Iterative Projection and Reflection Methods: Theory and Practice

Matthew K. Tam
BMath(Hon) BSci(Chem)

A thesis submitted for the degree of
Doctor of Philosophy (Mathematics)

Submitted: December, 2015
Revised: February, 2016

School of Mathematical and Physical Sciences
Faculty of Science and Information Technology
University of Newcastle, Australia
Statement of Originality

The thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to the final version of my thesis being made available worldwide when deposited in the University’s Digital Repository, subject to the provisions of the Copyright Act 1968.

Statement of Collaboration

I hereby certify that the work embodied in this thesis has been done in collaboration with other researchers, or carried out in other institutions. I have included as part of the thesis a statement clearly outlining the extent of collaboration, with whom and under what auspices.

Statement of Authorship

I hereby certify that the work embodied in this thesis contains a published papers of which I am a joint author. I have included as part of the thesis a written statement, endorsed by my supervisor, attesting to my contribution to the joint publications.

Matthew K. Tam

Date
Extent of Collaboration and Authorship

The majority of this dissertation has appeared or been accepted for publication. In every such case, I have been a significant and integral contributor involved in both the work undertaken and the preparation of the manuscript. For their inclusion here, these works have been appropriately edited and new material has been included. In what follows, I give a summary of these works.

The cyclic Douglas–Rachford method, which has since been referred to as the Borwein–Tam method in the literature, is the main motivator in Chapter 2. It is based on joint work with J.M. Borwein which falls within the usual scope of student-supervisor relations.


The following work is based on joint work with J.M. Borwein and B. Sims and also falls within the usual scope of student-supervisor relations, and appears in Chapter 4.


The following joint works represent an extensive collaborative effort with F.J. Aragón Artacho and J.M. Borwein. Numerical experiments were performed in applications which include combinatorial optimisation and matrix completion. Their content appears in Chapter 4.


In a follow-up investigation, the aforementioned applications were further refined. The following appears in Chapter 4 and is based on joint work with J.M. Borwein which falls within the usual scope of student-supervisor relations.

In the following, yet unpublished work of which I am sole author, a theoretical foundation to explain matrix completion applications is developed. Its content appears in Chapters 3 and 4.


The following joint work represents a continuing collaboration with F.J. Aragón Artacho and J.M. Borwein. Its content appears in Chapter 3.


Declaration by the candidate
I declare the details provided above to be correct.

Matthew K. Tam
Date

Endorsement by the supervisor
I, as supervisor of the candidate, certify the provided details to be correct.

Jonathan M. Borwein
Date
Acknowledgements

To my supervisor Jonathan Borwein, I wish to express my deepest gratitude. Your patience and vast knowledge has allowed me to become a better a mathematician. Without your encouragement, many of the wonderful opportunities and experiences of the past few years would simply not have been possible. It has been a great privilege to have you as supervisor and mentor.

To my co-supervisors Brailey Sims and Regina Burachik, I also wish to express my gratitude. Brailey particularly for the fondness of mathematics which he incited in me when I first encountered him as lecturer during my undergraduate studies. To Russell Luke and his group, thank you for your hospitality during my stay at the University of Göttingen.

To my parents, Kaye and Michael, thank you for your unconditional encouragement and support. To mum, thank you for making me attentive to details, and to dad, thank you for making sure I was never too attentive to details.

To Jacqueline, thank you for your love!
“The instrument that mediates between theory and practice, between thought and observation, is mathematics; it builds the connecting bridge and makes it stronger and stronger. Thus it happens that our entire present-day culture, insofar as it rests on intellectual insight into and harnessing of nature, is founded on mathematics.”

David Hilbert
Contents

Abstract xi

1 Introduction 1
  1.1 Basic Definitions and Notation 1
  1.2 Projectors and Reflectors 1
  1.3 The Feasibility Problem 4
  1.4 Product Space Reformulations 7
  1.5 Fundamental Algorithms 9

2 Convex Theory 11
  2.1 Nonexpansive Maps and Their Fixed Points 11
  2.2 The Cyclic Projection Method 18
  2.3 The Douglas–Rachford Feasibility Method 25
    2.3.1 Connections to Monotone Operator Splitting 28
  2.4 The Cyclic Douglas–Rachford Method 30
  2.5 Other Douglas–Rachford Variants 34
  2.6 Finer Behaviour of the Cyclic Douglas–Rachford Method 36

3 Nonconvex Theory 43
  3.1 Regularity Notions and Local Convergence 44
  3.2 Regularity of Sparsity Constraints 51
    3.2.1 Non-Negative Sparse Vector Sets 54
    3.2.2 Low-Rank Positive Semi-Definite Matrix Sets 61
    3.2.3 Characterisations of Regularity Conditions 63
  3.3 Global Convergence for Half-Spaces 65
    3.3.1 Properties of the Douglas–Rachford Operator 68
    3.3.2 Convergence of the Douglas–Rachford Algorithm 72
    3.3.3 Examples and Counter-Examples 81
4 Applications

4.1 Reconstruction from Non-Negative Moments

4.1.1 Strongly Convergent Algorithms

4.1.2 Linear Systems with Sparsity Bounds

4.2 Combinatorial Optimisation

4.2.1 Sudoku Puzzles

4.2.2 Nonogram Puzzles

4.2.3 Hadamard matrices

4.3 Euclidean Distance Matrix Reconstruction

4.3.1 Protein Conformation Determination

4.3.2 Performance Improving Heuristics

4.3.3 Additional Distance Data

4.3.4 Ionic Liquid Bulk Structure Determination

4.3.5 Regularity Properties of Constraints

4.4 Matrix Completion: A Unified Framework

4.4.1 Positive semi-definite matrices

4.4.2 Correlation matrices

4.4.3 Stochastic matrices

5 Conclusions and Open Questions

Bibliography

Index

Notation and Symbols
Abstract

This thesis investigates the family of so-called projection and reflection methods. These methods form the basis for a class of iterative algorithms which can be used to solve the feasibility problem which asks for a point in the intersection of a collection of constraint sets. Many optimisation and reconstruction problems can be profitably modelled within this framework, although the formulation is not always immediately obvious. In a typical feasibility problem the target intersection set is difficult to deal with directly. Projection and reflection algorithms overcome this difficulty by exploiting relatively simpler structure in each of the individual constraint sets from the collection.

In recent times, a particular member of the projection method family, the Douglas–Rachford method, has received extra attention. This is, in part, due to its empirically observed ability to successfully solve a variety of difficult non-convex problems including those of a combinatorial nature. Furthermore, even in the classical setting of convexity, there is some divergence in the features of the Douglas–Rachford method as compared to other members of the projection method family such as the alternating projection method. Until the work of this thesis, it was only possible to apply the Douglas–Rachford algorithm to feasibility problems involving only two constraints whereas virtually every other projection-type algorithm exhibited a natural many-set extension.

The organisation of this work is as follows: Chapter 1 introduces basic definitions, notation and background, before formally introducing the feasibility problem framework and fundamental projection-type algorithms. Chapter 2 focuses on theory in the presence of convex constraint sets, and introduces the recently developed cyclic Douglas–Rachford method which has since been referred to as the Borwein–Tam method in the literature. Chapter 3 focuses on theory in the absence of convexity. In this case, theoretical underpinnings are still in development and rather more cumbersome. Specific classes or instances of non-convex feasibility problems must be considered separately. Chapter 4 investigates applications, particularly of the Douglas–Rachford algorithm to settings without convexity. Chapter 5 summarises the contributions of this work as well as indicating open problems for future research.
Chapter 1

Introduction

The aim of this opening chapter is to provide a concise introduction to the central focus of this thesis and to place its contributions in context. Stated abstractly, we study a class of iterative algorithms which are useful for finding a point in the intersection of finitely many constraint sets. As we soon see, a great deal many problems can be modelled within this framework, although the formulation is not always obvious; there is some art involved. The particular class of algorithms upon which we focus iterate by solving a best approximation sub-problem at each stage, with the desired solution attained in the limit.

1.1 Basic Definitions and Notation

Unless otherwise stated, our setting is an arbitrary real Hilbert space \( \mathcal{H} \), by which we mean a linear space over \( \mathbb{R} \) equipped with an inner product \( \langle \cdot, \cdot \rangle \) such that the space is complete with respect to the norm induced given by \( \|x\| := \sqrt{\langle x, x \rangle} \) for all \( x \in \mathcal{H} \). The set of non-negative real numbers is denoted by \( \mathbb{R}_+ := \{ x \in \mathbb{R} : x \geq 0 \} \), and the set of positive real numbers by \( \mathbb{R}_{++} := \mathbb{R}_+ \setminus \{0\} \).

1.2 Projectors and Reflectors

Given an initial point and a set, the best approximation problem is to find a point contained within the set, nearest to the initial point. Central questions which arise in best approximation theory involve existence, uniqueness, and characterisations of best approximates. In this thesis, we consider algorithms which iterate by using the solutions to one or more best approximation problems at each stage. For this reason, it is important to study the family of operators which map points to sets of best approximates. These operators are called projectors or projection operators.
A detailed treatment of best approximation problems more generally can be found in Deutsch’s book [66].

The projector onto a set \( S \subseteq \mathcal{H} \) is the mapping \( P_S : \mathcal{H} \rightarrow S \) given by

\[
P_S(x) := \{ y \in S : \| x - y \| = d(x, S) \}.
\]

An element of \( P_S(x) \) is said to be a projection of \( x \) onto \( S \). A set \( S \) is said to be proximal if \( P_S(x) \neq \emptyset \) for every \( x \in \mathcal{H} \). If \( P_S(x) \) is singleton for every \( x \in \mathcal{H} \), then \( S \) is said to be Chebyshev. In an abuse of notation, we often write \( P_S(x) = p \) when \( P_S(x) = \{ p \} \).

**Proposition 1.2.1** (Proximal sets are closed). Every proximal subset of a Hilbert space is closed. Conversely, every non-empty locally compact set is proximal.

**Proof.** See, for instance, [66, Ch. 3].

A set \( S \subseteq \mathcal{H} \) is convex if, for any \( x, y \in S \),

\[
\lambda x + (1 - \lambda) y \in S \quad \text{for all } \lambda \in [0, 1].
\]

The class of Chebyshev sets for which the theory is richest is the class of closed convex sets. Moreover, for convex sets, the projector can be usefully characterised by a variational inequality.

