(see, for example, [6, 135, 139, 179] and the references therein).

Algorithms for nonlinear equations aim to choose α_k that significantly reduce the norm of the function value, $\|F(x_k)\|$, in each iteration. Choosing a step-length α_k such that

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

results in an **exact line search**, with α_k being the **optimal step length**. If we select a step-length α_k such that

$$\|F(x_k + \alpha_k d_k)\| < \|F(x_k)\|.$$

This type of line search is referred to as an **inexact line search**, whereas α_k refers to the **approximate step-length**.

Obtaining the optimal step-size in actual computation is often extremely costly if not impossible. Inexact line search is a preferred alternative due to its lower computing effort. Solodov and Svaiter [150] developed an inexact line search strategy for nonlinear systems of equations. They claimed that the step length α_k should fulfil

$$-F(x_k + \alpha_k d_k)^T d_k \geq \lambda (1 - \rho_k) \mu_k \|d_k\|^2, \quad \mu_k > 0, \quad \rho_k \in [0, 1), \quad \lambda \in (0, 1). \quad (2.36)$$

This line search technique has undergone some adjustments in order to improve its efficiency, see [18, 39, 70, 72, 175]. Finding a suitable stepsize α_k using an inexact line search can be challenging, as it must avoid being too long or too short. Inexact line search methods focus on selecting a suitable stepsize, ensuring that α_k is neither too long or too short, and creating a sequence of updates that meets these condition.

2.5 Overview of Solution Methods for Unconstrained Optimization Problems

In this section, we give some of the most important solution methods for unconstrained optimization problems based on the gradient computation, insisting on their definition, their advantages and disadvantages, as well as on their convergence properties. The main difference among these methods is the procedure for the computation of the search direction d_k . For stepsize α_k computation, the most used procedure is that of Wolfe (standard). The following methods are discussed: the steepest descent, Newton, quasi-Newton, conjugate gradient, spectral gradient and spectral-conjugate gradients. Before discussing the solution methods, we begin by recalling the formulation of a general optimization problem and give an overview of optimality conditions with respect to solution methods.

Many applications give rise to the unconstrained optimization problem of the form of (1.8) given by

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

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function at $x \in \mathbb{R}^n$ and bounded from below. By continuously differentiable, we mean $\left(\frac{\partial f}{\partial x_i}\right)(x)$ exists and is continuous, $i = 1, 2, \dots, n$. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice continuously differentiable at $x \in \mathbb{R}^n$ if $\left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)(x)$ exists and is continuous. In order to formulate optimality conditions for problem (1.8), it is necessary to introduce some concepts which characterize an improving direction along which the values of the function f decrease.

Definition 2.5.1. (Descent direction, Andrei [11]). Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous at x^* . A vector $d \in \mathbb{R}^n$ is a descent direction for f if there exists $\delta > 0$ such that $f(x^* + \lambda d) < f(x^*)$ for any $\lambda \in (0, \delta)$. The cone of descent directions at x^* , denoted by $K_{dd}(x^*)$ is given by

$$K_{dd}(x^*) := \{d : \text{there exists } \delta > 0 \text{ such that } f(x^* + \lambda d) < f(x^*), \text{ for any } \lambda \in (0, \delta)\}.$$

Assume that f is a differentiable function. Then, to get an algebraic characterization for a descent direction for f at x^* , we define the set

$$K_0(x^*) := \{d : \nabla f(x^*)^T d < 0\}.$$

The following result shows that every $d \in K_0(x^*)$ is a descent direction at x^* .

Proposition 2.5.2. (Algebraic Characterization of a Descent Direction, Andrei [11]). Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable at x^* . If there exists a vector d such that $\nabla f(x^*)^T d < 0$, then d is a descent direction for f at x^* , that is, $K(x^*) \subseteq K_{dd}(x^*)$.

In the next results, we provide the first and second-order necessary conditions for a local minimum. However, one can also use Theorem 2.2.26(2) in place of Theorem 2.5.3, because both theorems are saying the same thing.

Theorem 2.5.3. (*First-Order Necessary Condition for a Local Minimum, Andrei [11]*). Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable at x^* . If x^* is a local minimum, then $\nabla f(x^*) = 0$.

Theorem 2.5.4. (*Second-Order Necessary Condition for a Local Minimum, Andrei [11]*). Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice differentiable at x^* . If x^* is a local minimum, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semi-definite.

In the above theorems, we have presented the *necessary* conditions for a point x^* to be a local minimum. However, a point satisfying these necessary conditions need not be a local minimum. In the following theorems, the *sufficient* conditions for a global minimum are given, provided that the objective function is *convex* on \mathbb{R}^n .

The following theorem shows that the convexity assumption is crucial in global nonlinear optimization.

Theorem 2.5.5. (*First-Order Sufficient Conditions for a Strict Local Minimum, Andrei [11]*). Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable at x^* and convex on \mathbb{R}^n . If $\nabla f(x^*) = 0$, then x^* is a global minimum of f on \mathbb{R}^n .

The following theorem gives the second-order sufficient conditions characterizing a local minimum point for those functions which are strictly convex in a neighborhood of the minimum point.

Theorem 2.5.6. (*Second-Order Sufficient Conditions for a Strict Local Minimum, Andrei [11]*). Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice differentiable at point x^* . If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, then x^* is a local minimum of f .

2.5.1 Steepest Descent Method

The steepest descent method is a fundamental approach for solving unconstrained optimization problems. The simplest method, proposed by Cauchy in 1847 (see [35]), requires the search direction d_k to be computed as

$$d_k := -g(x_k), \quad (2.37)$$

where $g(x_k)$ is the gradient of the function f at a point x_k . At the current point x_k , the negative gradient points in the most optimal path to search for the minimum of the function f . However, once we start moving in this direction, it no longer remains the optimal choice and gradually worsens until it becomes orthogonal to $-g(x_k)$, meaning that the procedure starts taking little steps without making substantial progress towards the minimum. The main disadvantage of this is that the steps it takes are too long. In other words, there are some points z_k on the line segment linking the points x_k and x_{k+1} , where $-\nabla f(z_k)$ gives a more optimal new search direction than $-\nabla f(x_{k+1})$. The steepest descent method exhibits global convergence when used with a wide range of inexact line search procedures. In addition, the rate at which it converges is linear (a restriction) and it is significantly impacted by ill-conditioning (Akaike,

[5]). The rate at which this method converges is heavily influenced by the distribution of the eigenvalues of the Hessian matrix of the function being minimized.

The algorithm exhibits global convergence to a local minimizer regardless of the initial starting point x_0 . Many other algorithms for optimization employ the steepest descent approach when they fail to achieve significant advancements. However, it also has the following drawbacks. The property of scale invariance is not present, meaning that altering the scalar product on \mathbb{R}^n will result in a modification of the concept of gradient. In addition, it typically exhibits a significant degree of slowness, namely in terms of its convergence rate being linear. In terms of numerical analysis, it frequently lacks convergence entirely. Andrei [7] provided an enhanced version of the steepest descent method with backtracking, which was further examined by Babaie-Kafaki and Rezaee [20].

2.5.2 Newton Methods

The Newton method is based on the quadratic approximation of the function f and the exact minimization of this quadratic approximation. Therefore, in the vicinity of the current iteration point x_k , the function f is estimated using the truncated Taylor series

$$f(x) \cong f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2} (x - x_k)^T \nabla^2 f(x_k) (x - x_k), \quad (2.38)$$

known as the *local quadratic model* of f about x_k . The search direction of the Newton method is calculated by minimizing the right-hand side of equation (2.38) using the formula

$$d_k := -\nabla^2 f(x_k)^{-1} g(x_k). \quad (2.39)$$

Therefore, the Newton method is defined as

$$x_{k+1} := x_k - \alpha_k \nabla^2 f(x_k)^{-1} g(x_k), \quad k = 0, 1, \dots, \quad (2.40)$$

where α_k represents the stepsize. In the context of the Newton method (2.40), it can be observed that d_k is a descent direction if and only if $\nabla^2 f(x_k)$ is a positive definite matrix. If the initial point x_0 is in close proximity to x^* , then $\{x_k\}$ generated by the Newton method converges to x^* at a quadratic rate.

The primary limitation of this method is the requirement to calculate and store the Hessian matrix, which is an $n \times n$ matrix. Occasionally, during the iteration process, the Hessian $\nabla^2 f(x_k)$ may become singular or non-positive definite. If the Hessian matrix is singular at the solution point, the quadratic convergence property of the Newton method is compromised. To address this situation, the solution is to choose a positive definite matrix M_k in a manner that ensures that the sum of the Hessian matrix $\nabla^2 f(x_k)$ and M_k is positive definite enough. Then, we may solve the equation $(\nabla^2 f(x_k) + M_k)d_k = -g(x_k)$ to find the solution. The selection of the regularization term M_k is commonly based on the spectral decomposition of the Hessian. An alternative approach to improving the Newton method involves utilizing the modified Cholesky factorization, as described by Gill and Murray [66]. Certainly, the Newton approach is unsuitable for solving problems of significant magnitude. Furthermore, if the solution is located far outside the region of convergence, the Newton method is likely to diverge. Put simply, the Newton method lacks the property of global convergence. This is because the search direction (2.39) may not be an acceptable descent direction, even if $g(x_k)^T d_k < 0$. Additionally, using a unit stepsize might not result in a

decrease in the function values when minimizing. To address this issue, one should employ globalization strategies. The first method is the line search, which adjusts the size of the step. The second method is the trust-region, which affects both the step size and the direction. Furthermore, after each iteration, the Newton method requires the computation of the Hessian matrix $\nabla^2 f(x_k)$, which can be difficult, particularly for problems with large-scale and when seeking to solve a linear system. An alternative approach is to substitute the analytical Hessian with a finite difference approximation, as demonstrated by Sun and Yuan [153]. However, this approach is costly because n additional evaluations of the minimizing function are required at each iteration. Quasi-Newton algorithms can be employed to decrease the computational effort. These techniques produce estimations of the Hessian matrix by using the information gathered from the previous iterations. In order to eliminate the need to solve a linear system for the computation of the search direction, one can use variants of the quasi-Newton methods which generate approximations to the inverse Hessian. Regardless, the Newton approach is the most optimal when executed.

2.5.3 Quasi-Newton Methods

It is commonly known that the Newton method, the most famous method for solving unconstrained optimization problems, computes its search direction using exact Hessian matrices. However, for a variety of problems, computing the Hessian matrices is quite expensive, and the Hessian may not always be available analytically. Quasi-Newton methods rely solely on the gradients of the objective function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, to construct a sequence of Hessian approximations. They also ensure a relatively rapid rate of convergence.

Quasi-Newton search directions provide an attractive alternative to Newton's approach by eliminating the need for correct computation of the exact Hessian matrices while yet achieving a superlinear rate of convergence. Instead of using the exact Hessian $\nabla^2 f(x_k)$, an approximation A_k of the Hessian is employed, which is updated at each iteration to incorporate the additional information gained during the preceding iteration. The update utilizes the fact that changes in the gradient $\nabla f(x)$ yield insights into the second derivative of f along the search direction.

The quasi-Newton search direction is obtained by using the formula

$$d_k := -A_k^{-1}g(x_k), \quad (2.41)$$

where the matrix A_k is an approximation of $\nabla^2 f(x_k)$ such that the following equation is satisfied

$$A_{k+1}s_k = y_k, \quad (2.42)$$

with $y_k := g(x_{k+1}) - g(x_k)$ and $s_k = x_{k+1} - x_k$. Equation (2.42) is commonly referred to as the quasi-Newton equation or the secant equation. There are several formulae for updating the value of A_k but the most widely used ones are the well-known Broyden–Fletcher–Goldfarb–Shanno (BFGS) and Symmetric Rank–One (SR1) methods, respectively. These methods are defined as follows:

$$A_{k+1} := A_k - \frac{A_k s_k s_k^T A_k}{s_k^T A_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \quad (2.43)$$

$$A_{k+1} := A_k + \frac{(y_k - A_k s_k)(y_k - A_k s_k)^T}{(y_k - A_k s_k)^T s_k}. \quad (2.44)$$

The two updating formulae (2.43) and (2.44) produce matrices that exhibit symmetry. If the initial approximation A_0 is positive definite and $s_k^T y_k > 0$, then the BFGS update (2.43) generates positive definite approximations of Hessian matrix. However, the SR1 update (2.44) does not guarantee positive definiteness.

If $Q_k = A_k^{-1}$ represents the approximated inverse of the Hessian matrix of function f at point x_k , then the BFGS and SR1 formulas are defined as follows:

$$Q_{k+1} = \left(I - \frac{s_k y_k^T}{s_k^T y_k} \right) Q_k \left(I - \frac{y_k s_k^T}{s_k^T y_k} \right) + \frac{s_k s_k^T}{s_k^T y_k} \quad (2.45)$$

$$Q_{k+1} = Q_k + \frac{(s_k - Q_k y_k)(s_k - Q_k y_k)^T}{(s_k - Q_k y_k)^T y_k}. \quad (2.46)$$

Quasi-Newton methods exhibit local superlinear convergence, and when combined with appropriate line search techniques, they achieve global convergence. Other quasi-Newton updating formulae include the Davidon–Fletcher–Powell (DFP) method, as well as diagonal updating formula, among others. One important drawback of the quasi-Newton approach is its avoidance of computing an exact Hessian matrix and solving linear systems of equations in each iteration, which are significant limitations of the Newton’s method. Nevertheless, the most crucial aspect of several quasi-Newton methods is the generation and subsequent storage of an $n \times n$ matrix approximation of the Hessian matrix, which can be an exceedingly costly task when solving problems of large scale. As a result, several researchers have modified certain quasi-Newton search directions to imitate the characteristics of widely used conjugate gradient algorithms. Conjugate gradient methods are efficient in handling large-scale problems because they do not need to generate or store matrices during the iteration processes.

Zhang et al. [175] proposed a three-term conjugate gradient algorithm for addressing general unconstrained optimization problems using the BFGS updating formula and the Polak–Ribière–Polyak conjugate gradient parameter [135, 137]. Their method was proven to be globally convergent using the Armijo-type line search. Furthermore, Andrei [10] introduced a simple three-term conjugate gradient technique for solving unconstrained optimization problems, which relies on the BFGS updating formula. The search direction given in reference [10] meets both the descent and conjugacy criterion, regardless of the line search approach employed. Awwal et al. [16] proposed a three-term derivative-free method that effectively solves problems (1.10) with convex constraints, using a similar approach. Their approach was created by integrating the modified Perry conjugate gradient parameter [134] into the modified BFGS updating formula. Their algorithm was also utilized to restore some disturbed signals. Abubakar et al. [4] recently introduced a new method for solving Problem (1.10) with convex constraints. This method is based on a modified scaled SR1 updating formula and the projection technique of Solodov and Svaiter [150], and it does not require the use of derivatives. Their method’s convergence analysis was established by relying on the assumption of monotonicity and Lipschitz continuity of the underlying mapping.

2.5.4 Conjugate Gradient Method

Conjugate gradient methods are widely used iterative techniques for optimizing large-scale problems. This is because they do not create or store matrices during the iteration process. The linear conjugate gradient approach was initially introduced by Hestenes and Stiefel [82] in the 1950s as a method to solve

linear systems that are symmetric and positive-definite. In the 1960s, Fletcher and Reeves [61] modified the methods used to minimize a positive definite quadratic function, resulting in the first nonlinear conjugate gradient method for unconstrained minimization problems.

Conjugate gradient algorithms possess the qualities of low memory usage, simple implementation, and global convergence properties. Given an initial point $x_0 \in \mathbb{R}^n$, a nonlinear conjugate gradient method generates a sequence of iterates $\{x_k\}$ using the update formula $x_{k+1} = x_k + \alpha_k d_k$, $k = 0, 1, 2, \dots$. The search direction d_k is determined according to the following rule:

$$d_k := \begin{cases} -g(x_k), & \text{if } k = 0, \\ -g(x_k) + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \quad (2.47)$$

$g(x_k)$ represents the gradient of the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at x_k and β_k is the conjugate gradient parameter that is updated at each iteration. The conjugate gradient parameter β_k can be calculated using many formulae, and each choice of β_k corresponds to a distinct conjugate gradient parameter. Whenever the value of $\beta_k = 0$, the conjugate gradient method employs the widely recognized steepest descent method to calculate its search direction d_k . The selection of the conjugate gradient parameter β_k is a well-established fact that significantly impacts the numerical performance of the conjugate gradient method.

Remark 2.5.7. *In the context of a system of nonlinear equations, the gradient $g(x_k)$ of the function f is substituted by the value of the function F at the point x_k , that is, $F(x_k)$. Recall that F is a vector-valued. This assertion is from the first-order optimality conditions (see, for example, Theorem 2.2.26(2) or Theorem 2.5.3), that is, x^* is a minimizer of $f \implies \nabla f(x^*) = 0$. Hence, we consider the system of nonlinear equation $F(x) = 0$, where the gradient of f is viewed as the function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, that is, $F := \nabla f$.*

A variety of nonlinear conjugate gradient algorithms are known. There are two main differences between them: the method used to update the search direction and the mechanism for computing the step size along this direction. The primary criterion for the search direction in conjugate gradient methods is to fulfill the descent-like or sufficient descent-like condition. The stepsize is determined by using the Wolfe line search conditions or certain variations of them. The conjugate gradient algorithms can be categorized into various types, including standard, hybrid, modified versions of the standard conjugate gradient algorithms, memoryless BFGS preconditioned, three-term conjugate gradient algorithms, among others.

The most significant standard conjugate gradient methods include Fletcher and Reeves (FR) [61] methods, the Dai and Yuan (DY) [46] method and the Conjugate Descent (CD) method proposed by Fletcher [59]. These methods with the specified parameters (where $g(x_{k-1}) \neq 0$, $d_{k-1}^T y_{k-1} \neq 0$ and $-d_{k-1}^T g(x_{k-1}) \neq 0$):

$$\beta_k^{FR} := \frac{\|g(x_k)\|^2}{\|g(x_{k-1})\|^2}, \quad \beta_k^{DY} := \frac{\|g(x_k)\|^2}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{CD} := \frac{\|g(x_k)\|^2}{-d_{k-1}^T g(x_{k-1})},$$

exhibit strong convergence properties but may have limited practical performance due to jamming [8]. However, the Polak–Ribière [135] and Polyak (PRP) [137] methods, the Hestenes and Stiefel (HS) [82]

method and the Liu and Storey (LS) [115] method with the parameters:

$$\beta_k^{PRP} := \frac{g(x_k)^T y_{k-1}}{\|g(x_{k-1})\|^2}, \quad \beta_k^{HS} := \frac{g(x_k)^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{LS} := \frac{g(x_k)^T y_{k-1}}{-d_{k-1}^T g(x_{k-1})},$$

may not always converge, but they often exhibit superior computational efficiency [8]. The denominators of the CG parameters FR, DY, CD, PRP, HS, and LS are presumed to be non-zero. Conjugate gradient methods using the above parameters do not always guarantee a direction of descent without the inclusion of a line search procedure. If the objective function is a strongly convex quadratic and the line search is performed exactly, then, theoretically, all possible selections for the search direction in standard conjugate gradient algorithms yield the same results. However, when dealing with non-quadratic functions, the choice of the search direction results in standard conjugate gradient algorithms that exhibit varying levels of performance.

Hybrid conjugate gradient algorithms aim to combine standard conjugate gradient methods to take advantage of the attractive features of each method. Hybrid conjugate gradient algorithms can be obtained by combining the standard schemes in two distinct manners. The first combination is based on the concept of projection. The concept behind these methods involves the use of a pair of standard conjugate gradient methods, with one of them being employed when a specific requirement is met. Once the condition has been violated, the alternative standard conjugate gradient from the pair is employed. The second category of hybrid conjugate gradient methods is derived from the convex combination of the standard methods. The purpose of each of these techniques is to select a pair of standard methods and merge them in a convex manner, with the parameter in the convex combination determined through the use of either the conjugacy condition or the Newton search direction. Generally speaking, hybrid methods that use a convex combination of standard schemes performs better than hybrid methods that rely on the projection concept. Hybrid methods exhibit superior efficiency and robustness compared to standard methods.

A significant category of conjugate gradient algorithms is derived by modifying the standard methods. It is possible to improve the numerical efficiency of any standard conjugate gradient method by modifying it to ensure that the search direction is always in the direction of descent. The most effective conjugate gradient algorithms currently available include the CGDESCENT algorithm developed by Hager and Zhang [72], and the DESCON algorithm developed by Andrei [9]. CG-DESCENT is an algorithm that uses conjugate gradient methods and ensures that the objective function decreases at each iteration. CG-DESCENT can be considered as a modified version of the Dai and Liao conjugate gradient method, where its parameter is assigned a specific value to make it adaptive. The search direction of CG-DESCENT is linked to the memoryless quasi-Newton direction of Perry-Shanno. DESCON is an iterative optimization technique that ensures both descent and conjugacy criteria, and incorporates a modified Wolfe line search. Primarily, it is a modification of the Hestenes–Stiefel conjugate gradient algorithm.

In 1976, Perry established in [134] the initial link between the conjugate gradient algorithms and the quasi-Newton algorithms. According to Perry, the Hestenes–Stiefel search direction is a matrix that multiplies the negative gradient. In 1978, Shanno [149] demonstrated that the conjugate gradient methods are precisely equivalent to the BFGS quasi-Newton methods. Each iteration of those methods resets the approximation to the inverse Hessian to the identity matrix. We can describe conjugate gradient methods

as memoryless quasi-Newton methods. This marked the inception of a highly productive field of study focused on memoryless quasi-Newton conjugate gradient algorithms. The problem at hand is how to include the second-order information from the minimizing function in the procedure for updating the search direction.

Beale [23] and Nazareth [127] proposed the three-term conjugate gradient algorithms. The convergence rate of the conjugate gradient method may be improved from linear to n -step quadratic by restarting the method with the negative gradient direction after n iterations. One such restart technique was proposed by Beale [23]. During the restarting method, the restart direction is determined by combining the negative gradient with the prior search direction. This previous search direction includes information about the second-order derivative, which is obtained by searching along the previous direction. Therefore, a conjugate gradient with three terms was obtained. Nazareth [127] suggested a conjugate gradient approach that utilizes three terms for the search direction, ensuring limited convergence for an arbitrary initial search direction. There are numerous algorithms for three-term conjugate gradients that are well-known. The concept of a three-term conjugate gradient is an interesting development. Nevertheless, the numerical performance of these methods is modest.

2.5.5 Spectral Gradient Method

The spectral gradient method is an appropriate strategy for handling large-scale unconstrained optimization problems. The first implementation of the spectral gradient approach is attributed to Barzilai and Bowein [22] for unconstrained optimization problems. Later, La Cruz and Raydan [108] extended this method to solve large-scale systems of equations. One attractive feature of this method is its simplicity in implementation and its property of achieving global convergence.

Spectral gradient method generates its sequence of iterates $\{x_k\}$ using $x_{k+1} := x_k + \alpha_k d_k$, $k = 0, 1, 2, \dots$, where the search direction d_k is defined as

$$d_k := \begin{cases} -g(x_k), & \text{if } k = 0, \\ -\lambda_k g(x_k), & \text{if } k \geq 1, \end{cases} \quad (2.48)$$

$\lambda_k > 0$ is the spectral gradient parameter and $g(x_k)$ is the gradient of the function that has to be minimized. The scalar parameter λ_k should be strictly positive. The spectral gradient method becomes the steepest descent method when $\lambda_k = 1$. Spectral gradient methods differ depending on the spectral parameter λ_k used.

2.5.6 Spectral–Conjugate Gradient Methods

Liu and Li [113] suggested that the spectral gradient method may cause the iterates to zigzag, which is a similar drawback to the steepest descent method. In order to address this limitation, it is logical to combine the spectral gradient parameter and the conjugate gradient parameter to create a formula that generates the search direction d_k . This search direction is a combination of a steepest descent and the

previous search direction. The formula is defined as follows

$$d_k := \begin{cases} -g(x_k), & \text{if } k = 0, \\ -\lambda_k g(x_k) + \beta_k d_{k-1}, \quad \lambda_k, \beta_k > 0, & \text{if } k \geq 1. \end{cases} \quad (2.49)$$

The iterative techniques that generate the search direction d_k described in (2.49) are called spectral conjugate gradient methods. If $\lambda_k = 1$ for all $k > 0$, the search direction d_k in (2.49) reduces to the conjugate gradient search direction d_k defined in (2.47), while if $\beta_k = 0$, the spectral-conjugate gradient search direction (2.49) reduces to the spectral gradient search direction (2.48).

Chapter 3

Derivative-Free Algorithms for Nonlinear Systems of Equations

In this chapter, we discuss the first part of our main results. Most of the results of this chapter are from our work published in Muhammad et al., [1, 123, 124]. We begin by recalling the problem under consideration. Iterative methods such as Newton methods, quasi-Newton methods and conjugate gradient methods, have widely been successfully applied to deal with the general unconstrained optimization problems of the form (1.8), that is,

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

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function and bounded from below. Unlike quasi-Newton methods, which require the storage of $n \times n$ matrices in every iteration, conjugate gradient (CG) methods require the storage of $n \times 1$ vectors at each iteration. Moreover, the CG methods are of particular interest due to their global convergence property and simplicity of implementation.

The CG method computes its sequence of iterates $\{x_k\}$ using the following updating formula:

$$x_{k+1} := x_k + \alpha_k d_k, \quad k \geq 0.$$

The scalar $\alpha_k > 0$ denotes the step-length that is obtained via a suitable line-search strategy and the vector d_k is a search direction given by

$$d_0 := -F(x_0) \quad \text{and} \quad d_k := -F(x_k) + \beta_k d_{k-1}, \quad \text{for all } k \in \mathbb{N},$$

where $F := \nabla f$ and β_k (called the CG parameter) is a scalar that is updated in every iteration.

Under the differentiability assumption, a well-known necessary optimality condition for problem (1.8) is:

$$x^* \text{ is a minimizer of } f \implies \nabla f(x^*) = 0.$$

This fact is called the first-order necessary optimality condition for the minimizer x^* of problem (1.8).

Therefore, we consider the following system of nonlinear equations:

In this context, we interpret the gradient of f in Problem (1.8) as the operator F in Problem (1.10), that is, $F := \nabla f$.

In the literature, many algorithms have been proposed and shown to be efficient for solving a convex constrained monotone operator equation of the form (1.10). However, some of these iterative algorithms may not be well-defined, have poor numerical performance, cannot generate descent search directions and therefore, fail to converge globally to the solution of Problem (1.10). As a result, we need to improve these iterative algorithms by either modifying the search directions (or proposing new search directions) and introducing new parameters so that we can have algorithms that are well-defined, satisfying the descent-like conditions, converge globally to the solution of Problem (1.10) and exhibit good numerical convergence. Sections 3.1, 3.2, 3.3 and 3.4 clearly demonstrate this.

This chapter is divided into four sections where Section 3.1 proposes a Modified Dai-Yuan (MDY1 and MDY2) algorithm. Section 3.2 introduces a Hybrid Conjugate Descent and a Liu and Storey algorithm simply represented by HCDLS algorithm while Section 3.3 establishes a Three-term Dai-Liao Projection algorithm denoted by TDLP algorithm. Lastly, we present a convergence analyses of the proposed algorithms of Sections 3.1, 3.2 and 3.3 in Section 3.4.

The following Assumption 3.0.1 is a standing assumption in the whole chapter.

Assumption 3.0.1. Consider problem (1.10) and suppose the following conditions on the underlying mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, which are useful in the analyses of our proposed algorithms throughout this chapter.

- A1. The solution set χ of Problem (1.10) is nonempty.
- A2. The mapping F is Lipschitz continuous.
- A3. The mapping F is monotone.
- A4. The mapping F is pseudomonotone.

3.1 Algorithm 1: Modified Dai–Yuan Projection Methods (MDY)

In [46], Dai and Yuan proposed a conjugate gradient algorithm for solving the unconstrained optimization problem of the form of (1.8). Dai and Yuan's CG method defined their search direction d_k , in the following way:

$$d_k := \begin{cases} -g(x_k) & \text{if } k = 0, \\ -g(x_k) + \beta_k^{DY} d_{k-1} & \text{if } k \geq 1, \end{cases} \quad (3.1)$$

where β_k^{DY} is a scalar called the CG parameter by Dai and Yuan method and is defined as

$$\beta_k^{DY} := \frac{\|g(x_k)\|^2}{d_{k-1}^T y_{k-1}},$$

and $g(x_k) := \nabla f(x_k)$ is the gradient of f at x_k , with $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\forall k \in \mathbb{N}_0$.

Liu and Feng [112] proposed a modified version of the Dai and Yuan's [46] conjugate gradient method (known as PDY) for solving monotone system of nonlinear equations of the form of (1.10) that is,

$$F(x) = 0 \text{ such that } x \in \Psi \subseteq \mathbb{R}^n,$$

where the function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is assumed to be Lipschitz continuous and monotone. The search direction for the PDY method is given by

$$d_k^{PDY} := \begin{cases} -F(x_k) & \text{if } k = 0, \\ -\theta_k F(x_k) + \beta_k^{PDY} d_{k-1}^{PDY} & \text{if } k \geq 1, \end{cases} \quad (3.2)$$

where

$$\beta_k^{PDY} := \frac{\|F(x_k)\|^2}{w_{k-1}^T d_{k-1}^{PDY}}, \quad \theta_k := c + \frac{F(x_k)^T d_{k-1}^{PDY}}{w_{k-1}^T d_{k-1}^{PDY}}, \quad c > 0, \quad (3.3)$$

$$w_{k-1} := y_{k-1} + t_{k-1} d_{k-1}^{PDY},$$

$$y_{k-1} := F(x_k) - F(x_{k-1}) \text{ and } t_{k-1} := 1 + \max \left\{ 0, -\frac{y_{k-1}^T d_{k-1}^{PDY}}{\|d_{k-1}^{PDY}\|^2} \right\}$$

We quickly give the following useful remark.

Remark 3.1.1. *It is easy to see that from the definition of d_k^{PDY} in (3.2) and w_{k-1} in (3.3) that*

$$w_{k-1}^T d_{k-1}^{PDY} \geq y_{k-1}^T d_{k-1}^{PDY} + \|d_{k-1}^{PDY}\|^2 - y_{k-1}^T d_{k-1}^{PDY} = \|d_{k-1}^{PDY}\|^2. \quad (3.4)$$

Following the success recorded by the work of Liu and Feng [112], we propose a new modification of the Dai and Yuan [46] conjugate gradient method for generating a solution of Problem (1.10). This new modification has helped greatly in improving the numerical performance of Dai and Yuan's method (see, for example, Section 4 of Abubakar et al. [1] or Section 5.1.1 of this thesis) as well as in applying the new method to compressive sensing (see also, Section 4.1 of Abubakar et al., [1] or Section 6.1 of this thesis). We refer to this method as **A Modified Dai–Yuan (MDY) Method** where the proposed new search direction is defined as

$$d_k := \begin{cases} -F(x_k) & \text{if } k = 0, \\ -F(x_k) + \beta_k^{PDY} d_{k-1} - \theta_k F(x_k) & \text{if } k \geq 1, \end{cases} \quad (3.5)$$

where for all $k \in \mathbb{N}$

$$\beta_k^{PDY} := \frac{\|F(x_k)\|^2}{d_{k-1}^T w_{k-1}}, \quad (3.6)$$

$$w_{k-1} := y_{k-1} + t_{k-1} d_{k-1}, \quad y_{k-1} := F(x_k) - F(x_{k-1}) + r s_{k-1},$$

$$s_{k-1} := x_k - x_{k-1}, \quad t_{k-1} := 1 + \max \left\{ 0, -\frac{d_{k-1}^T y_{k-1}}{d_{k-1}^T d_{k-1}} \right\}.$$

In our new approach, the coefficient θ_k in (3.5) will be obtained in two different ways, such that, d_k satisfies the descent-like condition given by the inequality

$$F(x_k)^T d_k \leq -t \|F(x_k)\|^2, \quad (3.7)$$

where t is a positive constant.

In what follows, we define two new different modified Dai–Yuan descent-like search directions for non-linear monotone operator equations depending on the value of θ_k .

First, we introduce the modified Dai-Yuan search direction called **MDY1 Direction**. From the definition of d_k in (3.5) and β_k^{PDY} in (3.6), we obtain for all $k \in \mathbb{N}$

$$\begin{aligned} F(x_k)^T d_k &= -\|F(x_k)\|^2 + \frac{\|F(x_k)\|^2}{d_{k-1}^T w_{k-1}} F(x_k)^T d_{k-1} - \theta_k \|F(x_k)\|^2 \\ &= \frac{-\|F(x_k)\|^2 (d_{k-1}^T w_{k-1})^2 + \|F(x_k)\|^2 (d_{k-1}^T w_{k-1}) F(x_k)^T d_{k-1} - \theta_k \|F(x_k)\|^2 (d_{k-1}^T w_{k-1})^2}{(d_{k-1}^T w_{k-1})^2}. \end{aligned} \quad (3.8)$$

Taking $u := \frac{1}{\sqrt{2}}(d_{k-1}^T w_{k-1})F(x_k)$, $v := \sqrt{2}\|F(x_k)\|^2 d_{k-1}$ and using the following relation

$$u^T v \leq \frac{1}{2}(\|u\|^2 + \|v\|^2),$$

we conclude from (3.8) that

$$\begin{aligned} F(x_k)^T d_k &\leq -\frac{3}{4} \frac{\|F(x_k)\|^2 (d_{k-1}^T w_{k-1})^2}{(d_{k-1}^T w_{k-1})^2} + \frac{\|F(x_k)\|^4 \|d_{k-1}\|^2 - \theta_k \|F(x_k)\|^2 (d_{k-1}^T w_{k-1})^2}{(d_{k-1}^T w_{k-1})^2} \\ &= -\frac{3}{4} \|F(x_k)\|^2 + \frac{\|F(x_k)\|^4 \|d_{k-1}\|^2}{(d_{k-1}^T w_{k-1})^2} - \theta_k \|F(x_k)\|^2. \end{aligned}$$

Choosing

$$\theta_k := \frac{\|F(x_k)\|^2 \|d_{k-1}\|^2}{(d_{k-1}^T w_{k-1})^2}, \quad (3.9)$$

it follows that

$$F(x_k)^T d_k \leq -t_1 \|F(x_k)\|^2, \quad \text{where } t_1 = \frac{3}{4}. \quad (3.10)$$

Definition 3.1.2. For every $k \in \mathbb{N}$, the search direction d_k given by (3.5), together with β_k^{PDY} given by (3.6) and θ_k given by (3.9) is called the *MDY1 direction*.

Furthermore, we introduce a second modified Dai-Yuan search direction called the **MDY2 Direction** analogously to the procedure given above. Taking into account (3.8) and choosing

$$\theta_k := \frac{F(x_k)^T d_{k-1}}{d_{k-1}^T w_{k-1}}, \quad (3.11)$$

for all $k \in \mathbb{N}$, it follows that $F(x_k)^T d_k = -\|F(x_k)\|^2$. So, the inequality (3.7) holds with $t = 1$.

Now, the coefficient θ_k of the MDY2 search direction in (3.5) is obtained in such a way that it consists of two parts: the first part is (3.11), while the second part is chosen such that the inequality (3.7) is satisfied, we deduce that

$$\theta_k := \frac{F(x_k)^T d_{k-1}}{d_{k-1}^T w_{k-1}} + \frac{\|F(x_k)\|^2}{(d_{k-1}^T w_{k-1})^2}. \quad (3.12)$$

Using θ_k obtained in (3.12), d_k given in (3.5) and β_k^{PDY} given in (3.6), the inequality (3.7) holds in the following

$$\begin{aligned} F(x_k)^T d_k &= -\|F(x_k)\|^2 + \frac{\|F(x_k)\|^2 F(x_k)^T d_{k-1}}{d_{k-1}^T w_{k-1}} - \left(\frac{F(x_k)^T d_{k-1}}{d_{k-1}^T w_{k-1}} + \frac{\|F(x_k)\|^2}{(d_{k-1}^T w_{k-1})^2} \right) \|F(x_k)\|^2 \\ &= -\|F(x_k)\|^2 - \frac{\|F(x_k)\|^4}{(d_{k-1}^T w_{k-1})^2} \\ &\leq -t_2 \|F(x_k)\|^2, \quad \text{with } t_2 = 1. \end{aligned} \quad (3.13)$$

Definition 3.1.3. For every $k \in \mathbb{N}$, the direction d_k given by (3.5), together with β_k^{PDY} given by (3.6) and θ_k given by (3.12) is called the MDY2 direction.