**Proposition 1.2.2** (Characterisation of convex projections). Let \( C \subseteq \mathcal{H} \) be non-empty closed and convex. Then \( C \) is a Chebyshev. Moreover, \( p = P_C(x) \) if and only if \( p \in C \) and

\[
\langle x - p, c - p \rangle \leq 0 \quad \text{for all } c \in C.
\]

**Proof.** See, for instance, [45, Ex. 2.3.17].

One of the most famous open problems in approximation theory is the Chebyshev problem. The Chebyshev problem is concerned with a converse of Proposition 1.2.2. It asks: *Is every Chebyshev set convex?* In finite dimensions, the answer lies in the affirmative and is known as the Motzkin–Bunt theorem. In infinite dimensions, the problem remains open. According to Borwein [35], one of the cleanest partial answers is the following.

**Proposition 1.2.3** (Chebyshev sets in Hilbert space). Every non-empty closed convex subset of a Hilbert space is Chebyshev. Conversely, every weakly closed Chebyshev set is non-empty, closed and convex.

**Proof.** See, for instance, [45, Fact 4.5.3].
1.2. PROJECTORS AND REFLECTORS

(a) A convex set $C_1$ and its single-valued projectors and reflectors at $x$ and $y$. Note the nonexpansivity of $P_{C_1}$ and $R_{C_1}$.

(b) A non-convex set $C_2$ with a multi-valued projector and reflector at $x$ \((P_{C_2}x = \{p_1, p_2\} \text{ and } R_{C_2}x = \{r_1, r_2\})\).

Figure 1.1: Examples of projectors and reflectors.

For further commentary and discussion of the Chebyshev problem, the reader is referred to \([35, 66]\).

It is natural to ask how properties of the distance function correspond to properties of the projector, and vice versa. As the following remark explains, there exists a correspondence between differentiability and single-valuedness.

Remark 1.2.4 (Differentiability of distance functions and projectors). For a non-empty closed convex set $C \subseteq \mathcal{H}$, the distance to the set $C$ is a differentiable function on $\mathcal{H} \setminus C$ \([45, \text{Ex. 2.3.20}]\) with gradient given by

$$\nabla d(x, C) = \frac{1}{d(x, C)}(x - P_Cx), \text{ for all } x \notin C.$$  

If $C$ is merely closed, differentiability of $d(\cdot, C)$ on $U \setminus C$ for some open neighbourhood, $U$, of a point $x \in C$ is equivalent to single-valuedness of $P_C$ around $x$ \([126, \text{Th. 1.3}]\).

It is also worth noting that, by modifying a construction of Shapiro \([135]\), a recent paper of Akmal et al. \([2]\) gives an example of a convex set $C \subseteq \mathbb{R}^2$ with smooth boundary for which $P_C$ is not directionally differentiable. ♦

Although it fundamentally contains the same information, for our purposes it will sometimes be more instructive to consider an extrapolated version of the projector known as the reflector or the reflection operator. Alternatively the projector can also be considered as an “averaged” or interpolated reflector.

The reflector with respect to $S$ is the mapping $R_S: \mathcal{H} \rightrightarrows \mathcal{H}$ given by

$$R_S := 2P_S - I,$$

where $I: \mathcal{H} \to \mathcal{H}$ denotes the identity operator which maps any $x \in \mathcal{H}$ to itself.
An element of $R_S(x)$ is said to be a reflection of $x$ with respect to $S$, and as before, in an abuse of notation, we often write $R_S(x) = r$ when $R_S(x) = \{r\}$.

The follow proposition is the reflector’s companion of Proposition 1.2.2.

**Proposition 1.2.5** (Characterisation of convex reflections). Let $C \subseteq \mathcal{H}$ be non-empty closed and convex. Then $r = R_C(x)$ if and only if $\frac{1}{2}(x + r) \in C$ and

$$
(x - r, c - r) \leq \frac{1}{2} \|x - r\|^2 \text{ for all } c \in C.
$$

**Proof.** See [42, Fact 2.1(c)].

Projector and reflectors have a strongly geometric flavour, and it is therefore unsurprising that they satisfying many nice geometric properties.

**Proposition 1.2.6** (Translation and dilation formulae). Let $S \subseteq \mathcal{H}$ be a non-empty set. Then the following hold.

(a) $P_{y+S}(x) = y + P_S(x - y)$ for all $x,y \in \mathcal{H}$.

(b) $P_{\alpha S}(x) = \alpha P_S(x/\alpha)$ for all $x \in \mathcal{H}$, and $\alpha \in \mathbb{R} \setminus \{0\}$.

(c) $R_{y+S}(x) = y + R_S(x - y)$ for all $x,y \in \mathcal{H}$.

(d) $R_{\alpha S}(x) = \alpha R_S(x/\alpha)$ for all $x \in \mathcal{H}$, and $\alpha \in \mathbb{R} \setminus \{0\}$.

**Proof.** See, for instance, [66, 2.7].

### 1.3 The Feasibility Problem

Given a finite collection of constraint sets $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ with non-empty intersection, the feasibility problem is to

$$
\text{find } x \in C := \bigcap_{j=1}^{N} C_j \neq \emptyset. \tag{1.1}
$$

Whilst it is sometimes possible to solve (1.1) directly, this is usually not the case. Furthermore, even when a method for dealing with $C$ directly is known, it might be computationally intractable due to, for instance, the sheer size of the problem. The feasibility problem framework is an attempt to deal with these issues by exploiting the property that $C$ can be realised as the intersection of a collection of “simple” sets. In the context of this work, “simple” is meant in the sense that their projectors can be efficiently computed or approximated, although there are other useful possibilities [17,51,53].
1.3. THE FEASIBILITY PROBLEM

A great deal many problems can be cast within the language of feasibility problems. However, the appropriate formulation is not necessarily obvious. As with many types of mathematical modelling, finding the “right” formulation is part art. For the purpose of illustrating the framework, we give three motivating examples which shall be revisited in detail later in this thesis.

Example 1.3.1 (Moment Problems). Let $\mathcal{H}$ be a Hilbert lattice with lattice cone denoted $\mathcal{H}_+$. Let $A: \mathcal{H} \rightarrow \mathbb{R}^m$ be the linear operator defined by $A = (\langle a_j, \cdot \rangle)_{j=1}^m$ where $a_1, a_2, \ldots, a_m \in \mathcal{H}$ are linearly independent.

Consider the problem of finding an unknown point $x \in \mathcal{H}_+$ from a finite number of measurements represented by the vector $b := Ax \in \mathbb{R}^m$. In other words, we are interested in the so-called moment problem of reconstructing a nonnegative solution to the linear system specified by $A$ and $b$.

This problem can be formulated as the two-set feasibility problem with constraints

$$C_1 := A^{-1}b = \{x \in \mathcal{H} : Ax = b\}, \quad C_2 := \mathcal{H}_+.$$ 

Moreover, the two projectors can be readily computed [16, Ch. 5]. The projector onto $C_1$ is given by

$$P_{C_1} = I - A^* (AA^*)^{-1} (Ax - b),$$

(1.2)

where $(\cdot)^*$ denotes the adjoint of a linear operator, and $P_{C_2}$ amounts to taking the positive part with respect to the lattice.

When $m$ is large, it may not be efficient to compute using (1.2). In these cases, it is possible to consider a feasibility problem with more constraints by replacing $C_1 = \bigcap_{j=1}^m H_j$ where, for $j \in \{1, 2, \ldots, m\}$,

$$H_j := \{x \in \mathcal{H} : \langle a_j, x \rangle = b_j\}.$$ 

The disadvantage is this alternative feasibility problem has $(m + 1)$ constraint sets as compared to the original two-set problem. The advantage here is that the projectors onto the hyperplanes $H_j$ are given by

$$P_{H_j} x = x + \frac{b - \langle a_j, x \rangle}{\|a_j\|^2} a_j,$$

which requires only simple vector arithmetic.

Example 1.3.2 (Distance matrix reconstruction). Consider the set of real $m \times m$ symmetric matrices denoted

$$\mathcal{S}^m := \{X \in \mathbb{R}^{m \times m} : X^T = X\}.$$ 

The symmetric matrices form a Hilbert space when equipped with the inner product given by $\langle X, Y \rangle := \text{tr}(XY)$, where $\text{tr}(\cdot)$ denotes the trace of a matrix.
CHAPTER 1. INTRODUCTION

Suppose that we wish to reconstruct an incomplete matrix $D \in S^m$ which is known to contain the squares of the pairwise Euclidean distances between an unknown set of points $p_1, p_2, \ldots, p_m \in \mathbb{R}^s$. That is,

$$D_{ij} = \|p_i - p_j\|^2 \text{ for all } i, j \in \{1, 2, \ldots, m\}.$$ 

Problems of this kind arise, for example, when the conformation of a biomolecule is to be determined. In this setting, the unknown set of points represent the atoms within the molecule in $\mathbb{R}^s$ where $s := 3$, and the incomplete distance matrix has been obtained through a technique, such as NMR spectroscopy, which is typically not able to resolve all pairwise distances.

The key to modelling this problem within the feasibility framework is a theorem due to Hayden & Wells [83, Th. 3.3] which characterises Euclidean distances matrices in terms of positive semi-definiteness and a certain linear operator $T: S^m \to S^{m-1}$. Denoting the set of indices for which entry $D_{ij}$ is known by $\Omega \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$, the characterisations can be used to formulate a two sets feasibility problem with constraints

$$C_1 := \{X \in S^m : X \geq 0, X_{ij} = D_{ij} \text{ for all } (i, j) \in \Omega\},$$

$$C_2 := T^{-1}(\{Y \in S^{m-1}_+ : \text{rank } Y \leq s\}).$$

The set $C_2$ is the inverse image of a low-rank positive semi-definite matrix set, and is therefore non-convex. It is interesting that both the mathematically challenging, and the physically meaningful cases arise when $1 < s \ll m$.

In this feasibility problem, both projectors can be computed. The projector onto $C_1$ is given by

$$(P_{C_1} X)_{ij} = \begin{cases} 
D_{ij} & (i, j) \in \Omega \\
\max\{0, X_{ij}\} & (i, j) \notin \Omega 
\end{cases},$$

and $P_{C_2}$ amounts to thresholding eigenvalues after a spectral decomposition.

Example 1.3.3 (Combinatorial optimisation). In Chapter 4, we shall focus in detail on combinatorial optimisation problems. For the purpose of illustration, we give a classical example.

The 0-1 knapsack problem is the binary program

$$\min \{\langle c, x \rangle : x \in \{0, 1\}^m, \langle a, x \rangle \leq b\}, \quad (1.3)$$

for non-negative vectors $a, c \in \mathbb{R}_+^m$ and a non-negative real number $b \in \mathbb{R}_+$.