Note: One can quickly choose three other different values of θ_k such that the inequality (3.7) holds. The three choices are:

- (i) $\bar{\theta}_k := \frac{F(x_k)^T d_{k-1}}{d_{k-1}^T w_{k-1}}$,
- (ii) $\hat{\theta}_k := \frac{\|F(x_k)\|}{\|d_{k-1}\|}$, $\|d_{k-1}\| \neq 0$, (this follows from the Cauchy–Schwarz inequality and Remark 3.1.1),
- (iii) $\tilde{\theta}_k := \frac{\|F(x_k)\|^2 \|d_{k-1}\|^2}{(d_{k-1}^T w_{k-1})^2}$, (this follows from squaring (i) and the Cauchy–Schwarz inequality).

Now, we state the Modified Dai–Yuan (MDY) algorithm for solving Problem (1.10), see Muhammad et al., [1] for more details about Algorithm 1 (MDY).

Algorithm 1: Modified Dai–Yuan (MDY) Algorithm [for solving Problem (1.10)]

Input: Given an initial point $x_0 \in \Psi$ and parameters $\kappa > 0$, $r > 0$, $0 < \gamma < 2$, $\sigma > 0$, $0 < \rho < 1$, $Tol > 0$. Set $k = 0$.

Step 1: Compute $F(x_k)$ and set $d_k := -F(x_k)$.

Step 2: If $\|F(x_k)\| \leq Tol$, x_k is a solution and the iteration process stops.

Step 3: Set

$$v_k := x_k + \eta_k d_k, \quad (3.14)$$

where $\eta_k := \kappa \rho^i$, such that i is the smallest nonnegative integer satisfying

$$-\langle F(x_k + \kappa \rho^i d_k), d_k \rangle \geq \sigma \kappa \rho^i \|d_k\|^2 \|F(x_k + \kappa \rho^i d_k)\|^{1/q}, \quad q \geq 1. \quad (3.15)$$

Step 4: If $\|F(v_k)\| \leq Tol$, stop. Else, compute the next iterate

$$x_{k+1} := P_\Psi \left[x_k - \gamma \frac{\langle F(v_k), x_k - v_k \rangle}{\|F(v_k)\|^2} F(v_k) \right], \quad \|F(v_k)\| \neq 0. \quad (3.16)$$

Step 5: Set $k := k + 1$ and update d_k using (3.5), (3.6) with (3.9) for MDY1 direction or with (3.12) for MDY2 direction, then repeat the process from **Step 2**.

Definition 3.1.4. Algorithm 1 with MDY1 direction is called MDY1 Algorithm and with MDY2 direction is called MDY2 Algorithm.

Next, we give a pictorial description of the metric projection of Algorithm 1 where we let

$$\zeta_k := \frac{\langle F(v_k), x_k - v_k \rangle}{\|F(v_k)\|^2}, \quad F(v_k) \neq 0.$$

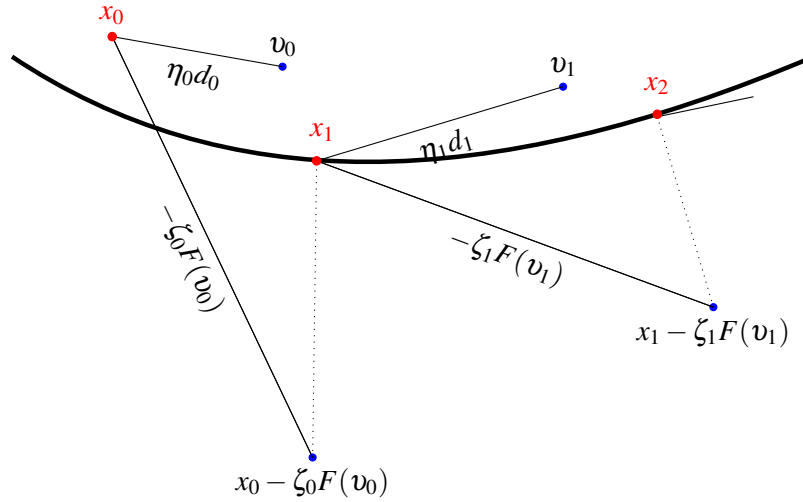


Figure 3.1: Illustration of the metric projection of Algorithm 1 adapted from the Figure 1 in Abubakar et al. [1]

Note: From Remark 3.1.1 and equation (3.7), we can conclude that for all $k \geq 0$, $d_{k-1}^T w_{k-1} \geq t^2 \|F_{k-1}\|^2$, which indicates the positivity of $d_{k-1}^T w_{k-1}$ when the solution of (1.10) is not achieved. In other words, it means that the parameters β_k^{PDY} and θ_k are well-defined.

Remark 3.1.5. The line search (3.15) is well-defined. That is, for all $k \geq 0$, there always exists a step-size η_k satisfying (3.15) in a finite number of iterations.

Suppose on the contrary that there exists some k_0 such that for any $i = 0, 1, 2, \dots$, (3.15) does not hold, that is

$$-\langle F(x_{k_0} + \kappa \rho^i d_{k_0}), d_{k_0} \rangle < \sigma \kappa \rho^i \|d_{k_0}\|^2 \|F(x_{k_0} + \kappa \rho^i d_{k_0})\|^{1/q}, \quad \kappa > 0 \text{ and } q \geq 1. \quad (3.17)$$

By the continuity of F and the fact that $0 < \rho^i < 1$, ($i = 0, 1, 2, \dots$), letting $i \rightarrow \infty$ together with (3.17) yields

$$\langle F(x_{k_0}), d_{k_0} \rangle \geq 0. \quad (3.18)$$

It is clear that the inequality (3.18) contradicts (3.7). Hence, the line search (3.15) is well-defined.

3.2 Algorithm 2: HCDLS Method

Let θ, β_1 and β_2 be real numbers. For $0 \leq \theta \leq 1$, we call $\beta^* = (1 - \theta)\beta_1 + \theta\beta_2$ the convex combination of the parameters β_1 and β_2 , where θ is called a convex combination parameter. It is not difficult to see that when $\theta = 1$, then $\beta^* = \beta_2$ and when $\theta = 0$, then $\beta^* = \beta_1$. This means that β^* takes both β_1 and β_2 as special cases. As a result, researchers often take the convex combination of two different CG parameters with the aim of enhancing the numerical performance of Problem (1.10). Recently, Abubakar et al. proposed in [87] a CG method for solving Problem (1.10) where their CG parameter is defined as a convex combination of Liu and Storey (LS) and Fletcher and Reeves (FR) CG parameters. However, the convex combination of their CG parameters may assume values that are outside the interval $[0, 1]$. Therefore, the authors have to attach some conditions to the convex combination of parameters. In order

to relax such conditions, we propose a new convex combination of two different CG parameters such that the convex combination of parameters is always in $[0, 1]$ independent of any condition. Our approach was found to improve the numerical performance of Problem (1.10) (see, Muhammad, Tammer and Abubakar [123] for more details).

Consider the conjugate descent (CD) conjugate gradient parameter $\beta_k^{CD} := \frac{\|F(x_k)\|^2}{-d_{k-1}^T F(x_{k-1})}$ proposed by Fletcher in [59] and the modified version of Liu and Storey (LS) [115] CG parameter $\beta_k^{LS} := \frac{F(x_k)^T y_{k-1}}{-d_{k-1}^T F(x_{k-1})}$, $-d_{k-1}^T F(x_{k-1}) \neq 0$, where $y_{k-1} := F(x_k) - F(x_{k-1}) + rs_{k-1}$, $r > 0$ and $s_{k-1} := x_k - x_{k-1}$, Andrei in [8] commented that the CD CG method possesses strong convergence properties, but may have poor practical performance due to the effect of jamming while the LS CG method may not always be convergent, they often have better computational performances. Consequently, we consider the convex combination of β_k^{CD} and β_k^{LS} with the aim of improving the numerical performance and achieving convergence of Problem (1.10).

Here, we propose the new search direction as follows:

$$d_k^H := \begin{cases} -F(x_k) & \text{if } k = 0, \\ -\tau_k F(x_k) + \beta_k^H s_{k-1} & \text{if } k \geq 1, \end{cases} \quad (3.19)$$

$$\beta_k^H := \theta_k \beta_k^{CD} + (1 - \theta_k) \beta_k^{LS}, \quad (3.20)$$

where τ_k is a positive spectral parameter such that (3.7) is fulfilled for d_k given by (3.19).

Let us consider the following choice for the convex combination parameter θ_k in (3.20)

$$\theta_k := \frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}^*}, \quad s_{k-1}^* := s_{k-1} + y_{k-1}, \quad y_{k-1} := F(x_k) - F(x_{k-1}) + rs_{k-1}, \quad r > 0. \quad (3.21)$$

Claim: $y_{k-1}^T s_{k-1}^* > 0$ and subsequently, $\theta_k \in [0, 1]$.

Proof of the claim: By the definition of s_{k-1}^* and y_{k-1} in (3.21) together with the monotonicity assumption on F , we obtain

$$\begin{aligned} y_{k-1}^T s_{k-1}^* &= y_{k-1}^T s_{k-1} + \|y_{k-1}\|^2 \\ &= (F(x_k) - F(x_{k-1}) + rs_{k-1})^T s_{k-1} + \|y_{k-1}\|^2 \\ &= (F(x_k) - F(x_{k-1}))^T (x_k - x_{k-1}) + r\|s_{k-1}\|^2 + \|y_{k-1}\|^2 \\ &\geq r\|s_{k-1}\|^2 + \|y_{k-1}\|^2. \end{aligned} \quad (3.22)$$

From (3.22), we can deduce the following two conditions:

$$y_{k-1}^T s_{k-1}^* \geq r\|s_{k-1}\|^2 > 0, \quad (3.23)$$

$$y_{k-1}^T s_{k-1}^* \geq \|y_{k-1}\|^2. \quad (3.24)$$

Inequality (3.23) shows that $y_{k-1}^T s_{k-1}^* > 0$, for all $k \in \mathbb{N}$.

Moreover, from inequality (3.24), we get

$$1 \geq \frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}^*} \geq 0. \quad (3.25)$$

Taking into account (3.21), it means that $\theta_k \in [0, 1]$ for all $k \in \mathbb{N}$.

Remark 3.2.1. In order to determine the spectral parameter τ_k such that (3.7) is fulfilled for d_k given by (3.19), we proceed as follows: Let $k \geq 1$ and $F(x_k) \neq 0$. Then, we obtain from the definition of d_k in (3.19) that

$$\begin{aligned} F(x_k)^T d_k &= -\tau_k \|F(x_k)\|^2 + \beta_k^H F(x_k)^T s_{k-1} \\ &= -\left(\tau_k - \frac{\beta_k^H F(x_k)^T s_{k-1}}{\|F(x_k)\|^2} \right) \|F(x_k)\|^2. \end{aligned} \quad (3.26)$$

If we set

$$\tau_k \geq 1 + \frac{\beta_k^H F(x_k)^T s_{k-1}}{\|F(x_k)\|^2} \quad \text{for all } k \in \mathbb{N},$$

then, the search direction d_k given by (3.19) satisfies the descent-like condition (3.7).

Without loss of generality, we choose the following value of τ_k in the proposed method as

$$\tau_k := 1 + \frac{\beta_k^H F(x_k)^T s_{k-1}}{\|F(x_k)\|^2} \quad \text{for all } k \in \mathbb{N}.$$

In what follows, we will derive a hybrid conjugate gradient algorithm with spectral parameter as a convex combination of CD and LS methods for solving the nonlinear monotone operator Problem (1.10). For the sake of simplicity, we refer to this algorithm as HCDLS algorithm, see Muhammad et al., [123] for more details about Algorithm 2 (HCDLS).

Algorithm 2: HCDLS Algorithm [for solving Problem (1.10)]

Input: Give the same inputs as in Algorithm 1.

Realize Step 1 to Step 4 of Algorithm 1, but replace Step 5 by:

Step 5: Set $k := k + 1$ and update d_k^H using (3.19), (3.20) and (3.21), then, repeat the process from Step 2.

Remark 3.2.2. From the definition of d_k^H in (3.19), it is easy to see that the following holds

$$F(x_k)^T d_k = -t_3 \|F(x_k)\|^2 \quad \text{for all } k \geq 0, \quad (3.27)$$

where $t_3 = 1$. This means that the search direction d_k^H in (3.19) generated by Algorithm 2 always satisfies the sufficient descent-like condition (3.7).

Remark 3.2.3. The search direction d_k^H generated by Algorithm 2 is well-defined. Taking into account (3.23) and (3.25), it can be observed that θ_k is well defined. Also, since Remark 3.2.2 holds for all $k \geq 0$, so is $-d_{k-1}^T F(x_{k-1}) = \|F(x_{k-1})\|^2 > 0$ for all $k \in \mathbb{N}$, provided the solution of Problem (1.10) has not been achieved. This implies that β_k^{CD} and β_k^{LS} are also well-defined.

3.3 Algorithm 3: Three-term Dai–Liao Projection (TDLP) Method

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable function. Suppose that at certain iteration say, $k - 1$, the Hessian of f , that is $\nabla^2 f$, is approximated by a certain matrix A_{k-1} such that A_k can be updated using updating formula

$$A_k := \lambda_k A_{k-1} + \left(1 + \frac{y_{k-1}^T A_{k-1} y_{k-1}}{y_{k-1}^T s_{k-1}} \right) \frac{s_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} - \left(\frac{s_{k-1} y_{k-1}^T A_{k-1} + A_{k-1} y_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} \right), \quad (3.28)$$

where $\lambda_k \in \mathbb{R}$, $s_{k-1} := x_k - x_{k-1}$, $y_{k-1} := F(x_k) - F(x_{k-1})$ and it is assumed that $y_{k-1}^T s_{k-1} \neq 0$. Note that when $\lambda_k = 1$ for all k , (3.28) reduces to the classical Broyden–Fletcher–Goldfarb–Shanno (BFGS) updating formula.

Consider the following search direction

$$d_k := -A_k F(x_k), \quad k \geq 0. \quad (3.29)$$

Now, replacing the matrix A_{k-1} in (3.28) with an identity matrix, that is, $A_{k-1} = I$, (3.28) becomes

$$A_k = \lambda_k I + \left(1 + \frac{y_{k-1}^T y_{k-1}}{y_{k-1}^T s_{k-1}} \right) \frac{s_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} - \left(\frac{s_{k-1} y_{k-1}^T + y_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} \right). \quad (3.30)$$

Multiplying equation (3.30) from the right hand side by $F(x_k)$ we get

$$A_k F(x_k) = \lambda_k F(x_k) + \left[\left(1 + \frac{\|y_{k-1}\|^2}{\langle y_{k-1}, s_{k-1} \rangle} \right) \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} - \frac{\langle y_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} \right] s_{k-1} - \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} y_{k-1}. \quad (3.31)$$

Substituting (3.31) into (3.29) gives

$$d_k = -\lambda_k F(x_k) + \left[\frac{\langle y_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} - \left(1 + \frac{\|y_{k-1}\|^2}{\langle y_{k-1}, s_{k-1} \rangle} \right) \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} \right] s_{k-1} + \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} y_{k-1}. \quad (3.32)$$

Since the coefficient of s_{k-1} and y_{k-1} are scalars, the search direction d_k in (3.32) can be written in the form of a three-term CG direction, that is,

$$d_k := -\lambda_k F(x_k) + \beta_k s_{k-1} + \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} y_{k-1}, \quad (3.33)$$

where the scalar β_k is a suitable CG parameter. In this context, the CG parameter β_k is replaced by the following Dai and Liao CG parameter [45]

$$\beta_k^{DL} := \frac{\langle F(x_k), y_{k-1} - t s_{k-1} \rangle}{\langle y_{k-1}, d_{k-1} \rangle}, \quad t \geq 0, \quad (3.34)$$

$y_{k-1} := F(x_k) - F(x_{k-1})$ and $s_{k-1} := x_k - x_{k-1}$. However, for some functions, the inner product $\langle y_{k-1}, d_{k-1} \rangle$ may become zero which will make the parameter β_k^{DL} (3.34) undefined. As a result, we slightly modify (3.34) as follows

$$\beta_k^{TDLP} := \frac{\langle F(x_k), \vartheta_{k-1} - t s_{k-1} \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle}, \quad (3.35)$$

where

$$\eta_{k-1} := \vartheta_{k-1} + \left(1 + \max \left\{ 0, -\frac{\langle s_{k-1}, \vartheta_{k-1} \rangle}{\|s_{k-1}\|^2} \right\} \right) s_{k-1}, \quad (3.36)$$

$\vartheta_{k-1} := F(x_k) - F(x_{k-1}) + r s_{k-1}$, $r > 0$, $s_{k-1} := x_k - x_{k-1}$ and $x_k \neq x_{k-1}$. Note that TDLP stands for **Three-term Dai–Liao Projection**). Obviously, it is clear that the CG parameter β_k^{TDLP} given by (3.35) is well-defined (see Remark 3.3.1 for more details).

Using similar approach of Li, Zhang and Zhou in [175], we propose the following new search direction

$$d_k := \begin{cases} -F(x_k) & \text{if } k = 0, \\ -\lambda_k F(x_k) + \beta_k^{TDLP} s_{k-1} + \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle} \vartheta_{k-1} & \text{if } k > 0, \end{cases} \quad (3.37)$$

where λ_k is a scalar parameter to be determined in such a way that the search direction d_k given by (3.37) satisfies the following inequality

$$\langle F(x_k), d_k \rangle \leq -t_4 \|F(x_k)\|^2, \quad t_4 > 0. \quad (3.38)$$

This means that the search direction d_k in (3.37) generated by Algorithm 3 always satisfies the sufficient descent-like condition (3.7).

Remark 3.3.1. To show that β_k^{TDLP} is well-defined, by the definition of η_{k-1} in (3.36), we consider two cases as follows:

If $\max \left\{ 0, -\frac{\langle s_{k-1}, \vartheta_{k-1} \rangle}{\|s_{k-1}\|^2} \right\} \neq 0$ in the definition of η_{k-1} in (3.36), we obtain

$$\begin{aligned} \langle s_{k-1}, \eta_{k-1} \rangle &= \left\langle s_{k-1}, \vartheta_{k-1} + s_{k-1} - \frac{\langle s_{k-1}, \vartheta_{k-1} \rangle}{\|s_{k-1}\|^2} s_{k-1} \right\rangle \\ &= \langle s_{k-1}, \vartheta_{k-1} \rangle + \|s_{k-1}\|^2 - \frac{\langle s_{k-1}, \vartheta_{k-1} \rangle}{\|s_{k-1}\|^2} \|s_{k-1}\|^2 \\ &= \|s_{k-1}\|^2 > 0, \quad \text{if } x_k \neq x_{k-1}. \end{aligned} \quad (3.39)$$

Otherwise, we get

$$\begin{aligned} \langle s_{k-1}, \eta_{k-1} \rangle &= \langle s_{k-1}, \vartheta_{k-1} \rangle + \|s_{k-1}\|^2 \\ &= \langle s_{k-1}, F(x_k) - F(x_{k-1}) + r s_{k-1} \rangle + \|s_{k-1}\|^2 \\ &= \langle s_{k-1}, F(x_k) - F(x_{k-1}) \rangle + r \|s_{k-1}\|^2 + \|s_{k-1}\|^2 \\ &\geq (r+1) \|s_{k-1}\|^2 > 0. \end{aligned} \quad (3.40)$$

The last inequality follows from the definition of s_{k-1} and the monotonicity of F . This means that $\langle s_{k-1}, \eta_{k-1} \rangle > 0$. Hence, β_k^{TDLP} is well-defined.

Remark 3.3.2. In order to determine λ_k such that (3.7) is fulfilled for d_k given by (3.37), we proceed as follows: Let $k > 0$, then by taking the inner product of $F(x_k)$ and d_k defined in (3.37), we obtain

$$\begin{aligned} \langle F(x_k), d_k \rangle &= -\lambda_k \|F(x_k)\|^2 + 2 \frac{\langle F(x_k), \vartheta_{k-1} \rangle \langle F(x_k), s_{k-1} \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle} - t \frac{\langle F(x_k), s_{k-1} \rangle^2}{\langle \eta_{k-1}, s_{k-1} \rangle} \\ &\leq -\lambda_k \|F(x_k)\|^2 + 2 \frac{\langle F(x_k), \vartheta_{k-1} \rangle \langle F(x_k), s_{k-1} \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle} \\ &\leq -\lambda_k \|F(x_k)\|^2 + 2 \frac{\|F(x_k)\|^2 \|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} \\ &= - \left(\lambda_k - 2 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} \right) \|F(x_k)\|^2. \end{aligned} \quad (3.41)$$

The first inequality in (3.41) is obtained by dropping the third term in the preceding line since from Remark 3.3.1, the denominator is positive. The second inequality is obtained by applying the Cauchy–Schwarz inequality. Therefore, for the search direction (3.37) to satisfy the inequality (3.7), we only need

$$\lambda_k \geq c + 2 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle}, \quad c > 0 \text{ for all } k \in \mathbb{N}.$$

Without loss of generality, we choose the following value of λ_k in the proposed method as

$$\lambda_k := c + 2 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} \text{ for all } k \in \mathbb{N}. \quad (3.42)$$

In what follows, we present the steps of the proposed Three-term Dai–Liao like Projection (TDLP) algorithm for solving a system of nonlinear monotone equations according to (1.10). See also Muhammad et al. [124] for more details of Algorithm 3 (TDLP).

Algorithm 3: Three-term Dai-Liao Projection (TDLP) Algorithm [for solving Problem (1.10)]

Input: Give the same inputs as in Algorithm 1.

Realize Step 1 to Step 4 of Algorithm 1, but replace Step 5 by:

Step 5: Set $k := k + 1$ and update d_k using (3.37), (3.35) and (3.42), then repeat the process from Step 2.

3.4 Convergence Analysis of Algorithms 1, 2 and 3

In this section, the convergence analysis of Algorithms 1, 2 and 3 (that is, MDY1, MDY2, HCDLS and TDLP methods) are provided. Throughout the section, it is assumed that $F(x_k) \neq 0$ for all $k \in \mathbb{N}_0$, otherwise, the solution of Problem (1.10) has been achieved.

Remark 3.4.1. Consider the relations (3.10), (3.13), (3.27) and (3.38) and setting $t := \text{Min}\{t_1, t_2, t_3, t_4\}$, then the sequence of the search directions generated by MDY1, MDY2, HCDLS or TDLP satisfy the inequality (3.7) given by

$$d_k^T F(x_k) \leq -t \|F(x_k)\|^2, \quad t > 0. \quad (3.43)$$

The assertion in the next lemma concerning Algorithm 1, 2 or 3 corresponds to the assertion in [1, Lemma 3.3] for the modified Dai–Yuan algorithm (MDY), [123, Lemma 3.3] for the hybrid conjugate gradient algorithm with spectral parameters (HCDLS) or [124, Lemma 2.2] for the Three-term Dai–Liao like Projection method (TDLP), respectively. We modify the proofs based on the corresponding results found in [1, 123, 124].

Lemma 3.4.2. Let $\gamma \in (0, 2)$ and $\sigma > 0$ be suitable constants. Suppose that $\{x_k\}$ is the sequence generated by Algorithm 1, 2 or 3. If the mapping F is pseudomonotone, then the following assertions hold

- (i) $\{x_k\}$ and $\{\|F(x_k)\|\}$ are bounded and $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$ exists.
- (ii) $\{v_k\}$ and $\{\|F(v_k)\|\}$ are bounded.
- (iii) The sequence of the search direction $\{\|d_k\|\}$ generated by Algorithm 1, 2 or 3 is bounded.
- (iv) $\lim_{k \rightarrow \infty} \eta_k \|d_k\| = 0$.
- (v) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$.

Proof. (i). From the nonexpansive property of projection operator, we obtain

$$\left\| P_{\Psi} \left(x_k - \gamma \frac{\langle F(v_k), x_k - v_k \rangle}{\|F(v_k)\|^2} F(v_k) \right) - \hat{x} \right\| \leq \left\| x_k - \gamma \frac{\langle F(v_k), x_k - v_k \rangle}{\|F(v_k)\|^2} F(v_k) - \hat{x} \right\|. \quad (3.44)$$

Since \hat{x} is a solution of Problem (1.10), then $F(\hat{x}) = 0$. Therefore, $\langle F(\hat{x}), \mathbf{v}_k - \hat{x} \rangle \geq 0$ and by the pseudomonotonicity of F , it implies $\langle F(\mathbf{v}_k), \mathbf{v}_k - \hat{x} \rangle \geq 0$. This means

$$\begin{aligned} \langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle &\leq \langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle + \langle F(\mathbf{v}_k), \mathbf{v}_k - \hat{x} \rangle \\ &= \langle F(\mathbf{v}_k), x_k - \mathbf{v}_k + \mathbf{v}_k - \hat{x} \rangle \\ &= \langle F(\mathbf{v}_k), x_k - \hat{x} \rangle. \end{aligned} \quad (3.45)$$

From (3.44), (3.45) and the definition of x_{k+1} in (3.16), we get

$$\begin{aligned} \|x_{k+1} - \hat{x}\|^2 &\leq \left\| x_k - \hat{x} - \gamma \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} F(\mathbf{v}_k) \right\|^2 \\ &= \|x_k - \hat{x}\|^2 - 2\gamma \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} \langle F(\mathbf{v}_k), x_k - \hat{x} \rangle + \gamma^2 \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle^2}{\|F(\mathbf{v}_k)\|^2} \\ &\leq \|x_k - \hat{x}\|^2 - 2\gamma \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} \langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle + \gamma^2 \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle^2}{\|F(\mathbf{v}_k)\|^2} \\ &= \|x_k - \hat{x}\|^2 - \gamma(2 - \gamma) \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle^2}{\|F(\mathbf{v}_k)\|^2} \\ &\leq \|x_k - \hat{x}\|^2. \end{aligned} \quad (3.46)$$

The relation (3.47) implies that for all $k \geq 0$, $\|x_{k+1} - \hat{x}\| \leq \|x_k - \hat{x}\| \leq \|x_{k-1} - \hat{x}\| \leq \dots \leq \|x_0 - \hat{x}\|$, where x_0 is one of the given starting points. This means that $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$ exists and so $\{x_k\}$ is bounded. Let $b_1 := L\|x_0 - \hat{x}\|$, since F is Lipschitz continuous, then for all $k \geq 0$, we obtain

$$\begin{aligned} \|F(x_k)\| &= \|F(x_k) - F(\hat{x})\| \\ &\leq L\|x_k - \hat{x}\| \\ &\vdots \\ &\leq L\|x_0 - \hat{x}\| \\ &= b_1. \end{aligned} \quad (3.48)$$

(ii). By the definition of \mathbf{v}_k in (3.14), (3.45) and the boundedness of $\{x_k\}$, we obtain for all $k \geq 0$, $\{\mathbf{v}_k\}$ is bounded. By the Lipschitz continuity of F and the boundedness of $\{\mathbf{v}_k\}$, there exists some constant, say b_2 , such that for all $k \geq 0$

$$\begin{aligned} \|F(\mathbf{v}_k)\| &= \|F(\mathbf{v}_k) - F(\hat{x})\| \\ &\leq L\|\mathbf{v}_k - \hat{x}\| \\ &\vdots \\ &\leq L\|\mathbf{v}_0 - \hat{x}\| \\ &= b_2. \end{aligned} \quad (3.49)$$

where $b_2 := L\|\mathbf{v}_0 - \hat{x}\|$.

(iii) We divide this proof into four cases as follows:

Case I: Consider the MDY1 search direction defined by (3.5), (3.6), (3.9), (3.10), (3.48) and Remark 3.1.1, we obtain

$$\begin{aligned}
\|d_k\| &= \left\| -F(x_k) + \frac{\|F(x_k)\|^2}{d_{k-1}^T w_{k-1}} d_{k-1} - \frac{\|F(x_k)\|^2 \|d_{k-1}\|^2}{(d_{k-1}^T w_{k-1})^2} F(x_k) \right\| \\
&\leq \|F(x_k)\| + \frac{\|F(x_k)\|^2 \|d_{k-1}\|}{d_{k-1}^T w_{k-1}} + \frac{\|F(x_k)\|^3 \|d_{k-1}\|^2}{(d_{k-1}^T w_{k-1})^2} \\
&\leq \|F(x_k)\| + \frac{\|F(x_k)\|^2 \|d_{k-1}\|}{\|d_{k-1}\|^2} + \frac{\|F(x_k)\|^3 \|d_{k-1}\|^2}{\|d_{k-1}\|^4} \\
&\leq \|F(x_k)\| + \frac{\|F(x_k)\|^2}{\|d_{k-1}\|} + \frac{\|F(x_k)\|^3}{\|d_{k-1}\|^2} \\
&\leq b_1 + \frac{b_1^2}{c\omega} + \frac{b_1^3}{(c\omega)^2} \\
&= M_1,
\end{aligned} \tag{3.50}$$

where $M_1 := b_1 + \frac{b_1^2}{c\omega} + \frac{b_1^3}{(c\omega)^2}$.

Case II: Similarly, consider the MDY2 search direction defined by (3.5), (3.6), (3.12), (3.13), (3.48) and Remark 3.1.1, we get

$$\begin{aligned}
\|d_k\| &= \left\| -F(x_k) + \frac{\|F(x_k)\|^2}{d_{k-1}^T w_{k-1}} d_{k-1} - \left(\frac{F(x_k)^T d_{k-1}}{d_{k-1}^T w_{k-1}} + \frac{\|F(x_k)\|^2}{(d_{k-1}^T w_{k-1})^2} \right) F(x_k) \right\| \\
&\leq \|F(x_k)\| + \frac{\|F(x_k)\|^2 \|d_{k-1}\|}{d_{k-1}^T w_{k-1}} + \frac{\|F(x_k)\|^2 \|d_{k-1}\|}{d_{k-1}^T w_{k-1}} + \frac{\|F(x_k)\|^3}{(d_{k-1}^T w_{k-1})^2} \\
&\leq \|F(x_k)\| + \frac{\|F(x_k)\|^2 \|d_{k-1}\|}{\|d_{k-1}\|^2} + \frac{\|F(x_k)\|^2 \|d_{k-1}\|}{\|d_{k-1}\|^2} + \frac{\|F(x_k)\|^3}{\|d_{k-1}\|^4} \\
&= \|F(x_k)\| + 2 \frac{\|F(x_k)\|^2}{\|d_{k-1}\|} + \frac{\|F(x_k)\|^3}{\|d_{k-1}\|^4} \\
&\leq b_1 + 2 \frac{b_1^2}{c\omega} + \frac{b_1^3}{(c\omega)^4} \\
&= M_2,
\end{aligned} \tag{3.51}$$

where $M_2 := b_1 + 2 \frac{b_1^2}{c\omega} + \frac{b_1^3}{(c\omega)^4}$.

Case III: Consider the HCDLS search direction defined by (3.37), (3.20) and (3.21), we obtain

$$d_k = - \left(1 + \beta_k^H \frac{F(x_k)^T s_{k-1}}{\|F(x_k)\|^2} \right) F(x_k) + \beta_k^H s_{k-1},$$

and

$$\beta_k^H := \theta_k \beta_k^{CD} + (1 - \theta_k) \beta_k^{LS}, \quad \beta_k^{CD} := \frac{\|F(x_k)\|^2}{-d_{k-1}^T F_{k-1}}, \quad \beta_k^{LS} := \frac{F(x_k)^T y_{k-1}}{-d_{k-1}^T F_{k-1}}.$$

Simplifying β_k^H , we get

$$|\beta_k^H| = |\theta_k \beta_k^{CD} + (1 - \theta_k) \beta_k^{LS}|,$$

since $\theta_k \in (0, 1)$, it implies that

$$|\beta_k^H| \leq |\beta_k^{CD}| + |\beta_k^{LS}|.$$

From (3.27) and Remark 3.2.3, it holds that for all $k \geq 0$,

$$|d_k^T F(x_k)| = \|F(x_k)\|^2 \geq r^2. \quad (3.52)$$

Therefore, from (3.20), (3.21), (3.37), (3.48) and (3.52), we get

$$\begin{aligned} \|d_k\| &= \left\| -\left(1 + \beta_k^H \frac{F(x_k)^T s_{k-1}}{\|F(x_k)\|^2}\right) F(x_k) + \beta_k^H s_{k-1} \right\| \\ &\leq \|F(x_k)\| + |\beta_k^H| \frac{\|F(x_k)\|^2 \|s_{k-1}\|}{\|F(x_k)\|^2} + |\beta_k^H| \|s_{k-1}\| \\ &= \|F(x_k)\| + 2|\beta_k^H| \|s_{k-1}\| \\ &\leq \|F(x_k)\| + 2(|\beta_k^{CD}| + |\beta_k^{LS}|) \|s_{k-1}\| \\ &= \|F(x_k)\| + 2 \frac{\|F(x_k)\|^2 \|s_{k-1}\|}{|d_{k-1}^T F_{k-1}|} + 2 \frac{\|F(x_k)\| \|y_{k-1}\| \|s_{k-1}\|}{|d_{k-1}^T F_{k-1}|} \\ &\leq \|F(x_k)\| + 2 \frac{\|F(x_k)\|^2 \|x_k - x_{k-1}\|}{|d_{k-1}^T F_{k-1}|} + 2L \frac{\|F(x_k)\| \|x_k - x_{k-1}\|^2}{|d_{k-1}^T F_{k-1}|} \\ &\leq \|F(x_k)\| + 2 \frac{\|F(x_k)\|^2 (\|x_k\| + \|x_{k-1}\|)}{|d_{k-1}^T F_{k-1}|} + 2L \frac{\|F(x_k)\| (\|x_k\| + \|x_{k-1}\|)^2}{|d_{k-1}^T F_{k-1}|} \\ &\leq b_1 + \frac{4b_1 \omega (b_1 + 2L\omega)}{r^2} \\ &= M_3, \end{aligned} \quad (3.53)$$

where $M_3 := b_1 + \frac{4b_1 \omega (b_1 + 2L\omega)}{r^2}$.

Case IV: Lastly, consider the TDLP search direction defined by (3.35), (3.37) and (3.42). For $k > 0$, by the definition of ϑ_{k-1} and the Lipschitz continuity of F , we get

$$\begin{aligned} \|\vartheta_{k-1}\| &= \|F(x_k) - F(x_{k-1}) + r(x_k - x_{k-1})\| \\ &\leq \|F(x_k) - F(x_{k-1})\| + r\|x_k - x_{k-1}\| \\ &\leq L\|x_k - x_{k-1}\| + r\|x_k - x_{k-1}\| \\ &= (L + r)\|s_{k-1}\|, \quad r > 0. \end{aligned} \quad (3.54)$$

Substituting (3.35) and (3.42) into (3.37) and using (3.48), we obtain

$$\begin{aligned} \|d_k\| &= \left\| -\lambda_k F(x_k) + \beta_k^{TDLP} s_{k-1} + \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle} \vartheta_{k-1} \right\| \\ &\leq c\|F(x_k)\| + 2 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} \|F(x_k)\| + 2 \frac{\|F(x_k)\| \|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} + \frac{t\|F(x_k)\| \|s_{k-1}\|^2}{\langle \eta_{k-1}, s_{k-1} \rangle} \\ &\leq c\|F(x_k)\| + 4 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\|s_{k-1}\|^2} \|F(x_k)\| + \frac{t\|F(x_k)\| \|s_{k-1}\|^2}{\|s_{k-1}\|^2} \\ &\leq c\|F(x_k)\| + 4 \frac{(L + r)\|s_{k-1}\|^2}{\|s_{k-1}\|^2} \|F(x_k)\| + \frac{t\|F(x_k)\| \|s_{k-1}\|^2}{\|s_{k-1}\|^2} \\ &= c\|F(x_k)\| + 4(L + r)\|F(x_k)\| + t\|F(x_k)\| \\ &= [c + 4(L + r) + t]\|F(x_k)\| \\ &\leq [c + 4(L + r) + t]b_1 \\ &= M_4, \end{aligned} \quad (3.55)$$

where $M_4 := [c + 4(L + r) + t]b_1$.