The 0-1 knapsack lower-bound feasibility problem is the decision problem which asks if there exists a feasible point of (1.3) such that the objective function value...
is greater than some prescribed $\lambda \in \mathbb{R}_+$. As we shall see in Chapters 3 and 4, this is closely related to the two set feasibility problem specified by

$$C_1 := \{x \in \{0, 1\}^m : \langle c, x \rangle \geq \lambda \}, \quad C_2 := \{x \in \mathbb{R}^m : \langle a, x \rangle \leq b\}.$$ 

As a decision problem the problem is NP-complete [147, I.5 Cor. 6.11]. Note that, in general, $P_{C_2}$ usually cannot be computed efficiently.

It is usually not known \textit{a priori} whether the constraint sets have non-empty intersection. The case of an empty intersection occurs, for instance, when the sets encode experimentally collected data which have been corrupted by noise or measurement errors. In these situations, there are at least two possible replacements to the \textit{consistent} feasibility problem defined in (1.1).

On one hand, we might like to detect the \textit{inconsistency} of (1.1). This is the case arising when the corresponding \textit{decision problem} is considered. On the other hand, if one insists on a “solution” to the inconsistent problem, then we are forced to look for an appropriate \textit{surrogate solution}. As an example, when $N = 2$, one possibility is the so called \textit{best approximation pair} which is a pair of points $(c_1, c_2) \in C_1 \times C_2$ such that

$$\|c_1 - c_2\| = d(C_1, C_2).$$

For $N > 2$ sets, the question of an appropriate surrogate is much more subtle [11], and shall be discussed in Section 2.6.

\section*{1.4 Product Space Reformulations}

When presented with a many-set feasibility problem but equipped with an algorithm which can only handle two-set instances, there are two possible resolutions. The first is to search for an appropriate many-set variant or extension of the two-set algorithm. This avenue of investigation is the main focus of Chapter 2. The second possibility entails reformulation as an equivalent two-set problem in a larger Hilbert space. The standard way of doing so is sometimes called the \textit{product space trick} and appears, for instance, in the work of Pierra [124, 125] for the case of convex sets. However, the reformulation also applies to non-convex sets, as we now explain.

Consider the $N$-fold Cartesian product

$$\mathcal{H}^N := \prod_{j=1}^N \mathcal{H} = \mathcal{H} \times \mathcal{H} \times \cdots \times \mathcal{H}.$$ 

Let $\mathbf{x} = (x_j)_{j=1}^N \in \mathcal{H}^N$ and $\mathbf{y} = (y_j)_{j=1}^N \in \mathcal{H}^N$. The set $\mathcal{H}^N$ is Hilbert space when
equipped with inner product
\[ \langle x, y \rangle := \sum_{j=1}^{N} \alpha_j \langle x_j, y_j \rangle, \quad (1.4) \]
where \( \alpha \in \mathbb{R}^{++} \) such that \( \sum_{j=1}^{N} \alpha_j = 1 \).

Given the finite family of constraint sets \( C_1, C_2, \ldots, C_N \subseteq \mathcal{H} \), we define the following two product space constraints
\[ C := \prod_{j=1}^{N} C_j := \{(x_j)_{j=1}^{N} \in \mathcal{H}^N : x_j \in C_j \text{ for } j = 1, 2, \ldots, N\}, \quad (1.5) \]
\[ D := \{(x_j)_{j=1}^{N} \in \mathcal{H}^N : x_1 = x_2 = \cdots = x_N\}. \]
The \( N \)-set feasibility problem (1.1), in \( \mathcal{H} \), is thus equivalent to the two-set feasibility problem, in \( \mathcal{H}^N \), specified by \( C \) and \( D \) in the sense that
\[ x \in \bigcap_{j=1}^{N} C_j \subseteq \mathcal{H} \iff (x, x, \ldots, x) \in C \cap D \subseteq \mathcal{H}^N. \quad (1.6) \]
Furthermore, the product space projectors (and hence reflectors) onto \( C \) and \( D \) can be readily computed whenever the same is true of the projectors onto the original constraints.

**Theorem 1.4.1** (Product space projectors). Let \( C_1, C_2, \ldots, C_N \subseteq \mathcal{H} \) be proximal sets, and let \( C \) and \( D \), as defined in (1.5), be subsets of the Hilbert space \( \mathcal{H}^N \) equipped with inner product given by (1.4). For any \( x = (x_j)_{j=1}^{N} \in \mathcal{H}^N \),
\[ P_C(x) = \prod_{j=1}^{N} P_{C_j}(x_j), \quad P_D(x) = \left( \sum_{i=1}^{N} \alpha_i x_i \right)^N_{j=1}. \]

**Proof.** See, for instance, [15, Sect. 2.6.3]. \( \square \)

The simplest and most used case of Theorem 1.4.1 is given by setting the weight vector \( \alpha := (1/N)_{j=1}^{N} \in \mathbb{R}_{++}^N \).

**Remark 1.4.2.** The product space has two potential drawbacks. Firstly, the dimension of the product Hilbert space in the reformulation is directly proportional to the number of constraints in the feasibility problem. Therefore, such a formulation can become computationally intractable when the number of constraints is large. Secondly, beyond the equivalence in (1.6), properties of the original constraints in \( \mathcal{H} \) need not transfer to the product constraints. For example, since the diagonal subspace, \( D \), is a closed subspace of codimension 1, it is always the case that \( \text{int}(C \cap D) = \emptyset \) even when \( \text{int} \left( \bigcap_{j=1}^{N} C_j \right) \neq \emptyset. \) \( \Diamond \)
1.5 Fundamental Algorithms

We now turn our attention to algorithms based on projectors and reflectors which can be used for solving the feasibility problem (1.1). In the literature, these methods are called projection methods. The term reflection methods is sometimes also used to refer to those methods which can be more naturally interpreted in terms of reflection operations, although the distinction is somewhat arbitrary.

In their most general form, many of these methods have interpretations as set-valued fixed point iterations. In order words, the algorithms can be described by the iteration of a set-valued operator followed by a selection. The hope here is that such a sequence converges to a fixed point of the operator, and indeed much of the literature is concerned with providing sufficient conditions for this to occur.

In this section, we give a brief introduction to two cornerstone algorithms from the projection and reflection method family which are at the heart of this work. We delay a discussion of their origins and history until Chapter 2.

The first algorithm which we consider is the method of alternating projections which can be applied to two-set feasibility problems. Given an initial point \( x_0 \in \mathcal{H} \), it generates a sequence \( (x_n) \) as follows:

\[
x_{n+1} \in P_{C_2}P_{C_1}(x_n).
\]

This method has an obvious extension to the many-set feasibility problem known as the method of cyclic projections. It generates a sequence \( (x_n) \) as follows:

\[
x_{n+1} \in P_{C_N} \ldots P_{C_2}P_{C_1}(x_n).
\]

That is, the algorithm is based on the operator \( (P_{C_N} \ldots P_{C_2}P_{C_1}) \). On the other hand, the feasibility problem constraint sets are precisely the fixed point sets of the individual projectors (i.e., \( \text{Fix } P_{C_j} = C_j \)). The relationship between the fixed point set of the composition of projectors, and the fixed point sets of the individual projectors is therefore important. As we shall see in Section 2.2, for closed convex sets having non-empty intersection the answer is most satisfactory. In fact, we have

\[
\text{Fix}(P_{C_N} \ldots P_{C_2}P_{C_1}) = \bigcap_{j=1}^{N} \text{Fix } P_{C_j} = \bigcap_{j=1}^{N} C_j.
\]

The second method which we shall consider is the Douglas–Rachford method. It can be applied to the two-set feasibility problem. Given an initial point \( x_0 \in \mathcal{H} \), it generates a sequence \( (x_n) \) as follows:

\[
x_{n+1} \in T_{C_1,C_2}(x_n) \text{ where } T_{C_1,C_2} := \frac{I + R_{C_2}R_{C_1}}{2}.
\]
Notice that unlike the method of alternating projection, the Douglas–Rachford method does not necessarily generate a sequence of points contained in the individual constraint sets. Consequently, it is not always possible to directly use the structure of the constraint sets, and is thus more difficult to analyse. Moreover, the sequence of interest is not the fixed point iterates themselves, but rather a sequence of their projections.

**Proposition 1.5.1** (Fixed points of $T_{C_1,C_2}$). Let $C_1, C_2 \subseteq \mathcal{H}$ be proximal sets. Then $x \in \text{Fix} T_{C_1,C_2}$ if and only if $P_{C_1}(x) \cap C_2 \neq \emptyset$.

*Proof.* Compute as follows: $x \in \text{Fix} T_{C_1,C_2} \iff x \in x + P_{C_2}(2p - x) - p$ for some $p \in P_{C_1}x \iff P_{C_1}(x) \cap C_2 \neq \emptyset$. The proof is now complete. \qed

Since it is not usually that case that the projector is weakly sequentially continuous, the sequence of projections can be difficult to deal with.

As we will see, extension of the Douglas–Rachford method to many-set feasibility problem is not so straightforward. The first such extension was proposed by Borwein and the present author [42] and is discussed in Chapter 2.
Chapter 2
Convex Theory

Most projection-type algorithms can be naturally extended to handle feasibility problems having more than two sets without significant modification of the iteration or recourse to a product space reformulation. For example, in Section 1.5, the method of alternating projections and, its natural extension, the method of cyclic projections were introduced. Despite having its origins in the work of Douglas and Rachford [68] from over half a century ago, it was only recently that a suitable many-set variant of the Douglas–Rachford method, the cyclic Douglas–Rachford method, was discovered by Borwein & Tam [42]. This variant and other possible extensions have since been investigated by Borwein & Tam [44], Bauschke et al. [30], Censor & Mansour [54] and Reich & Zalas [127].

This chapter introduces the cyclic Douglas–Rachford method, along with its variants, and investigates their behaviour in the presence of convexity. In this setting, these methods can be understood using the theory of nonexpansive maps.

2.1 Nonexpansive Maps and Their Fixed Points

Perhaps the simplest idea for finding a fixed point of a mapping is using its fixed point iteration. That is, starting from some initial point, generate a sequence by repeatedly applying the mapping, with the hope that a fixed point is attained in the limit. A classical result due to Banach [12] known as Banach’s contraction principle shows that, for contraction mappings, a sequence constructed in this way converges to the necessarily unique fixed point of the mapping.

**Theorem 2.1.1** (Banach’s contraction principle). Let \((X, d)\) be a non-empty complete metric space, and let \(T: X \to X\) be a contraction mapping (i.e., there exists \(\kappa \in [0, 1)\) such that \(d(Tx, Ty) \leq \kappa d(x, y)\) for all \(x, y \in X\)). Then \(T\) has exactly one fixed point \(x \in X\). Furthermore, for any \(x_0 \in X\) the sequence defined by \(x_{n+1} := Tx_n\), for all \(n \in \mathbb{N}\), converges to \(x\) with the \(R\)-linear rate \(\kappa\).
Described abstractly, most algorithms based on convex projectors and reflectors are fixed point iterations. Unfortunately, however, they typically do not satisfy the assumptions of Theorem 2.1.1. More precisely, the corresponding mappings are always nonexpansive but are rarely contractive nor are their fixed points unique. On the other hand, in general, nonexpansivity alone is not sufficient to ensure convergence.

The purpose of this section is to introduce nonexpansivity notions for mappings, explore the structure of their fixed point sets, and give conditions which guarantee convergence of the corresponding fixed point iterations. It is worth noting that there exists a vast literature concerned with these topics which we shall not touch on here. For further details, the interested reader is referred to [19, 31, 51, 99]. In this section we simply provide enough of the necessary background and tools needed for our subsequent analyses whilst, at the same time, giving enough of the broader picture to place the work in context.

We begin with three definitions. Let $D \subseteq \mathcal{H}$, and let $T: D \rightarrow \mathcal{H}$. The mapping $T$ is nonexpansive if

$$\|Tx - Ty\| \leq \|x - y\| \quad \forall x, y \in D.$$ 

The mapping $T$ is firmly nonexpansive if

$$\|Tx - Ty\|^2 + \|(I - T)x - (I - T)y\|^2 \leq \|x - y\|^2 \quad \forall x, y \in D.$$ 

Let $\alpha \in [0, 1]$. The mapping $T$ is $\alpha$-averaged if, there exists a nonexpansive mapping $R: D \rightarrow \mathcal{H}$ such that

$$T = (1 - \alpha)I + \alpha R.$$ 

From the definitions, it immediately follows that the families of firmly nonexpansive mappings and $\alpha$-averaged mappings are contained with the family of nonexpansive mappings. Furthermore, every $\alpha$-averaged mappings is a $\beta$-averaged mapping for any $\beta \in [\alpha, 1]$.

The following proposition provides various characterisations of firmly nonexpansive maps.

**Proposition 2.1.2** (Characterisations of firm nonexpansivity). Let $D \subseteq \mathcal{H}$ and let $T: D \rightarrow \mathcal{H}$. Then the following are equivalent.

(a) $T$ is firmly nonexpansive.
(b) $I - T$ is firmly nonexpansive.
(c) $2T - I$ is nonexpansive.
(d) $T$ is $1/2$-averaged.
(e) $\|Tx - Ty\|^2 \leq \langle Tx - Ty, x - y \rangle$ for all $x, y \in D$. 

2.1. NONEXPANSIVE MAPS AND THEIR FIXED POINTS

(f) \( 0 \leq \langle Tx - Ty, (I - T)x - (I - T)y \rangle \) for all \( x, y \in D \).

(g) \( \|Tx - Ty\| \leq \|\alpha(x - y) + (1 - \alpha)(Tx - Ty)\| \) for all \( x, y \in D \) and \( \alpha \in [0, 1] \).

Proof. See, for instance, [19, Prop. 4.2] and [19, Remark 4.24].

The following example shows that nonexpansive maps need not be averaged. A second graphical example is offered in Figure 2.1.

Example 2.1.3 (A nonexpansive map which is not averaged). The mapping \(-I\) is non-expansive but not \(\alpha\)-averaged for any \(\alpha \in ]0, 1]\), as can be seen directly from the definition. Moreover, if \(A := \{0\} \subseteq H\) and \(H \neq \{0\}\), then \(P_A = 0\) and hence \(R_A = -I\). That is, \(-I\) can be interpreted as a reflector.

\[\begin{proof}
\end{proof}\]

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.1}
\caption{A fixed point iteration starting at \(x \in \mathbb{R}^2\) for a nonexpansive map \(R\) (anti-clockwise rotation), and an averaged map of the form \(T = (1 - \alpha)I + \alpha R\).}
\end{figure}

We now introduction one further property. Let \(D \subseteq H\). The mapping \(T : D \to D\) is asymptotically regular at \(x \in D\) if

\[\lim_{n \to \infty} \|T^{n+1}x - T^nx\| = 0.\]

We say \(T\) is asymptotically regular if it is asymptotically regular at every \(x \in D\).

As the following proposition shows, every averaged map and hence, by Proposition 2.1.2, every firmly nonexpansive map is asymptotically regular. On the other hand, Example 2.1.5 shows a merely nonexpansive map need not be asymptotically regular.
Proposition 2.1.4 (Averaged maps are asymptotically regular). Let $D \subseteq H$, and let $T: D \to D$ be $\alpha$-averaged with $\alpha \in [0, 1]$. If $\text{Fix } T \neq \emptyset$ then $T$ is asymptotically regular.

Proof. See, for instance, [51, Th. 3.4.3].

Example 2.1.5 (A nonexpansive map which is not asymptotically regular). Recall from Example 2.1.3, that the reflector $R_A = -I$ where $A = \{0\}$ is nonexpansive. For any $n \in \mathbb{N}$ and $x \in H \setminus \{0\}$,

$$R_A^n x - R_A^{n+1} x = R_A^n (I - R_A) x = R_A^n (2x) = (-1)^n (2x) \not\to 0 \text{ as } n \to \infty.$$  

Thus $R_A$ is not asymptotically regular at $x$. 

Remark 2.1.6. Any composition or convex combination of averaged mappings is also an averaged map, as can be seen by a straightforward induction argument (see also Proposition 2.1.9). As a consequence of Proposition 2.1.4, it therefore follows that any composition or convex combination of averaged mapping with a common fixed point is also asymptotically regular. A recent result of Bauschke, Martín-Márquez, Moffat & Wang [27] shows that compositions and convex combinations of asymptotically regular firmly nonexpansive mappings are also asymptotically regular. Their result holds even when fixed point sets are empty.

We are now ready to state a fixed point theorem for nonexpansive maps due to Opial [118]. It yields a useful recipe for building iterative algorithms. More precisely, design an operator satisfying the assumptions of Theorem 2.1.7 with a fixed point set that can be used to generate solutions to the problem at hand.

Theorem 2.1.7 (Opial’s theorem). Let $C \subseteq H$ be nonempty closed and convex, and let $T: C \to C$ be nonexpansive and asymptotically regular with $\text{Fix } T \neq \emptyset$. Then, for any $x_0 \in C$, the sequence $T^n x_0 \overset{w}{\to} x \in \text{Fix } T$. Moreover, $x$ is the unique fixed point of $T$ satisfying

$$\lim_{n \to \infty} \|T^n x_0 - x\| = \inf_{z \in C} \lim_{n \to \infty} \|T^n x_0 - z\|.$$  

Proof. See, for instance, [118, Th. 1] or [51, Th. 3.5.1].

Remark 2.1.8. Opial’s Theorem remains true in any uniformly convex Banach space with weakly continuous duality map [118, Th. 2]. Other generalisations of the theorem, still in Hilbert space, can be found in Cegielski’s book [51, Ch. 3].

It is frequently the case that we would like to form new operators from existing ones. The two most useful ways of doing so involve taking convex combinations or compositions. Furthermore, whenever the original operators possess a given property, it is desirable for the derived operators to possesses the same property.
2.1. NONEXPANSIVE MAPS AND THEIR FIXED POINTS

It is straightforward to see that the family of nonexpansive maps is closed under the operations of convex combination and composition. The following propositions shows that averaged mappings behave similarly.

**Proposition 2.1.9** (Convex combinations and compositions of averaged maps). Let $D \subseteq H$ be a nonempty set, and suppose $T_j : D \to H$ is $\alpha_j$-averaged for each $j \in J := \{1, 2, \ldots, m\}$. The following two assertions hold.

(a) If $\alpha_j \in [0, 1]$, for each $j \in J$, then $\sum_{j \in J} \omega_j T_j$ is $\alpha$-averaged where $\alpha := \max_{j \in J} \alpha_j$, whenever $\omega \in \mathbb{R}_+^m$ and $\sum_{j \in J} \omega_j = 1$.

(b) If $\alpha_j \in [0, 1]$, for each $j \in J$, then $(T_m \ldots T_1)$ is $\alpha$-averaged where

$$\alpha := \frac{m}{m - 1 + \frac{1}{\max_{j \in J} \alpha_j}}.$$

*Proof.* See, for instance, [19, Prop. 4.30 & Prop. 4.32].

Combining Propositions 2.1.2 and 2.1.9 it follows that the class of firmly nonexpansive maps is closed under convex combinations. The same, however, is not true of compositions even when restricting ourselves to projectors in $\mathbb{R}^2$ as the following example due to Bauschke et al. [18, Ex. 4.2.5] shows.

**Example 2.1.10** (Firmly nonexpansive maps are not closed under composition). Consider the Hilbert space $H := \mathbb{R}^2$, and the constraint sets $C_1 := \mathbb{R} \times \{0\}$ and $C_2 := \mathbb{R}(1, 1)$. Then

$$P_{C_2} P_{C_1} (x, y) = P_{C_2} (x, 0) = (x, x)/2,$$

and

$$\langle P_{C_2} P_{C_1} (1, -2) - P_{C_2} P_{C_1} (0, 0), (1, -2) - (0, 0) \rangle = -1/2 < 0 \leq \| (2, -1) - (0, 0) \|^2.$$

By Proposition 2.1.2, the composition $P_{C_2} P_{C_1}$ is not firmly nonexpansive. ♦

We now turn our attention to the structure of the fixed point sets of nonexpansive operators.

**Proposition 2.1.11.** Let $D \subseteq H$ be a nonempty set, and let $T : D \to H$ be nonexpansive. Then $\text{Fix} T$ is closed and convex.

*Proof.* See, for instance, [51, Lem. 2.1.11].

For collections of averaged mappings possessing a common fixed point, the following two propositions apply and can be used, for instance, in combination with Opial’s theorem.
CHAPTER 2.  CONVEX THEORY

Proposition 2.1.12 (Fixed points of convex combinations of nonexpansive operators). Let $D \subseteq \mathcal{H}$ be a nonempty set, and let $\omega \in \mathbb{R}^m_{++}$ with $\sum_{j=1}^m \omega_j = 1$. Suppose $T_j : D \to \mathcal{H}$ is nonexpansive, for each $j \in J := \{1, 2, \ldots, m\}$. If $\cap_{j \in J} \text{Fix} T_j \neq \emptyset$ then $\text{Fix} \sum_{j \in J} \omega_j T_j = \cap_{j \in J} \text{Fix} T_j$.