Now, from (3.50), (3.51), (3.53) and (3.55), if we let $\mathbb{M} := \text{Min}\{M_1, M_2, M_3, M_4\}$ we obtain

$$\|d_k\| \leq \mathbb{M} \text{ for all } k \in \mathbb{N}_0. \quad (3.56)$$

(iv). From (3.46), we can deduce that

$$\langle F(\mathbf{v}_k), \eta_k d_k \rangle^2 \leq \frac{\|F(\mathbf{v}_k)\|^2}{\gamma(2-\gamma)} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2). \quad (3.57)$$

By the definition of η_k in Algorithms 1, 2 or 3 and (3.15), we get

$$\sigma^2 \eta_k^4 \|d_k\|^4 \|F(\mathbf{v}_k)\|^{2/q} \leq \langle F(\mathbf{v}_k), \eta_k d_k \rangle^2, \quad q \geq 1. \quad (3.58)$$

Combining (3.57) and (3.58), we obtain

$$\sigma^2 \eta_k^4 \|d_k\|^4 \|F(\mathbf{v}_k)\|^{2/q} \leq \frac{\|F(\mathbf{v}_k)\|^2}{\gamma(2-\gamma)} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2). \quad (3.59)$$

Using (3.49) and the fact that $\sigma > 0$, $0 < \gamma < 2$ and $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$ exists, then from (3.59), we obtain

$$\begin{aligned} \lim_{k \rightarrow \infty} \eta_k^4 \|d_k\|^4 &\leq \frac{1}{\gamma(2-\gamma)\sigma^2} \lim_{k \rightarrow \infty} \|F(\mathbf{v}_k)\|^{2-2/q} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2) \\ &\leq \frac{b_2^{2-2/q}}{\gamma(2-\gamma)\sigma^2} \lim_{k \rightarrow \infty} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2) \\ &= 0. \end{aligned}$$

This implies that,

$$\lim_{k \rightarrow \infty} \eta_k \|d_k\| = 0. \quad (3.60)$$

(v). Using the nonexpansive property of the projection operation, (3.14), (3.60) and the Cauchy–Schwarz inequality, we obtain

$$\begin{aligned} \lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| &= \lim_{k \rightarrow \infty} \left\| P_{\Psi} \left[x_k - \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} F(\mathbf{v}_k) \right] - P_{\Psi}(x_k) \right\| \\ &\leq \lim_{k \rightarrow \infty} \left\| x_k - \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} F(\mathbf{v}_k) - x_k \right\| \\ &\leq \lim_{k \rightarrow \infty} \frac{\|F(\mathbf{v}_k)\| \|x_k - \mathbf{v}_k\|}{\|F(\mathbf{v}_k)\|^2} \|F(\mathbf{v}_k)\| \\ &= \lim_{k \rightarrow \infty} \|x_k - \mathbf{v}_k\| \\ &= \lim_{k \rightarrow \infty} \eta_k \|d_k\| \\ &= 0. \end{aligned}$$

□

Using the same idea as in the proof of [1, Theorem 3.5] for the MDY algorithm, [123, Theorem 3.5] for the HCDLS algorithm or [124, Theorem 2.1] for the TDLP algorithm, we establish the following convergence result for Algorithm 1, 2 or 3, respectively. We modify the proofs presented in the relevant literature [1, 123, 124].

Theorem 3.4.3. *Suppose that Assumption 3.0.1 holds. For $\kappa > 0$, let $\{x_k\}$ be the sequence of iterates and d_k be the search direction generated by Algorithm 1, 2 or 3. Furthermore, let the sequence $\{x_k\}$ be convergent to a point \hat{x} . Then,*

$$\liminf_{k \rightarrow \infty} \|F(x_k)\| = 0 \text{ and } F(\hat{x}) = 0. \quad (3.61)$$

Proof. We claim that the sequence of iterates $\{x_k\}$ generated by Algorithm 1, 2 or 3 satisfies (3.61). Let us assume that (3.61) does not hold. Then, there exists a sufficiently large K such that for all $k \geq K$

$$\|F(x_k)\| \geq \varepsilon, \quad (3.62)$$

where $\varepsilon > 0$ is a given positive constant. Suppose $\eta'_k := \rho^{-1}\eta_k$, $\rho \in (0, 1)$ and define $v'_k := x_k + \eta'_k d_k$. If $\eta_k \neq \kappa$, then, η'_k does not satisfy (3.15), that is,

$$F(v'_k)^T d_k + \sigma \eta'_k \|F(v'_k)\|^{1/q} \|d_k\|^2 > 0, \quad q \geq 1.$$

Since \hat{x} is in the solution set of Problem (1.10) it implies that $F(\hat{x}) = 0$. By the Lipschitz continuity of F and the boundedness of $\|d_k\|$ in (3.56), we obtain for all $k \geq 0$

$$\begin{aligned} \|F(v'_k)\| &= \|F(v'_k) - F(\hat{x})\| \\ &\leq L \|x_k + \eta'_k d_k - \hat{x}\| \\ &\leq L \|x_k - \hat{x}\| + L \eta'_k \|d_k\| \\ &\leq L \|x_0 - \hat{x}\| + L \eta'_k \|d_k\| \\ &\leq u_1 + L \rho^{-1} \eta_k \mathbb{M} \\ &\leq u_1 + L \rho^{-1} \mathbb{M}, \end{aligned}$$

where $u_1 := L \|x_0 - \hat{x}\|$. The last inequality follows from the fact that $\eta_k \leq 1$, for all $k \geq 0$. Now, letting $u_3 := u_1 + L \rho^{-1} \mathbb{M}$, we get

$$\|F(v'_k)\| \leq u_3. \quad (3.63)$$

From the inequality (3.43), Lipschitz continuity of F and by Cauchy–Schwarz inequality, we obtain for all $k \geq 0$,

$$\begin{aligned} t \|F(x_k)\|^2 &\leq -\langle F(x_k), d_k \rangle \\ &< -\langle F(x_k), d_k \rangle + \langle F(v'_k), d_k \rangle + \sigma \eta'_k \|F(v'_k)\|^{1/q} \|d_k\|^2 \\ &= \langle F(v'_k) - F(x_k), d_k \rangle + \sigma \eta'_k \|F(v'_k)\|^{1/q} \|d_k\|^2 \\ &\leq \|F(v'_k) - F(x_k)\| \|d_k\| + \sigma \eta'_k \|F(v'_k)\|^{1/q} \|d_k\|^2 \\ &\leq L \eta'_k \|d_k\|^2 + \sigma \eta'_k \|F(v'_k)\|^{1/q} \|d_k\|^2 \\ &= \rho^{-1} \eta_k (L + \sigma \|F(v'_k)\|^{1/q}) \|d_k\|^2. \end{aligned}$$

This further gives

$$\eta_k \|d_k\| \geq \frac{\rho t \|F(x_k)\|^2}{(L + \sigma \|F(v'_k)\|^{1/q}) \|d_k\|} \geq \frac{\rho t \varepsilon^2}{(L + \sigma u_3^{1/q}) \mathbb{M}}, \quad (3.64)$$

where the last inequality follows from (3.56) and (3.63).

By taking the limit in (3.64), we obtain

$$\lim_{k \rightarrow \infty} \eta_k \|d_k\| \geq \frac{\rho t \varepsilon^2}{(L + \sigma u_3^{1/q}) \mathbb{M}}, \quad (3.65)$$

which clearly contradicts (3.60). Hence, the assertion (3.61) holds.

Furthermore, by the continuity of F and the boundedness of $\{x_k\}$, there exists an accumulation point \hat{x} of $\{x_k\}$ such that $F(\hat{x}) = 0$. Since $\{\|x_k - \hat{x}\|\}$ converges (according to Lemma 3.4.2) and \hat{x} is an accumulation point of $\{x_k\}$, it follows that $\{x_k\}$ converges to \hat{x} . \square

However, in order to see the numerical performance and the robustness of all the algorithms proposed in this chapter (that is, Algorithm 1, 2 and 3), we provide their numerical comparison in Section 5.1.1. Secondly, a numerical experiment of the best performing algorithm in terms of efficiency and robustness (that is, Algorithm 3) will be chosen and compare with the state-of-the-art algorithms in Section 5.1.2. Furthermore, the application of Algorithm 1, 2 and 3 in compressive sensing is given in Section 6.1. Lastly, we incorporate Algorithm 3 (TDLP) in **Step 3** of Algorithm 8 for solving vector-valued approximation problem (P^m) in Section 7.5 of Chapter 7.

Chapter 4

Two–Step Projection Methods

In this chapter, we present the second part of our main results by considering two-step methods. In Section 4.1, with the help of the hyperplane projection of Solodov and Svaiter [150] and the inertial-like algorithms, we propose a two-step inertial-type projection method where we use two starting points say x_{-1} and x_0 to update our sequence of iterates, unlike the classical iterative algorithms (such as Newton’s method and its variants as well as in quasi–Newton methods and so on) that use the formula

$$x_{k+1} := x_k + \alpha_k d_k, \quad k \geq 0, \quad \alpha_k > 0, \quad (4.1)$$

to update their sequence of iterations. Here, the search direction d_k in (4.1) is usually updated using x_k and its preceding point x_{k-1} as well as their images $F(x_k)$ and $F(x_{k-1})$ (see, for example, Sections 3.1, 3.2 and 3.3 or [1, 114, 123, 124]). Going by the fact that incorporating an inertial step

$$w_k := x_k + \alpha_k (x_k - x_{k-1}), \quad \alpha_k > 0, \quad (4.2)$$

into algorithms for solving variational inequalities, split feasibility problems, and so on, speeds up the numerical performance, which is why we incorporated it in our search directions and its effect can be seen clearly in Section 5.2. Secondly, using Solodov and Svaiter’s hyperplane projection techniques [150], we proposed a two-step hybrid algorithm that uses two search directions defined using the well-known Barzilai and Borwein (BB) spectral parameters to solve a system of nonlinear monotone operator equations in the form of (1.10). The BB spectral parameters can be viewed as the approximations of Jacobians with scalar multiples of identity matrices. Section 4.2.2 provides the convergence analysis of the two-step hybrid projection algorithm.

4.1 Inertial–Type Projection Method

Given two starting points, say x_0 and x_{-1} , we consider an inertial step (4.2)

$$w_k := x_k + \alpha_k (x_k - x_{k-1}), \quad k \geq 1, \quad \alpha_k > 0.$$

Iterative algorithms that incorporate the inertial step (4.2) are popularly referred to as *inertial–type algorithms*. These algorithms originated from the heavy–ball method of the second–order–in–time dissipative dynamical system. In 1964, Polyak [136] began by considering inertial extrapolation as a speed–up

method to solve smooth convex minimization problems. Inertial-type methods are two-step iterative schemes and the next iterate is defined by making use of the previous two iterates, see [25]. In order to speed up the iteration process, an inertial extrapolation term is required to boost the iterative sequence. These inertial-type methods are basically used to accelerate the iterative sequence towards the required solution. Recently, there has been a growing interest in studying inertial-type algorithms for optimization, variational inequalities and monotone inclusions, see, for example, [17, 50, 143, 144, 146, 162] and the references therein. Various studies have shown that iterative algorithms for solving the above nonlinear problems with an inertial step have better numerical performance in terms of the number of iterations and a time of execution compared to their counterparts without the inertial step. These two impressive advantages enhance the researcher's interest in developing new inertial-like methods.

Given a starting point, say x_0 , classical iterative algorithms (such as Newton's method and its variants as well as quasi-Newton methods among others) use the formula (4.1) to update their sequence of iterates. The search direction d_k in (4.1) is usually updated using x_k and its preceding point x_{k-1} as well as their images $F(x_k)$ and $F(x_{k-1})$ (see, for example, [1, 73, 167, 168]). As stated above, going by the fact that incorporating the inertial step (4.2) into algorithms for solving variational inequalities, split feasibility problems and so on, speeds up the numerical performance, so we pose the following question: Does incorporating a search direction with the inertial effect (4.2) enhance the numerical performance of the conjugate gradient-like algorithms? To answer this question, given a step size $\alpha_k \in [0, 1]$ and any two starting points, say x_{-1} and x_0 , we compute the sequence of inertial steps $\{w_k\}$ as well as their images using $w_k := x_k + \alpha_k(x_k - x_{k-1})$ and then use them to build the search direction of the proposed algorithm.

4.1.1 Conjugate Gradient and Spectral Gradient Algorithms with Inertial-Step

Let x_0 and x_{-1} be the given two starting points and let

$$w_k := x_k + \alpha_k(x_k - x_{k-1}), \quad k \geq 1,$$

be an inertial step with $\alpha_k \in [0, 1]$. We begin this section by recalling Problem (1.10) (that is, for $x \in \Psi \subseteq \mathbb{R}^n$ nonempty, closed and convex, $F : \Psi \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ continuous and monotone, $F(x) = 0$ is fulfilled) and suppose the following assumptions are vital in the proof of convergence analysis of the proposed algorithms throughout the chapter.

Assumption 4.1.1.

We suppose the following assumptions throughout the chapter

- A. *The solution set χ of Problem (1.10) is nonempty.*
- B. *The function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is monotone and Lipschitz continuous.*
- C. *The sequence $\{\alpha_k\} \in [0, 1]$ fulfill $\lim_{k \rightarrow \infty} \alpha_k = 0$.*

In what follows, we derive two new Inertial-Step Projection Algorithms for solving Problem (1.10).

Algorithm 4: CG Algorithm with Inertial-Step (CGAIS) [for solving Problem (1.10)]

Input: Given $x_{-1}, x_0 \in \Psi$, $\kappa, r > 0$, $\gamma \in (0, 2)$, $\sigma, \rho \in (0, 1)$, $\alpha_k \in [0, 1]$ and $Tol > 0$.

Step 0: Set $k = 0$, compute $d_0 := -F(x_0)$ and $w_0 := x_0 + \alpha_0(x_0 - x_{-1})$.

Step 1: If $\|F(x_k)\| \leq Tol$, then x_k is a solution and the iteration process stops.

Step 2: Set

$$v_k := x_k + \eta_k d_k \quad \text{and} \quad \eta_k := \kappa \rho^i, \quad (4.3)$$

where i is the smallest non-negative integer such that

$$-\langle F(x_k + \kappa \rho^i d_k), d_k \rangle \geq \sigma \kappa \rho^i \|d_k\|^2 \|F(x_k + \kappa \rho^i d_k)\|^{1/q}, \quad q \geq 1. \quad (4.4)$$

Step 3: If $\|F(v_k)\| \leq Tol$, stop. Else, compute the next iterate using the following formula

$$x_{k+1} := P_\Psi \left[x_k - \gamma \frac{\langle F(v_k), x_k - v_k \rangle}{\|F(v_k)\|^2} F(v_k) \right], \quad \|F(v_k)\| \neq 0. \quad (4.5)$$

Step 4: Compute $w_{k+1} := x_{k+1} + \alpha_k(x_{k+1} - x_k)$.

Step 5: Set $k := k + 1$, update the search direction and repeat the process from *Step 1*.

$$d_k := -\widehat{\theta}_k F(x_k) + \beta_k(w_k - w_{k-1}) - u_{k-1}z_{k-1}, \quad (4.6)$$

$$\beta_k := \frac{\langle z_{k-1}, F(x_k) \rangle}{\langle z_{k-1}, w_k - w_{k-1} \rangle}, \quad w_k \neq w_{k-1}, \quad (4.7)$$

$$u_{k-1} := \frac{\langle w_k - w_{k-1}, F(x_k) \rangle}{\langle z_{k-1}, w_k - w_{k-1} \rangle}, \quad w_k \neq w_{k-1}, \quad (4.8)$$

where

$$z_{k-1} := F(w_k) - F(w_{k-1}) + r(w_k - w_{k-1}), \quad (4.9)$$

$$\widehat{\theta}_k := \begin{cases} \lambda_k, & \text{if } \theta_k \leq 0 \text{ or } \langle z_{k-1}, F(x_k) \rangle = 0, \\ \theta_k, & \text{otherwise,} \end{cases} \quad (4.10)$$

$$\theta_k := \frac{1}{\langle z_{k-1}, F(x_k) \rangle} \left\langle F(x_k), w_k - w_{k-1} + z_{k-1} - \frac{w_k - w_{k-1}}{\langle w_k - w_{k-1}, z_{k-1} \rangle} \|z_{k-1}\|^2 \right\rangle, \quad (4.11)$$

$$\lambda_k := \frac{\|w_k - w_{k-1}\|^2}{\langle z_{k-1}, w_k - w_{k-1} \rangle}. \quad (4.12)$$

Algorithm 5: Spectral Algorithm with Inertial-Step (SAIS) [for solving Problem (1.10)]

Input: Given the same inputs as in Algorithm 4.

Realize *Step 0* to *Step 4* of Algorithm 4, but replace *Step 5* by:

Step 5: Compute

$$d_k := -\widehat{\theta}_k F(x_k), \quad (4.13)$$

where $\widehat{\theta}_k$ is defined in (4.10).

Remark 4.1.1. Observe that Algorithm 5 is obtained by setting $\beta_k = u_{k-1} = 0$ in (4.6) of Algorithm 4.

Remark 4.1.2. By the definition of z_{k-1} and the monotonicity assumption on F , we get

$$\begin{aligned} \langle z_{k-1}, w_k - w_{k-1} \rangle &= \langle F(w_k) - F(w_{k-1}) + r(w_k - w_{k-1}), w_k - w_{k-1} \rangle \\ &= \langle F(w_k) - F(w_{k-1}), w_k - w_{k-1} \rangle + r \langle w_k - w_{k-1}, w_k - w_{k-1} \rangle \\ &\geq r \|w_k - w_{k-1}\|^2 > 0, \quad \text{if } w_k \neq w_{k-1}. \end{aligned} \quad (4.14)$$

Next, we show that θ_k is well-defined. We begin by showing that λ_k (according to (4.12)) is bounded. By the Lipschitz continuity of F , we get

$$\langle z_{k-1}, w_k - w_{k-1} \rangle = \langle F(w_k) - F(w_{k-1}), w_k - w_{k-1} \rangle + r \|w_k - w_{k-1}\|^2 \leq (L+r) \|w_k - w_{k-1}\|^2. \quad (4.15)$$

From (4.14) and (4.15) we get

$$\frac{1}{L+r} \leq \frac{\|w_k - w_{k-1}\|^2}{\langle z_{k-1}, w_k - w_{k-1} \rangle} \leq \frac{1}{r}. \quad (4.16)$$

This implies that λ_k is bounded.

Therefore, from (4.10), we can find some constants, say $t_1 > 0$ and $t_2 > 0$ such that

$$t_1 \leq \theta_k \leq t_2. \quad (4.17)$$

Combining (4.16) and (4.17) gives

$$p \leq \widehat{\theta}_k \leq q, \quad (4.18)$$

where $p \in [t_1, \frac{1}{L+r}]$ and $q \in [\frac{1}{r}, t_2]$.

If the search direction d_k defined by (4.6) or (4.13) satisfies the following sufficient descent-like condition

$$F(x_k)^T d_k \leq -t \|F(x_k)\|^2, \quad t > 0, \quad (4.19)$$

then, it is said to be a descent-like direction (see [6, 135, 139, 179]). It is worth mentioning that the inequality (4.19) is very important for conjugate gradient-like iterative algorithms to be globally convergent.

The following lemma shows that the proposed search directions satisfy (4.19) independent of the line search strategy used and the result can be found in [125, Lemma 2.1].

Lemma 4.1.3. The search directions (4.6) and (4.13) generated by Algorithms 4 and 5 satisfy the descent-like condition defined in (4.19).

Proof. Taking the inner product of the search direction d_k defined by (4.6) with $F(x_k)$, for $k = 0$, it follows that $\langle F(x_0), d_0 \rangle \leq -\|F(x_0)\|^2$. For $k > 0$, we get

$$\begin{aligned} \langle F(x_k), d_k \rangle &= -\widehat{\theta}_k \|F(x_k)\|^2 + \frac{\langle z_{k-1}, F(x_k) \rangle}{\langle z_{k-1}, w_k - w_{k-1} \rangle} \langle F(x_k), w_k - w_{k-1} \rangle \\ &\quad - \frac{\langle w_k - w_{k-1}, F(x_k) \rangle}{\langle z_{k-1}, w_k - w_{k-1} \rangle} \langle F(x_k), z_{k-1} \rangle \\ &= -\widehat{\theta}_k \|F(x_k)\|^2 \\ &\leq -p \|F(x_k)\|^2. \end{aligned} \quad (4.20)$$

The last inequality follows from (4.18). This shows that the search direction d_k is a descent-like direction. \square

Remark 4.1.4. *The line search (4.4) is well-defined. That is, for all $k \geq 0$, there always exists a step-size η_k satisfying (4.4) in a finite number of iterations.*

Suppose on the contrary that there exists some k_0 such that for any $i = 0, 1, 2, \dots$, (4.4) does not hold, that is,

$$-\langle F(x_{k_0} + \kappa \rho^i d_{k_0}), d_{k_0} \rangle < \sigma \kappa \rho^i \|d_{k_0}\|^2 \|F(x_{k_0} + \kappa \rho^i d_{k_0})\|^{1/q}, \quad \kappa > 0 \text{ and } q \geq 1. \quad (4.21)$$

By the continuity of F and the fact that $0 < \rho^i < 1$, ($i = 0, 1, 2, \dots$), letting $i \rightarrow \infty$ together with (4.21) yields

$$\langle F(x_{k_0}), d_{k_0} \rangle \geq 0. \quad (4.22)$$

It is clear that the inequality (4.22) contradicts (4.19). Hence the line search (4.4) is well-defined.

4.1.2 Convergence Analysis of Algorithm 4 (CGAIS) and Algorithm 5 (SAIS)

In this section, we show the global convergence of Algorithm 4 (CGAIS) and Algorithm 5 (SAIS). In what follows, we assume that $F(x_k) \neq 0$ and $F(v_k) \neq 0$ for all $k \in \mathbb{N}_0$. If $F(x_k) = 0$, x_k is already a solution of Problem (1.10).

The results in the next lemma are presented in our paper [125, Lemma 2.2] with a similar proof.

Lemma 4.1.5. *Suppose that Assumption 4.1.1 holds and let $0 < \gamma < 2$. Let \hat{x} be an element of the solution set of Problem (1.10). If the sequences $\{w_k\}$, $\{v_k\}$, $\{d_k\}$ and $\{x_k\}$ are generated by (4.2), (4.3), (4.5) and (4.6) as well as the sequences of scalars $\{\alpha_k\}$ and $\{\eta_k\}$ in Algorithm 4 or 5 with the Lipschitz constant L , then the following assertions hold:*

- (i) $\{x_k\}$ and $\{w_k\}$ are bounded and $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$ exists.
- (ii) The sequence of the search direction $\{\|d_k\|\}$ is bounded.
- (iii) $\{v_k\}$ and $\{\|F(v_k)\|\}$ are bounded.
- (iv) $\lim_{k \rightarrow \infty} \eta_k \|d_k\| = 0$.

Proof. (i). Since \hat{x} is a solution of Problem (1.10), then $F(\hat{x}) = 0$, that is, $\langle F(\hat{x}), v_k - \hat{x} \rangle = 0$. Monotonicity of F gives $\langle F(\hat{x}), v_k - \hat{x} \rangle \leq \langle F(v_k), v_k - \hat{x} \rangle$. Furthermore,

$$\begin{aligned} \langle F(v_k), x_k - v_k \rangle &= \langle F(v_k), x_k - v_k \rangle + \langle F(\hat{x}), v_k - \hat{x} \rangle \\ &\leq \langle F(v_k), x_k - v_k \rangle + \langle F(v_k), v_k - \hat{x} \rangle \\ &= \langle F(v_k), x_k - v_k + v_k - \hat{x} \rangle \\ &= \langle F(v_k), x_k - \hat{x} \rangle. \end{aligned} \quad (4.23)$$

By the definition of x_{k+1} in Step 3 of Algorithm 4, the nonexpansive property of the metric projection and (4.23), we get

$$\begin{aligned}
\|x_{k+1} - \hat{x}\|^2 &= \left\| P_{\Psi} \left(x_k - \gamma \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} F(\mathbf{v}_k) \right) - P_{\Psi}(\hat{x}) \right\|^2 \\
&\leq \left\| x_k - \hat{x} - \gamma \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} F(\mathbf{v}_k) \right\|^2 \\
&= \|x_k - \hat{x}\|^2 - 2\gamma \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} \langle F(\mathbf{v}_k), x_k - \hat{x} \rangle + \gamma^2 \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle^2}{\|F(\mathbf{v}_k)\|^2} \\
&\leq \|x_k - \hat{x}\|^2 - 2\gamma \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle}{\|F(\mathbf{v}_k)\|^2} \langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle + \gamma^2 \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle^2}{\|F(\mathbf{v}_k)\|^2} \\
&= \|x_k - \hat{x}\|^2 - \gamma(2 - \gamma) \frac{\langle F(\mathbf{v}_k), x_k - \mathbf{v}_k \rangle^2}{\|F(\mathbf{v}_k)\|^2} \tag{4.24}
\end{aligned}$$

$$\leq \|x_k - \hat{x}\|^2. \tag{4.25}$$

The relation (4.25) implies that for all $k \geq 0$, $\|x_{k+1} - \hat{x}\| \leq \|x_k - \hat{x}\| \leq \|x_{k-1} - \hat{x}\| \leq \dots \leq \|x_0 - \hat{x}\|$, where x_0 is one of the given starting points. This means that $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$ exists and so $\{x_k\}$ is bounded. From Assumption 4.1.1(C), since $\alpha_k \in [0, 1]$ and the sequence $\{x_k\}$ is bounded, then it implies that $\{w_k\}$ is also bounded.

Let $b_1 := L\|x_0 - \hat{x}\|$. Since F is Lipschitz continuous, then for all $k \geq 0$, we get

$$\begin{aligned}
\|F(x_k)\| &= \|F(x_k) - F(\hat{x})\| \\
&\leq L\|x_k - \hat{x}\| \\
&\vdots \\
&\leq L\|x_0 - \hat{x}\| \\
&= b_1. \tag{4.26}
\end{aligned}$$

To show (ii), let $k = 0$, by the definition of the search direction d_k in (4.6), we get

$$\|d_0\| = \|F(x_0)\| \leq b_1. \tag{4.27}$$

By the Lipschitz continuity of F , we get

$$\begin{aligned}
\|z_{k-1}\| &= \|F(w_k) - F(w_{k-1}) + r(w_k - w_{k-1})\| \\
&\leq L\|w_k - w_{k-1}\| + r\|w_k - w_{k-1}\| \\
&= (L + r)\|w_k - w_{k-1}\|. \tag{4.28}
\end{aligned}$$

Suppose $k > 0$,

$$\begin{aligned}
\|d_k\| &= \left\| -\widehat{\theta}_k F(x_k) + \beta_k(w_k - w_{k-1}) - u_{k-1}z_{k-1} \right\| \\
&\leq \widehat{\theta}_k \|F(x_k)\| + |\beta_k| \|w_k - w_{k-1}\| + |u_{k-1}| \|z_{k-1}\| \\
&= \widehat{\theta}_k \|F(x_k)\| + \left| \frac{\langle z_{k-1}, F(x_k) \rangle}{\langle z_{k-1}, w_k - w_{k-1} \rangle} \right| \|w_k - w_{k-1}\| + \left| \frac{\langle w_k - w_{k-1}, F(x_k) \rangle}{\langle z_{k-1}, w_k - w_{k-1} \rangle} \right| \|z_{k-1}\| \\
&\leq \widehat{\theta}_k \|F(x_k)\| + \frac{\|z_{k-1}\| \|F(x_k)\|}{\langle z_{k-1}, w_k - w_{k-1} \rangle} \|w_k - w_{k-1}\| + \frac{\|w_k - w_{k-1}\| \|F(x_k)\|}{\langle z_{k-1}, w_k - w_{k-1} \rangle} \|z_{k-1}\| \\
&= \widehat{\theta}_k \|F(x_k)\| + 2 \frac{\|z_{k-1}\| \|F(x_k)\|}{\langle z_{k-1}, w_k - w_{k-1} \rangle} \|w_k - w_{k-1}\| \\
&\leq \frac{1}{r} \|F(x_k)\| + 2 \frac{(L+r) \|w_k - w_{k-1}\|^2}{r \|w_k - w_{k-1}\|^2} \|F(x_k)\| \\
&= \left[\frac{1}{r} + 2 \frac{(L+r)}{r} \right] \|F(x_k)\|. \tag{4.29}
\end{aligned}$$

The first and the second inequalities follow from triangle inequality and Cauchy–Schwarz inequality, respectively. The third inequality follows from (4.14) and (4.28).

If we let $\widehat{b}_2 := \left[\frac{1}{r} + 2 \frac{(L+r)}{r} \right]$, then (4.29) becomes

$$\|d_k\| \leq \widehat{b}_2 \|F(x_k)\|. \tag{4.30}$$

Combining (4.30) and (4.27) gives

$$\|d_k\| \leq b \quad \text{for all } k \geq 0, \tag{4.31}$$

where $b := b_1 \widehat{b}_2$.

(iii). By the definition of v_k in (4.3), the boundedness of d_k in (4.31) and the boundedness of $\{x_k\}$, we obtain for all $k \geq 0$, $\{v_k\}$ is bounded. Hence, we can find some constant, say \bar{b}_2 , such that $\|v_k - \hat{x}\| \leq \bar{b}_2$. Subsequently, by the Lipschitz continuity of F , there exists some constant, say b_2 such that for all $k \geq 0$,

$$\|F(v_k)\| = \|F(v_k) - F(\hat{x})\| \leq L \|v_k - \hat{x}\| \leq b_2, \tag{4.32}$$

where $b_2 := L\bar{b}_2$.

(iv). From (4.24), we can deduce that

$$\langle F(v_k), \eta_k d_k \rangle^2 \leq \frac{\|F(v_k)\|^2}{\gamma(2-\gamma)} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2). \tag{4.33}$$

By the definition of η_k in Step 3 of Algorithm 4 and (4.4), we get

$$\sigma^2 \eta_k^4 \|d_k\|^4 \|F(v_k)\|^{2/q} \leq \langle F(v_k), \eta_k d_k \rangle^2, \quad q \geq 1. \tag{4.34}$$

By combining (4.33) and (4.34), we get

$$\sigma^2 \eta_k^4 \|d_k\|^4 \|F(v_k)\|^{2/q} \leq \frac{\|F(v_k)\|^2}{\gamma(2-\gamma)} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2). \tag{4.35}$$

Using (4.32) and the fact that $\sigma > 0$, $0 < \gamma < 2$, and the existence of $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$, then from (4.35), we get

$$\begin{aligned} \lim_{k \rightarrow \infty} \eta_k^4 \|d_k\|^4 &\leq \frac{1}{\gamma(2-\gamma)\sigma^2} \lim_{k \rightarrow \infty} \|F(v_k)\|^{2-2/q} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2) \\ &\leq \frac{b_2^{2-2/q}}{\gamma(2-\gamma)\sigma^2} \lim_{k \rightarrow \infty} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2) \\ &= 0. \end{aligned}$$

This implies

$$\lim_{k \rightarrow \infty} \eta_k \|d_k\| = 0. \quad (4.36)$$

□

A corresponding result to the next theorem is shown in [125, Theorem 2.1].

Theorem 4.1.6. *Suppose that Assumption 4.1.1 holds. For $\kappa > 0$, let $\{x_k\}$ be the sequence of iterates and d_k the search direction generated by Algorithm 4 or 5. Furthermore, let the sequence $\{x_k\}$ be convergent to a point \hat{x} . Then,*

$$\liminf_{k \rightarrow \infty} \|F(x_k)\| = 0 \text{ and } F(\hat{x}) = 0. \quad (4.37)$$

Proof. Suppose that for all $k > 0$, $\|d_k\| \neq 0$. Then, from (4.30), we get

$$\frac{\|F(x_k)\|}{\|d_k\|} \geq \frac{1}{\widehat{b}_2}, \quad \|d_k\| \neq 0, \quad (4.38)$$

with $\widehat{b}_2 := \left[\frac{1}{r} + 2\frac{(L+r)}{r} \right]$, L is a Lipschitz constant and $r > 0$ is a positive constant.

Let $v'_k := x_k + \eta'_k d_k$ and suppose that $\eta_k \neq \kappa$, $\kappa > 0$. Then, for $\rho \in (0, 1)$, $\eta'_k := \rho^{-1} \eta_k$ does not satisfy (4.4), that is,

$$\langle F(v'_k), d_k \rangle + \sigma \eta'_k \|F(v'_k)\|^{1/q} \|d_k\|^2 > 0, \quad q \geq 1. \quad (4.39)$$

By Assumption 4.1.1, \hat{x} is an element of the solution set of Problem (1.10). By the Lipschitz continuity of F , the boundedness of $\|d_k\|$ in (4.31) with $b = b_1 \widehat{b}_2$ and for any given starting point, say x_0 , we get

$$\begin{aligned} \|F(v'_k)\| &= \|F(v'_k) - F(\hat{x})\| \\ &\leq L \|x_k + \eta'_k d_k - \hat{x}\| \\ &\leq L \|x_k - \hat{x}\| + L \eta'_k \|d_k\| \\ &\leq L \|x_0 - \hat{x}\| + L \eta'_k \|d_k\| \\ &\leq b_1 + L \rho^{-1} \eta_k b, \\ &\leq b_1 + L \rho^{-1} b, \end{aligned}$$

where $b_1 := L \|x_0 - \hat{x}\|$. Now, letting $b_3 := b_1 + L \rho^{-1} b$, we obtain for all $k \geq 0$,

$$\|F(v'_k)\| \leq b_3. \quad (4.40)$$

Substituting (4.39) in the descent-like condition (4.19) and applying the Lipschitz continuity of F together with the Cauchy–Schwarz inequality, we get for all $k \geq 0$,

$$\begin{aligned}
t\|F(x_k)\|^2 &\leq -\langle F(x_k), d_k \rangle \\
&< -\langle F(x_k), d_k \rangle + \langle F(v'_k), d_k \rangle + \sigma\eta'_k\|F(v'_k)\|^{1/q}\|d_k\|^2 \\
&= \langle F(v'_k) - F(x_k), d_k \rangle + \sigma\eta'_k\|F(v'_k)\|^{1/q}\|d_k\|^2 \\
&\leq \|F(v'_k) - F(x_k)\|\|d_k\| + \sigma\eta'_k\|F(v'_k)\|^{1/q}\|d_k\|^2 \\
&\leq L\eta'_k\|d_k\|^2 + \sigma\eta'_k\|F(v'_k)\|^{1/q}\|d_k\|^2 \\
&= \rho^{-1}\eta_k(L + \sigma\|F(v'_k)\|^{1/q})\|d_k\|^2.
\end{aligned}$$

This implies that

$$\eta_k \geq \frac{\rho t\|F(x_k)\|^2}{(L + \sigma\|F(v'_k)\|^{1/q})\|d_k\|^2} \geq \frac{\rho t}{(L + \sigma b_3^{1/q})\widehat{b}_2^2}, \quad (4.41)$$

where the last inequality follows from (4.38) and (4.40). By combining (4.36) and (4.41), we get

$$\liminf_{k \rightarrow \infty} \|d_k\| = 0. \quad (4.42)$$

Furthermore, we can deduce the following relation from (4.19),

$$t\|F(x_k)\|^2 \leq -F(x_k)^T d_k \leq \|F(x_k)\|\|d_k\|, \quad t > 0,$$

which further gives

$$\|d_k\| \geq t\|F(x_k)\|, \quad t > 0. \quad (4.43)$$

Therefore,

$$0 = \liminf_{k \rightarrow \infty} \|d_k\| \geq t \liminf_{k \rightarrow \infty} \|F(x_k)\|, \quad (4.44)$$

which gives (4.37).

Furthermore, there is an accumulation point of $\{x_k\}$, say \hat{x} , for which $\|F(\hat{x})\| = 0$, since F is Lipschitz continuous and the sequence $\{x_k\}$ is bounded. Because $\{x_k\}$ is bounded, we can identify a subsequence of $\{x_k\}$, say $\{x_{k_j}\}$, for which $\lim_{k \rightarrow \infty} \|x_{k_j} - \hat{x}\| = 0$. Lemma 4.1.5 states that $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$ exists, hence we can deduce that $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\| = 0$ and the proof is complete. \square

4.2 Two-Step Hybrid Spectral Gradient Projection Method

In order to further improve the results from the literature, we propose a two-step hybrid iterative algorithm based on the projection technique for solving a system of nonlinear monotone equations (1.10). The proposed algorithm uses two search directions, which are defined using the well-known Barzilai and Borwein (BB) spectral parameters. The BB spectral parameters are approximations of Jacobians using scalar multiples of identity matrices. It has been demonstrated that convex combinations of the BB parameters improve the numerical performance of iterative algorithms, thus, we raise the following question: Can the numerical performance of the method proposed by Awwal et al. in [18] be improved when incorporated with a convex combination of the BB parameters? To answer this question, we propose a two-step hybrid method as a convex combination of BB1 and BB2 spectral parameters, as we shall see in the following section. In Section 4.2.1, we state the proposed algorithm, while in Section 4.2.2, we present the convergence analysis of the proposed two-step hybrid spectral gradient algorithm.

4.2.1 Proposed Algorithm

The search direction in (4.1) is usually defined as $d_k := -B_k^{-1}F(x_k)$ where B_k is either the approximate Hessian matrix in the quasi-Newton method or the exact Hessian matrix $\nabla^2 f(x_k)$ in the case of Newton's method. It is necessary to approximate the Hessian matrix, B_k , in order to satisfy the following secant equation

$$B_k s_{k-1} = y_{k-1}, \quad k > 0, \quad (4.45)$$

$$s_{k-1} := x_k - x_{k-1} \text{ and } y_{k-1} := F(x_k) - F(x_{k-1}).$$

One of the primary shortcomings of the well-known Newton method is the requirement to compute the second derivative of the objective function in each iteration, which led to the development of the quasi-Newton method. However, it is not suitable for large-scale problems because it necessitates storing $n \times n$ matrices during the iteration process. The matrix-free method proposed by Barzilai and Borwein (BB) [22] is an important solution developed to address the quasi-Newton method's storage difficulty. The BB method generate the next iteration using (4.1), with the search direction given by $d_k := -F(x_k)$, $k \geq 0$ and the stepsize taken as a diagonal matrix $D_k := \lambda_k I$, which is meant to fulfill the secant equation (4.45). However, since $\lambda_k I$ produces diagonal matrices with identical diagonal elements, it is usually very difficult to find λ_k for which $D_k^{-1} = \lambda_k^{-1} I$ satisfies (4.45) when the dimension is greater than one. Consequently, Barzilai and Borwein required that D_k^{-1} approximately satisfies (4.45) by finding $\lambda_k \in \mathbb{R}$ that minimizes the following least squares problems

$$\min_{\lambda} \|\lambda s_{k-1} - y_{k-1}\|^2, \quad (4.46)$$

and

$$\min_{\lambda} \|s_{k-1} - \lambda y_{k-1}\|^2. \quad (4.47)$$

The solutions of the minimization problems (4.46) and (4.47) are respectively given as

$$\lambda_k^{BB1} := \frac{\|s_{k-1}\|^2}{\langle y_{k-1}, s_{k-1} \rangle} \quad \text{and} \quad \lambda_k^{BB2} := \frac{\langle y_{k-1}, s_{k-1} \rangle}{\|y_{k-1}\|^2}. \quad (4.48)$$

By Cauchy-Schwarz inequality, we see that the stepsize produced by λ_k^{BB1} in (4.48) is always greater than or equal to the one produced by λ_k^{BB2} whenever $\langle y_{k-1}, s_{k-1} \rangle > 0$. Barzilai and Borwein proved that the iterative scheme (4.1) with $\alpha_k := \lambda_k^{BB1}$ and $d_k := -F(x_k)$ converge with an R-superlinear rate for two-dimensional strictly convex quadratic problems.

However, if the objective function is not convex, the stepsizes λ_k^{BB1} and λ_k^{BB2} could go negative, which is a drawback of the BB approach. As a result, Dai et al. [42] developed and analyzed the subsequent positive stepsize

$$\widehat{\lambda}_k := \frac{\|s_{k-1}\|}{\|y_{k-1}\|}. \quad (4.49)$$

For λ_k^{BB1} and λ_k^{BB2} , the geometric mean is the stepsize (4.49). The iterative scheme (4.1) with $\alpha_k = \widehat{\lambda}_k$ has been demonstrated to have an equivalent rate of convergence with the stepsize λ_k^{BB1} in (4.48) for strictly convex quadratic functions that are two-dimensional under certain conditions. A family of gradient

algorithms with a stepsize that is a convex combination of λ_k^{BB1} and λ_k^{BB2} was recently proposed by Dai et al. in [43]. One obtains the stepsize by resolving the subsequent problem

$$\Psi_\xi(\lambda) = \|\xi[(1/\lambda)s_{k-1} - y_{k-1}] + (1 - \xi)[s_{k-1} - \lambda y_{k-1}]\|^2. \quad (4.50)$$

The equation $\frac{d\Psi_\xi(\lambda)}{d\lambda} = 0$, was demonstrated to have a unique solution in the closed interval $[\lambda_k^{BB1}, \lambda_k^{BB2}]$ for $0 \leq \xi \leq 1$ and $\langle y_{k-1}, s_{k-1} \rangle > 0$. They also demonstrated that for two-dimensional strictly convex quadratics and any finite-dimensional case, respectively, their method is R-superlinearly and R-linearly convergent. Readers who are interested in the convergence analysis of the BB stepsizes may look at the following references [30, 44, 41, 60, 141, 142].

Conversely, La Cruz and Raydan [108] have expanded the BB method with the stepsize λ_k^{BB1} in (4.48) to solve unconstrained nonlinear equations. Their technique is based on the nonmonotone line search strategy, which ensures the method's global convergence. The method they have provided competes with certain well-established methods, as indicated by the numerical tests. On the other hand, their algorithm needs descent-like directions relative to the residual's squared norm. This implies that each iteration requires the computation of a directional derivative or an approximation of it. As a result, La Cruz et al. [106] proposed an additional BB method for solving unconstrained nonlinear equations that uses a new nonmonotone line-search strategy. Their method has an advantage over the previous one in that the directional derivative computations are completely avoided. Zhang and Zhou [174] developed an intriguing projection spectral method, which can be seen as a variant of the method given in [106, 108], based on the projection technique of Solodov and Svaiter [150]. They proposed a new line search technique that considers the monotonicity of F and does not require a merit function. They presented various numerical experiments that demonstrated the method's computational advantage and established the method's global convergence under some appropriate assumptions. Zhang and Zhou's [174] method for solving nonlinear systems of equations with convex constraints was further extended by Yu et al. [170]. Under some moderate conditions, their method is globally convergent, and preliminary numerical results demonstrate that the method performs well and is far better than the projection method proposed in [164]. Mohammad and Abubakar [120] proposed a positive spectral gradient method for unconstrained nonlinear monotone operator equations. It was based on the projection technique that Solodov and Svaiter [150] proposed. In their work, they propose a convex combination of the modified λ_k^{BB1} and $\hat{\tau}_k$ for the spectral parameter. Their method works well and was extended to solve monotone operator equations with convex constraints in Reference [15] as well as signal and image restoration by Abubakar et al. in [3]. Recently, Awwal et al. [18] proposed a novel two-step spectral gradient method for system of nonlinear monotone equations based on the projection technique which is first of its kind. They also introduced a new line search technique for generating the separating hyperplane projection step of Solodov and Svaiter [150] that generalizes the one used in most of the existing literature. Their method is globally convergent under some suitable conditions, followed with some numerical experiments that demonstrated the computational advantages of their proposed method. Finally, their method was applied for image deblurring problems in compressive sensing.

In order to improve the numerical performance of the method proposed in [18], we propose a two-step hybrid iterative scheme based on the projection technique for solving a system of monotone nonlinear

equations with convex constraints. The new scheme proposed here is a convex combination of the Barzilai and Borwein (BB1 and BB2) spectral parameters with modifications. Let I be an identity map in \mathbb{R}^n , if we set $d := (F - I)$, then equation (4.1) is closely related to the well-known Mann iterative scheme [118]

$$u_{k+1} := u_k + \alpha_k(F(u_k) - u_k), \quad (4.51)$$

where $0 \leq \alpha_k < 1$. Mann iteration has been used to address a variety of nonlinear problems with success. Its convergence speed is somewhat modest, nevertheless. Numerous investigations have demonstrated that the well-known two-step Ishikawa iterative method [88]

$$\begin{aligned} v_k &:= (1 - \alpha_k)u_k + \alpha_k F(u_k), \\ u_{k+1} &:= (1 - \beta_k)u_k + \beta_k F(v_k), \end{aligned} \quad (4.52)$$

where $\alpha_k, \beta_k \in [0, 1)$, converges faster than the one-step Mann iteration.

Let $\bar{d}_k := (F - I)u_k$ and $\hat{d}_k := F(v_k) - u_k$, then the Ishikawa iterative scheme can be rewritten as follows

$$\begin{aligned} v_k &:= u_k + \alpha_k \bar{d}_k, \\ u_{k+1} &:= u_k + \beta_k \hat{d}_k. \end{aligned} \quad (4.53)$$

In this thesis, we propose a two-step hybrid iterative scheme incorporating nonnegative BB parameters with a projection strategy to solve nonlinear monotone systems of equations with convex constraints. This scheme is based on the observation that the two-step Ishikawa iterative scheme has a faster convergence speed than the one-step Mann iterative scheme. For the proposed two-step hybrid system, given a starting point $x_0 \in \Psi$ and $\alpha_k, \beta_k \in (0, 1]$, we define the updating formula as follows

$$\begin{aligned} w_k &:= x_k + \alpha_k d_k^I, \\ x_{k+1} &:= P_\Psi \left[x_k - \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} F(z_k) \right], \end{aligned} \quad (4.54)$$

where $z_k := x_k + \beta_k d_k^II$, $P_\Psi(\cdot)$ is a projection operator and

$$\begin{cases} d_k^I := -F(x_k) & \text{if } k = 0, \\ d_k^I := -\tau^I(x_k)F(x_k) & \text{if } k > 0, \\ d_k^II := -\tau^II(w_k)F(x_k) & \text{for } k \geq 0. \end{cases} \quad (4.55)$$

For simplicity, we denote $\tau^I(x_k) := \tau_k^I$ and $\tau^II(w_k) := \tau_k^II$. The parameters τ_k^I and τ_k^II are modifications of the BB parameters (4.48) given as follows

$$\begin{cases} \tau_k^I := \frac{\|s_k^I\|^2}{\langle y_k^I, s_k^I \rangle}, \\ \tau_k^II := (1 - \theta_k) \frac{\langle y_k^II, s_k^II \rangle}{\|y_k^II\|^2} + \theta_k \frac{\|s_k^II\|}{\|y_k^II\|}, \end{cases} \quad (4.56)$$

where

$$\begin{cases} y_k^I := F(x_{k+1}) - F(x_k) + rs_k^I, & r > 0, \\ y_k^{II} := F(w_k) - F(x_k) + ts_k^{II}, & t > 0, \\ s_k^I := x_{k+1} - x_k, \\ s_k^{II} := w_k - x_k. \end{cases} \quad (4.57)$$

Lemma 4.2.1. *If $\{\alpha_k\} \in (0, 1)$ and $\lim_{k \rightarrow \infty} \alpha_k = 0$, then $\eta \leq \tau_k^I \leq \mu$ and $\delta \leq \tau_k^{II} \leq \gamma$, where $\eta := \frac{1}{L+r}$, $\mu := \frac{1}{r}$, $\delta := \max \left\{ \frac{t}{(t+L)^2}, \frac{1}{t+L}, \bar{t} \frac{1}{t+L} \right\}$ and $\gamma := \min \left\{ \frac{t+L}{t^2}, \frac{1}{t}, \frac{t+L}{t^2} + \frac{1}{t} \right\}$.*

Proof. The monotonicity of F gives $\langle F(x_{k+1}) - F(x_k), x_{k+1} - x_k \rangle \geq 0$. Therefore, by the definition of y_k^I and y_k^{II} in (4.57), we get

$$\langle y_k^I, s_k^I \rangle = \langle F(x_{k+1}) - F(x_k), s_k^I \rangle + r \langle s_k^I, s_k^I \rangle \geq r \|s_k^I\|^2. \quad (4.58)$$

$$\langle y_k^{II}, s_k^{II} \rangle = \langle F(w_k) - F(x_k), s_k^{II} \rangle + t \langle s_k^{II}, s_k^{II} \rangle \geq t \|s_k^{II}\|^2. \quad (4.59)$$

On the other hand, by the Lipschitz continuity and the Cauchy–Schwarz inequality, we obtain

$$\langle y_k^I, s_k^I \rangle = \langle F(x_{k+1}) - F(x_k), s_k^I \rangle + r \langle s_k^I, s_k^I \rangle \leq (L+r) \|s_k^I\|^2. \quad (4.60)$$

$$\langle y_k^{II}, s_k^{II} \rangle = \langle F(w_k) - F(x_k), s_k^{II} \rangle + t \langle s_k^{II}, s_k^{II} \rangle \leq (t+L) \|s_k^{II}\|^2. \quad (4.61)$$

$$\|y_k^{II}\| = \|F(w_k) - F(x_k) + t(w_k - x_k)\| \leq (t+L) \|s_k^{II}\|. \quad (4.62)$$

Also, by the monotonicity of F , we obtain

$$\begin{aligned} \|y_k^{II}\|^2 &= \langle F(w_k) - F(x_k) + t(w_k - x_k), F(w_k) - F(x_k) + t(w_k - x_k) \rangle \\ &= \|F(w_k) - F(x_k)\|^2 + 2t \langle F(w_k) - F(x_k), (w_k - x_k) \rangle + t^2 \|w_k - x_k\|^2 \\ &\geq \|F(w_k) - F(x_k)\|^2 + t^2 \|w_k - x_k\|^2 \\ &\geq t^2 \|(w_k - x_k)\|^2. \end{aligned} \quad (4.63)$$

This together with (4.62) gives

$$t^2 \|s_k^{II}\|^2 \leq \|F(w_k) - F(x_k) + t(w_k - x_k)\|^2 \leq (t+L)^2 \|s_k^{II}\|^2. \quad (4.64)$$

Therefore, from (4.59), (4.61) and (4.64) we get

$$\frac{t}{(t+L)^2} \leq \frac{\langle y_k^{II}, s_k^{II} \rangle}{\|y_k^{II}\|^2} \leq \frac{t+L}{t^2}. \quad (4.65)$$

Also, from (4.64), we get

$$\frac{1}{t+L} \leq \frac{\|s_k^{II}\|}{\|y_k^{II}\|} \leq \frac{1}{t}. \quad (4.66)$$

Now, if for all k , $\theta_k = 0$, then $\tau_k^{II} = \frac{\langle y_k^{II}, s_k^{II} \rangle}{\|y_k^{II}\|^2}$ and so it follows from (4.65) that $\frac{t}{(t+L)^2} \leq \tau_k^{II} \leq \frac{t+L}{t^2}$. Furthermore, if for all k , $\theta_k = 1$, then $\tau_k^{II} = \frac{\|s_k^{II}\|}{\|y_k^{II}\|}$ and therefore, (4.66) gives $\frac{1}{t+L} \leq \tau_k^{II} \leq \frac{1}{t}$. Lastly, if for all k , $0 < \theta_k < 1$, then there exists a constant $\bar{t} > 0$ such that $\theta_k > \bar{t}$ and $1 - \theta_k > 0$. Thus, we obtain $\bar{t} \frac{\|s_k^{II}\|}{\|y_k^{II}\|} < \tau_k^{II} \leq \frac{\langle y_k^{II}, s_k^{II} \rangle}{\|y_k^{II}\|^2} + \frac{\|s_k^{II}\|}{\|y_k^{II}\|}$ and from (4.65) and (4.66), it follows that $\bar{t} \frac{1}{t+L} < \tau_k^{II} \leq \frac{t+L}{t^2} + \frac{1}{t}$. \square

Remark 4.2.2. *The following observations are made*

(i) *The search directions d_k^I and d_k^{II} satisfy the descent-like condition (4.19) based on Lemma 4.2.1, that is,*

$$\begin{cases} \langle d_k^I, F(x_k) \rangle \leq -\eta \|F(x_k)\|^2, \\ \langle d_k^{II}, F(x_k) \rangle \leq -\delta \|F(x_k)\|^2. \end{cases} \quad (4.67)$$

(ii) *The two search directions d_k^I and d_k^{II} satisfy the following inequalities*

$$\begin{cases} \eta \|F(x_k)\| \leq \|d_k^I\| \leq \mu \|F(x_k)\|, \\ \delta \|F(x_k)\| \leq \|d_k^{II}\| \leq \gamma \|F(x_k)\|. \end{cases} \quad (4.68)$$

We now state the steps of the proposed Two-step Hybrid Spectral gradient Projection algorithm (denoted as THSP) for solving Problem (1.10).

Algorithm 6: THSP method [for solving Problem (1.10)]

Input: Given $x_0 \in \Psi$, $0 < \kappa \leq 1$, $r, t > 0$, $\sigma, \rho \in (0, 1)$ and $\{\alpha_k\} \in (0, 1)$. Set $k = 0$.

1 **Step 1:** Compute $d_k^I := -F(x_k)$.

2 **Step 2:** if $d_k^I = 0$, then

3 | x_k is a solution and the iteration process stops.

end

4 **Step 3:** Compute $w_k := x_k + \alpha_k d_k^I$.

5 **Step 4:** Compute $d_k^{II} := -\tau_k^{II} F(x_k)$.

6 **Step 5:** Define a set $\{\kappa, \kappa\rho, \kappa\rho^2, \dots, \kappa\rho^n\}$,

7 **while**

$$-\langle F(x_k + \kappa\rho^i d_k^{II}), d_k^{II} \rangle \geq \sigma \kappa \rho^i \|d_k^{II}\|^2 \|F(x_k + \kappa\rho^i d_k^{II})\|^{1/q}, \quad q \geq 1, \quad (4.69)$$

do

8 | set $\beta_k := \kappa\rho^i$ where i is the smallest nonnegative integer.

end

9 **Step 6:** Set $z_k := x_k + \beta_k d_k^{II}$,

10 **if** $z_k \in \Psi$ and $\|F(z_k)\| \leq Tol$, **then**

11 | stop,

else

12 |

$$x_{k+1} := P_{\Psi} \left[x_k - \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} F(z_k) \right], \quad \|F(z_k)\| \neq 0. \quad (4.70)$$

end

13 **Step 7:** Set $k := k + 1$, update $d_k^I := -\tau_k^I F(x_k)$ and go to Step 2.

Remark 4.2.3. We claim that there exists a step-size β_k satisfying the line search (4.69) for any $k \geq 0$. Assume, however, that there exists some k_0 such that for each $i = 0, 1, 2, \dots$, the line search (4.69) is not satisfied, that is

$$-\langle F(x_{k_0} + \kappa \rho^i d(w_{k_0})), d(w_{k_0}) \rangle < \sigma \kappa \rho^i \|d(w_{k_0})\|^2 \|F(x_{k_0} + \kappa \rho^i d(w_{k_0}))\|^{1/q}. \quad (4.71)$$

Since F is continuous and τ_k^H is bounded for all k , letting $i \rightarrow \infty$ yields

$$\|F(x_{k_0})\| \leq 0. \quad (4.72)$$

It is clear that the inequality (4.72) cannot hold. Hence the line search (4.69) is well-defined.

The next Lemma is highly important for the convergence of Algorithm 6.

4.2.2 Convergence Analysis of THSP Algorithm

Lemma 4.2.4. The sequences $\{w_k\}$, $\{z_k\}$ and $\{x_k\}$ generated by Algorithm 6 are bounded if Assumption 4.1.1 is satisfied. Furthermore, there exist some positive constants b_1 , b_2 and b_3 such that

$$\begin{cases} \|F(x_k)\| \leq b_1 \\ \|F(w_k)\| \leq b_2 \\ \|F(z_k)\| \leq b_3. \end{cases} \quad (4.73)$$

Additionally,

$$\lim_{k \rightarrow \infty} \beta_k \|d_k^H\| = 0, \quad (4.74)$$

and

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0. \quad (4.75)$$

Proof. Let \hat{x} be a solution of Problem (1.10), then by monotonicity of F , we get

$$\begin{aligned} \langle F(z_k), x_k - \hat{x} \rangle &= \langle F(z_k), x_k - z_k + z_k - \hat{x} \rangle \\ &= \langle F(z_k), x_k - z_k \rangle + \langle F(z_k), z_k - \hat{x} \rangle \\ &\geq \langle F(z_k), x_k - z_k \rangle. \end{aligned} \quad (4.76)$$

By the definition of x_{k+1} in Step 6 of Algorithm 6 and (4.76) we obtain

$$\begin{aligned} \|x_{k+1} - \hat{x}\|^2 &= \left\| P_{\Psi} \left[x_k - \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} F(z_k) \right] - P_{\Psi}(\hat{x}) \right\|^2 \\ &\leq \left\| x_k - \hat{x} - \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} F(z_k) \right\|^2 \\ &= \|x_k - \hat{x}\|^2 - 2 \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} \langle F(z_k), x_k - \hat{x} \rangle + \frac{\langle F(z_k), x_k - z_k \rangle^2}{\|F(z_k)\|^2} \\ &\leq \|x_k - \hat{x}\|^2 - 2 \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} \langle F(z_k), x_k - z_k \rangle + \frac{\langle F(z_k), x_k - z_k \rangle^2}{\|F(z_k)\|^2} \\ &= \|x_k - \hat{x}\|^2 - \frac{\langle F(z_k), x_k - z_k \rangle^2}{\|F(z_k)\|^2} \\ &\leq \|x_k - \hat{x}\|^2. \end{aligned} \quad (4.77)$$

This implies that $\|x_k - \hat{x}\| \leq \|x_0 - \hat{x}\|$ for all k , and therefore the sequence $\{x_k\}$ is bounded and $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$ exists. Let b_1 be a positive constant such that $\|x_0 - \hat{x}\| = b_1/L$, since F is Lipschitz continuous, we get

$$\begin{aligned} \|F(x_k)\| &= \|F(x_k) - F(\hat{x})\| \\ &\leq L\|x_k - \hat{x}\| \\ &\leq L\|x_0 - \hat{x}\| \\ &= b_1. \end{aligned} \tag{4.78}$$

$\|d_k^I\| \leq \mu b_1$ and $\|d_k^{II}\| \leq \gamma b_1$ are deduced from (4.68). Moreover, (4.54) implies that $\{w_k\}$ is bounded. Since F is Lipschitz continuous, then $\|F(w_k)\| \leq b_2$ for some positive constant $b_2 > 0$. $\{z_k\}$ is bounded since $\{d_k^{II}\}$ is bounded, which follows from the definition of z_k in Step 6 of Algorithm 6. Also, there is a constant b_3 for which there exists a Lipschitz continuity of F , such that

$$\|F(z_k)\| \leq b_3. \tag{4.79}$$

Since the step-size β_k in Step 5 of Algorithm 6 satisfies $\beta_k \leq 1$ for all k , then from (4.69), we obtain

$$\sigma^2 \beta_k^4 \|d_k^{II}\|^4 \|F(z_k)\|^{2/q} \leq \sigma^2 \beta_k^2 \|d_k^{II}\|^4 \|F(z_k)\|^{2/q} \leq \langle F(z_k), \beta_k d_k^{II} \rangle^2, \quad q \geq 1.$$

Combining with (4.77) gives

$$\frac{\sigma^2 \beta_k^4 \|d_k^{II}\|^4 \|F(z_k)\|^{2/q}}{\|F(z_k)\|^2} \leq \frac{\langle F(z_k), \beta_k d_k^{II} \rangle^2}{\|F(z_k)\|^2} \leq \|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2, \quad q \geq 1. \tag{4.80}$$

From (4.79) and (4.80), we get

$$\sigma^2 \beta_k^4 \|d_k^{II}\|^4 \leq \|F(z_k)\|^{2-\frac{2}{q}} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2) \leq b_3^{2-\frac{2}{q}} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2).$$

Now, as $k \rightarrow \infty$, we obtain

$$\sigma^2 \lim_{k \rightarrow \infty} \beta_k^4 \|d_k^{II}\|^4 = 0. \tag{4.81}$$

Hence

$$\lim_{k \rightarrow \infty} \beta_k \|d_k^{II}\| = 0. \tag{4.82}$$

This, together with the definition of z_k in Step 6 of Algorithm 6 yields

$$\lim_{k \rightarrow \infty} \|z_k - x_k\| = 0. \tag{4.83}$$

From the nonexpansive property of metric projection, we get

$$\begin{aligned} \lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| &= \lim_{k \rightarrow \infty} \left\| P_{\Psi} \left[x_k - \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} F(z_k) \right] - x_k \right\| \\ &\leq \lim_{k \rightarrow \infty} \left\| x_k - \frac{\langle F(z_k), x_k - z_k \rangle}{\|F(z_k)\|^2} F(z_k) - x_k \right\| \\ &\leq \lim_{k \rightarrow \infty} \|x_k - z_k\| \\ &= 0. \end{aligned} \tag{4.84}$$

□

Theorem 4.2.5. *Suppose that Assumption 4.1.1 holds. Let $\{x_k\}$ be the sequence of iterates and d_k be the search direction generated by Algorithm 6. Furthermore, let the sequence $\{x_k\}$ be convergent to a point \hat{x} . Then $\liminf_{k \rightarrow \infty} \|F(x_k)\| = 0$ and $F(\hat{x}) = 0$.*

Proof. We begin by proving that

$$\liminf_{k \rightarrow \infty} \|F(x_k)\| = 0. \quad (4.85)$$

Suppose on the contrary that (4.85) does not hold, then there exists $\varepsilon > 0$ for which

$$\|F(x_k)\| \geq \varepsilon \text{ for all } k \geq 0. \quad (4.86)$$

Algorithm 6 employs a backtracking approach to compute β_k starting from κ . When $\beta_k \neq \kappa$, then $\rho^{-1}\beta_k$ does not satisfy (4.69), that is,

$$-\langle F(x_k + \rho^{-1}\beta_k d_k^{II}), d_k^{II} \rangle < \sigma \rho^{-1}\beta_k \|d_k^{II}\|^2 \|F(x_k + \rho^{-1}\beta_k d_k^{II})\|^{1/q}, \quad q \geq 1. \quad (4.87)$$

Consequently, we obtain from Remark 4.2.2 (i) that

$$\begin{aligned} \delta \|F(x_k)\|^2 &\leq -\langle d_k^{II}, F(x_k) \rangle \\ &= -\langle d_k^{II}, F(x_k) - F(x_k + \rho^{-1}\beta_k d_k^{II}) + F(x_k + \rho^{-1}\beta_k d_k^{II}) \rangle \\ &= -\langle d_k^{II}, F(x_k) - F(x_k + \rho^{-1}\beta_k d_k^{II}) \rangle - \langle d_k^{II}, F(x_k + \rho^{-1}\beta_k d_k^{II}) \rangle \\ &< -\langle d_k^{II}, F(x_k) - F(x_k + \rho^{-1}\beta_k d_k^{II}) \rangle + \sigma \rho^{-1}\beta_k \|d_k^{II}\|^2 \|F(x_k + \rho^{-1}\beta_k d_k^{II})\|^{1/q} \\ &\leq \|d_k^{II}\| \|F(x_k + \rho^{-1}\beta_k d_k^{II}) - F(x_k)\| + \sigma \rho^{-1}\beta_k \|d_k^{II}\|^2 \|F(x_k + \rho^{-1}\beta_k d_k^{II})\|^{1/q} \\ &\leq L \|d_k^{II}\| \|x_k + \rho^{-1}\beta_k d_k^{II} - x_k\| + \sigma \rho^{-1}\beta_k \|d_k^{II}\|^2 \|F(x_k + \rho^{-1}\beta_k d_k^{II})\|^{1/q} \\ &\leq L \rho^{-1}\beta_k \|d_k^{II}\|^2 + \sigma \rho^{-1}\beta_k \|d_k^{II}\|^2 \|F(x_k + \rho^{-1}\beta_k d_k^{II})\|^{1/q} \\ &\leq (L \rho^{-1}b + \sigma \rho^{-1}b b_4^{1/q}) \beta_k \|d_k^{II}\|, \end{aligned}$$

where b is the boundedness of d_k^{II} and $\|F(x_k + \rho^{-1}\beta_k d_k^{II})\|$ is bounded from above by a positive constant say b_4 . This means

$$\begin{aligned} \beta_k \|d_k^{II}\| &\geq \frac{\rho \delta \|F(x_k)\|^2}{b(L + \sigma b_4^{1/q})} \\ &\geq \frac{\rho \delta q^2}{b(L + \sigma b_4^{1/q})}. \end{aligned}$$

By considering the limits on both sides as $k \rightarrow \infty$, we arrive at the following

$$\lim_{k \rightarrow \infty} \beta_k \|d_k^{II}\| > 0. \quad (4.88)$$

This contradicts (4.82). Hence (4.85) must hold. Since F is continuous and the sequence $\{x_k\}$ is bounded, then there exists an accumulation point of $\{x_k\}$, say \hat{x} , for which $\|F(\hat{x})\| = 0$. The boundedness of $\{x_k\}$, allows us to find a subsequence $\{x_{k_j}\}$ of $\{x_k\}$ such that $\lim_{j \rightarrow \infty} \|x_{k_j} - \hat{x}\| = 0$. The proof of Lemma 4.2.4 demonstrates that $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$ exists. Thus, we can conclude that $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\| = 0$ and the proof is complete. \square

Furthermore, to demonstrate the efficiency and robustness of Algorithms 4 and 5, we provide their numerical experiments in Chapter 5 (particularly, Sections 5.2.1 and 5.2.2). Section 6.2 describes the applications of Algorithms 4 and 5 in motion control problems.

Chapter 5

Numerical Experiments

This chapter focuses extensively on the computing efficiency of the theoretical outcomes of the algorithms proposed in Chapters 3 and 4 of this thesis. It includes comparisons for solving large-scale convex constrained optimization problems. A great deal of Dolan and Morè [49] performance profiles and Morè and Wild [122] data profiles have been provided to demonstrate the behavior of the proposed algorithms. The main purpose of this chapter is to illustrate the computational capabilities of the algorithms presented in Chapters 3 and 4. These algorithms were tested in solving large-scale constrained monotone nonlinear equations and were compared to state-of-the-art algorithms, as discussed in various sections of the chapter. Most of these results are derived from the research conducted by Muhammad et al. in references [1, 123, 124, 125].

5.1 Experiments with the Derivative-Free Algorithms

In this section, we present some numerical experiments to assess the performance of our proposed algorithms as well as their computational advantages. The experiment is divided into two parts where the first part compares the performance of Algorithms 1, 2 and 3, that is,

- A Modified Descent Dai–Yuan Conjugate Gradient Method for Constraint Nonlinear Monotone Operator Equations proposed by Abubakar, Ibrahim, Muhammad and Tammer in [1] denoted as MDY1 and MDY2
- A Hybrid Conjugate Gradient Algorithm With Spectral Parameters for Solving Monotone Operator Equations With Convex Constraints and Application proposed by Muhammad, Tammer and Abubakar in [123] denoted as HCDLS
- A Dai–Liao–Like Projection Method for Solving Convex Constrained Nonlinear Monotone Equations and Minimizing the ℓ_1 –Regularized Problem proposed by Muhammad, Tammer, Awwal and Elster in [124] denoted as TDLP

for solving constrained monotone nonlinear equations while the second part demonstrates the performance of the best algorithm amongst our proposed algorithms, that is, Algorithm 3 (TDLP) in comparison to

- A Perry-Type Derivative-Free Algorithm for Solving Nonlinear System of Equations and Minimizing ℓ_1 -Regularized Problem, proposed in Awwal et al. [16] (denoted as DPP) and
- A Conjugate Gradient Projection Method for Solving Equations With Convex Constraints proposed in Zheng et al. [178] denoted as CGPM.

The executed algorithms were written on Windows 10 ASUSTek personal computer with Intel(R) Core(TM) i7-7500U processor with 8.00GB of RAM and CPU of 2.70GHz using MATLAB R2017a software.

5.1.1 Numerical comparisons of Algorithms 1, 2 and 3

In this experiment, we compare the performances and efficiency of Algorithms 1, 2 and 3, that is, MDY (MDY1, MDY2), HCDLS and TDLP methods on **Test Problems 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.1.6, 5.1.8, 5.1.9, 5.1.10 and 5.1.11**. In order to get the best possible results, the following parameters: $\kappa = 1$, $\rho = 0.4$, $\gamma = 1.99$, $\sigma = 0.0001$, $r = 0.01$, $q = 5$, $c = 10$ and $t = 0.1$ were chosen for Algorithm 1 (MDY1 & MDY2), Algorithm 2 (HCDLS) and Algorithm 3 (TDLP). Furthermore, all runs were terminated whenever $\|F(x_k)\| < 10^{-6}$. If this condition is not satisfied after 1000 iterations, failure is declared.

The **Test Problems** were taken from the existing literature and the function F is taken as $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, for $i = 1, 2, \dots, n$. However, the associated initial points for these **Test Problems** are given in Table 5.1. All the **Test Problems** were solved using the dimension (DIM) of $n = 1000, 5000, 10000, 50000$ and 100000 .