Proof. See, for instance, [19, Prop. 4.34] or [51, Th. 2.1.14].

Proposition 2.1.13 (Fixed points of compositions of averaged operators). Let $D \subseteq \mathcal{H}$ be a nonempty set. Suppose $T_j : D \rightarrow D$ is $\alpha_j$-averaged for each $j \in \{1, 2, \ldots, m\}$ with $\alpha_j \in ]0, 1[$. If $\cap_{j=1}^m \text{Fix} T_j \neq \emptyset$ then $\text{Fix}(T_m \ldots T_2 T_1) = \cap_{j=1}^m \text{Fix} T_j$.

Proof. See, for instance, [19, Prop. 4.36].

Again, the conclusions of Proposition 2.1.13 need not hold if the operators involved are assumed to be merely nonexpansive.

Example 2.1.14 (Fixed points of compositions of nonexpansive operators). Recall the setting of Example 2.1.3. Set $T_1 = T_2 = R_A = -I$ where $A = \{0\}$. Then the composition $T_2 T_1 = I$, and thus $\text{Fix} T_1 \cap \text{Fix} T_2 = \{0\} \subset \mathcal{H} = \text{Fix}(T_2 T_1)$. \hfill \Box

When a finite family of averaged maps possesses no common fixed point, it is still possible that their composition may. The following theorem is a suitable refinement of Opial’s theorem which covers this case.

Theorem 2.1.15 (Weak convergence of averaged iterates). Let $D \subseteq \mathcal{H}$ be a nonempty closed convex set. For $j \in \{1, 2, \ldots, m\}$ suppose $T_j : D \to D$ is an $\lambda_j$-averaged mapping with $\lambda_j \in ]0, 1[$, and that $\text{Fix}(T_m \ldots T_1)$ is nonempty. For any $x_0 \in \mathcal{H}$ define

$$x_{n+1} := (T_m \ldots T_1)x_n.$$  

Then $x_n - (T_m \ldots T_1)x_n \to 0$, and there exists points

$$y_1 \in \text{Fix}(T_m \ldots T_1),$$

$$y_2 \in \text{Fix}(T_1 T_m \ldots T_2),$$

$$\vdots$$

$$y_m \in \text{Fix}(T_{m-1} \ldots T_1 T_m),$$

such that

$$x_n \overset{w}{\rightharpoonup} y_1 = T_m y_m,$$

$$T_1 x_n \overset{w}{\rightharpoonup} y_2 = T_1 y_1,$$

$$T_2 T_1 x_n \overset{w}{\rightharpoonup} y_3 = T_2 y_2,$$

$$\vdots$$

$$T_{m-2} \ldots T_1 x_n \overset{w}{\rightharpoonup} y_{m-1} = T_{m-2} y_{m-2},$$

$$T_{m-1} T_{m-2} \ldots T_1 x_n \overset{w}{\rightharpoonup} y_m = T_{m-1} y_{m-1}.$$
2.1. NONEXPANSIVE MAPS AND THEIR FIXED POINTS

Proof. See, for instance, [19, Th. 5.22].

Before turning our attention to the case in which the fixed point set of the operator is empty, we observe that fixed points of $T$ correspond to zeros of the operator $(I - T)$. In particular, we have that $\text{Fix } T \neq \emptyset$ implies $0 \in \text{range}(I - T)$. Conversely, whenever $0 \in \text{cl range}(I - T) \setminus \text{range}(I - T)$, it is possible to find a sequence of points such that the distance between the points and their images under $T$ becomes arbitrarily small (i.e., $T$ possesses an “approximate fixed point”).

The following trichotomy theorem due to Pazy [120] characterises asymptotic behaviour of nonexpansive fixed point iterations.

**Theorem 2.1.16** (Pazy’s trichotomy theorem). Let $D \subseteq \mathcal{H}$ be a nonempty closed convex set, $T: D \to D$ be nonexpansive, and let $x \in D$. Then the vector $v := \text{P}_{\text{cl range}(I - T)}(0)$ is well-defined and

$$
\lim_{n \to \infty} \frac{T^n x}{n} = -v.
$$

Moreover, exactly one of the following three alternatives holds.

(a) $0 \in \text{range}(I - T)$ and $(T^n y)_{n=1}^\infty$ is a bounded sequence for every $y \in D$.

(b) $0 \notin \text{cl range}(I - T)$ and $\|T^n y\|/n \to \alpha > 0$ for every $y \in D$.

(c) $0 \in \text{cl range}(I - T) \setminus \text{range}(I - T)$ and $\|T^n y\| \to +\infty$ for every $y \in D$.

**Proof.** See [120, Cor. 3 & Cor. 6].

**Remark 2.1.17.** Cases (b) and (c) of Theorem 2.1.16 are considered in further detail in a recent paper of Bauschke, Douglas and Moursi [23]. More specifically, they investigate cosmic convergence of the sequence $(T^n x)_{n=1}^\infty$ (i.e., convergence of the normalised sequence $(\frac{T^n x}{\|T^n x\|})_{n=1}^\infty$).

The following theorem gives a further refinement of the asymptotic behaviour when the nonexpansive map is averaged.

**Theorem 2.1.18** (Asymptotic behaviour of averaged iterates). Let $D \subseteq \mathcal{H}$ be a nonempty closed convex set, and let $T: D \to D$ be $\alpha$-averaged. Then, for any $x \in D$ and any $k \in \mathbb{N}$,

$$
\lim_{n \to \infty} \|T^{n+1} x - T^n x\| = \frac{1}{k} \lim_{n \to \infty} \|T^{n+k} x - T^n x\| = \lim_{n \to \infty} \frac{1}{n} \|T^n x\|,
$$

In particular, $T$ is asymptotically regular at $x$ whenever $(T^n x)$ is bounded.

**Proof.** See [10, Th. 2.1].
Remark 2.1.19. Whilst much of the general structure theory discussed in this section can be generalised to larger classes of operators having nonexpansive-type properties, the three classes of mappings considered will be sufficient for our purposes. Moreover, to ensure clarity and simplicity of exposition we avoid these further generalisations and variants, referring the interested reader to [19, 51] for further details.

2.2 The Cyclic Projection Method

The prototypical member of the projection method family is known as the cyclic projection method. As the name suggests, the method iterates by cyclic applications of the projection operator onto individual constraint sets. The method is generally credited to von Neumann [145] who studied the method applied to two closed subspaces, but has also been independently discovered at least a few times [146]. Von Neumann’s result was later extended, using a different approach, to cover the setting of finitely many subspaces by Halperin [81]. Applied to arbitrary convex sets, Bregman [48] showed that, whenever they exist, the method produces a sequence which weakly converges to solutions of the feasibility problem.

In this section, we present elements of the convergence theory of the cyclic projection method for the purpose of comparison with the new methods proposed in the subsequent sections. We focus mainly on the question of whether a sequence generated by the method converges. Of course, there is a vast literature dealing with questions of convergence rates and the like. For further details regarding the method, we refer the reader to surveys [52, 76] and the references therein.

We begin by exploring some properties of projectors onto convex sets.

Proposition 2.2.1 (Properties of convex projectors). Let $C_1, C_2, \ldots, C_m \subseteq \mathcal{H}$ be nonempty closed convex sets. Then the following hold.

(a) $P_{C_1}$ is firmly nonexpansive (or equivalently $1/2$-averaged).

(b) $(P_{C_m} \cdots P_{C_2} P_{C_1})$ is $m/(m + 1)$-averaged.

(c) If $\cap_{j=1}^m C_j \neq \emptyset$ then $\text{Fix}(P_{C_m} \cdots P_{C_2} P_{C_1}) = \cap_{j=1}^m C_j$.

(d) $\text{Fix}(P_{C_2} P_{C_1}) = \{x \in C_2 : d(x, C_1) = d(C_1, C_2)\}$ where

$$d(C_1, C_2) := \inf_{(c_1, c_2) \in C_1 \times C_2} \|c_1 - c_2\|.$$

Proof. (a): By Proposition 1.2.2, for any $x, y \in \mathcal{H},$

$$\langle P_{C_1} x - x, P_{C_1} x - P_{C_1} y \rangle \leq 0, \quad \langle y - P_{C_1} y, P_{C_1} x - P_{C_1} y \rangle \leq 0.$$
2.2. THE CYCLIC PROJECTION METHOD

Summing these two inequalities gives
\[ \|P_{C_1}x - P_{C_2}y\|^2 \leq \langle x - y, P_{C_1}x - P_{C_2}y \rangle, \]
and the result now follows from Proposition 2.1.2. (b): Combine (a) and Proposition 2.1.9. (c): Combine (a) and Proposition 2.1.13 noting that, for any set \( C \), the fixed point set of its projector is given by \( \text{Fix} P_C = C \). (d): Let \( x \in \text{Fix}(P_{C_2}P_{C_1}) \subseteq C_2 \) and \( y = P_{C_1}x \). Then \( x = P_{C_2}y \). By Proposition 1.2.2, for any \( c_1 \in C_1 \) and \( c_2 \in C_2 \),
\[ \langle y - x, y - c_1 \rangle \leq 0, \quad \langle y - x, c_2 - x \rangle \leq 0. \]
Summing these two inequalities gives
\[ \|y - x\|^2 \leq \langle y - x, c_1 - c_2 \rangle \leq \|x - y\| \|c_2 - c_1\|, \]
which proves inclusion in the forward direction. Conversely, let \( x \in C_2 \) with \( d(x, C_1) = d(C_1, C_2) \). Set \( y = P_{C_2}P_{C_1}x \in C_2 \). Since \((I - P_{C_2})x = 0 \) and \( P_{C_2} \) is firmly nonexpansive,
\[ \|P_{C_1}x - y\|^2 = \|(I - P_{C_2})P_{C_1}x - (I - P_{C_2})x\|^2 \leq \|P_{C_1}x - x\|^2. \]
Hence
\[ d(C_2, C_1) \leq \|y - P_{C_1}y\| \leq \|y - P_{C_1}x\| \leq \|x - P_{C_1}x\| = d(C_2, C_1). \]
Since the projection of \( P_{C_1}x \) onto \( C_2 \) is unique, it follows that \( y = x \). That is, \( x \in \text{Fix}(P_{C_2}P_{C_1}). \)
\[
\text{Remark 2.2.2. Proposition 2.2.1(d) is originally due to Cheney & Goldstein [57, Th. 2].} \]

We are now in a position to prove the following theorem which summarises the basic behaviour of the cyclic projection method. In a moment, we shall give a more complete, albeit more complex, characterisation. It will, however, be instructive to first give the following simpler version.