Table 5.1: The initial points used for Algorithm 1, 2, 3, 4 and 5

Initial Points (INP)	Values
x_1	$(1, 1, 1, \dots, 1)^T$
x_2	$(\frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \dots, \frac{1}{10})^T$
x_3	$(\frac{1}{2}, \frac{1}{2^2}, \frac{1}{2^3}, \dots, \frac{1}{2^n})^T$
x_4	$(1 - \frac{1}{n}, 1 - \frac{2}{n}, 1 - \frac{3}{n}, \dots, 0)^T$
x_5	$(0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n})^T$,
x_6	$(1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{n})^T$
x_7	$(\frac{n-1}{n}, \frac{n-2}{n}, \frac{n-3}{n}, \dots, 0)^T$
x_8	$(\frac{1}{n}, \frac{2}{n}, \frac{3}{n}, \dots, 1)^T$
x_9	rand(0, 1)
x_{10}	$(\frac{3}{2}, \frac{3}{2}, \frac{3}{2}, \dots, \frac{3}{2})^T$
x_{11}	$(2, 2, 2, \dots, 2)^T$
x_{12}	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})^T$
x_{13}	$5 \min(ih, 1 - ih)$, $1 \leq i \leq n$, $h = 1/(n+1)$
x_{14}	$(\frac{-1}{4}, \frac{2}{5}, \frac{-3}{6}, \dots, \frac{(-1)^n}{n+3})^T$

Test Problems:**Problem 5.1.1. [107]:**

$$f_1(x) = e^{x_1} - 1,$$

$$f_i(x) = e^{x_i} + x_{i-1} - 1, \text{ for } i = 1, 2, \dots, n-1 \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5.1.2. [107]:

$$f_i(x) = \ln(x_i + 1) - \frac{x_i}{n}, \text{ for } i = 2, 3, \dots, n,$$

$$\text{and } \Psi = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq n, x_i > -1, i = 1, 2, \dots, n\}.$$

Problem 5.1.3. [13]:

$$f_i(x) = 2x_i - \sin|x_i|, \text{ } i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5.1.4. [18]:

$$f_i(x) = e^{x_i} - 1, \text{ } i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5.1.5. [27]:

$$f_1(x) = x_1 - e^{\cos(h(x_1+x_2))},$$

$$f_i(x) = x_i - e^{\cos(h(x_{i-1}+x_i+x_{i+1}))}, \text{ } i = 2, \dots, n-1,$$

$$f_n(x) = x_n - e^{\cos(h(x_{n-1}+x_n))}, \text{ where } h = \frac{1}{n+1} \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5.1.6. [168]:

$$f_i(x) = x_i - \sin(|x_i - 1|), \text{ } i = 1, 2, \dots, n-1,$$

$$\text{and } \Psi = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq n, x_i \geq -1, i = 1, 2, \dots, n\}.$$

Problem 5.1.7. [16]:

$$f_i(x) = (e^{x_i})^2 + \frac{3}{2} \sin(2x_i) - 1, \text{ } i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5.1.8. [2]:

$$f_1(x) = \frac{5}{2}x_1 + x_2 - 1,$$

$$f_i(x) = x_{i-1} + \frac{5}{2}x_i + x_{i+1} - 1, \text{ } i = 2, \dots, n-1,$$

$$f_n(x) = x_{n-1} + \frac{5}{2}x_n - 1 \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5.1.9. [19]:

$$f_1(x) = 2x_1 - x_2 + e^{x_1} - 1,$$

$$f_i(x) = -x_{i-1} + 2x_i - x_{i+1} + e^{x_i} - 1, \text{ } i = 2, \dots, n-1,$$

$$f_n(x) = -x_{n-1} + 2x_n + e^{x_n} - 1 \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5.1.10. [107]:

$$f_i(x) = \frac{i}{n} e^{x_i} - 1, \text{ for } i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5.1.11. [178]:

$$f_1(x) = x_1 + \sin x_1 - 1,$$

$$f_i(x) = -x_{i-1} + 2x_i + \sin x_i - 1, \quad i = 2, \dots, n-1,$$

$$f_n(x) = x_n + \sin x_n - 1 \text{ and } \Psi \in [-3, +\infty].$$

The metrics used for the comparison include ITER (number of iterations), FVAL (number of function evaluations), and TIME (CPU time in seconds), along with NORM (norm of the objective function at the solution). The table of the numerical results is available in the following link <https://github.com/AMBakoji/CGAIS-CGWOIS>. The NORM values represent whether or not an algorithm effectively produces an approximate solution to a certain problem.

Based on the NORM values presented in the Tables, it is evident that the MDY1, MDY2, HCDLS, and TDLP methods effectively solved all of the **Test Problems**. Considering the ITER values reported in the Tables, though all the four methods recorded relatively low ITER, it can be seen that the TDLP method recorded the least values of ITER in most cases. When comparing the FVAL values obtained from the MDY1, MDY2, HCDLS and TDLP methods, it is observed that the TDLP method requires the lowest FVAL in order to acquire the solutions for the **Test Problems**. Finally, the Tables demonstrate that the four methods are able to solve the majority of the **Test Problems** in less than one second. However, for problems with a dimension of 100000, the TDLP method takes longer to compute than the HCDLS method, while the MDY1 and MDY2 methods require more computing time than both the TDLP and HCDLS methods. This could be due to multiple parameters that need to be computed using the MDY1 and MDY2 methods. Upon careful evaluation, it is evident that the TDLP method demonstrates superior performance compared to the MDY1, MDY2 and HCDLS methods.

In order to obtain a graphical representation of the numerical performance of the four methods, that is, MDY1, MDY2, HCDLS, and TDLP, we summarized the information on the Tables of our numerical experiments and used the performance profile proposed by Dolan and Morè in [49]. The performance profile takes into account the number of iterations (ITER), the CPU time in seconds (TIME), and the number of function evaluations (FVAL). The process is as follows:

Let P be the set of test problems and n_p the number of test problems. S denotes the collection of methods/solvers, whereas n_s represents the number of solvers. For each solver $s \in S$ and problem $p \in P$, let $j_{p,s}$ denote either the ITER, FVAL or TIME required to solve a problem p by a solver s . To compare a solver's performance on a problem p to that of other solvers, we use the performance ratio defined as

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

The performance ratio of a solver s is calculated using the (cumulative) distribution function. The ρ profile is defined as follows:

$$\rho_s(\tau) := \frac{1}{n_p} \text{size}\{p \in P : \log_2 r_{p,s} \leq \tau\},$$

where $\tau > 0$ and $r_{p,s}$ represents the performance ratio. $\rho(\tau)$ is the probability for solvers $s \in S$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ (which is in \log_2 scale) of the best possible ratio. The performance profile demonstrates the *efficiency* and *robustness* of the proposed algorithms compared to other state-of-the-art algorithms, *SOTA*. We define an algorithm as *efficient* when it has the lowest relative cost of the *metric*, number of function evaluations, and *robust* when it has the most solved problems in comparison to others. In general, the solver with the highest performance profile value $\rho_s(\tau)$ is regarded as the effective method for a given τ value. Put simply, the solver that exhibits the highest degree of dominance at the top of the curve is the most efficient method when compared to the others.

Figure 5.1(a) shows the performance profile of the four solvers based on ITER, Figure 5.1(b) shows the performance profile based on FVAL while Figure 5.1(c) shows the performance profile based on TIME (CPU time in seconds). Figures 5.1(a), 5.1(b) and 5.1(c) show that the TDLP solver outperforms the MDY1 and MDY2 solvers, as well as the HCDLS solver, in terms of iteration and function evaluations.

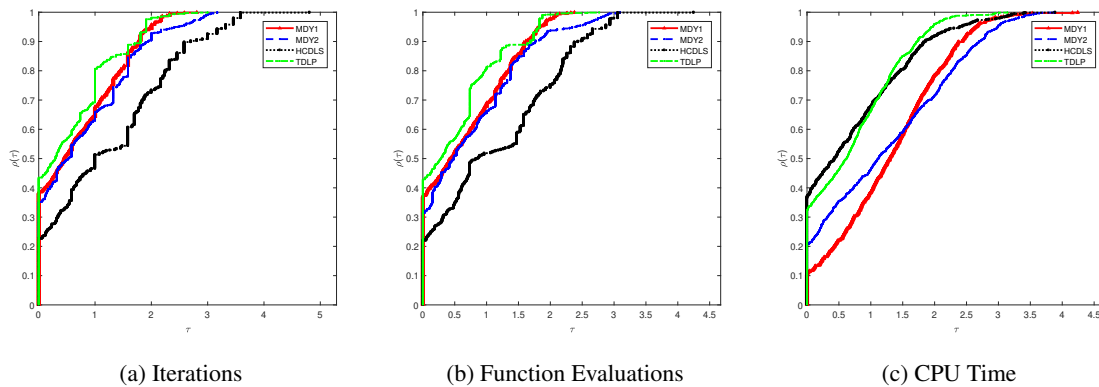


Figure 5.1: Performance profile based on number of: Iterations, Function Evaluations CPU Time.

However, in order to know the cost optimization of the solvers for the Test Problems with expensive function evaluations, we use the Data Profile proposed by Moré and Wild in [122]. In this scenario, the performance metric is determined by the number of function evaluations (FVAL) as it is considered to be the primary cost per iteration. Performance profiles provide a precise assessment of the comparative performance of solvers based on a specific number of function evaluations, as depicted in Figure 5.1(b). Performance profiles, unfortunately, do not provide adequate information for a user dealing with a costly optimization problems.

Users who have expensive optimization problems that need significant resources are frequently concerned about the performance of solvers in relation to the number of function evaluations. In other words, these users are interested in the percentage of problems that can be solved (given a tolerance τ) using a specific number of μ function evaluations. Performance profiles and data profiles are cumulative distribution functions, which means they are step functions that increase monotonically and have a range between 0 and 1. However, performance profiles provide a comparison of various solvers, whereas data profiles present the raw data without any modifications. Specifically, performance profiles do not indicate the exact number of function evaluations needed to solve any of the problems. It is important to understand that the data profile of a specific solver say $s \in S$, is not influenced by other solvers. However,

this is not true for performance profiles [122].

Data profiles are useful to users with a specific computational budget who need to choose a solver that is likely to reach a given reduction in function value. The user must specify the computational budget using simplex gradients and analyze the data profile values for all solvers. For instance, if the user has a budget of 10 simplex gradients (FVAL), then the data profiles in Figure 5.2 (top-left, top-right and bottom-left) indicate that the MDY2 solver successfully solves approximately 24% of the problems at this level of accuracy. In comparison, the HCDLS solver solves around 11% of the problems and the TDLP solver solves about 28% of all the problems. In addition, when a user is limited to 30 simplex gradients (FVAL), the data profiles indicate that MDY2 solver successfully solves approximately 60% of the problems while HCDLS and TDLP solvers solve about 50% and 62%, respectively. Finally, if a user has a budget of 50 simplex gradients, then the data profiles show that MDY2 solver solves about 69% of the problems while HCDLS and TDLP solvers solve about 85% and 86% of the **Test Problems**, respectively. This means that at some point, all the three solvers are in a high competition with MDY2 and HCDLS methods been more computationally expensive compared to the TDLP method in order to successfully solve certain percentage of the entire seven hundred (700) **Test Problems** we have considered. However, this information is not available from the performance profile in Figure 5.1(b). This interpretation means that data profile measures the reliability of the solver for a given tolerance as a function of the budget.

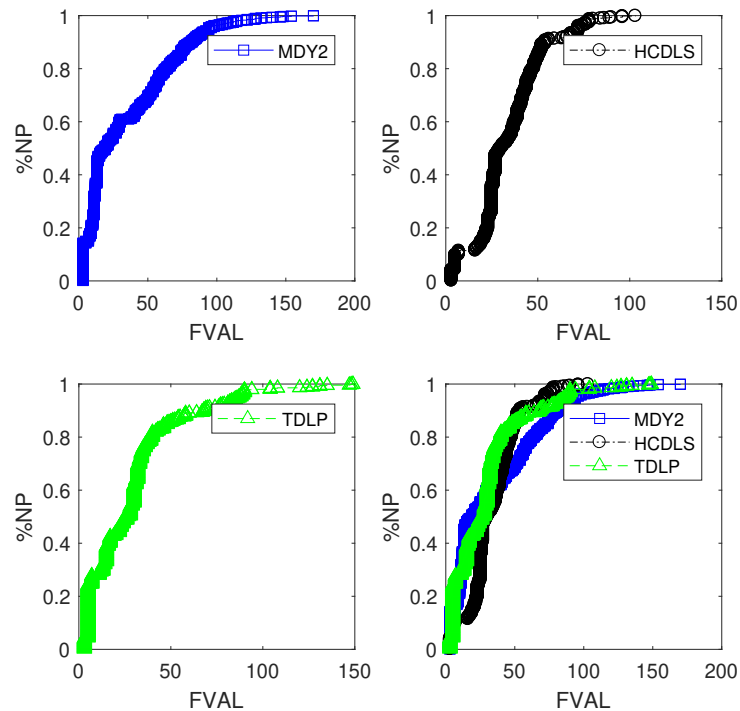


Figure 5.2: Percentage Data Profile of MDY2, HCDLS and TDLP methods.

5.1.2 Numerical Comparisons of Algorithm 3 (TDLP) with DPP and CGPM Methods

In this experiment, we are comparing the performance of Algorithm 3 (TDLP) with two other algorithms. The first algorithm, proposed by Awwal et al., [16], is a derivative-free algorithm for solving nonlinear

systems of equations and minimizing ℓ_1 -regularized problems (denoted as DPP). The second algorithm, proposed by Zheng et al., [178] is a conjugate gradient projection method for solving equations with convex constraints, (denoted as CGPM). We have successfully solved **Test Problems: 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.6, 5.1.7, 5.1.8, 5.1.9, 5.1.10 and 5.1.11** using the three solvers. The parameters for Algorithm 3 (TDLP) were selected as follows: $\kappa = 1$, $\rho = 0.4$, $\gamma = 1.99$, $\sigma = 0.0001$, $r = 0.01$, $q = 5$, $c = 10$ and $t = 0.1$. The values for the DPP and CGPM methods were set according to the information provided in references [16] and [178], respectively. Seven hundred (700) **Test Problems** were solved using the same initial points and dimensions as specified in Table 5.1.

The metrics employed for comparison in this experiment are ITER (number of iterations), FVAL (number of function evaluations), and TIME (CPU time in seconds). The information regarding these metrics, along with NORM (norm of the objective function at the solutions), is presented. The numerical findings can be found in the table provided at the following link: <https://github.com/AMBakoji/CGAIS-CGWOIS>. We summarized these information in Figure 5.3(a), 5.3(b) and 5.3(c) using the Dolan and Moré performance profile [49]. Based on the experimental results, it is evident that the proposed TDLP algorithm consistently outperforms the two other algorithms in terms of ITER, FVAL, and CPU TIME. The superiority of the proposed TDLP algorithm is clearly demonstrated in Figure 5.3(a), 5.3(b) and 5.3(c) where it outperforms other two algorithms with approximately 73% higher ITER, 80% higher FVAL, and improved CPU TIME.

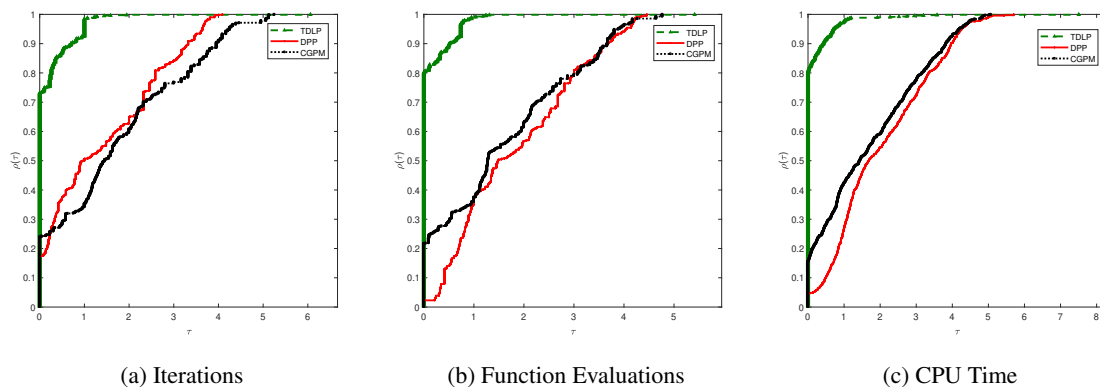


Figure 5.3: Performance profile based on number of: Iterations, Function Evaluations CPU Time .

5.2 Numerical Experiments and Comparisons for Algorithms 4 and 5

In this section, we present some numerical experiments to assess the performance of Algorithms 4 and 5 as well as their computational advantages in comparison with some existing methods. We implement these algorithms to solve a collection of monotone system of nonlinear equations, see **Test Problems** for Algorithm 4 (Section 5.2.1), also known as **Conjugate Gradient Algorithm with Inertial Step (CGAIS)** as well as Algorithm 5 (Section 5.2.2) that is, **Spectral Algorithm with Inertial Step (SAIS)**.

5.2.1 Numerical Comparisons of Algorithm 4 (CGAIS) With CGWOI Method

This experiment aims to evaluate the performance of Algorithm 4 (CGAIS) in comparison to a modified version of the same algorithm called CGWOI (Conjugate Gradient WithOut Inertial step) as well as Algorithm 1 of Awwal et al. in [19], denoted as HSS. The evaluation is conducted on **Test Problems**: 5.1.1, 5.1.3, 5.1.4, 5.1.5, 5.1.6 and 5.1.10. In other words, we aim to evaluate the numerical efficiency of a method that includes an inertial step in comparison to two other methods that do not incorporate an inertial step. In order to have best possible results, the following parameters were chosen for the implementation of CGAIS and CGWOI methods used the following parameters to get optimal results: $\sigma = 10^{-4}$, $r = 0.01$, $c = 2$, $\rho = 0.50$, $\gamma = 1.99$, $\kappa = 1$ and $\alpha_k = \frac{1}{(k+1)^2}$. The settings for the HSS method were selected according to the information provided in reference [19]. In addition, the iteration process for the **Test Problems** is terminated whenever the inequality $\|F(x_k)\| < 10^{-6}$ or $\|F(v_k)\| < 10^{-6}$ is met. If the number of iterations exceeds 1000 and the aforementioned termination requirement has not been satisfied, it is considered a failure.

However, since our proposed algorithm uses two starting points, that is, x_{-1} and x_0 , then for each x_j , $j = 1, 2, \dots, 14$, taken from Table 5.1, we set $x_{-1} := \{x_j^1 + i, x_j^2 + i, \dots, x_j^n + i\}$, $i \geq 0$, and update them subsequently. All the Test Problems were solved using the same dimension (DIM) of $n = 1000, 5000, 10000, 50000$ and 100000 as in Section 5.1.1.

In this section, the same metrics used for the comparison in Section 5.1.1 are utilized. These metrics include ITER (number of iterations), FVAL (number of function evaluations), and TIME (CPU time in seconds). Additionally, the NORM (norm of the objective functions at the solutions) is also supplied along with this information. The Table of the numerical results can be accessed using the following link: <https://github.com/AMBakoji/CGAIS-CGWOIS>.

The NORM values reported indicate that each solver successfully obtained solutions of virtually all the **Test Problems** with least ITER and FVAL. These information are summarized in Figures 5.4 and 5.5 based on the Dolan and Moré performance profile [49]. Figure 5.4 (a) compare the performance between CGAIS and CGWOI methods based on ITER while Figure 5.4 (b) illustrates the Dolan and Moré performance profile of CGAIS and HSS methods based on ITER, respectively. Figure 5.5 (a) and (b) demonstrates the performance profile based on FVAL of CGAIS with CGWOI and FVAL of CGAIS with HSS methods, respectively. We see from Figure 5.4 ((a) and (b)) and Figure 5.5 ((a) and (b)) that CGAIS solver perform better with higher percentage win of ITER and FVAL than CGWOI and HSS solvers for solving all the four hundred and twenty (420) Test Problems. In other words, this experiment reveals that CGAIS solver has advantage over CGWOI and HSS solvers with regards to ITER and FVAL. Thus, we may conclude that the inertial step incorporated in Algorithm 4 have impacted positively in improving the numerical performance of the proposed algorithm.

However, to determine the cost optimization of the solvers for the **Test Problems** with expensive function evaluations, we once again utilize the Data Profile proposed by Moré and Wild in [122] as previously explained. In this context as well, the performance metric is determined by the number of function evaluations (FVAL) as it is considered to be the primary cost per iteration. Performance profiles offer a precise assessment of the comparative performance of solvers based on a specific number of function

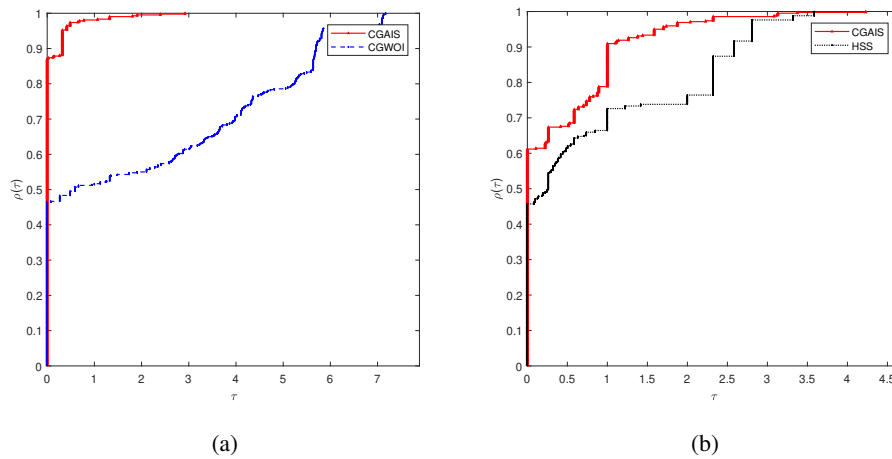


Figure 5.4: Performance profiles based on number of iterations (ITER) [reprinted from [125, Figure 1]].

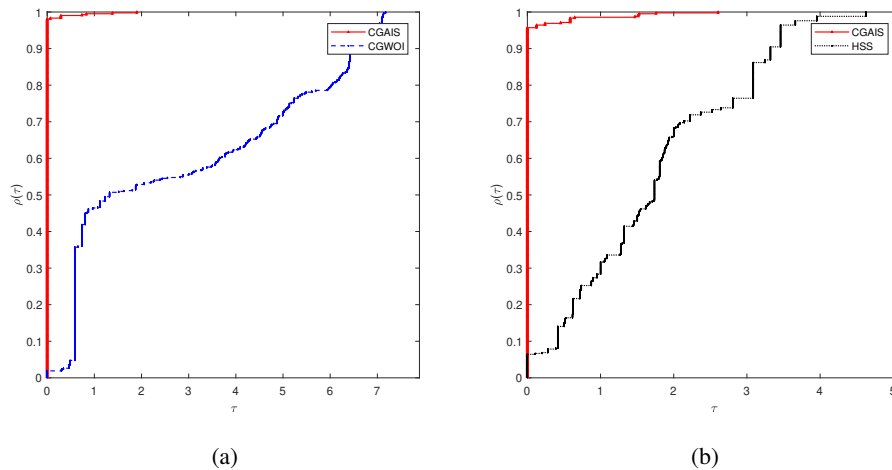


Figure 5.5: Performance profiles based on number of function evaluations (FVAL) [reprinted from [125, Figure 2]].

evaluations, as depicted in Figure 5.5. For instance, if the user's budget is 20 simplex gradients (FVAL), the data profiles in Figure 5.6 (top-left, top-right, and bottom-left) indicate that the proposed CGAIS solver achieves a success rate of approximately 99% at this level of precision. In contrast, the CGWOI solver solves around 50% of the problems while the HSS solver solves about 63% of all the problems. The CGWOI and HSS methods are more computationally costly than the proposed CGAIS method when solving a specified percentage of the total four hundred and twenty 420 **Test Problems** we have examined. Unfortunately, the performance profile in Figure 5.5 does not provide this information. This interpretation implies that the data profile measures the reliability of the solver based on a specified tolerance level and budget constraints.

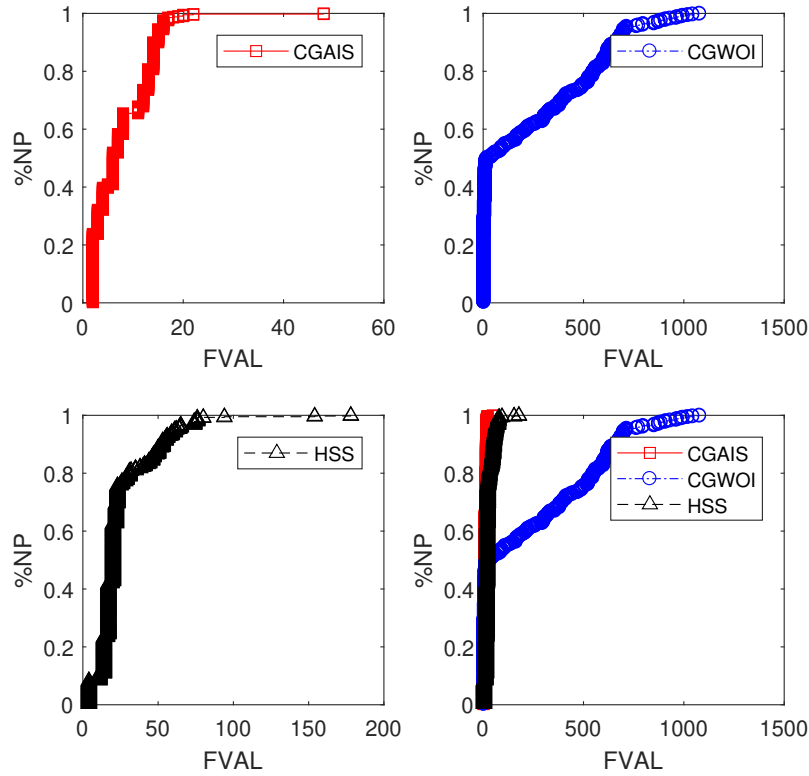


Figure 5.6: Percentage Data Profile of CGAIS, CGWOI and HSS methods [reprinted from [125, Figure 3]].

5.2.2 Numerical Comparisons of Algorithm 5 (SAIS) With DAIS 1 and DAIS 2 Methods

Similar to the experiment one, we assess the performance of Algorithm 5 (SAIS) in comparison to Algorithms 1 and 2 of Awwal et al. [17], which are referred to as DAIS 1 and DAIS 2, respectively. All three solvers are spectral gradients and each of them includes an inertial step. This indicates that the three solvers share comparable characteristics. The objective of this experiment is to assess and contrast their numerical capabilities. We have successfully solved **Test Problems 5.1.1, 5.1.3, 5.1.7, 5.1.8, 5.1.9** and **5.1.10** using all three solvers. The values used for Algorithm 5 (SAIS) are as follows: $\sigma = 10^{-4}$, $r = 0.01$, $c = 2$, $\gamma = 1.99$, $\rho = 0.45$, $\kappa = 1$ and $\alpha_k = \frac{1}{(k+1)^2}$. The parameters for DAIS 1 and DAIS 2 were chosen according to the information provided in reference [17]. A total total of 420 **Test Problems** were solved, using same initial points and dimensions as in experiment one (see, for example, Section 5.1.1). Specific numerical findings from the table are provided and can be accessed through the following link: <https://github.com/AMBakoji/CGAIS-CGWOIS>.

To visually compare the numerical performance of the SAIS solver with the DAIS 1 and DAIS 2 solvers in terms of the ITER and FVAL metrics, we utilize the Dolan and Moré performance profile. Figure 5.7(a) depicts the performance profile of the three solvers using ITER, whereas Figure 5.7(b) showcases the performance profile using FVAL. By examining Figure 5.7(a) and 5.7(b), it is evident that the SAIS solver achieved the lowest number of iterations (ITER) and function values (FVAL), winning almost 90%

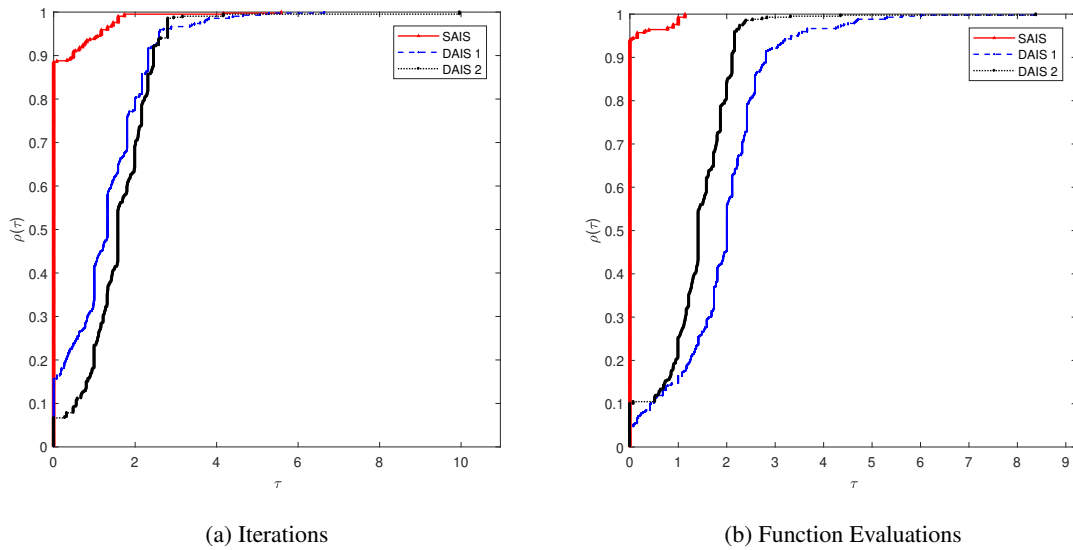


Figure 5.7: Performance profiles based on number of iterations and function evaluations [reprinted from [125, Figures 4 and 5]].

of all the experiments conducted. Our proposed algorithm (SAIS) demonstrated superior performance compared to DAIS 1 and DAIS 2 algorithms, respectively. To put it simply, the experiment demonstrates that the SAIS solver outperforms the DAIS 1 and DAIS 2 solvers in terms of the ITER and FVAL metrics. To obtain more comprehensive information regarding the numerical experiments discussed in this chapter, we direct readers to the research conducted by Muhammad et al. as documented in [1, 123, 124, 125] and the related references.

Chapter 6

Applications in Compressive Sensing and Motion Control Problems

In this chapter, we provide a practical application of Algorithm 1 (MDY), Algorithm 2 (HCDLS) and Algorithm 3 (TDLP) described in Chapter 3 in the reconstruction of sparse signals and image restoration in compressive sensing. Furthermore, the chapter also presents application of Algorithm 4 (CGAIS) and Algorithm 5 (SAIS) studied in Chapter 4 of this thesis in motion control problem involving a two-joint planar robotic manipulator as can be seen in Section 6.2. These results are mostly from the work by Muhammad et al., in [1, 123, 124, 125].

6.1 Applications in Compressive Sensing

Let us quickly recall the description and reformulations ℓ_1 -norm in compressive sensing, which were discussed in Chapters 1 and 2 and are brought to the attention of the readers for their convenience. Compressive sensing (CS) is an important function that is utilized in a variety of scientific and engineering fields, including medical sciences, biological engineering, and other fields [21, 36]. The CS problem involves the process of recovering a sparse signal, denoted as x , from a linear system $Ax = b$. In this system, the vector $b \in \mathbb{R}^k$ represents the observations, and the linear operator $A \in \mathbb{R}^{k \times n}$ ($k \ll n$) represents the sensing matrix within the system. The preceding system of linear equation is transformed into the following so-called " ℓ_0 -norm" optimization problem with constraints

$$\min_x \{\|x\|_0 : Ax = b\},$$

where the expression $\|x\|_0$ is determined by the number of non-zero components of a vector $x \in \mathbb{R}^n$. The objective of this conversion is to search for the solution set that contains the fewest elements. Regrettably, the set of values $\|\cdot\|_0$ does not exhibit a positive homogeneity and does not constitute a norm. On the other hand, because of the challenges that are connected with problems involving the ℓ_0 -norm, an alternative approach is to replace it with the ℓ_1 -norm (see [38]), that is,

$$\min_x \{\|x\|_1 : Ax = b\},$$

where $\|x\|_1 := \sum_{i=1}^n |x_i|$. It has been demonstrated that the aforementioned problem can give the necessary outcomes with some degree of precision that is acceptable, provided that certain assumptions are made. When the measurements are influenced by a certain amount of noise, the constraint in the ℓ_1 -norm is usually relaxed to the following regularized least squares (also known as Lasso) problem

$$\min_x \tau \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2, \quad (P^\tau)$$

where τ represent a positive regularization parameter and $\|\cdot\|_2$ represents the Euclidean norm of \mathbb{R}^n . This is exactly the repetition of Problem (1.3), which will also be studied in Chapter 7 as a scalarization of a vector-valued approximation problem (see Section 7.5).

Iterative methods for solving (P^τ) have been presented in several papers, (see [25, 29, 58, 75]). Due to the fact that the proposed algorithms are derivative-free, they are able to successfully handle nonsmooth problems that are in the form of (1.10). Therefore, in order to solve the nonsmooth Problem (P^τ) , we change (P^τ) into the form of (1.10). Hu, Wang, and Xiao [167] were able to successfully translate the Lasso Problem (P^τ) into a nonlinear system of equations. This translation was based on the work of Figueiredo, Nowak, and Wright [58].

Let x be any vector in \mathbb{R}^n . Then, x can be divided into positive and negative components, defined as $x := u - v$, $u \geq 0$, $v \geq 0$, where $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$ and $u_i = (x_i)_+$, $v_i = (-x_i)_+$, for all $i = 1, 2, \dots, n$ with $(\cdot)_+ = \max\{0, \cdot\}$. The ℓ_1 -norm of a vector x can be represented as $\|x\|_1 = e_n^T u + e_n^T v$, where $e_n = (1, 1, \dots, 1)^T \in \mathbb{R}^n$. Hence, the ℓ_1 -norm Problem (P^τ) can be transformed as

$$\min_{u,v} \frac{1}{2} \|b - A(u - v)\|_2^2 + \tau e_n^T u + \tau e_n^T v, \quad \text{such that } \tau \geq 0, u \geq 0, v \geq 0.$$

However, according to reference [58], the above mentioned problem can alternatively be reformulated as a quadratic programming problem with box constraints

$$\min_z \frac{1}{2} z^T D z + c^T z, \quad \text{such that } z \geq 0,$$

$$\text{where } z = \begin{bmatrix} u \\ v \end{bmatrix}, \quad c = \tau e_{2n} + \begin{bmatrix} -x \\ x \end{bmatrix}, \quad x = A^T b \quad \text{and} \quad D = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}.$$

Clearly, D is a positive semi-definite matrix, which implies that the aforementioned problem is a convex quadratic problem.

Hu, Wang and Xiao [167] translated $\min_z \frac{1}{2} z^T D z + c^T z$, $z \geq 0$, into a linear variable inequality problem which is equivalent to a linear complementary problem. Furthermore, they underlined that z can be viewed as a solution to the linear complementary problem if and only if it also solves the nonlinear equation

$$F(z) = \min\{z, D z + c\} = 0,$$

where $F(\cdot)$ is considered to be continuous and monotone as referenced in [131, 167]. So, the solution to Problem (P^τ) is the same as the solution to Problem (1.10), which is to say that $F(x) = 0$. Consequently, the algorithms that we have proposed, namely Algorithm 1 (MDY), Algorithm 2 (HCDLS) and Algorithm 3 (TDLP), have the capability to accomplish the task of solving (P^τ) in an efficient manner.

The process of signal reconstruction has involved the development of a variety of algorithms over the course of time. On the other hand, with the intention of working with a small number of measurements, there are still proposals being made for algorithms that are both efficient and robust. Accordingly, there is a research field that is both interesting and significant, and it is committed to the process of developing optimal explicit measurement matrices as well as all known "good" matrix constructions that consist of randomness. In the field of signal and image processing, this discovery offers a great deal of potential applications.

Since they do not necessitate the knowledge of the derivative of the objective function, the algorithms that we proposed are suitable for treating nonsmooth functions. This is the primary advantage that our algorithms propose. In order to apply Algorithm 1 (MDY1), Algorithm 2 (HCDLS) and Algorithm 3 (TDLP) in the reconstruction of a sparse signal in compressive sensing, we take into consideration a typical compressive sensing scenario. In this scenario, the ultimate objective is to reconstruct a sparse signal of length n from k observations ($k \ll n$) with Gaussian noise. The number of samples required for this reconstruction is significantly smaller than the size of the original signal. In this test, we considered a small size of signal with $n = 2^{12}$, $k = 2^{10}$ because our personal computer has a restricted amount of memory. For the original signal, x , there are 2^7 elements that are not zero and are chosen at random. The quality assessment of the restored signal is based on the mean squared error (MSE) to the original signal x , which is calculated using the following equation:

$$MSE := \frac{1}{n} \|x - x^*\|^2$$

where x^* represents the restored or recovered signal. This method is similar to the ones described in [163, 167, 168]. In this test, we generate the random matrix A using the Matlab command `rand(n,k)`. In addition, the noise is appropriately added to the observed data computed by

$$b := Ax + \mu$$

where μ represents the Gaussian noise that is normally distributed with a mean of 0 and a variance of 10^{-4} . The parameters that are used in the implementation of the TDLP, HCDLS, and MDY1 methods are as follows: $\kappa = 10$, $\rho = 0.8$, $\sigma = 10^{-4}$, $q = 5$, $c = 9$ and $\gamma = 1.99$. The parameter τ in the merit function $f(x) = \tau \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2$ was selected in accordance with reference [103]. The value of τ was determined to be $\tau = 0.008 \|A^T b\|_\infty$ and the initial points for all the methods begin at $x_0 = A^T b$. The process are terminated whenever the inequality holds

$$Tol := \left| \frac{f(x_k) - f(x_{k-1})}{f(x_{k-1})} \right| < 10^{-5},$$

where $f(x_k)$ denotes the function value at x_k . For the purpose of this test, it is important to note that we only observe the convergence behavior of each method in order to achieve a solution with comparable accuracy.