**Theorem 2.2.3 (Basic behaviour of the cyclic projection method).** Let the sets \( C_1, C_2, \ldots, C_m \subseteq H \) be nonempty closed and convex. Let \( x_0 \in H \) and set
\[ x_{n+1} := (P_{C_m} \ldots P_{C_2}P_{C_1})x_n. \]
Then one of the following alternatives holds.

(a) \( \text{Fix}(P_{C_m} \ldots P_{C_2}P_{C_1}) \neq \emptyset \) and \( x_n \xrightarrow{w} x \in \text{Fix}(P_{C_m} \ldots P_{C_2}P_{C_1}). \) In particular, \( x \in \bigcap_{j=1}^m C_j \) whenever \( \bigcap_{j=1}^m C_j \neq \emptyset \).
Lemma 2.2.4. Let \((P_{C_m} \ldots P_{C_2} P_{C_1})\) = \(\emptyset\) and \(\|x_n\| \to \infty\).

**Proof.** By Proposition 2.2.1(b), the mapping \((P_{C_m} \ldots P_{C_2} P_{C_1})\) is \(m/(m+1)\)-averaged. 

(a): If \(\text{Fix}(P_{C_m} \ldots P_{C_2} P_{C_1}) \neq \emptyset\) then by Proposition 2.1.4, \((P_{C_m} \ldots P_{C_2} P_{C_1})\) is asymptotically regular. By Theorem 2.1.7, the sequence \(x_n \xrightarrow{w} x \in \text{Fix}(P_{C_m} \ldots P_{C_2} P_{C_1})\).

In particular, if \(\cap_{i=1}^m C_i \neq \emptyset\) then, by Proposition 2.2.1(c), \(\text{Fix}(P_{C_m} \ldots P_{C_2} P_{C_1}) = \cap_{j=1}^m C_j\). (b): Follows from Theorem 2.1.16. 

In order to take a deeper look at Theorem 2.2.3(a) in the case that the target intersection is potentially empty, we first need to introduce some further notation. In this case, it is necessary to consider each of the individual projection operators individually rather than their entire composition.

Let \(C_1, C_2, \ldots, C_m \subseteq H\) denote nonempty closed convex sets, and let \(x_0 \in H\) be an initial point. The method of cyclic projections can be described as the iterative scheme given by

\[
\begin{align*}
    x^1_n &:= P_{C_1} x_0, \\
    x^{i+1}_n &:= P_{C_{i+1}} x^i_n, \\
    x^{m+1}_n &:= x^{m+1}_n, \\
\end{align*}
\]

for all \(n \in \mathbb{N}\) and all \(i \in I := \{1, 2, \ldots, m\}\), and the set \(C_{m+1} := C_1\). The sequences \((x^1_n)_{n=1}^\infty, (x^2_n)_{n=1}^\infty, \ldots, (x^m_n)_{n=1}^\infty\) are called the cyclic projection sequences.

For convenience, we use the operator \(Q_i : H \to C_i\) to denote

\[
Q_i := P_{C_i} P_{C_{i-1}} \ldots P_{C_{m+1}} P_{C_m} \ldots P_{C_{i+1}} \quad \forall i \in I.
\]

The cyclic projection sequence \((x^i_n)_{n=1}^\infty\) can thus be described by

\[
x^{i+1}_n = Q_i x^i_n \quad \forall n \in \mathbb{N}.
\]

Finally, suppose \(\text{Fix} Q_1 \neq \emptyset\) and let \(q^1 \in \text{Fix} Q_1\). Define the sequence \((q^i)_{i=1}^m\) by

\[
q^{i+1} := P_{C_{i+1}} q^i \in \text{Fix} Q_{i+1} \quad \forall i \in I \setminus \{m\}.
\]

The sequence of difference vectors, denoted \((d^i)_{i=1}^m\), is then defined by

\[
d^i := q^{i+1} - q^i \quad \forall i \in I,
\]

where \(q^{m+1} := q^1\). The sequence of difference vectors is well-defined in the sense that it is independent of the choice of the starting point \(q_1 \in \text{Fix} Q_1\), as observed in [18, 148]. Before proving this fact, we shall require the following lemma which can be found in [18, Prop. 2.1.3].

**Lemma 2.2.4.** Let \((x^i_n)_{n=1}^\infty\) (resp. \((y^i_n)_{n=1}^\infty\)) be a cyclic projection sequence, as defined in (2.1), starting at \(x_0 \in H\) (resp. \(y_0 \in H\)). For each \(i \in I := \{1, 2, \ldots, m\}\),

\[
\lim_{n \to \infty} [(x^i_n - y^i_n) - (x^{i+1}_n - y^{i+1}_n)] = 0.
\]
2.2. THE CYCLIC PROJECTION METHOD

Proof. For each $i \in I$, the projector $P_{C_i}$ is firmly nonexpansive, and thus

$$
\|x_n^i - y_n^i\|^2 - \|x_{n+1}^{i+1} - y_{n+1}^{i+1}\|^2 \geq \|(x_n^i - y_n^i) - (x_{n+1}^{i+1} - y_{n+1}^{i+1})\|^2 \quad \forall n \in \mathbb{N}.
$$

Summing over $i \in \{1, \ldots, m\}$ yields

$$
\|x_n^1 - y_n^1\|^2 - \|x_{n+1}^1 - y_{n+1}^1\|^2 \geq \sum_{i=1}^{m} \|(x_n^i - y_n^i) - (x_{n+1}^{i+1} - y_{n+1}^{i+1})\|^2 \quad \forall n \in \mathbb{N}.
$$

Summing over $n \in \{1, \ldots, k\}$ yields

$$
\|x_1^1 - y_1^1\|^2 - \|x_{k+1}^1 - y_{k+1}^1\|^2 \geq \sum_{n=1}^{k} \sum_{i=1}^{m} \|(x_n^i - y_n^i) - (x_{n+1}^{i+1} - y_{n+1}^{i+1})\|^2.
$$

The result now follows.

Proposition 2.2.5 (Difference vectors). The difference vectors $(d^i)^{m}_{i=1}$, as defined in (2.3), are independent of the choice of $q^1 \in \text{Fix } Q_1$ and satisfy $\sum_{i=1}^{m} d^i = 0$.

Proof. Let $q^1, \bar{q}^1 \in \text{Fix } Q_1$ be two distinct points. Let $(d^i)^{m}_{i=1}$ be the sequence of difference vectors obtained from the starting point $q^1$. Similarly, let $(\bar{d}^i)^{m}_{i=1}$ be the sequence of difference vectors obtained from the starting point $\bar{q}^1$. By Lemma 2.2.4, it follows that $d^i = \bar{d}^i$ for all $i \in I$.

We are now ready to state and prove our main convergence theorem for the method of cyclic projections which generalises Theorem 2.2.3. It can be found in [18, Th. 5.2.1].

Theorem 2.2.6 (Cyclic projection method dichotomy). Let $C_1, C_2, \ldots, C_m \subseteq \mathcal{H}$ be nonempty closed and convex sets. Let $(x_n^1)_{n=1}^{\infty}, \ldots, (x_n^m)_{n=1}^{\infty}$ be the cyclic projection sequences, as defined in (2.1), starting at $x_0 \in \mathcal{H}$. Exactly one of the following alternatives hold.

(a) For each $i \in I$, $\text{Fix } Q_i \neq \emptyset$ and the sequence $(x_n^i)_{n=1}^{\infty}$ weakly converges to a point $x^i$ with $x^{i+1} = P_{C_{i+1}} x^i$. Furthermore, $(x_n^{i+1} - x_n^{i})_{n=1}^{\infty}$ converges strongly to the $i^{th}$ difference vector $d^i$.

(b) For each $i \in I$, the set $\text{Fix } Q_i = \emptyset$ and $\|x_n^i\| \rightarrow +\infty$ as $n \rightarrow \infty$.

Proof. By Proposition 2.2.1(b), the operator $Q_i$ is $m/(m+1)$-averaged for each $i \in I$. By combining Theorems 2.1.15 and 2.1.16, we establish two possible cases: either (a) $\text{Fix } Q_i \neq \emptyset$ and $x_n^i \rightharpoonup x^i$ such that $x^{i+1} = P_{C_{i+1}} x^i$ for all $i \in I$, or (b) $\text{Fix } Q_i = \emptyset$ and $\|x_n^i\| \rightarrow +\infty$ for all $i \in I$. 
In case (a), the difference vectors are well-defined (Proposition 2.2.5) and hence Lemma 2.2.4 implies
\[ \lim_{n \to \infty} \left[ (x_{i+1}^n - x^n_i) - d^i \right] = \lim_{n \to \infty} \left[ (x_{i+1}^n - x^n_i) - (x_{i+1}^n - x^i) \right] = 0, \]
for each \( i \in I \). This completes the proof.

In the special case that the feasibility problem contains only two constraints, the situation can be simplified significantly using the notion of the displacement vector which describes the distance and relative positions between the two sets.

**Proposition 2.2.7** (The geometry of two sets). Let \( C_1, C_2 \subseteq \mathcal{H} \) be nonempty closed convex sets, and define the displacement vector \( v := P_{C_1} - C_2(0) \). Suppose that the set \( \text{Fix}(P_{C_1}P_{C_2}) \) (or equivalently the set \( \text{Fix}(P_{C_2}P_{C_1}) \)) is nonempty. Then

\[ F_1 := C_1 \cap (C_2 + v) = \text{Fix}(P_{C_1}P_{C_2}), \quad F_2 := C_2 \cap (C_1 - v) = \text{Fix}(P_{C_2}P_{C_1}), \tag{2.4} \]

and the difference vectors, defined in (2.2), are given by \( v = d^1 = -d^2 \).

**Proof.** Since \( C_1 \) and \( C_2 \) are non-empty closed convex sets, the closure of their Minkowski difference is also nonempty closed and convex. As a consequence, the displacement vector \( v := P_{C_1 - C_2}(0) \) is well-defined. Furthermore, \( \|v\| = d(C_1, C_2) \) and, by the characterisation of convex projectors, we deduce that, for any \( c \in C_1 - C_2 \),

\[ d^2(C_1, C_2) - \langle v, c \rangle = \|v\|^2 - \langle v, c \rangle = \langle 0 - v, c - v \rangle \leq 0. \tag{2.5} \]

We now prove the characterisation of \( F_1 \). The characterisation of \( F_2 \) is performed similarly. The forward inclusion follows immediately from Proposition 2.2.1.