It is not difficult to understand, in light of the plots that are displayed in Figure 6.1, that the original signal is effectively recovered by the three methods with the least amount of computing time (TIME), the least number of iterations (ITER), and the least mean square error (MSE). When it came to decoding

the sparse signal, however, the TDLP succeeded. Iterations (ITER), computing time (TIME), and most significantly, mean squared error (MSE) are all lower, which is a reflection of this fact. Using twenty distinct noise samples, we repeated the experiment in order to provide a more comprehensive illustration of the effectiveness of the TDLP in comparison to the HCDLS and MDY1. Each time the experiment is carried out, the TDLP method is shown to be more effective in terms of the amount of time utilized by the CPU, the number of iterations and most importantly the mean square error (MSE). In Table 6.1, you can find a summary.

Table 6.1: Twenty experimental results for sparse signal recovery.

	TDLP			HCDLS			MDY1		
	ITER	TIME	MSE	ITER	TIME	MSE	ITER	TIME	MSE
	53	1.05	4.59E-05	80	1.64	2.19E-05	63	1.59	2.09E-05
	59	1.02	4.83E-05	100	1.64	2.19E-05	56	0.94	2.20E-05
	72	1.20	5.76E-05	96	1.52	2.51E-05	67	1.11	2.54E-05
	58	1.00	6.57E-05	96	1.45	2.60E-05	64	1.05	2.66E-05
	58	1.08	6.02E-05	91	1.52	2.60E-05	58	1.02	2.61E-05
	57	1.03	4.59E-05	88	1.30	1.69E-05	60	1.02	1.68E-05
	58	1.03	3.51E-05	87	1.34	1.95E-05	65	1.05	4.99E-05
	57	1.05	4.30E-05	94	1.48	2.28E-05	61	1.03	2.28E-05
	50	0.77	5.77E-05	83	1.28	2.42E-05	69	1.20	2.38E-05
	53	0.89	4.26E-05	93	1.44	1.81E-05	63	1.06	3.32E-05
	64	1.14	2.14E-05	77	1.19	2.07E-05	66	1.16	4.65E-05
	55	1.27	4.36E-05	89	1.56	2.07E-05	68	1.22	1.56E-05
	57	1.06	2.03E-05	86	1.31	1.52E-05	64	1.03	1.50E-05
	51	0.81	2.15E-05	89	1.41	2.12E-05	68	1.03	2.13E-05
	60	1.00	4.30E-05	89	1.36	2.29E-05	66	1.02	6.11E-05
	55	1.02	5.85E-05	83	1.22	2.17E-05	55	0.91	3.93E-05
	62	1.00	7.40E-05	100	1.50	3.76E-05	61	0.97	3.76E-05
	71	1.27	6.65E-05	91	1.52	2.57E-05	69	1.11	2.61E-05
	79	1.28	7.00E-05	89	1.36	2.83E-05	111	1.80	5.68E-05
	57	0.94	4.72E-05	87	1.34	2.63E-05	68	1.11	4.27E-05
Average	59.3	1.05	4.84E-05	89.4	1.42	2.31E-05	66.1	1.12	3.15E-05

Table 6.2: Approximate solutions $\bar{x} := x_k \in \mathbb{R}^n$ for the sparse signal recovery experiments of Algorithm 1 (MDY1), Algorithm 2 (HCDLS) and Algorithm 3 (TDLP).

TDLP	HCDLS	MDY1
\bar{x}_{TDLP}	\bar{x}_{HCDLS}	\bar{x}_{MDY1}
0	0	0
0.002046	0	0
0	0	0
0	0	0
0	0	0
-0.00126	-7.56E-05	-2.00E-06
-0.00134	-0.00012	-2.95E-06
0	0	0
0.000773	-0.00012	-4.73E-06
-0.00053	-0.0001	-3.51E-06
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
-0.00928	8.55E-05	1.75E-05
0	0	0
0	0	0
0	0	0
...
0.001939	0.000118	3.07E-06
-4.42E-05	-1.98E-05	-5.75E-07
0	0	0
0	0	0
0.002397	0.000221	5.58E-06
0.000924	3.45E-05	-5.86E-06
0	0	0
0	0	0
-0.00274	-7.41E-05	-2.22E-06
-0.00243	-0.00017	-4.37E-06
-0.00175	-0.00017	-3.80E-06
0	0	0
0	0	0
0	0	0
-0.0017	-9.20E-05	-8.53E-07
0	0	0
-0.012	0.000185	1.06E-05
0	0	0
0	0	0
-0.00096	-0.00011	-3.14E-06

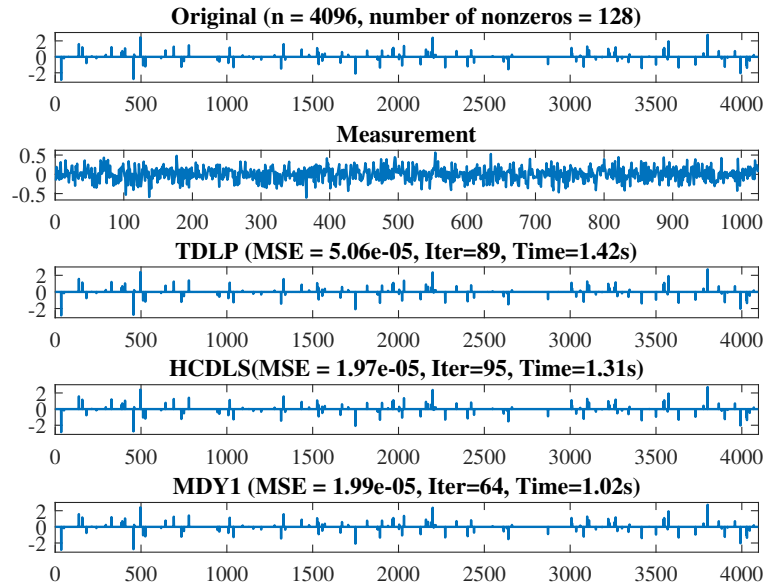


Figure 6.1: Reconstruction of sparse signal. From top to bottom is the original signal, the measurement and the reconstructed signals by the: TDLP, HCDLS and MDY1.

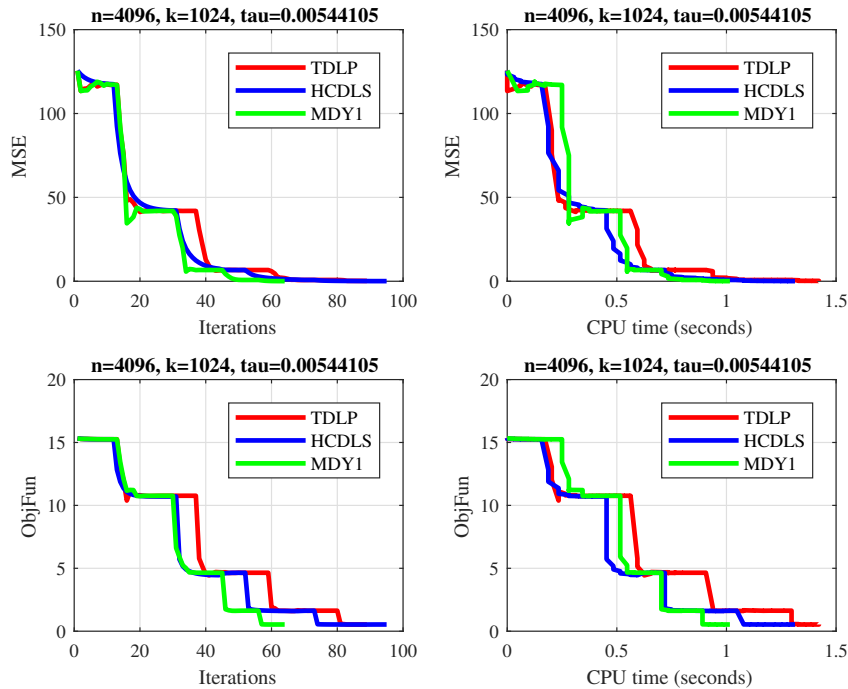


Figure 6.2: Comparison results of the TDLP, HCDLS and MDY1 Algorithms. The x -axis represents the number of iterations (**top and bottom left**) and the CPU time in seconds (**top and bottom right**), while the y -axis represents the MSE (**top left and right**) and the function values (**bottom left and right**).

Next, we illustrate the effectiveness and robustness of Algorithms 1 (MDY2), Algorithm 2 (HCDLS) and Algorithm 3 (TDLP) in image de-blurring problems. We carried out our experiment using some personal images. The performance evaluation criteria to measure the quality of restoration by the methods are

measured by the signal-to-noise ratio (SNR) defined by

$$SNR := 20 \times \log_{10} \left(\frac{\|x\|}{\|x^* - x\|} \right),$$

the peak signal-to-noise ratio (PSNR) [32] and the structural similarity index (SSIM) [110]. To ensure fairness when comparing algorithms, all iterations begin at $x_0 = A^T b$ and end at $Tol < 10^{-5}$. For the image de-blurring problem, the following parameters were used in our implementation: $\kappa = 0.001$, $\rho = 0.8$, $\sigma = 10^{-4}$, $q = 5$ and $\gamma = 1.99$. We tested several images including: $P1(256 \times 256)$, $P2(256 \times 256)$, $P3(256 \times 256)$ and $P4(256 \times 256)$ which were degraded by Gaussian blur and 10% noise.

In Table 6.3, we report the performance for SNR, PSNR and SSIM of TDLP, HCDLS and MDY2 methods in the recovery of blurred and noisy images. We can see that the SNR, PSNR and SSIM of the test images calculated by the HCDLS method are a bit higher than that by the TDLP and MDY2 methods, respectively. The higher the value of SNR, PSNR and SSIM the better the quality of image restoration is reflected.

Based on the performance reported in Table 6.3, all of the three methods were able to restore the blurred and noisy images with a better quality of restoration. However, the HCDLS method obtains better quality reconstructed images in an efficient manner. Figure 6.3 shows the original images (first column), the blurred images (second column), the restored images by: TDLP (third column), the HCDLS (fourth column) and the MDY2 (fifth column), respectively.

Table 6.3: Numerical results of TDLP, HCDLS and MDY2 methods in image restorations.

Image	TDLP			HCDLS			MDY2		
	SNR	PSNR	SSIM	SNR	PSNR	SSIM	SNR	PSNR	SSIM
P1	24.46	29.12	0.89	24.54	30.11	0.89	24.34	28.68	0.88
P2	24.64	29.14	0.89	24.80	30.18	0.89	23.82	27.73	0.88
P3	25.34	29.88	0.88	25.75	30.23	0.89	24.28	28.22	0.88
P4	24.98	29.74	0.83	25.67	30.21	0.84	25.14	28.82	0.84
Average	24.86	29.47	0.87	25.19	30.18	0.88	24.40	28.36	0.87

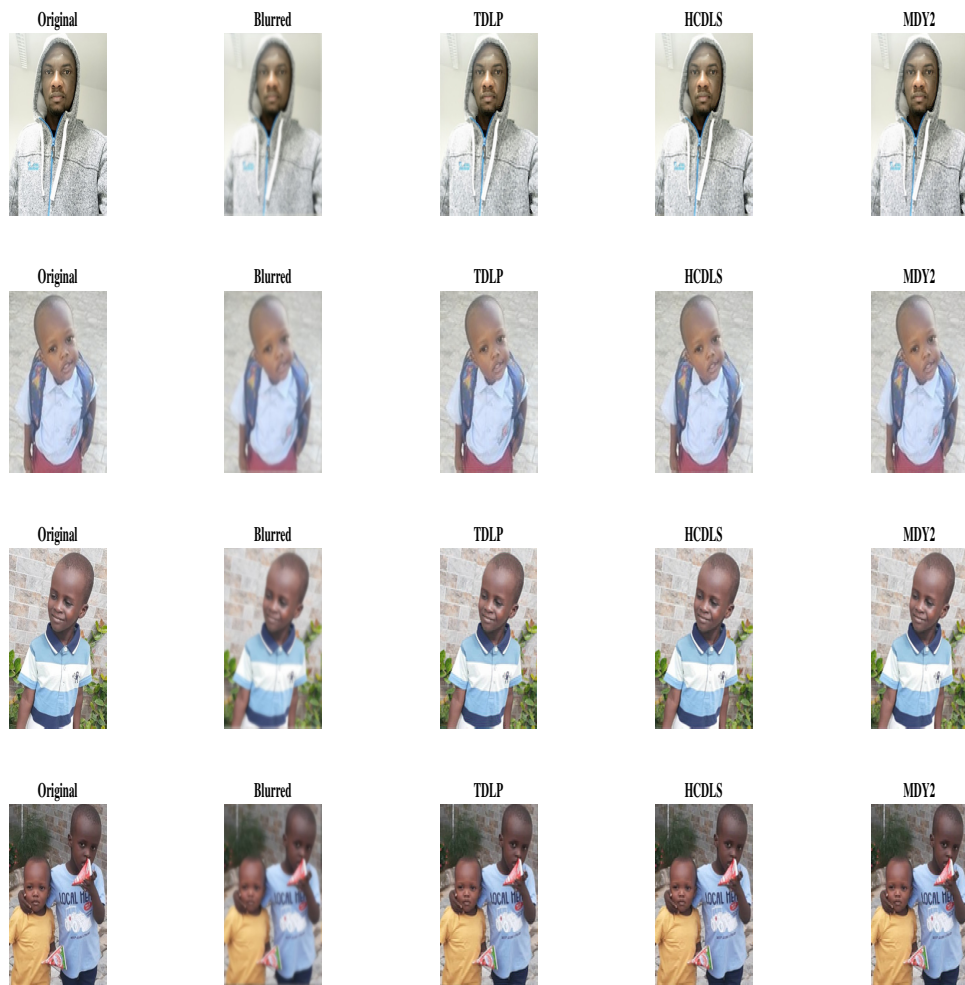


Figure 6.3: Original images (first column), blurred images (second column), restored images by: the TDLP Algorithm (third column), the HCDLS Algorithm (fourth column) and the MDY2 Algorithm (fifth column)

In the next section, we present application of Algorithm 4 (CGAIS) and Algorithm 5 (SAIS) in motion control models.

6.2 Applications in Motion Control Problems

The problems that arise in the idea of a robot system are currently receiving a lot of attention, and several approaches for addressing these issues have been suggested [119]. Zhang et al. [177] examined the fundamental aspects of n-link robotics, specifically focusing on the 1-link robot system, which has a broad range of applications. Motor dynamics, as described in reference [140], must be considered while planning robot movement in order to meet stability and accuracy criteria. One of the requirements for a motor's dynamic is that the actual output of the system should closely follow the anticipated output with a small error that is considered acceptable [160]. Several approaches have been proposed to address the tracking control issues of a nonlinear system. Here, we can discuss the use of the proportional-integral-

derivative (PID) control [91, 133], feedback linearization [37, 126], and optimal output tracking control using an approximation technique [161]. Furthermore, some motion control models can be expressed as a planar location problem in which distances are measured using norms relevant to the robot's setup and motion should be regulated. These location problems can be classified as special approximation problems. Locational analysis provides excellent solutions for this type of scenario (see Hamacher [76]). This experiment focuses on a motion control problem that involves a two-joint planar robotic manipulator. Algorithm 4 is adapted to address the problems of the form

$$\min\{f(x) : x \in \mathbb{R}\}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable and convex function. The modified algorithm is applied to solve the subsequent motion control model.

Algorithm 7: Modified CGAIS (MCGAIS) [for solving Problem (1.8)] [125]

Input: For all $k \geq 0$, give the same inputs as in Algorithm 4 but with $\alpha_k = 0$. Let $\sigma, \kappa > 0$, $\rho \in (0, 1)$ and $F(x_k) := \nabla f(x_k)$.

Replace *Step 2* and *Step 3* of Algorithm 4 with the subsequent instructions:

Step 3: Calculate the step size η_k by setting $\eta_k := \kappa \rho^i$ where i is the smallest non-negative integer that satisfies the condition

$$f(x_k + \kappa \rho^i d_k) - f(x_k) \leq \sigma \kappa \rho^i F(x_k)^T d_k. \quad (6.1)$$

Step 4: Compute the next iterate using the following

$$x_{k+1} := x_k + \eta_k d_k. \quad (6.2)$$

Remark 6.2.1. Consider the bounded level set $\{x : f(x) \leq f(x_0)\}$ and the problem $\min\{f(x) : x \in \mathbb{R}\}$, where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ has a solution. Since $F(x)$ is supposed to be Lipschitz continuous on \mathbb{R}^n , [152, Theorem 2] yields $\liminf_{k \rightarrow \infty} \|F(x_k)\| = 0$.

According to the description provided in reference [176], the equation for the discrete-time kinematics of a two-joint planar robot manipulator at the position level is expressed as

$$f(\theta_k) = q_k. \quad (6.3)$$

The joint angle vector, denoted by $\theta_k \in \mathbb{R}^2$, and the end effector position vector, denoted by $q_k \in \mathbb{R}^2$, are respectively represented by the vectors described above. The kinematics mapping represented by the function $f(\cdot)$ is characterized by the following known structure

$$f(\theta_k) = \begin{bmatrix} \ell_1 c_1 + \ell_2 c_2 \\ \ell_1 s_1 + \ell_2 s_2 \end{bmatrix}, \quad (6.4)$$

where ℓ_i ($i = 1, 2$) represents the length of the i^{th} rod, $c_1 = \cos(\theta_1)$, $c_2 = \cos(\theta_1 + \theta_2)$, $s_1 = \sin(\theta_1)$ and $s_2 = \sin(\theta_1 + \theta_2)$. When considering robotic control, it is necessary for us to find a solution to the

following optimization problem

$$\min_{q_k \in \mathbb{R}^2} f(q_k), \text{ where } f(q_k) = \frac{1}{2} \|q_k - q_{dk}\|^2, \quad (6.5)$$

q_{dk} represents the end effector control track. Consider the computing time intervals with t_k belonging to the interval $[0, t_f]$, where t_f represents the end of task duration.

In accordance with the approach described in [152, 169], we determine the length of the rod to be $\ell_i = 1$, ($i = 1, 2$). The end effector is then controlled to follow the two Lissajous curves that are expressed as

$$q_{dk}^{(1)} = \begin{bmatrix} \frac{3}{2} + \frac{1}{5} \sin(3t_k) \\ \frac{\sqrt{3}}{2} + \frac{1}{5} \sin(2t_k) \end{bmatrix}, \quad (6.6)$$

and

$$q_{dk}^{(2)} = \begin{bmatrix} \frac{3}{2} + \frac{1}{5} \sin(t_k) \\ \frac{\sqrt{3}}{2} + \frac{1}{5} \sin(2t_k) \end{bmatrix}. \quad (6.7)$$

In the implementation of Algorithm 7 (MCGAIS), we established the parameters $\rho = 0.6$, $\sigma = 0.08$ and the task duration $t_f = 10$ seconds. To determine the initial point, we select $\theta_0 = [0, \frac{\pi}{3}]^T$ and split the duration of the task $t = [0, 10]$, into 200 equal parts. For the Lissajous curves that are presented in $q_{dk}^{(1)}$ and $q_{dk}^{(2)}$, respectively, the numerical results that are generated by Algorithm 7 are depicted in Figures 6.4 and 6.5. It is demonstrated in Figures 6.4(A), 6.4(B), 6.5(A) and 6.5(B) that Algorithm 7 (MCGAIS) is capable of successfully completing the task at hand for $q_{dk}^{(1)}$ and $q_{dk}^{(2)}$, respectively (compare Figures 7 and 8 in [169]). Figures 6.4(C) and 6.4(D) illustrate the residual errors $\varepsilon(t_{k+1})$ along the x and y axes, respectively for $q_{dk}^{(1)}$. On the other hand, Figures 6.5(C) and 6.5(D) illustrate the residual errors along the x and y axes, respectively for $q_{dk}^{(2)}$. When we examine the residual errors on both x and y axes of Figures 6.4(C), 6.4(D), 6.5(C) and 6.5(D), we can conclude that Algorithm 7 successfully recorded an error of around 10^{-5} , which is considered to be acceptable. Therefore, based on the two figures, we are able to conclude that Algorithm 7 is capable of being used to successfully handle motion control models, which are problems that occur in the real world.

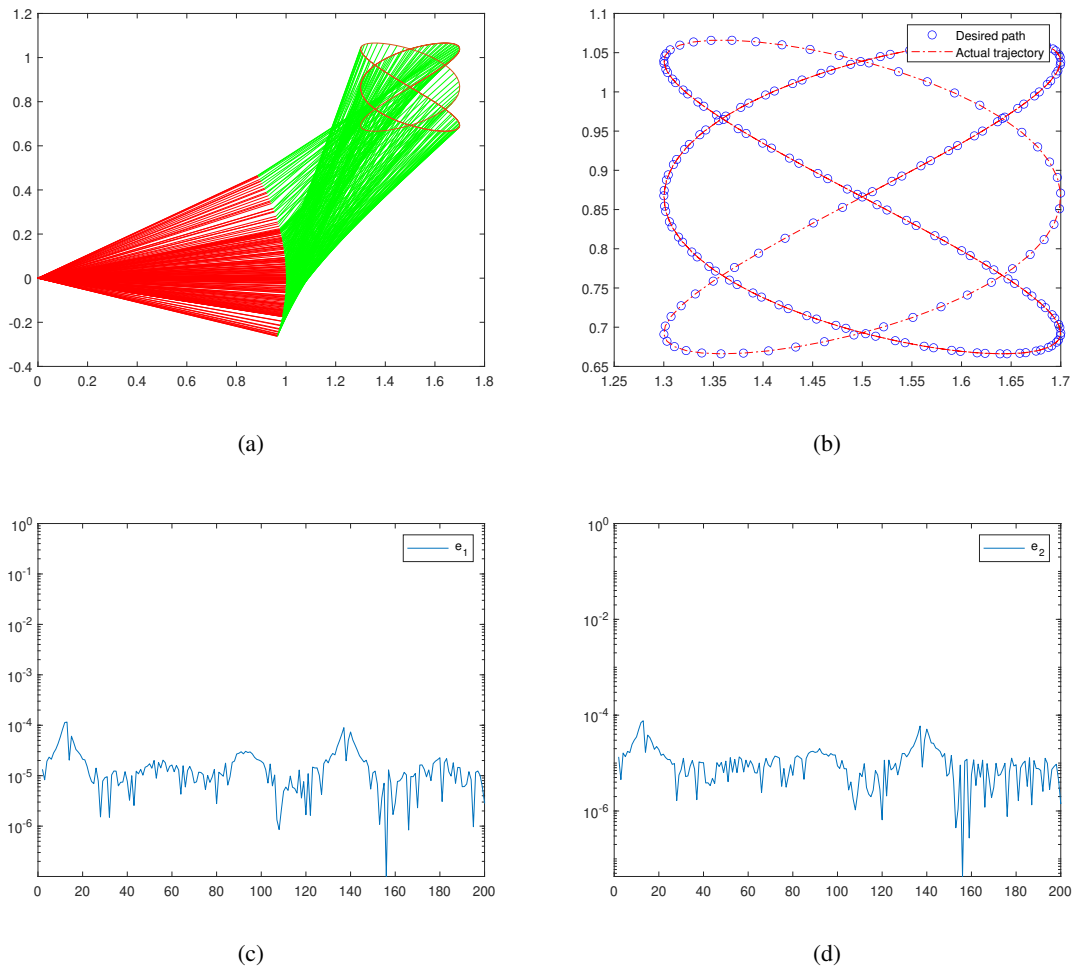


Figure 6.4: The robot's trajectories path and residual errors $\varepsilon(t_{k+1})$ along x and y axes of the motion control model for the Lissajous curve $q_{dk}^{(1)}$ of Algorithm 7 (MCGAIS) [reprinted from [125, Figure 6]].

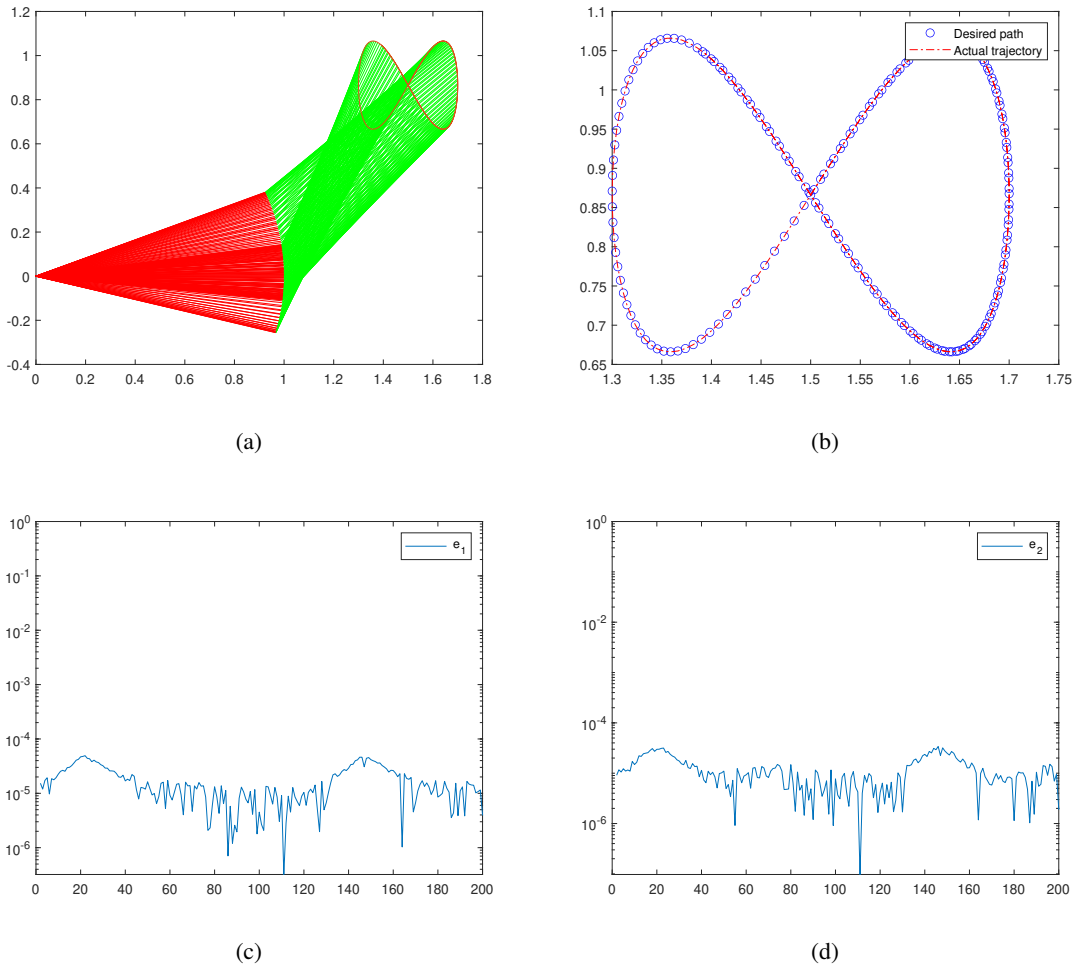


Figure 6.5: The robot's trajectories path and residual errors $\varepsilon(t_{k+1})$ along x and y axes of the motion control model for the Lissajous curve $q_{dk}^{(2)}$ of Algorithm 7 (MCGAIS) [reprinted from [125, Figure 7]].

Chapter 7

Application for Solving Vector-Valued Approximation Problems

Vector optimization also known as multiobjective optimization is a method for handling many real-world problems occurring in control theory, decision making, operations research, networking, economics and other branches of science and engineering.

The Irish economist Francis Ysidro Edgeworth (1881) (see [51]) and the Swiss economist Vilfredo Pareto (1896) (see [132]) were the first to develop a solution concept for multiobjective optimization problems. It is based on the concept of finding a good compromise rather than a single solution, as with scalar instances. In this work, we primarily apply Pareto optimum concepts in the fields of approximation and location theory.

Several books and works have introduced the foundations of multiobjective optimization, including [52, 67, 68, 93, 98, 158]. For some examples of its various applications, see [53, 55, 77, 84, 105].

In this Chapter, we introduce some solution concepts in multiobjective optimization (vector optimization) in Section 7.2. Then, a weighted sum scalarizations approach for multiobjective optimization (vector-valued approximation) problems is given in Section 7.3. Furthermore, we formulate the general multi-objective approximation problem in Section 7.4 and finally close the chapter by applying Algorithm 3 (TDLP) (see, for example, Chapter 3, Section 3.3) in an interactive procedure to solve multiobjective optimization (vector-valued approximation) problems in Section 7.5.

In the following section, we formulate the vector-valued approximation problem.

7.1 Formulation of the Problem

In order to formulate the problem, let \mathbb{R}^n be a linear space with $f_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, p\}$, $p \in \mathbb{N}$, $p \geq 2$. We consider the vector-valued objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$ where

$$f(x) := \begin{pmatrix} f_1(x) \\ \vdots \\ f_p(x) \end{pmatrix}. \quad (7.1)$$

A general multiobjective optimization problem is usually described as

$$\begin{cases} \text{Minimize} & f(x) \\ \text{subject to} & x \in E, \end{cases} \quad (MOP)$$

for a given nonempty feasible set $E \subseteq \mathbb{R}^n$.

Notice that, in the formulation of (MOP), the components objective functions are conflicting in nature. Therefore, one cannot find a single vector from the feasible set E that minimize them simultaneously. Hence, there is need for us to study the solution concepts concerning the (MOP), in order to understand the "minimization" in the image space \mathbb{R}^p .

7.2 Solution Concepts in Vector Optimization

In this section, we will introduce some well-known solution concepts in vector optimization, such as [52, 67, 93].

In vector optimization, we employ cones with specific features to generalize the order relation in \mathbb{R} . $K \subset \mathbb{R}^p$ is a proper, convex, closed and pointed cone representing a partial ordering relation in \mathbb{R}^p . This is referred to as an ordering cone.

The cone K generates the following order relationship:

$$y^1 \leq_K y^2 \iff y^2 - y^1 \in K, \text{ for } y^1, y^2 \in \mathbb{R}^p.$$

The definition of the cone K incorporates the reflexivity of the aforementioned relation, $0 \in K$. The convexity of the cone K implies transitivity of the relation defined above, whereas its pointedness results in antisymmetry of this relation. In several works of literature, the antisymmetry is sometimes ignored by not requiring the cone to be pointed; however, in our research, the ordering cone should be always pointed. Often, the ordering cone used in \mathbb{R}^p is the standard ordering structure, which is \mathbb{R}_+^p .

We now introduce the first concept of a solution, known as **Pareto Optimality**.

Definition 7.2.1. ([98, Efficiency]). *Let $F \subset \mathbb{R}^p$ be a set and $K \subset \mathbb{R}^p$ a proper, convex, pointed and closed cone. An element $y^0 \in F$ is said to be **efficient** with respect to the cone K if*

$$F \cap (y^0 - (K \setminus \{0\})) = \emptyset.$$

We represent by $\text{Eff}(F, K)$ the set of all efficient elements in F relative to the cone K .

Remark 7.2.2. *In the preceding definition, we did not use an index to denote the set of efficient elements in F relative to the cone K for the minimization problem $\text{Eff}(F, K)$. Later, we will use indices to distinguish between the minimization problem, represented by Eff_{Min} , and the maximization problem indicated by Eff_{Max} .*

The following definition introduces weakly efficient elements under the assumption that the ordering cone has a nonempty interior; this is important mathematically. Furthermore, numerical algorithms often generate weakly efficient elements.

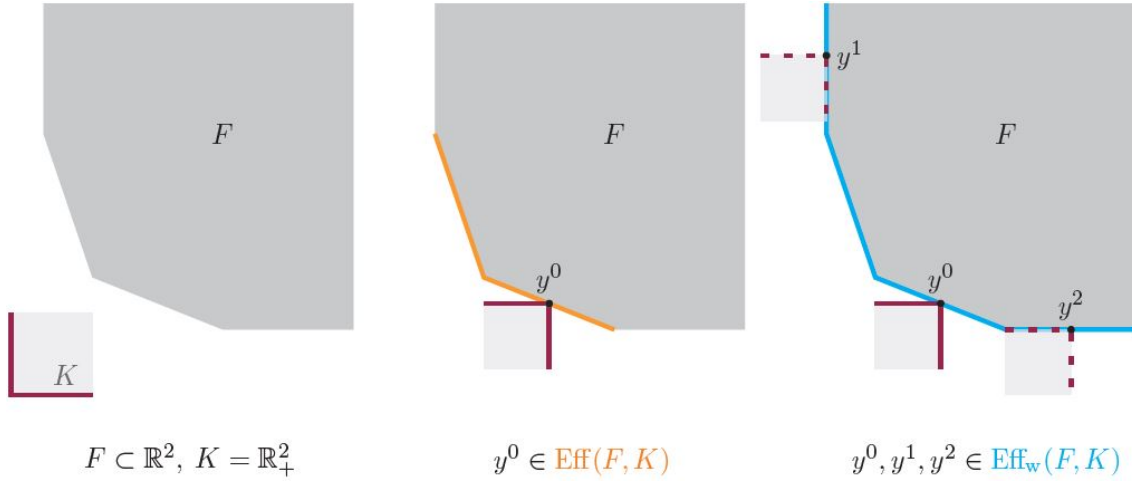


Figure 7.1: The set $\text{Eff}(F, K)$ and the set $\text{Eff}_w(F, K)$ of a set F w.r.t. $K = \mathbb{R}_+^2$ with $y^1, y^2 \notin \text{Eff}(F, K)$.

Definition 7.2.3. ([98, Weak Efficiency]). Let $F \subset \mathbb{R}^p$ be a set and $K \subset \mathbb{R}^p$ a proper, convex, pointed and closed cone. Let $\text{int}K \neq \emptyset$. An element $y^0 \in F$ is said to be **weakly efficient** with respect to the cone K if

$$F \cap (y^0 - \text{int}K) = \emptyset.$$

The set of all weakly efficient elements in F with respect to the cone K is denoted by $\text{Eff}_w(F, K)$.

Figure 7.1 is an illustration of Definitions 7.2.1 and 7.2.3.

Let $X = \mathbb{R}^n$. Consider the vector optimization Problem (**MOP**) with $E \subseteq X$ and $f : E \rightarrow \mathbb{R}^p$, where \mathbb{R}^p is partially ordered by proper, convex, pointed and closed cones. In Definitions 7.2.1 and 7.2.3, we use $F := f[E] := \{f(x) \mid x \in E\}$.

An element $x^0 \in E$ for which $f(x^0) \in \text{Eff}(f[E], K)$ is called a **minimal solution**. We denote the set of minimal solutions of E with respect to the objective function f and the cone K by $\text{Min}(f[E], K)$. This means that

$$\text{Min}(f[E], K) := \{x \in E \mid f(x) \in \text{Eff}(f[E], K)\}. \quad (7.2)$$

The set of **weakly minimal solutions** for E with respect to the objective function f and the cone K is denoted by $\text{Min}_w(f[E], K)$, that is,

$$\text{Min}_w(f[E], K) := \{x \in E \mid f(x) \in \text{Eff}_w(f[E], K)\}. \quad (7.3)$$

Using the prior solution concepts and notations, we formulate our optimization Problem (**MOP**) in the following way:

$$\text{Determine the set } \text{Eff}(f[E], K). \quad (P)$$

The formulations (**P**) and (**MOP**) are widely used in the literature.

Next, we present further efficiency results.

Definition 7.2.4. ([79, Proper Efficiency]). Let $F \subset \mathbb{R}^p$ be a set and $K \subset \mathbb{R}^p$ a proper, convex, pointed and closed cone. An element $y^0 \in F$ is said to be a **properly efficient** element of F with respect to the cone K , if there exists a proper convex cone $\tilde{K} \subset \mathbb{R}^p$ with $K \setminus \{0\} \subset \text{int} \tilde{K}$ such that y^0 is an efficient element of F with respect to \tilde{K} , that is,

$$F \cap (y^0 - (\tilde{K} \setminus \{0\})) = \emptyset.$$

The set of all properly efficient elements in F relative to the cone K is denoted by $\text{Eff}_p(F, K)$.

Assuming $\text{int} K \neq \emptyset$, we can simply demonstrate that every properly efficient element of a set F is an efficient element of F . Additionally, every efficient element of F is a weakly efficient element of F with respect to a certain cone K , that is,

$$\text{Eff}_p(F, K) \subseteq \text{Eff}(F, K) \subseteq \text{Eff}_w(F, K). \quad (7.4)$$

Note: Taking $F := f(\mathbb{R}^n)$ in Definitions 7.2.1, 7.2.3 and 7.2.4, which is the image of \mathbb{R}^n under f , the solution concepts holds for (MOP).

Proper efficiency can be described in a different way when convexity assumptions are taken into account. This can be achieved through scalarization using linear continuous functionals λ^* that correspond to the interior of the dual cone K^* to K (K^* was introduced in Section 2.2, specifically Problem (2.9)), as given in the following definition.

Definition 7.2.5. ([148, Schönfeld]). Let $F \subset \mathbb{R}^p$ be a convex set and $K \subset \mathbb{R}^p$ is a proper, convex, pointed and closed cone. We call $y^0 \in F$ a properly efficient element of F with respect to the cone K (in the sense of Schönfeld), if an element $\alpha^* \in \text{int} K^*$ exists such that

$$\forall y \in F : \alpha^*(y^0) \leq \alpha^*(y).$$

The set of all properly efficient elements of F with respect to the cone K in the sense of Schönfeld is denoted by $\text{Eff}_{\text{PSch}}(F, K)$.

7.3 Weighted Sum Scalarization Approach

The idea behind scalarization is to replace a multiobjective optimization problem with a suitable scalar problem, that is, an optimization problem with a real-valued objective function (compare Definition 7.2.5). The solutions to the multiobjective optimization problem can, under some assumptions, be characterized through the solutions to the scalar problem. This is very interesting, as the study of a scalar optimization problem is greatly developed. There are many possibilities for formulating the surrogate scalar problem (see e.g. [52, 56, 67]). One of the well-known methods is the weighted sum scalarization, where a multiobjective optimization problem like (MOP) with the objective function $f = (f_1, \dots, f_p)^T$ (see (7.1)) can be transformed into a scalar problem of the form:

$$\begin{cases} \text{Minimize} & \sum_{i=1}^p w_i f_i(x) \\ \text{subject to} & x \in E, \end{cases} \quad (SP)$$

by choosing some suitable weights $w_i \geq 0$, $i \in \{1, \dots, p\}$, $p \in \mathbb{N}$ and $p \geq 2$ which are to be determined by the decision maker. Weights represent the importance of a particular objective function. In this section, we give some properties and conditions for characterizing (weakly, properly) efficient elements (cf. Definition 7.2.5) through scalarization (e.g. by using a scalarizing functional belonging to the dual cone). To this end we introduce monotonicity properties of a scalarizing functional $z: \mathbb{R}^p \rightarrow \mathbb{R}$:

Definition 7.3.1. Let $K \subset \mathbb{R}^p$ be a proper, convex, pointed and closed cone with $y^1, y^2 \in \mathbb{R}^p$. The functional $z: \mathbb{R}^p \rightarrow \mathbb{R}$ is said to be:

- **K -monotone increasing** if $y^2 \in y^1 + K \implies z(y^1) \leq z(y^2)$.
- **Strictly K -monotone increasing** if $y^2 \in y^1 + (K \setminus \{0\}) \implies z(y^1) < z(y^2)$.

The following example demonstrates the monotonicity properties of elements belonging to the dual cone of K .

Example 7.3.2. Consider the standard ordering cone in \mathbb{R}^p :

$$\mathbb{R}_+^p := \{y \in \mathbb{R}^p \mid \forall i = 1, \dots, p, y_i \geq 0\}.$$

The dual cone of \mathbb{R}_+^p is given by

$$(\mathbb{R}_+^p)^* = \{z^* \in (\mathbb{R}^p)^* = \mathbb{R}^p \mid \forall y \in \mathbb{R}_+^p : (z^*)^T y \geq 0\} = \mathbb{R}_+^p.$$

The element $z^* \in \mathbb{R}_+^p \setminus \{0\}$ is a \mathbb{R}_+^p -monotone increasing functional. Taking $y^2 \in y^1 + \mathbb{R}_+^p$ implies that $z^*(y^2 - y^1) \geq 0$, based on the definition of the dual cone. It is clear that $z^*(y^1) \leq z^*(y^2)$. On the other hand, $z^* \in \text{int } \mathbb{R}_+^p$ is strictly \mathbb{R}_+^p -monotone increasing, since $y^2 \in y^1 + (\mathbb{R}_+^p \setminus \{0\})$ yields $z^*(y^2 - y^1) > 0$, implying $z^*(y^1) < z^*(y^2)$.

Theorem 7.3.3. ([67]). Let $F \subset \mathbb{R}^p$ be a nonempty set, $K \subset \mathbb{R}^p$ is a proper, convex, pointed and closed cone and $z: \mathbb{R}^p \rightarrow \mathbb{R}$. Suppose that for an element $y^0 \in F$, it holds that for all $y \in F$, $z(y^0) \leq z(y)$. If z is strictly K -monotone, then $y^0 \in \text{Eff}(F, K)$.

For the other direction, in order to apply the classical separation arguments, we suppose convexity assumptions which are represented in the definition and theorem that follows.

Definition 7.3.4. Let $K \subset \mathbb{R}^p$ be a proper, convex, pointed and closed cone. Let $E \subseteq X$ be a convex set. The function $f: X \rightarrow \mathbb{R}^p$ is considered **K -convex**, if

$$f(\alpha x^1 + (1 - \alpha)x^2) \in \alpha f(x^1) + (1 - \alpha)f(x^2) - K \quad (7.5)$$

holds for all $x^1, x^2 \in X$ and all $\alpha \in [0, 1]$.

Remark 7.3.5. For $p = 1$ we get

$$\forall x^1, x^2 \in X, \forall \alpha \in [0, 1] : f(\alpha x^1 + (1 - \alpha)x^2) \leq \alpha f(x^1) + (1 - \alpha)f(x^2),$$

which is the classic definition of the convexity of a function $f: X \rightarrow \mathbb{R}$.

Now we investigate the convexity of the set $f[X]$ in the image space under the convexity assumptions for X and f .

Theorem 7.3.6. ([67, Theorem 2.11]). *Let $K \subseteq \mathbb{R}^p$ be a proper, convex, pointed and closed cone. Let $X \subseteq \mathbb{R}^n$ be a convex set and $f : X \rightarrow \mathbb{R}^p$ be K -convex. Then the set $f[X] + K$ is convex.*

Under convexity assumptions, we present a well known scalarization result: for any minimal solution x^0 (with $f(x^0) \in \text{Eff}(f[X], K)$), there exists a functional $\alpha^* \in K^* \setminus \{0\}$ such that x^0 is a minimal solution of the scalarized problem. See also [93, Theorem 5.4].

Theorem 7.3.7. (Necessary Condition for an Efficient Element). *Let $K \subseteq \mathbb{R}^p$ be a proper, convex, pointed and closed cone with $\text{int} K \neq \emptyset$. Let $E \subseteq X \subseteq \mathbb{R}^n$ be a convex set. Let the function $f : X \rightarrow \mathbb{R}^p$ be K -convex. Then*

$$f(x^0) \in \text{Eff}(f[X], K) \implies (\exists \alpha^* \in K^* \setminus \{0\}, \forall x \in X : \alpha^* f(x^0) \leq \alpha^* f(x)).$$

For readers convenience, we repeat the proof as follows.

Proof. If $f(x^0) \in \text{Eff}(f[X], K)$, then $f[X] \cap (f(x^0) - (K \setminus \{0\})) = \emptyset$. This means that

$$(f[X] + K) \cap (f(x^0) - (K \setminus \{0\})) = \emptyset.$$

Suppose on the contrary the last statement does not hold, that is,

$$(f[X] + K) \cap (f(x^0) - (K \setminus \{0\})) \neq \emptyset,$$

this implies the existence of $x^1 \in X$ and $k^1 \in K$ with $f(x^1) + k^1 \in f(x^0) - (K \setminus \{0\}) \implies$

$$f(x^1) \in f(x^0) - (K \setminus \{0\}) - k^1.$$

Since K is a convex and pointed cone, it follows that $f(x^1) \in f(x^0) - (K \setminus \{0\})$. This contradicts the statement that $f(x^0) \in \text{Eff}(f[X], K)$. To apply the Separation Theorem 2.2.11, consider the two sets

$$B := (f(x^0) - K) \text{ and } C := f[X] + K,$$

B is nonempty and convex, with $\text{int} B \neq \emptyset$ while C is also nonempty and convex, with $\text{int} B \cap C = \emptyset$ (refer to Theorem 7.3.6). Theorem 2.2.11 states that for all $x \in X$ and all $k^1, k^2 \in K$,

$$\alpha^*(f(x^0) - k^1) \leq \lambda \leq \alpha^*(f(x) + k^2). \quad (7.6)$$

Since K is a cone, we get $\alpha^* \in K^* \setminus \{0\}$. Alternatively, if $\alpha^* \notin K^*$, then $\alpha^*(k) < 0$ for some $k \in K$. For some $n \in \mathbb{N}$, we set $\alpha^*(n \cdot k)$ small enough so that $\alpha^*(n \cdot k) < \lambda - \alpha^*(f(x))$ for some fixed $x \in X$, contradicting (7.6). As a result,

$$\forall x \in X : \alpha^*(f(x^0)) \leq \alpha^*(f(x)).$$

□

However, the converse requirement in Theorem 7.3.7 does not apply since α^* does not have strict monotonicity. In the following theorem, we provide the necessary and sufficient condition for weakly efficient element. The following result can be traced back to Focke [62, 63] as well as more recently to Jahn [93] and Ehrgott [52], respectively.

Theorem 7.3.8. *Consider the Problem (MOP). Let $K \subset \mathbb{R}^p$ be a proper, convex, pointed and closed cone. Let $E \subset \mathbb{R}^n$ be a convex set and the function $f : E \rightarrow \mathbb{R}^p$ be K -convex. Then $f(x^0)$ is a weakly efficient element of the image set $f[E]$ of Problem (MOP) if and only if $\exists \alpha^* \in K^* \setminus \{0\}$ such that*

$$\forall x \in E : \alpha^* f(x) \geq \alpha^* f(x^0).$$

The corresponding characterization for properly efficient elements in the sense of Schönfeld (see Definition 7.2.5) is given in Remark 7.3.9.

Remark 7.3.9. *Consider the Problem (MOP). Let $K \subset \mathbb{R}^p$ be a proper, convex, pointed and closed cone. Let $E \subset \mathbb{R}^n$ be a convex set and the function $f : E \rightarrow \mathbb{R}^p$ be K -convex. In Definition 7.2.5, we get for $F = f[E]$ and $y^0 = f(x^0)$ the following characterization: $f(x^0)$ is a properly efficient element of the image set $f[E]$ of Problem (MOP) if and only if $\exists \alpha^* \in \text{int} K^*$ such that*

$$\forall x \in E : \alpha^* f(x) \geq \alpha^* f(x^0).$$

7.4 Formulation of the Multiobjective Approximation Problem

Many real-world problems can be described as an approximation problem. Besides problems with one objective function, several authors have investigated vector-valued or multiobjective approximation problems (see, for example, the book by Göpfert, Riahi, Tammer and Zalinescu [68] and the references therein). Here, we will consider a general vector control approximation problem and derive necessary conditions for approximate solutions of this problem.

In this section, we assume the following

$$(B1) \quad X = \mathbb{R}^n, \quad Y = \mathbb{R}^p \quad \text{and} \quad Z = \mathbb{R}^m$$

$$(B2) \quad K \subset \mathbb{R}^p \text{ is a pointed closed convex cone.}$$

If $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$ is **convex**, then the Fenchel's subdifferential of f at $x^0 \in \text{dom } f$ (in the sense of convex analysis) according to Definition 2.2.18 is the set

$$\partial f(x^0) = \{x^* \in X^* \mid \forall x \in X : x^*(x - x^0) \leq f(x) - f(x^0)\},$$

and for $x^0 \in X$ with $f(x^0) = +\infty$, we put $\partial f(x^0) = \emptyset$.

It is well known that the minimality condition for convex function $f : X \rightarrow \mathbb{R} \cup \{+\infty\}$ (according to Theorem 2.2.28) is:

$$\begin{cases} x^0 \in X \text{ is a minimal solution of} \\ \text{the optimization problem } \min_{x \in X} f(x) \end{cases} \iff 0 \in \partial f(x^0).$$

Moreover, we assume that K is a cone with $\text{int}K \neq \emptyset$.

Next, we consider the general vector-valued approximation problem (P_{app}):

$$\text{Determine the set } \text{Eff}(f[E], \mathbb{R}_+^p), \quad (P_{app})$$

the objective function $f(x)$ is defined as follows

$$f(x) := f_1(x) + \begin{pmatrix} \|A_1(x) - a^1\|_1^{\beta_1} \\ \vdots \\ \|A_p(x) - a^p\|_p^{\beta_p} \end{pmatrix},$$

where for each norm $\|\cdot\|_i$, we consider the exponents $\beta_i \geq 1$, ($i = 1, \dots, p$) for the different norms, $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}^p$ is a convex function, $E \subseteq \mathbb{R}^n$ with $Y = \mathbb{R}^p$, $a^i \in \mathbb{R}^m$, $A_i \in L(\mathbb{R}^n, \mathbb{R}^m)$, ($i = 1, \dots, p$), (cf. Jahn [92], Gerth and Pöhler [65], Henkel and Tammer [80, 81], Jahn and Krabs [95], Tammer [156, 157], Wanka [165], Oettli [130]).

Furthermore, we consider the scalarized approximation problem (see Section 7.3) to (P_{app})

$$f_\lambda(x) := \lambda^T(f_1(x)) + \sum_{i=1}^p \lambda_i \|A_i(x) - a^i\|_i^{\beta_i} \rightarrow \inf_{x \in E}, \quad (P_\lambda)$$

where $f : E \subset \mathbb{R}^n \rightarrow \mathbb{R}^p$, $f_1 \in L(\mathbb{R}^n, \mathbb{R}^p)$, $a^i \in \mathbb{R}^m$, $\lambda_i \geq 0$, $\beta_i \geq 1$ and $A_i \in L(\mathbb{R}^n, \mathbb{R}^m)$, ($i = 1, \dots, p$).

If in problem (P_λ), $f_1 \equiv 0$ and for all $i = 1, \dots, p$, $A_i = I$, $E = \mathbb{R}^n$, we get the special case of the **real-valued location problem**

$$\tilde{f}_\lambda(x) := \sum_{i=1}^p \lambda_i \|x - a^i\|_i^{\beta_i} \rightarrow \inf_{x \in \mathbb{R}^n}. \quad (\tilde{P}_\lambda)$$

In the following, we describe important special cases of Problem (P_{app}):

1. Vector-valued optimal control problems given by (cf. Section 2.3, precisely Remark 2.3.1 with $p = 2$, $A_1 = I$, $\beta_1 = 1$, $\beta_2 = 2$, $Y = \mathbb{R}^2$ and $K = \mathbb{R}_+^2$): For

$$\hat{f}(x) := \begin{pmatrix} \|x\|_1 \\ \|Ax - b\|_2^2 \end{pmatrix}, \quad x \in \mathbb{R}^n,$$

determine

$$\text{Eff}(\hat{f}[\mathbb{R}^n], \mathbb{R}_+^2),$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ are the ℓ_1 - and ℓ_2 -norms, $A \in L(\mathbb{R}^n, \mathbb{R}^m)$, $b \in \mathbb{R}^m$, $X \subseteq \mathbb{R}^n$ is a nonempty closed convex set, and \mathbb{R}_+^2 denotes the usual ordering cone in \mathbb{R}^2 . Here x denotes the so-called control variable; the image $z = Ax$ denotes the state variable. We will apply Algorithm 3 (TDLP) in an interactive procedure for generating representatives of the solution set of this multiobjective approximation problem in Section 7.5.

2. Scalar location and approximation problems ($Y = \mathbb{R}$, $f_1 \equiv 0$, $\beta_i = 1$, $i = 1, 2, \dots, p$):

$$\sum_{i=1}^p \lambda_i \|A_i(x) - a^i\| \rightarrow \inf_{x \in \mathbb{R}^n}, \quad (P_\lambda^0)$$

where $\|\cdot\|$ is a norm in Z (see for example, [68, Section 3.7 and 4.2]).

3. Linear vector optimization problems (for all $i = 1, \dots, p$, $\lambda_i = 0$).
4. Surrogate problems for linear vector optimization problems with an objective function $f(x) := f_1(x)$ subject to $x \in \mathbb{R}^n$ and $A(x) = a$, for which the feasible set is empty.
5. Perturbed linear vector optimization problems.
6. Tychonoff regularization for linear vector optimization problems.

7.5 Application of Algorithm 3 (TDLP) for Solving Vector-Valued Approximation Problems

In this section, we consider the vector approximation Problem (P_{app}). A special case of this problem was discussed in Chapter 1; see, for example, **Problem 1**. For instance, in Section 2.3, we mentioned in Remark 2.3.1 that a special approximation problem of this type exists. We should discuss this type of problem here but without the regularization parameter τ . The least squares problem (in the literature, also named Lasso problem) was studied in Chapter 6, in particular, Section 6.1 as Problem (P^τ), i.e.,

$$\min_{x \in \mathbb{R}^n} \tau \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2,$$

where τ is a positive regularization parameter and $\|\cdot\|_2$ denotes the Euclidean norm of \mathbb{R}^m . Iterative methods for solving (P^τ) have been presented in several papers, (see [25, 29, 58, 75]). Due to the fact that the proposed algorithm is derivative-free, it can be applied to handle nonsmooth problems that are in the form of (1.10) effectively.

The main difficulty associated with the Lasso Problem (P^τ) is that the solution solely depends on the choice of the regularization parameter τ and most of the research reports only demonstrate results based on a limited number of a priori selected values; see [86, 117] and the references therein. To get rid of this difficulty, we consider instead of the least squares Problem (P^τ) a multiobjective optimization problem without a regularization parameter τ is essential. The corresponding formulation takes the following form of a vector optimization problem with two objective functions, as mentioned in Remark 2.3.1:

$$\min_{x \in \mathbb{R}^n} \begin{pmatrix} \|x\|_1 \\ \|Ax - b\|_2^2 \end{pmatrix}, \quad (P^m)$$

where $A \in L(\mathbb{R}^n, \mathbb{R}^m)$, $b \in \mathbb{R}^m$, $f_1(x) := \|x\|_1$, $f_2(x) := \|Ax - b\|_2^2$, $f = (f_1, f_2)^T : \mathbb{R}^n \rightarrow \mathbb{R}^2$, the function $f : \mathbb{R}^n \rightarrow \mathbb{R}^2$ is vector-valued, $\|\cdot\|_1$ denotes the ℓ_1 -norm in \mathbb{R}^n and $\|\cdot\|_2$ denotes the Euclidean-norm in \mathbb{R}^m , respectively. Clearly, Problem (P^m) is a special case of Problem (P_{app}). In addition, (P^τ) could be interpreted as a special scalarization of (P^m) with weights $\lambda_1 = \tau$ and $\lambda_2 = \frac{1}{2}$ (see Problem (P_λ) or its special case (P_λ^0) in Section 7.4). Hence, (P^m) can be scalarized using weights $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, such that we consider the scalar problem

$$\min_{x \in \mathbb{R}^n} \lambda_1 \|x\|_1 + \lambda_2 \|Ax - b\|_2^2. \quad (P_\lambda^m)$$

Furthermore, we can use the interactive procedure proposed in [68] for generating representatives of the solution set of (P^m) by using a scalarization (see (P_λ^m)) and Algorithm 3 (TDLP) derived in Section

3.3. The procedure for the *Interactive Algorithm* is repeated for reader convenience, as given in [68] as follows: In what follows, consider $\Lambda \subseteq \mathbb{R}^2$ as a given set of parameters: $\Lambda = (\lambda_1, \lambda_2)^T$ with $\lambda_i > 0$ for $i = 1, 2$.

Algorithm 8: Interactive Algorithm for Generating Representatives of the Solution set of (P^m)

Step 1: Choose $\bar{\lambda} \in \Lambda$. Compute an approximate solution x^0 to the Problem $(P_{\bar{\lambda}}^m)$ with Algorithm 3 (TDLP). If x^0 is accepted by the decision-maker, then **stop**. Otherwise, go to **Step 2**.

Step 2: Put $k = 0$, $t_0 = 0$. Choose $\hat{\lambda} \in \Lambda$, $\hat{\lambda} \neq \bar{\lambda}$. Go to **Step 3**.

Step 3: Choose t_{k+1} with $t_k < t_{k+1} \leq 1$ and compute the approximate solution x^{k+1} of

$$\min_{x \in \mathbb{R}^n} \{(\bar{\lambda}_1 + t_{k+1}(\hat{\lambda}_1 - \bar{\lambda}_1))\|x\|_1 + (\bar{\lambda}_2 + t_{k+1}(\hat{\lambda}_2 - \bar{\lambda}_2))\|Ax - b\|_2^2\}, \quad (P(t_{k+1}, \bar{\lambda}, \hat{\lambda}))$$

with Algorithm 3 (TDLP) and use x^k as a starting point. If an approximate solution of $P(t, \bar{\lambda}, \hat{\lambda})$ cannot be found for $t > t_k$, then go to **Step 1**. Otherwise, go to **Step 4**.

Step 4: The point x^{k+1} is to be evaluated by the decision-maker. If it is accepted by the decision-maker, then **Stop**. Otherwise, go to **Step 5**.

Step 5: If $t_{k+1} \geq 1$, then go to **Step 1**. Otherwise, set $k = k + 1$ and go to **Step 3**.

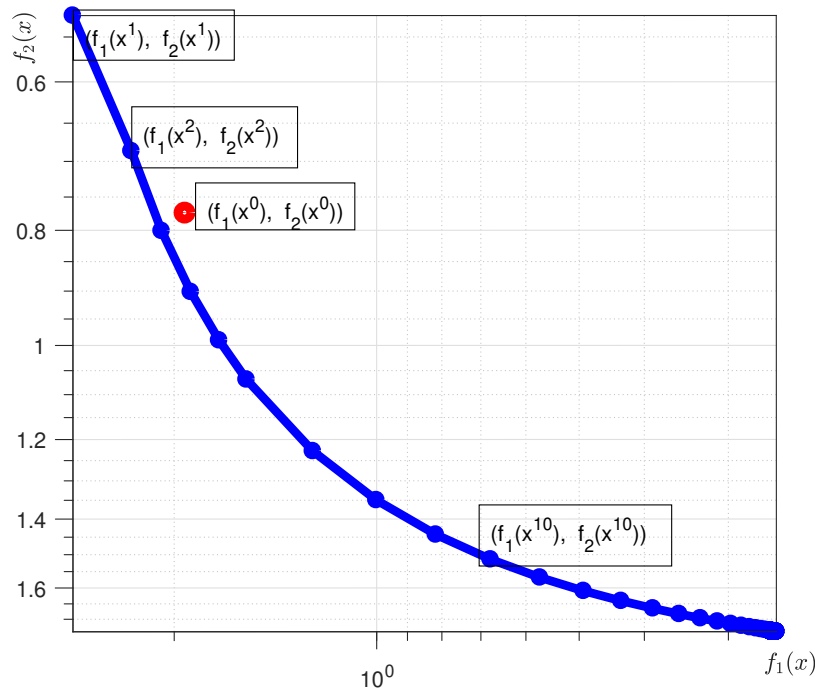


Figure 7.2: Representatives of the set of approximate solutions of Problem (P^m) generated for $\bar{\lambda} = [1, 1]$, $\hat{\lambda} = [4, 4]$ using Algorithm 8 (where Algorithm 3 (TDLP) is involved in *Step 3*) and the involved Problems $(P(t_{k+1}, \bar{\lambda}, \hat{\lambda}))$. The element $(f_1(x^0), f_2(x^0))$ (in red) is the approximate solution of (P^c) with $\tau = 0.008\|A^T b\|_\infty$ generated in Section 6.1 using Algorithm 3.

Remark 7.5.1. The point x^0 generated using Algorithm 3 is an approximate solution of (P^τ) for $\tau = 0.008\|A^T b\|_\infty$ with the stopping criterion being $\text{Tol} \leq 10^{-6}$. This means that we have a certain approximation error and the element $(f_1(x^0), f_2(x^0)) = (1.926, 0.778)$ is an approximate solution of the vector optimization Problem (P^m) . Consequently, from Figure 7.2, one can see that the approximate solution $(f_1(x^0), f_2(x^0))$ is closed to the efficient frontiers. Taking into account Remark 7.3.9, the objective function value of an exact solution to the scalarized Problem (P^τ) belongs to the efficient frontiers of Problem (P^m) . However, the objective function value $(f_1(x^0), f_2(x^0))$ does not belong to the efficient frontiers (only to an approximation of the efficient frontiers). Furthermore, in Figure 7.2, we can observe that there are much more minimal solutions to the Problem (P^m) that could be of interest to the decision maker.

Remark 7.5.2. The generated objective function values $(f_1(x^1), f_2(x^1)) = (2.828, 0.5277)$, $(f_1(x^2), f_2(x^2)) = (2.315, 0.686)$, ... are shown in Figure 7.2. The points x^1, x^2, \dots are approximate solutions of the scalarized Problem $(P(t_{k+1}, \bar{\lambda}, \hat{\lambda}))$ generated in Step 3 of Algorithm 8 for the parameters $\bar{\lambda}, \hat{\lambda}$ chosen as $\bar{\lambda} = [1, 1]^T$, $\hat{\lambda} = [4, 4]^T$, and the sequence t_k is $t_k := \frac{1}{1+e^{k^2}}$, $k \geq 0$. Algorithm 8 is used to obtain these points as representatives of the approximate solution set of (P^m) . Step 3 of the procedure for solving the Problem $(P(t_{k+1}, \bar{\lambda}, \hat{\lambda}))$ involves the use of Algorithm 3 (TDLP) (see Section 3.3).

Recall that Problem (P_λ^m) is the scalarization of (P^m) . By choosing the parameters in (P_λ^m) , we can obtain solutions that correspond with the decision maker's preferences. Therefore, Algorithm 8 generates representative of an approximation of the efficient frontiers of the Problem (P^m) . The points $(f_1(x^i), f_2(x^i))$, $i = 1, 2, \dots, n$ on the efficient frontiers (see Figure 7.2) are approximate solutions corresponding to the decision maker's preference, depending on the parameters chosen in (P_λ^m) . If the weights $\bar{\lambda} = [1, 1]^T$ and $\hat{\lambda} = [4, 4]^T$, the decreasing sequence $t_k := \frac{1}{1+e^{k^2}}$, $k \geq 0$, and all other parameters used for solving $(P(t_{k+1}, \bar{\lambda}, \hat{\lambda}))$ are positive real numbers, one can conclude that the solutions generated using Algorithm 8 belong to the set of properly efficient points of Problem (P^m) , taking into account Remark 7.3.9.

It is also possible to use Algorithm 8 to generate approximate solutions to other important multiobjective approximation problems, particularly in radiotherapy treatment (see, for example, [34, 40, 54, 85, 111] and the references therein) and special approximation problems (see [68, 71, 93, 98]) where the goal is to generate Pareto optimal solution(s) based on the decision maker's preferences.

Chapter 8

Conclusion and Outlook

8.1 Conclusion

In this dissertation, we have developed new algorithms for solving nonlinear systems of equations. These algorithms are all matrix-free, thereby making them suitable for problems with a large sets of variables. One of the notable advantages of our proposed algorithms is that the functions under consideration need not be differentiable. This allowed the proposed new algorithms to comfortably deal with nonsmooth problems. Interestingly, we have demonstrated the applicability of the proposed new algorithms in handling problems arising from compressive sensing and two–arms robotic motion control. In the following, we highlight some of our new results:

- In Chapter 3, we have introduced three new iterative projection algorithms. One of them is the Modified Dai and Yuan (MDY) algorithm in Section 3.1. Unlike the classical Dai–Yuan’s CG parameter where the denominator may go to zero as the iteration progresses, we succeeded in re-defining the search direction in such a way that it remains well-defined throughout the iteration process. This further made the MDY algorithm well-defined. The second algorithm defined its CG parameter as a convex combination of two CG parameters. It is noteworthy that, in contrast to certain previous research in the literature, it was demonstrated that the convex combination parameter θ_k , was shown to lie within the interval $[0, 1]$ without imposing additional conditions, unlike what is obtained in some existing works in the literature. The last algorithm proposed in this chapter incorporated the popular Dai-Liao CG parameter into the modified BFGS updating formula. Remark 3.3.1 demonstrates the well-defined nature of the new search direction. Furthermore, we constructed the spectral parameter associated with this algorithm to ensure the search direction satisfied the important property known as the sufficient descent condition. Interestingly, the proposed new algorithms mentioned above assumed a pseudomonotone function for global convergence. This is a weaker assumption compared to the monotonicity condition usually imposed in most existing methods found in the literature, (see, for example, [45, 46, 112, 175]).
- Chapter 4 leverages the fact that incorporating inertial steps into algorithms for solving variational inequalities, split feasibility problems, and so on, speeds up their numerical performances established in the literature. In this light, two multi-step inertial projection algorithms that use two

preceding iterates, say x_{k-1} and x_k , as well as their respective images to update the sequence of search directions have been proposed in Section 4.1. As we have demonstrated in the numerical experiments, this approach has tremendously increased the numerical performance of the proposed new inertial-like conjugate and spectral gradient algorithms, CGAIS and SAIS, respectively. Additionally, we have introduced a new version of multi-step algorithm that mimic the Ishikawa iterative scheme approach. This algorithm updates its sequence of iterations using two different search directions, hence the name two-step hybrid algorithm. This approach confirmed the claim made by some articles that the two step Ishikawa scheme converges faster than the one step Mann iterative scheme, (see Chapter 5). Respective comprehensive convergence analyses of these new inertial-like schemes have been discussed under mild conditions.

- Chapter 5 of this thesis focused on the numerical performances of the newly proposed algorithms. We coded the proposed new algorithms using MATLAB, along with some existing state-of-the-art algorithms selected from the literature for comparison. We have implemented each algorithm to solve a collection of benchmark test problems. The performances recorded have been illustrated with the aid of Dolan and Moré [49] Performance Profiles as well as Data Profiles of Moré and Wild [122]. It is evident from the figures generated by these profiles that the newly proposed algorithms performed better in almost all the metrics considered for the comparisons.
- We have demonstrated the applicability of Algorithm 1 (MDY), Algorithm 2 (HCDLS), and Algorithm 3 (TDLP) in the reconstruction of sparse signals and the restoration of blurred images in compressive sensing. Furthermore, we have applied the inertial-like conjugate and spectral gradient projection algorithms to motion control problems.
- In the last chapter of this thesis, we introduce multiobjective optimization problems and their solution concepts, that is, the concept of Pareto (weak, proper) efficiency. Furthermore, we studied scalarization techniques, especially the weighted sum scalarization approach. In addition, we formulated the general multiobjective approximation problem and described some of its important special cases. Lastly, we closed the chapter by applying Algorithm 3 (from Section 3.3) in an interactive procedure to solve multiobjective approximation problems for generating representatives of the approximate solution sets for properly efficient elements. These points were obtained with the help of an interactive algorithm for generating representatives of the (approximate) solution set of (P^m) , where Algorithm 3 (TDLP) was involved in *Step 3* of Algorithm 8 for solving Problem $(P(t_{k+1}, \bar{\lambda}, \hat{\lambda}))$ corresponding to the decision maker's preferences for appropriate weight selections.

8.2 Future Outlook

This research generates several ideas for further research.

- One can consider exploring the use of the proposed new algorithms or their modifications for addressing a variety of problems including:
 - Time-varying nonlinear equations.
 - Problems arising from financial forecasting [47, 48].
- To derive optimality conditions for solutions to the Problem (P_{app}) and to develop the corresponding algorithms based on these optimality conditions.
- In general, since most applications lead to a Problem (P_{app}) with more than two objective functions, it is our goal in the future to consider a more general Problem (P_{app}), where $p > 2$.
- Furthermore, it would be an important area of interest to employ Algorithm 8 for applications especially for generating approximate solutions of inverse problems in radiotherapy treatment formulated as an approximation problem of type (P_{app}).
- Despite the good features associated with the proposed methods, theoretical convergence analyses hugely depend on the line search procedure for obtaining the step length as the iteration progresses. Therefore, we encourage further research on how to achieve global convergence with a constant step length.

8.3 Summary of Contributions

Several findings presented in this thesis are based on the following four articles, which have been published in international peer-reviewed journals:

MTAE2021a : A. B. Muhammad, C. Tammer, A. M. Awwal and R. Elster. A Dai–Liao–like projection method for solving convex constrained nonlinear monotone equations and minimizing the ℓ_1 –regularized problem. *Appl. Set-Valued Anal. Optim.*, 3 (2021), 259–279. Available online at <http://asvao.biemdas.com>, <https://doi.org/10.23952/asvao.3.2021.3.02>.

MTAEM2021b : A. B. Muhammad, C. Tammer, A. M. Awwal, R. Elster and Z. Ma. Inertial–type Projection Method for Solving Convex Constrained Monotone Nonlinear Equations With Application in Robotic Motion Control. *J. Nonlinear Var. Anal.*, 5 (2021), 831–849, Available online at <http://jnva.biemdas.com>, <https://doi.org/10.23952/jnva.5.2021.5.13>.

AIMT2020 : A. B. Abubakar, A. H. Ibrahim, A. B. Muhammad and C. Tammer. A Modified Descent Dai–Yuan Conjugate Gradient Method for Constraint Nonlinear Monotone Operator Equations. *Applied Analysis and Optimization*, 4 (2020), 1–24.

MTA2019 : A. B. Muhammad, C. Tammer and A.B. Abubakar. A Hybrid Conjugate Gradient Algorithm With Spectral Parameters for Solving Monotone Operator Equations With Convex Constraints and Application. *Nonlinear Analysis and Convex Analysis & International Conference on Optimization: Techniques and Application (NACA-ICOTA2019)*, Yokohama Publishers, Japan, 2 (2019), 23–51.

In what follows, we summarize the author's main new contributions to each chapter of this thesis:

- Chapter 3 is new, however within this chapter we use some ideas that were presented in the articles **AIMT2020**, **MTA2019** and **MTAE2021a**. Lemma 3.4.2 and Theorem 3.4.3 are new.
- In Chapter 4, Section 4.1 is based on the joint work **MTAEM2021b**, while Section 4.2 is new.
- The discussion of the numerical experiments given in Chapter 5 is based on the results from the papers **AIMT2020**, **MTA2019** and **MTAE2021a**. Section 5.1 is new while Section 5.2 is mainly based on the paper **MTAEM2021b**.
- In Chapter 6, Section 6.1 is completely new and is the sole work of the author. Furthermore, the results in Section 6.2 is based on the work **MTAEM2021b**.
- Chapter 7 is the sole work of the author.