Conversely, let \( f_1 \in \text{Fix}(P_{C_1}P_{C_2}) \subseteq C_1 \). Then \( f_1 - P_{C_2}f_1 \in C_1 - C_2 \) and, by Proposition 2.2.1, \( \|f_1 - P_{C_2}f_1\| = d(C_1, C_2) \). Together with (2.5), we therefore deduce

\[ \|v - (f_1 - P_{C_2}f_1)\|^2 = \|v\|^2 - 2\langle v, f_1 - P_{C_2}f_1 \rangle + \|f_1 - P_{C_2}f_1\|^2 = 2(d^2(C_1, C_2) - \langle v, f_1 - P_{C_2}f_1 \rangle) \leq 0, \]

and hence we deduce that \( f_1 = P_{C_2}f_1 + v \in C_2 + v \). The proof is complete.

We therefore have the following important special case of Theorem 2.2.6.

**Corollary 2.2.8** (Alternating projection method dichotomy). Let \( C_1, C_2 \subseteq \mathcal{H} \) be nonempty closed and convex sets. Let \( (x^1_n)_{n=1}^\infty, (x^2_n)_{n=1}^\infty \) be the cyclic projection sequences, as defined in (2.1), starting at \( x_0 \in \mathcal{H} \). Exactly one of the following alternatives hold.
2.2. THE CYCLIC PROJECTION METHOD

(a) \( \text{Fix}(P_{C_2}P_{C_1}) \neq \emptyset \) (or equivalently \( \text{Fix}(P_{C_1}P_{C_2}) \neq \emptyset \)), \( x^1_{n} \overset{\omega}{\rightharpoonup} x^1 \in F_1 \) and \( x^2_{n} \overset{\omega}{\rightharpoonup} x^2 \in F_2 \) such that \( v = x^1 - x^2 \). Moreover, the sequence \( (x^1_{n} - x^2_{n})_{n=1}^{\infty} \) converges strongly to \( v \).

(b) \( \text{Fix}(P_{C_2}P_{C_1}) = \text{Fix}(P_{C_1}P_{C_2}) = \emptyset \) and \( \| x^1_{n} \| , \| x^2_{n} \| \to +\infty \) as \( n \to \infty \).

Proof. Follows from Proposition 2.2.7 and Theorem 2.2.6.

Remark 2.2.9 (Limit cycles of the cyclic projection method). The weak limit points \( (x^i)_{i=1}^{m} \) of the cyclic projection method are sometimes called a limit cycle. In this language, Corollary 2.2.8 shows that the limit cycles of the method of alternating projections are best approximation pairs relative to \( (C_1, C_2) \). That is, they are precisely the solutions of the minimisation problem given by

\[
\min_{c_1 \in C_1, c_2 \in C_2} \| c_1 - c_2 \|. \tag{2.6}
\]

This interpretation is of interest in applications, since, unbeknownst to the practitioner, it is possible that an otherwise consistent feasibility problem has been corrupted by noise or measurement errors. In these situations, best approximations pairs offer a suitable surrogate solution.

One might hope for a similar characterisation of the limit cycles for three or more sets. However, a surprising result of Baillon et al. [11, Cor. 2.2] shows that, in general, there exists no such variational characterisation.

Regarding the difference vectors, an open problem, known as the geometry conjecture, proposed by Bauschke, Borwein & Lewis [18, Conj. 5.1.6], conjectures that the difference vectors can be defined without reference to the fixed point sets \( \text{Fix} Q_i \), and thus still exist even when these fixed point sets are empty. Furthermore, they conjecture that

\[
\sum_{j=1}^{m} d_j = 0 \quad \text{and} \quad \text{Fix} Q_m = C_m \cap (C_{m-1} + d_{m-1}) \cap \cdots \cap (C_1 + d_1 + \cdots + d_{m-1});
\]

with corresponding formulae holding for \( \text{Fix} Q_{1}, \ldots, \text{Fix} Q_{m-1} \).

A long standing open question asked whether it is possible for the cyclic projection method to fail to converge in norm. This was answered in the affirmative by Hundal [93] using an explicit counter-example involving a hyperplane and a convex cone. Verification of the counter-example is elementary, in the sense that it uses no heavy machinery, albeit involved. We shall be satisfied with simply stating the example.

Example 2.2.10 (The Hundal counter-example). Let \( \mathcal{H} := \ell_2(\mathbb{N}) \) with orthonormal basis \( (e_n) \), and define a function \( v : [0, +\infty) \to \mathcal{H} \) by

\[
v(r) := \exp(-100r^3)e_1 + \cos((r - \lfloor r \rfloor)\pi/2) e_{\lfloor r \rfloor + 2} + \sin((r - \lfloor r \rfloor)\pi/2) e_{\lfloor r \rfloor + 3},
\]
where the floor function is given by $[r] := \max\{s \in \mathbb{N} : s \leq r\}$.

Consider two constraint sets

\[ C_1 := \{e_1\}^\perp = \{x \in \mathcal{H} : \langle e_1, x \rangle = 0\}, \quad C_2 := \text{cl cone } v(\mathbb{R}_+). \]

The intersection of these sets is $C_1 \cap C_2 = \{0\}$ [93, Lem. 3]. By Theorem 2.2.3, it is necessarily the case that $(P_{C_2}P_{C_1})^n x \overset{w}{\rightharpoonup} 0$, for any $x \in \mathcal{H}$. However, Hundal [93, Th. 1] shows if $x_0 := v(1)$ then

\[ \lim_{n \to \infty} \|(P_{C_2}P_{C_1})^n x_0\| > 0. \]

In other words, the cyclic projection method fails to converge in norm. 

Since Hundal’s original counter-example, a number of variants and simplifications have been published [14, 28, 100, 113]. In light of this example, it is natural to ask what conditions are sufficient to guarantee norm convergence. We provide some partial answers to this question in Section 4.1.

To conclude this section, we offer the following averaged variant of the method.

**Theorem 2.2.11** (Averaged projection method). Let $C_1, C_2, \ldots, C_m \subseteq \mathcal{H}$ be nonempty closed and convex. Let $\omega \in \mathbb{R}^m_{++}$ with $\sum_{j=1}^m \omega_j = 1$. Let $x_0 \in \mathcal{H}$ and set

\[ x_{n+1} := Tx_n \text{ where } T := \sum_{j=1}^m \omega_j P_{C_j}. \] (2.7)

Exactly one of the following alternatives hold.

(a) Fix $T \neq \emptyset$ and $x_n \overset{w}{\rightharpoonup} x \in \text{Fix } T$. In particular, $x \in \cap_{j=1}^m C_j$ whenever $\cap_{j=1}^m C_j \neq \emptyset$.

(b) Fix $T = \emptyset$ and $\|x_n\| \to +\infty$.

**Proof.** By Proposition 2.2.1(b), the mapping $T$ is $1/2$-averaged. (a): If $\text{Fix } T \neq \emptyset$ then, by Proposition 2.1.4, $T$ is asymptotically regular. By Theorem 2.1.7, $x_n \overset{w}{\rightharpoonup} x \in \text{Fix } T$. In particular, if $x \in \cap_{j=1}^m C_j$ then, by Proposition 2.1.13,

\[ \text{Fix } T = \cap_{j=1}^m \text{Fix } P_{C_j} = \cap_{j=1}^m C_j. \]

(b): Follows from Theorem 2.1.16. The proof is now complete. 

---

1Hundal’s paper [93] uses the half-space $\{x \in \mathcal{H} : \langle e_1, x \rangle \leq 0\}$ for the first set rather than its boundary hyperplane. However, the “substitution does not affect the iterates generated by the algorithm.”
Remark 2.2.12 (Parallel implementation of (2.7)). The averaged projection method of Theorem 2.2.11 lends itself to a parallel implementation in the following sense. Consider a computing architecture consisting of a single head node, and several cluster nodes. The head node sends a copy of the current iterate, $x_n$, to each cluster node who, in parallel, each compute the projectors, $P_{C_j}x_n$. Each cluster node sends the result back to the master node which performs the averaging step which only requires multiplication and additions, hence is relatively fast.

This scheme can be particularly useful when either the number of constraints or the time cost of computing the projectors is large, and the time of computing each of the projectors $P_{C_1}, \ldots, P_{C_m}$ is similar.

2.3 The Douglas–Rachford Feasibility Method

The Douglas–Rachford method is an alternative approach to solving feasibility problems. The method was considered by Peaceman, Douglas & Rachford [68,121] to solve problems arising in nonlinear heat flow. Stated in full generality in the convex setting, the Douglas–Rachford method is an operator splitting method which finds zeros in the sum of two maximally monotone operators (see Section 2.3.1). Weak convergence of the scheme was originally proven by Lions & Mercier [110]; a result which was later improved by Svaiter [138]. When necessary, we refer to the method as the Douglas–Rachford feasibility method to distinguish it from its operator splitting interpretation. The method is notoriously difficult to analyse partly because, unlike the alternating projection method, it need not generate a sequence contained in the respective constraint sets. Consequently it is more difficult to use the geometry of the constraint sets directly.

The Douglas–Rachford feasibility method can be described as the fixed point iteration of the 2-set Douglas–Rachford operator. Given two nonempty closed convex sets $C_1, C_2 \subseteq \mathcal{H}$, the 2-set Douglas–Rachford operator is the mapping denoted $T_{C_1,C_2} : \mathcal{H} \to \mathcal{H}$ given by

$$T_{C_1,C_2} := \frac{I + R_{C_2}R_{C_1}}{2} = I + P_{C_2}R_{C_1} - P_{C_1}. \quad (2.8)$$

Unlike the method of cyclic projections, the fixed point set of the 2-set Douglas–Rachford operator does not directly solve the feasibility problem, the following proposition shows a correspondence. To proceed, let $C \subseteq \mathcal{H}$ be a convex set. The (convex) normal cone to $C$ at a point $x \in C$ is the set

$$N_C(x) := \begin{cases} \{ \phi \in \mathcal{H} : \langle c - x, \phi \rangle \leq 0, \forall c \in C \} & x \in C, \\ \emptyset & x \notin C. \end{cases}$$
Proposition 2.3.1 (Properties of the 2-set Douglas–Rachford operator). Let \( C_1, C_2, \subseteq \mathcal{H} \) be nonempty closed convex sets. Then the following hold.

(a) \( R_{C_1} \) is nonexpansive.

(b) \( T_{C_1,C_2} \) is firmly nonexpansive (or equivalently 1/2-averaged).

(c) \( \text{Fix} T_{C_1,C_2} = C_1 \cap C_2 + N_{C_1-C_2}(0) \).

(d) \( P_{C_1} \text{Fix} T_{C_1,C_2} = C_1 \cap C_2 \).