Bibliography

- [1] A. B. ABUBAKAR, A. H. IBRAHIM, A. B. MUHAMMAD, AND C. TAMMER, *A modified descent Dai–Yuan conjugate gradient method for constraint nonlinear monotone operator equations*, Applied Analysis and Optimization, 4 (2020), pp. 1–24.
- [2] A. B. ABUBAKAR AND P. KUMAM, *A descent Dai-Liao conjugate gradient method for nonlinear equations*, Numerical Algorithms, (2018, DOI:10.1007/s11075-018-0541-z), pp. 1–14.
- [3] A. B. ABUBAKAR, P. KUMAM, AND H. MOHAMMAD, *A note on the spectral gradient projection method for nonlinear monotone equations with applications*, Computational and Applied Mathematics, 39 (2020), pp. 1–35.
- [4] A. B. ABUBAKAR, J. SABU’U, P. KUMAM, AND A. SHAH, *Solving nonlinear monotone operator equations via modified SR1 update*, Journal of Applied Mathematics and Computing, 67 (2021), pp. 343–373.
- [5] H. AKAIKE, *On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method*, Annals of the Institute of Statistical Mathematics, 11 (1959), pp. 1–16.
- [6] M. AL-BAALI, *Descent property and global convergence of the Fletcher–Reeves method with inexact line search*, IMA Journal of Numerical Analysis, 5 (1985), pp. 121–124.
- [7] N. ANDREI, *An acceleration of gradient descent algorithm with backtracking for unconstrained optimization*, Numerical Algorithms, 42 (2006), pp. 63–73.
- [8] N. ANDREI, *Another hybrid conjugate gradient algorithm for unconstrained optimization*, Numer. Algorithms, 47 (2008), pp. 143–156.
- [9] N. ANDREI, *Another conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization*, Journal of Optimization Theory and Applications, 159 (2013), pp. 159–182.
- [10] N. ANDREI, *A simple three-term conjugate gradient algorithm for unconstrained optimization*, Journal of Computational and Applied Mathematics, 241 (2013), pp. 19–29.
- [11] N. ANDREI, *Nonlinear conjugate gradient methods for unconstrained optimization*, Springer, 2020.

- [12] L. ARMIJO, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific Journal of mathematics, 16 (1966), pp. 1–3.
- [13] A. M. AWWAL, P. KUMAM, AND A. B. ABUBAKAR, *A modified conjugate gradient method for monotone nonlinear equations with convex constraints*, Applied Numerical Mathematics, 145 (2019), pp. 507–520.
- [14] A. M. AWWAL, P. KUMAM, AND A. B. ABUBAKAR, *Spectral modified Polak–Ribière–Polyak projection conjugate gradient method for solving monotone systems of nonlinear equations*, Applied Mathematics and Computation, 362 (2019), p. 124514.
- [15] A. M. AWWAL, P. KUMAM, A. B. ABUBAKAR, A. WAKILI, AND N. PAKKARANANG, *A new hybrid spectral gradient projection method for monotone system of nonlinear equations with convex constraints*, Thai Journal of Mathematics, 16 (2018).
- [16] A. M. AWWAL, P. KUMAM, H. MOHAMMAD, W. WATTHAYU, AND A. B. ABUBAKAR, *A Perry-type derivative-free algorithm for solving nonlinear system of equations and minimizing ℓ_1 -regularized problem*, Optimization, 70 (2021), pp. 1231–1259.
- [17] A. M. AWWAL, P. KUMAM, L. WANG, S. HUANG, AND W. KUMAM, *Inertial-based derivative-free method for system of monotone nonlinear equations and application*, IEEE Access, 8 (2020), pp. 226921–226930.
- [18] A. M. AWWAL, L. WANG, P. KUMAM, AND H. MOHAMMAD, *A two-step spectral gradient projection method for system of nonlinear monotone equations and image deblurring problems*, Symmetry, 12 (2020), p. 874.
- [19] A. M. AWWAL, L. WANG, P. KUMAM, H. MOHAMMAD, AND W. WATTHAYU, *A projection Hestenes–Stiefel method with spectral parameter for nonlinear monotone equations and signal processing*, Mathematical and Computational Applications, 25 (2020), p. 27.
- [20] S. BABAIE-KAFAKI AND S. REZAEI, *Two accelerated nonmonotone adaptive trust region line search methods*, Numerical Algorithms, 78 (2018), pp. 911–928.
- [21] M. R. BANHAM AND A. K. KATSAGGELOS, *Digital image restoration*, IEEE signal processing magazine, 2 (1997), pp. 24–417.
- [22] J. BARZILAI AND J. M. BORWEIN, *Two-point step size gradient methods*, IMA Journal of Numerical Analysis, 8 (1988), pp. 141–148.
- [23] E. BEALE, *A deviation of conjugate gradients.*, Numerical methods for nonlinear optimization, (1972), pp. 39–43.
- [24] A. BECK, *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*, SIAM, 2014.

- [25] A. BECK AND M. TEBoulLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences, 2 (2009), pp. 183–202.
- [26] D. BERTSEKAS, *Nonlinear programming, athena sci*, Optim. Comput. Ser., 3rd ed., Athena Scientific, Belmont, MA, (2016).
- [27] Y. BING AND G. LIN, *An efficient implementation of Merrills method for sparse or partially separable systems of nonlinear equations*, SIAM Journal on Optimization, 1 (1991), pp. 206–221.
- [28] E. G. BIRGIN AND J. M. MARTÍNEZ, *A spectral conjugate gradient method for unconstrained optimization*, Applied Mathematics and optimization, 43 (2001), pp. 117–128.
- [29] E. G. BIRGIN, J. M. MARTÍNEZ, AND M. RAYDAN, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM Journal on Optimization, 10 (2000), pp. 1196–1211.
- [30] E. G. BIRGIN, J. M. MARTÍNEZ, AND M. RAYDAN, *Spectral projected gradient methods: review and perspectives*, Journal of Statistical Software, 60 (2014), pp. 1–21.
- [31] C. M. BISHOP, *Pattern recognition and machine learning*, Springer google schola, 2 (2006), pp. 1122–1128.
- [32] A. C. BOVIK, *Handbook of image and video processing*, Academic press, 2010.
- [33] S. P. BOYD AND L. VANDENBERGHE, *Convex optimization*, Cambridge university press, 2004.
- [34] F. CARLSSON AND A. FORSGREN, *Iterative regularization in intensity-modulated radiation therapy optimization*, Medical physics, 33 (2006), pp. 225–234.
- [35] A. CAUCHY, *Méthode générale pour la résolution des systemes d'équations simultanées*, Comp. Rend. Sci. Paris, 25 (1847), pp. 536–538.
- [36] C. L. CHAN, A. K. KATSAGGELOS, AND A. V. SAHAKIAN, *Image sequence filtering in quantum-limited noise with applications to low-dose fluoroscopy*, IEEE Transactions on Medical Imaging, 12 (1993), pp. 610–621.
- [37] B. S. CHEN, H. J. UANG, AND C. S. TSENG, *Robust tracking enhancement of robot systems including motor dynamics: A fuzzy-based dynamic game approach*, IEEE Transactions on Fuzzy Systems, 6 (1998), pp. 538–552.
- [38] S. S. CHEN, D. L. DONOHO, AND M. A. SAUNDERS, *Atomic decomposition by basis pursuit*, SIAM review, 43 (2001), pp. 129–159.
- [39] W. CHENG, *A PRP type method for systems of monotone equations*, Mathematical and Computer Modelling, 50 (2009), pp. 15–20.
- [40] D. CIARDO, M. PERONI, M. RIBOLDI, D. ALTERIO, G. BARONI, AND R. ORECCHIA, *The role of regularization in deformable image registration for head and neck adaptive radiotherapy*, Technology in cancer research & treatment, 12 (2013), pp. 323–331.

- [41] Y. H. DAI, *A new analysis on the Barzilai-Borwein gradient method*, Journal of the operations Research Society of China, 1 (2013), pp. 187–198.
- [42] Y. H. DAI, M. AL-BAALI, AND X. YANG, *A positive Barzilai–Borwein-like stepsize and an extension for symmetric linear systems*, in Numerical Analysis and Optimization, Springer, 2015, pp. 59–75.
- [43] Y. H. DAI, Y. HUANG, AND X. W. LIU, *A family of spectral gradient methods for optimization*, Computational Optimization and Applications, 74 (2019), pp. 43–65.
- [44] Y. H. DAI AND C. X. KOU, *A Barzilai–Borwein conjugate gradient method*, Science China Mathematics, 59 (2016), pp. 1511–1524.
- [45] Y. H. DAI AND L. Z. LIAO, *New conjugacy conditions and related nonlinear conjugate gradient methods*, Applied Mathematics and Optimization, 43 (2001), pp. 87–101.
- [46] Y. H. DAI AND Y. YUAN, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM Journal on optimization, 10 (1999), pp. 177–182.
- [47] Z. DAI AND H. ZHOU, *Prediction of stock returns: Sum-of-the-parts method and economic constraint method*, Sustainability, 12 (2020), p. 541.
- [48] Z. DAI AND H. ZHU, *Stock return predictability from a mixed model perspective*, Pacific-Basin Finance Journal, 60 (2020), p. 101267.
- [49] E. D. DOLAN AND J. J. MORÉ, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.
- [50] Q. DONG, Y. CHO, L. ZHONG, AND T. M. RASSIAS, *Inertial projection and contraction algorithms for variational inequalities*, Journal of Global Optimization, 70 (2018), pp. 687–704.
- [51] F. Y. EDGEWORTH, *Mathematical psychics: An essay on the application of mathematics to the moral sciences*, CK Paul, 1881.
- [52] M. EHRGOTT, *Multicriteria optimization*, vol. 491, Springer Science & Business Media, 2005.
- [53] M. EHRGOTT AND M. BURJONY, *Radiation therapy planning by multicriteria optimization*, in Proceedings of the 36th Annual Conference of the Operational Research Society of New Zealand, 2001, pp. 244–253.
- [54] M. EHRGOTT, Ç. GÜLER, H. W. HAMACHER, AND L. SHAO, *Mathematical optimization in intensity modulated radiation therapy*, Annals of Operations Research, 175 (2010), pp. 309–365.
- [55] M. EHRGOTT, K. KLAMROTH, AND C. SCHWEHM, *An MCDM approach to portfolio optimization*, European Journal of Operational Research, 155 (2004), pp. 752–770.
- [56] G. EICHFELDER, *Adaptive scalarization methods in multiobjective optimization*, Vector Optimization, Springer-Verlag, Berlin, 2008.

- [57] F. FACCHINEI AND J. S. PANG, *Finite-dimensional variational inequalities and complementarity problems*, Springer, 2003.
- [58] M. FIGUEIREDO, R. D. NOWAK, AND S. J. WRIGHT, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of selected topics in signal processing, 1 (2007), pp. 586–597.
- [59] R. FLETCHER, *Practical methods of optimization*, A Wiley-Interscience Publication, John Wiley & Sons, Ltd., Chichester, second ed., 1987.
- [60] R. FLETCHER, *On the Barzilai-Borwein method*, in Optimization and control with applications, Springer, 2005, pp. 235–256.
- [61] R. FLETCHER AND C. M. REEVES, *Function minimization by conjugate gradients*, The computer journal, 7 (1964), pp. 149–154.
- [62] J. FOCKE, *Vektormaximumproblem und parametrische optimierung*, Mathematische Operationsforschung und Statistik, 4 (1973), pp. 365–369.
- [63] J. FOCKE, *Strict linear inequalities with respect to conic semiorders*, Statistics: A Journal of Theoretical and Applied Statistics, 6 (1975), pp. 881–900.
- [64] J. GARETH, W. DANIELA, H. TREVOR, AND T. ROBERT, *An introduction to statistical learning: with applications in R*, Spinger, 2013.
- [65] C. GERTH AND K. PÖHLER, *Duality and algorithmic application in the vector location problem*, Optimization, 19 (1988), pp. 491–512.
- [66] P. E. GILL AND W. MURRAY, *Newton-type methods for unconstrained and linearly constrained optimization*, Mathematical Programming, 7 (1974), pp. 311–350.
- [67] A. GÖPFERT AND R. NEHSE, *Vector optimization: theory, methods and applications*, vol. 74, knight, 1990.
- [68] A. GÖPFERT, H. RIAHI, C. TAMMER, AND C. ZALINESCU, *Variational methods in partially ordered spaces*, vol. 17, Springer, 2003.
- [69] A. GÖPFERT, T. RIEDRICH, AND C. TAMMER, *Angewandte Funktionalanalysis*, Springer, 2009.
- [70] N. Z. GU AND J. T. MO, *Incorporating nonmonotone strategies into the trust region method for unconstrained optimization*, Computers & Mathematics with Applications, 55 (2008), pp. 2158–2172.
- [71] C. GÜNTHER AND C. TAMMER, *On generalized-convex constrained multi-objective optimization*, Martin-Luther-Universität Halle-Wittenberg, Naturwissenschaftliche Fakultät, 2017.
- [72] W. W. HAGER AND H. ZHANG, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM Journal on optimization, 16 (2005), pp. 170–192.

- [73] W. W. HAGER AND H. ZHANG, *A survey of nonlinear conjugate gradient methods*, Pacific journal of Optimization, 2 (2006), pp. 35–58.
- [74] B. HAIM, *Analyse fonctionnelle, théorie et applications*, collection mathématiques appliquées pour la maîtrise. ANNEXE, (1983).
- [75] E. T. HALE, W. YIN, AND Y. ZHANG, *A fixed-point continuation method for ℓ_1 -regularized minimization with applications to compressed sensing*, CAAM TR07-07, Rice University, 43 (2007), p. 44.
- [76] H. W. HAMACHER, *Mathematical solution methods for planar location problems*, Springer-Verlag, 2019.
- [77] H. W. HAMACHER AND K. H. KÜFER, *Inverse radiation therapy planning: A multiple objective optimisation approach*, in Monitoring, Evaluating, Planning Health Services, World Scientific, 1999, pp. 177–189.
- [78] T. HASTIE, R. TIBSHIRANI, J. H. FRIEDMAN, AND J. H. FRIEDMAN, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2, Springer, 2009.
- [79] M. HENIG, *Proper efficiency with respect to cones*, Journal of Optimization Theory and Applications, 36 (1982), pp. 387–407.
- [80] E. C. HENKEL AND C. TAMMER, *ε -variational inequalities for vector approximation problems*, Optimization, 38 (1996), pp. 11–21.
- [81] E. C. HENKEL AND C. TAMMER, *ε -variational inequalities in partially ordered spaces*, Optimization, 36 (1996), pp. 105–118.
- [82] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, NBS Washington, DC, 1952.
- [83] H. HEUSER, *Funktionalanalysis*, Teubner, Stuttgart, 1986.
- [84] C. HILLERMEIER AND J. JAHN, *Multiobjective optimization: survey of methods and industrial applications*, Surv. Math. Ind, 11 (2005), pp. 1–42.
- [85] E. HODGSON, *A Textbook of Modern Toxicology*, Wiley Online Library, 2004.
- [86] Y. HONJO AND N. KUDO, *Matching objective and subjective information in geotechnical inverse analysis based on entropy minimization*, in Inverse problems in engineering mechanics, Elsevier, 1998, pp. 263–271.
- [87] A. H. IBRAHIM, P. KUMAM, A. B. ABUBAKAR, W. JIRAKITPUWAPAT, AND J. ABUBAKAR, *A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing*, Heliyon, 6 (2020), pp. 1–17.

- [88] S. ISHIKAWA, *Fixed points by a new iteration method*, Proceedings of the American Mathematical Society, 44 (1974), pp. 147–150.
- [89] N. A. IUSEM AND V. M. SOLODOV, *Newton-type methods with generalized distances for constrained optimization*, Optimization, 41 (1997), pp. 257–278.
- [90] A. F. IZMAILOV AND M. V. SOLODOV, *Newton-type methods for optimization and variational problems*, vol. 3, Springer, 2014.
- [91] E. M. JAFAROV, M. A. PARLAKCI, AND Y. ISTEфанOPULOS, *A new variable structure PID-controller design for robot manipulators*, IEEE Transactions on Control Systems Technology, 13 (2004), pp. 122–130.
- [92] J. JAHN, *Mathematical vector optimization in partially ordered linear spaces*, vol. 31, Peter Lang GmbH, Internationaler Verlag Der Wissenschaften, 1985.
- [93] J. JAHN, *Vector optimization, theory, applications, and extensions*, 2004.
- [94] J. JAHN, *Introduction to the theory of nonlinear optimization*, Springer Nature, 2020.
- [95] J. JAHN AND W. KRABS, *Applications of multicriteria optimization in approximation theory*, in Multicriteria Optimization in Engineering and in the Sciences, Springer, 1988, pp. 49–75.
- [96] G. JAMESON, *Ordered linear spaces*, in Ordered linear spaces, Springer, 1970, pp. 1–39.
- [97] J. C. JOHN, *Introduction to robotics: Mechanics and control*, 2005.
- [98] A. A. KHAN, C. TAMMER, AND C. ZALINESCU, *Set-valued optimization*, Springer, 2016.
- [99] P. D. KHANH, B. S. MORDUKHOVICH, V. T. PHAT, AND D. B. TRAN, *Generalized damped Newton algorithms in nonsmooth optimization with applications to Lasso problems*, arXiv preprint arXiv:2101.10555, (2021).
- [100] P. D. KHANH, B. S. MORDUKHOVICH, V. T. PHAT, AND D. B. TRAN, *Generalized damped Newton algorithms in nonsmooth optimization via second-order subdifferentials*, Journal of Global Optimization, 86 (2023), pp. 93–122.
- [101] P. D. KHANH, B. S. MORDUKHOVICH, V. T. PHAT, AND D. B. TRAN, *Globally convergent coderivative-based generalized Newton methods in nonsmooth optimization*, Mathematical Programming, (2023), pp. 1–57.
- [102] P. D. KHANH, B. S. MORDUKHOVICH, AND D. B. TRAN, *Inexact reduced gradient methods in nonconvex optimization*, arXiv preprint arXiv:2204.01806, (2022).
- [103] S. J. KIM, K. KOH, M. LUSTIG, S. BOYD, D. GORINEVSKY, ET AL., *A method for large-scale ℓ_1 -regularized least squares*, IEEE Journal on Selected Topics in Signal Processing, 1 (2007), pp. 606–617.

- [104] F. KOHSAKA AND W. TAKAHASHI, *Fixed point theorems for a class of nonlinear mappings related to maximal monotone operators in Banach spaces*, Archiv der Mathematik, 91 (2008), pp. 166–177.
- [105] K. H. KÜFER, A. SCHERRER, M. MONZ, F. ALONSO, H. TRINKAUS, T. BORTFELD, AND C. THIEKE, *Intensity-modulated radiotherapy—a large scale multi-criteria programming problem*, OR spectrum, 25 (2003), pp. 223–249.
- [106] W. LA CRUZ, J. MARTÍNEZ, AND M. RAYDAN, *Spectral residual method without gradient information for solving large-scale nonlinear systems of equations*, Mathematics of Computation, 75 (2006), pp. 1429–1448.
- [107] W. LA CRUZ, J. M. MARTÍNEZ, AND M. RAYDAN, *Spectral residual method without gradient information for solving large-scale nonlinear systems of equations*, Mathematics of Computation, 75 (2006), pp. 1429–1448.
- [108] W. LA CRUZ AND M. RAYDAN, *Nonmonotone spectral methods for large-scale nonlinear systems*, Optimization Methods and Software, 18 (2003), pp. 583–599.
- [109] M. LAHANAS, E. SCHREIBMANN, AND D. BALTAS, *Multiobjective inverse planning for intensity modulated radiotherapy with constraint-free gradient-based optimization algorithms*, Physics in Medicine & Biology, 48 (2003), p. 2843.
- [110] S. M. LAJEVARDI, *Structural similarity classifier for facial expression recognition*, Signal, Image and Video Processing, 8 (2014), pp. 1103–1110.
- [111] T. T. LE, *Multiobjective approaches based on variable ordering structures for intensity problems in radiotherapy treatment*, Investigación Operacional, 39 (2018).
- [112] J. LIU AND Y. FENG, *A derivative-free iterative method for nonlinear monotone equations with convex constraints*, Numerical Algorithms, 82 (2019), pp. 245–262.
- [113] J. LIU AND S. LI, *Spectral DY-type projection method for nonlinear monotone systems of equations*, Journal of Computational Mathematics, 33 (2015), pp. 341–355.
- [114] J. K. LIU AND S. J. LI, *A projection method for convex constrained monotone nonlinear equations with applications*, Computers & Mathematics with Applications, 70 (2015), pp. 2442–2453.
- [115] Y. LIU AND C. STOREY, *Efficient generalized conjugate gradient algorithms. I. Theory*, J. Optim. Theory Appl., 69 (1991), pp. 129–137.
- [116] T. L. MAGNANTI AND G. PERAKIS, *Solving variational inequality and fixed point problems by line searches and potential optimization*, Math. program, 101 (2004), pp. 435–461.
- [117] R. MAHNKEN, *Gradient-based methods for parameter identification of viscoplastic materials*, Inverse problems in engineering mechanics, 137 (1994).

- [118] W. R. MANN, *Mean value methods in iteration*, Proceedings of the American Mathematical Society, 4 (1953), pp. 506–510.
- [119] A. MIELE, T. WANG, AND S. MANCUSO, *Optimization of missions to Mars for robotic and manned spacecraft*, Nonlinear Analysis: Theory, Methods & Applications, 47 (2001), pp. 1425–1443.
- [120] H. MOHAMMAD AND A. B. ABUBAKAR, *A positive spectral gradient-like method for large-scale nonlinear monotone equations*, Bull. Comput. Appl. Math., 5 (2017), pp. 97–113.
- [121] B. S. MORDUKHOVICH, X. YUAN, S. ZENG, AND J. ZHANG, *A globally convergent proximal Newton-type method in nonsmooth convex optimization*, Math. Program., 198 (2023), pp. 899–936.
- [122] J. J. MORÉ AND S. M. WILD, *Benchmarking derivative-free optimization algorithms*, SIAM Journal on Optimization, 20 (2009), pp. 172–191.
- [123] A. B. MUHAMMAD, C. TAMMER, AND A. B. ABUBAKAR, *A hybrid conjugate gradient algorithm with spectral parameters for solving monotone operator equations with convex constraints and application*, Nonlinear Analysis and Convex Analysis & International Conference on Optimization: Techniques and Application (NACA-ICOTA2019), Yokohama Publishers, Japan, 2 (2019), pp. 23–51.
- [124] A. B. MUHAMMAD, C. TAMMER, A. M. AWWAL, AND R. ELSTER, *A Dai-Liao-like projection method for solving convex constrained nonlinear monotone equations and minimizing the ℓ_1 -regularized problem*, Appl. Set-Valued Anal. Optim., 3 (2021), pp. 259–279.
- [125] A. B. MUHAMMAD, C. TAMMER, A. M. AWWAL, R. ELSTER, AND Z. MA, *Inertial-type projection methods for solving convex constrained monotone nonlinear equations with applications to robotic motion control*, JOURNAL OF NONLINEAR AND VARIATIONAL ANALYSIS, 5 (2021), pp. 831–849.
- [126] J. NA, X. REN, AND D. ZHENG, *Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer*, IEEE transactions on neural networks and learning systems, 24 (2013), pp. 370–382.
- [127] L. NAZARETH, *A conjugate direction algorithm without line searches*, Journal of Optimization Theory and Applications, 23 (1977), pp. 373–387.
- [128] J. NOCEDAL AND S. WRIGHT, *Numerical optimization*, Springer Science & Business Media, 2006.
- [129] J. NOCEDAL AND S. J. WRIGHT, *Line search methods*, Numerical optimization, (2006), pp. 30–65.

- [130] W. OETTLI, *Approximate solutions of variational inequalities*, Quantitative Wirtschaftsforschung, (1977), pp. 535–538.
- [131] J. PANG, *Inexact Newton methods for the nonlinear complementarity problem*, Mathematical Programming, 36 (1986), pp. 54–71.
- [132] V. PARETO, *Manuale di economia politica (manual of political economy)*, Milan, Italy: Societa Editrice Libraiia, (1906).
- [133] V. PARRA-VEGA, S. ARIMOTO, Y. H. LIU, G. HIRZINGER, AND P. AKELLA, *Dynamic sliding PID control for tracking of robot manipulators: Theory and experiments*, IEEE Transactions on Robotics and Automation, 19 (2003), pp. 967–976.
- [134] A. PERRY, *A modified conjugate gradient algorithm*, Operations Research, 26 (1978), pp. 1073–1078.
- [135] E. POLAK AND G. RIBIERE, *Note sur la convergence de methodes de directions conjuguées*, Rev Française Informat Recherche Operationelle, 3e Année, 16 (1969), pp. 35–43.
- [136] B. T. POLYAK, *Some methods of speeding up the convergence of iteration methods*, USSR Computational Mathematics and Mathematical Physics, 4 (1964), pp. 1–17.
- [137] B. T. POLYAK, *The conjugate gradient method in extremal problems*, USSR Computational Mathematics and Mathematical Physics, 9 (1969), pp. 94–112.
- [138] B. T. POLYAK, *Introduction to optimization. optimization software*, Inc., Publications Division, New York, 1 (1987), p. 32.
- [139] M. J. POWELL, *Nonconvex minimization calculations and the conjugate gradient method*, in Numerical analysis, Springer, 1984, pp. 122–141.
- [140] Y. QIANG, F. JING, J. ZENG, AND Z. HOU, *Dynamic modeling and vibration mode analysis for an industrial robot with rigid links and flexible joints*, in 2012 24th Chinese Control and Decision Conference (CCDC), IEEE, 2012, pp. 3317–3321.
- [141] M. RAYDAN, *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA Journal of Numerical Analysis, 13 (1993), pp. 321–326.
- [142] M. RAYDAN, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM Journal on Optimization, 7 (1997), pp. 26–33.
- [143] H. U. REHMAN, P. KUMAM, A. B. ABUBAKAR, AND Y. J. CHO, *The extragradient algorithm with inertial effects extended to equilibrium problems. comput*, Appl. Math, 39 (2020), pp. 1–26.
- [144] H. U. REHMAN, P. KUMAM, I. K. ARGYROS, W. DEEBANI, AND W. KUMAM, *Inertial extragradient method for solving a family of strongly pseudomonotone equilibrium problems in real Hilbert spaces with application in variational inequality problem*, Symmetry, 12 (2020), p. 503.

- [145] R. T. ROCKAFELLAR, *Convex analysis*, Princeton university press, 2015.
- [146] D. SAHU, Y. CHO, Q. DONG, M. KASHYAP, AND X. LI, *Inertial relaxed CQ algorithms for solving a split feasibility problem in Hilbert spaces*, Numerical Algorithms, (2020), pp. 1–21.
- [147] H. H. SCHAEFER, *Topological vector spaces (4th printing)*, New York, (1980).
- [148] P. SCHÖNFELD, *Some duality theorems for the non-linear vector maximum problem*, Unternehmensforschung, 14 (1970), pp. 51–63.
- [149] D. F. SHANNO, *Conjugate gradient methods with inexact searches*, Mathematics of operations research, 3 (1978), pp. 244–256.
- [150] M. V. SOLODOV AND B. F. SVAITER, *A globally convergent inexact Newton method for systems of monotone equations*, in Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Springer, 1998, pp. 355–369.
- [151] L. STELLA, A. THEMELIS, AND P. PATRINOS, *Forward–backward quasi-Newton methods for nonsmooth optimization problems*, Computational Optimization and Applications, 67 (2017), pp. 443–487.
- [152] M. SUN, J. LIU, AND Y. WANG, *Two improved conjugate gradient methods with application in compressive sensing and motion control*, Mathematical Problems in Engineering, 2020 (2020).
- [153] W. SUN AND Y.-X. YUAN, *Optimization theory and methods: nonlinear programming*, vol. 1, Springer Science & Business Media, 2006.
- [154] W. TAKAHASHI, *Nonlinear Functional Analysis: Fixed Point Theory and its Applications*, Yokohama Publishers, 2000.
- [155] W. TAKAHASHI, *Introduction to nonlinear and convex analysis*, Yokohama Publishers, 2009.
- [156] C. TAMMER, *Necessary conditions for approximately efficient solutions of vector approximation problems. in: Approximation and optimization in the Caribbean II (Havana, 1993)*, Approx. Optim., 8, Lang, Frankfurt am Main, (1995), pp. 651–663.
- [157] C. TAMMER, *Approximate solutions of vector-valued control approximation problems*, Studies in Locational Analysis, 10 (1996), pp. 151–162.
- [158] C. TAMMER AND A. GÖPFERT, *Theory of vector optimization*, in Multiple criteria optimization: state of the art annotated bibliographic surveys, vol. 52 of Internat. Ser. Oper. Res. Management Sci., Kluwer Acad. Publ., Boston, MA, 2002, pp. 1–70.
- [159] C. TAMMER AND P. WEIDNER, *Scalarization and Separation by Translation Invariant Functions*, Springer, 2020.

- [160] G. Y. TANG, L. SUN, C. LI, AND M. Q. FAN, *Successive approximation procedure of optimal tracking control for nonlinear similar composite systems*, *Nonlinear Analysis: Theory, Methods & Applications*, 70 (2009), pp. 631–641.
- [161] G. Y. TANG, Y. D. ZHAO, AND B. L. ZHANG, *Optimal output tracking control for nonlinear systems via successive approximation approach*, *Nonlinear Analysis: Theory, Methods & Applications*, 66 (2007), pp. 1365–1377.
- [162] N. T. VINH AND L. D. MUU, *Inertial extragradient algorithms for solving equilibrium problems*, *Acta Mathematica Vietnamica*, 44 (2019), pp. 639–663.
- [163] Z. WAN, J. GUO, J. LIU, AND W. LIU, *A modified spectral conjugate gradient projection method for signal recovery*, *Signal, Image and Video Processing*, 12 (2018), pp. 1455–1462.
- [164] C. WANG, Y. WANG, AND C. XU, *A projection method for a system of nonlinear monotone equations with convex constraints*, *Mathematical Methods of Operations Research*, 66 (2007), pp. 33–46.
- [165] G. WANKA, *On duality in the vectorial control-approximation problem*, *Zeitschrift für Operations Research*, 35 (1991), pp. 309–320.
- [166] P. WOLFE, *Convergence conditions for ascent methods*, *SIAM review*, 11 (1969), pp. 226–235.
- [167] Y. XIAO, Q. WANG, AND Q. HU, *Non-smooth equations based method for ℓ_1 -norm problems with applications to compressed sensing*, *Nonlinear Analysis: Theory, Methods & Applications*, 74 (2011), pp. 3570–3577.
- [168] Y. XIAO AND H. ZHU, *A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing*, *Journal of Mathematical Analysis and Applications*, 405 (2013), pp. 310–319.
- [169] M. M. YAHAYA, P. KUMAM, A. M. AWWAL, AND S. AJI, *A structured quasi-Newton algorithm with nonmonotone search strategy for structured NLS problems and its application in robotic motion control*, *Journal of Computational and Applied Mathematics*, (2021), p. 113582.
- [170] Z. YU, J. LIN, J. SUN, Y. XIAO, L. LIU, AND Z. LI, *Spectral gradient projection method for monotone nonlinear equations with convex constraints*, *Applied numerical mathematics*, 59 (2009), pp. 2416–2423.
- [171] C. ZALINESCU, *Convex analysis in general vector spaces*, World scientific, 2002.
- [172] E. H. ZARANTONELLO, *Projections on convex sets in Hilbert space and spectral theory: Part i. projections on convex sets: Part ii. spectral theory*, in *Contributions to nonlinear functional analysis*, Elsevier, 1971, pp. 237–424.
- [173] E. ZEIDLER, *IJA linear monotone operators, IIB nonlinear monotone operators*, *Nonlinear Functional Analysis and Its Application I*, (1990).

- [174] L. ZHANG AND W. ZHOU, *Spectral gradient projection method for solving nonlinear monotone equations*, Journal of Computational and Applied Mathematics, 196 (2006), pp. 478–484.
- [175] L. ZHANG, W. ZHOU, AND D. H. LI, *A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence*, IMA Journal of Numerical Analysis, 26 (2006), pp. 629–640.
- [176] Y. ZHANG, L. HE, C. HU, J. GUO, J. LI, AND Y. SHI, *General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization*, Journal of Computational and Applied Mathematics, 347 (2019), pp. 314–329.
- [177] Y. ZHANG, W. LI, B. QIU, Y. DING, AND D. ZHANG, *Three-state space reformulation and control of MD-included one-link robot system using direct-derivative and hang-dynamics methods*, in 2017 29th Chinese Control And Decision Conference (CCDC), IEEE, 2017, pp. 3724–3729.
- [178] L. ZHENG, L. YANG, AND Y. LIANG, *A conjugate gradient projection method for solving equations with convex constraints*, J. Comput. Appl. Math., 375 (2020), p. 112781.
- [179] G. ZOUTENDIJK, *Nonlinear programming, computational methods*, Integer and nonlinear programming, (1970), pp. 37–86.

Selbständigkeitserklärung

Hiermit erkläre Ich Muhammad, Abubakar Bakoji an Eides statt, dass ich die vorliegende Dissertation selbständig und ohne fremde Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht.

Halle (Saale), den 29. November. 2024

Curriculum Vitae

Abubakar Bakoji Muhammad

Oct. 2018 - Oct. 2024

Doctoral Studies in Applied Mathematics
Martin-Luther-Universität Halle-Wittenberg

December 2015

Master of Science in Mathematics

Oct. 2014 - Dec. 2015

Master Studies in Applied Mathematics
University of Ilorin, Ilorin, Nigeria

September 2012

Bachelor of Science in Mathematics

Oct. 2006 - Sept. 2012

Bachelor Studies in Mathematics
Gombe State University, Nigeria

June 1999

Abitur

Sept. 1999 - July. 2005

Government Science Secondary School Gombe, Nigeria

Halle (Saale), December 4, 2024

(Abubakar Bakoji Muhammad)

Publications

1. A. B. Muhammad, C. Tammer and A.B. Abubakar. A Hybrid Conjugate Gradient Algorithm With Spectral Parameters for Solving Monotone Operator Equations With Convex Constraints and Application. *Nonlinear Analysis and Convex Analysis & International Conference on Optimization: Techniques and Application (NACA-ICOTA2019)*, Yokohama Publishers, Japan, 2 (2019), 23–51.
2. A. B. Abubakar, A. H. Ibrahim, A. B. Muhammad and C. Tammer. A Modified Descent Dai-Yuan Conjugate Gradient Method for Constraint Nonlinear Monotone Operator Equations. *Applied Analysis and Optimization*, 4 (2020), 1–24.
3. A. B. Muhammad, C. Tammer, A. M. Awwal, R. Elster and Z. Ma. Inertial-type Projection Method for Solving Convex Constrained Monotone Nonlinear Equations With Application in Robotic Motion Control. *J. Nonlinear Var. Anal.*, 5 (2021), 831–849, Available online at <http://jnva.biemdas.com>, <https://doi.org/10.23952/jnva.5.2021.5.13>.
4. A. B. Muhammad, C. Tammer, A. M. Awwal and R. Elster. A Dai-Liao-like projection method for solving convex constrained nonlinear monotone equations and minimizing the ℓ_1 -regularized problem. *Appl. Set-Valued Anal. Optim.*, 3 (2021), 259–279. Available online at <http://asvao.biemdas.com>, <https://doi.org/10.23952/asvao.3.2021.3.02>.