Proof. (a): By Proposition 2.2.1, the operator \( P_{C_1} \) is 1/2-averaged. It therefore follows that \( 2P_{C_1} - I \) is nonexpansive. (b): Since \( R_{C_1} \) and \( R_{C_2} \) are nonexpansive, so is their composition \( R_{C_2}R_{C_1} \). We therefore see that operator \( T_{C_1,C_2} \) is 1/2-averaged, and hence firmly nonexpansive by Proposition 2.1.9. (c): See [20, Cor. 3.9]. (d): Since \( \text{Fix} T_{C_1,C_2} \supseteq C_1 \cap C_2 \) and \( P_{C_1}(C_1 \cap C_2) = C_1 \cap C_2 \), it follows that \( P_{C_1} \text{Fix} T_{C_1,C_2} \supseteq C_1 \cap C_2 \). The converse inclusion follows by specialising Proposition 1.5.1 to the single-valued case.

Remark 2.3.2. As the basis of an algorithm for solving feasibility problems, the Douglas–Rachford feasibility method is of interest for a few reasons. Firstly, the two-set iteration is asymmetric in the sense that interchanging the roles of the constraints \( C_1 \) and \( C_2 \) gives rise to a different algorithm. Secondly, the iteration does not solve the feasibility problem directly. Instead, the fixed point set of the underlying operator can be used to generate solutions through a further projection step.

The basic behaviour of the method can be summarised as follows.

Theorem 2.3.3 (Basic behaviour of the Douglas–Rachford method). Let \( C_1, C_2, \subseteq \mathcal{H} \) be non-empty closed convex sets. Let \( x_0 \in \mathcal{H} \) and set \( x_{n+1} := T_{C_1,C_2}x_n \). Then:

(a) If \( C_1 \cap C_2 \neq \emptyset \) then \( x_n \rightharpoonup x \in \text{Fix} T_{C_1,C_2} \) such that \( P_{C_1}x \in C_1 \cap C_2 \).

(b) If \( C_1 \cap C_2 = \emptyset \) then \( \|x_n\| \to +\infty \).

Proof. We note that the operator \( T_{C_1,C_2} \) is 1/2-averaged (Proposition 2.3.1). By Proposition 2.3.1(c), \( \text{Fix} T_{C_1,C_2} \neq \emptyset \) if and only if \( C_1 \cap C_2 \neq \emptyset \), thus establishing the dichotomy. (a): By Proposition 2.1.4, \( T_{C_1,C_2} \) is asymptotically regular. It then follows, by Theorem 2.1.7, that \( x_n \rightharpoonup x \in \text{Fix} T_{C_1,C_2} \) and, by Proposition 2.3.1(d), \( P_{C_1}x \in C_1 \cap C_2 \). (b): Follows from Theorem 2.1.16.

The following example shows that convex projectors, in general, need not be weakly continuous. Thus unless \( \mathcal{H} \) is finite dimensional, in which case the weak and strong topologies coincide, weak convergence of the shadow sequence \( (P_{C_1}x_n)_{n=1}^{\infty} \) cannot be deduced from Theorem 2.3.3 alone.
2.3. THE DOUGLAS–RACHFORD FEASIBILITY METHOD

Example 2.3.4 (A convex projector that is not weakly continuous). Let \( \mathcal{H} = \ell_2(\mathbb{N}) \) with orthonormal basis \((e_1)_n\). Let \( C := \{ x \in \mathcal{H} : \|x\| \leq 1 \} \), and let \( x_n := e_1 + e_n \) for all \( n \in \mathbb{N} \). Then \( x_n \rightharpoonup e_1 \) but \( P_Cx_n = x_n/\sqrt{2} \rightharpoonup e_1/\sqrt{2} \neq e_1 = P_Ce_1 \).

We make two further remarks regarding the Douglas–Rachford method in light of Theorem 2.3.3.

Remark 2.3.5 (Comparison with the cyclic projection algorithm). Using Proposition 2.3.1 we have observed that \( C_1 \cap C_2 = \emptyset \) if and only if \( \text{Fix} T_{C_1,C_2} = \emptyset \). As a consequence, the dichotomy established in Theorem 2.3.3 shows that the Douglas–Rachford method converges weakly if and only if the corresponding feasibility problem is consistent. In contrast, the cyclic projection method can still converge even when the intersection is empty (Theorem 2.2.6). This is possible because emptiness of the intersection does not imply emptiness of the underlying operator’s fixed point set.

Remark 2.3.6 (Failure of norm convergence). No explicit Hundal-type counterexample to norm convergence is known for the Douglas–Rachford method. However, the existence of such an example seems likely.

The following theorem is the main convergence theorem for the Douglas–Rachford method and represents an amalgam of Bauschke, Combettes and Luke [20] and Svaiter [138]. For convenience, we denote the sets

\[
F_1 := \{ c_1 \in C_1 : d(c_1, C_2) = d(C_1, C_2) \}, \quad F_2 := \{ c_2 \in C_2 : d(c_2, C_1) = d(C_1, C_2) \}.
\]

Theorem 2.3.7 (Behaviour of the Douglas–Rachford method). Let \( C_1, C_2 \subseteq \mathcal{H} \) be nonempty closed and convex, and set \( v := P_{C_1-C_2}(0) \). For any \( x_0 \in \mathcal{H} \), set \( x_{n+1} := T_{C_1,C_2}x_n \) for all \( n \in \mathbb{N} \). Then:

(a) \( x_n - x_{n+1} = P_{C_1}x_n - P_{C_2}R_{C_1}x_n \to v \) and \( P_{C_1}x_n - P_{C_2}P_{C_1}x_n \to v \).

(b) Exactly one of the following two alternatives holds.

(i) \( C_1 \cap C_2 \neq \emptyset \), \( x_n \rightharpoonup x \in \text{Fix} T_{C_1,C_2} \) and \( P_{C_1}x_n \rightharpoonup P_{C_1}x \in C_1 \cap C_2 \).

(ii) \( C_1 \cap C_2 = \emptyset \) and \( \|x_n\| \to \infty \).

(c) Exactly one of the following two alternatives holds.

(i) \( F_1 = \emptyset \) (or equivalently \( F_2 = \emptyset \)), \( \|P_{C_1}x_n\| \to \infty \), and \( \|P_{C_2}P_{C_1}x_n\| \to \infty \).

(ii) \( F_1 \neq \emptyset \) (or equivalently \( F_2 \neq \emptyset \)), the sequences \( (P_{C_1}x_n)_{n=1}^{\infty} \) and \( (P_{C_2}P_{C_1}x_n)_{n=1}^{\infty} \) are bounded, and their weak cluster points belong to \( F_1 \) and \( F_2 \), respectively. Moreover, the weak cluster points of

\[
((P_{C_1}x_n, P_{C_2}R_{C_1}x_n))_{n=1}^{\infty}, \quad ((P_{C_1}x_n, P_{C_2}P_{C_1}x_n))_{n=1}^{\infty}, \quad (2.9)
\]

are best approximation pairs relative to \((C_1, C_2)\).
Proof. Combine [20, Th. 3.13] and Theorem 2.3.12 specialised to the setting of Examples 2.3.10 and 2.3.11.

Remark 2.3.8 (A regularised iteration). Motivated by applications in diffraction imaging, Luke [111] proposed and analysed a variant of the Douglas–Rachford method which he called the relaxed averaged alternating reflections (RAAR). The RAAR algorithm generates a sequence by iterating an operator which is a strict convex combination of a projector and the two-set Douglas–Rachford operator.

2.3.1 Connections to Monotone Operator Splitting

The Douglas–Rachford method has a more general interpretation in the area of monotone operator theory as we now explain. An operator $A: \mathcal{H} \rightrightarrows \mathcal{H}$ is monotone if
\[
\langle x - y, u - v \rangle \geq 0 \quad \forall (x, u), (y, v) \in \text{gra} A.
\]

A maximal monotone operator is a monotone operator $A: \mathcal{H} \rightrightarrows \mathcal{H}$ such that there exists no monotone operator $B: \mathcal{H} \rightrightarrows \mathcal{H}$ such that $\text{gra} B$ properly contains $\text{gra} A$. In fact, a monotone operator always admits a (possibly non-unique) maximally monotone extension.

Theorem 2.3.9. Let $A: \mathcal{H} \rightrightarrows \mathcal{H}$ be monotone. Then there exists a maximally monotone extension of $A$. That is, there exists a maximally monotone operator $\tilde{A}: \mathcal{H} \rightrightarrows \mathcal{H}$ such that $\text{gra} \tilde{A} \subseteq \text{gra} A$.

Proof. The proof uses Zorn’s lemma to deduce the existence of $\tilde{A}$. For details, see [19, Th. 20.21].

Given two maximally monotone operators $A, B: \mathcal{H} \rightrightarrows \mathcal{H}$, a problem of central interest involves finding a zero of their sum. That is,
\[
\text{find } x \in \text{zer}(A + B) := \{ x \in \mathcal{H} : 0 \in Ax + Bx \}. \tag{2.10}
\]

Example 2.3.10 (Feasibility as a special case of (2.10)). Let $C_1, C_2 \subseteq \mathcal{H}$ be nonempty closed convex sets. Then the corresponding convex normal cones $N_{C_1}, N_{C_2}: \mathcal{H} \rightrightarrows \mathcal{H}$ are maximally monotone operators [19, Th. 20.40].

If a point $x \in C_1 \cap C_2$ then $0 \in N_{C_1}(x) \cap N_{C_2}(x)$, hence $x \in \text{zer}(N_{C_1} + N_{C_2})$. Conversely, if $x \in \text{zer}(N_{C_1} + N_{C_2})$ then there exists $v \in \mathcal{H}$ such that $v \in N_{C_1}(x) \cap (-N_{C_2}(x))$ hence $x \in C$. Altogether we have shown that
\[
x \in \text{zer}(N_{C_1} + N_{C_2}) \iff x \in C_1 \cap C_2.
\]

That is, the problem of finding a zero of the sum of two normal cones is equivalent to the two set feasibility problem.
2.3. THE DOUGLAS–RACHFORD FEASIBILITY METHOD

The Douglas–Rachford splitting method is the monotone operator generalisation of the Douglas–Rachford feasibility method. It has the same general structure but with the projectors onto the constraint sets replaced with resolvents of monotone operators. The resolvent of a set-valued map \( \mathcal{A} : \mathcal{H} \rightrightarrows \mathcal{H} \) is the set-valued map \( \mathcal{J}_\mathcal{A} : \mathcal{H} \rightrightarrows \mathcal{H} \) given by

\[ \mathcal{J}_\mathcal{A} := (I + \mathcal{A})^{-1}. \]

The reflected resolvent is the set-valued map \( \mathcal{R}_\mathcal{A} := 2\mathcal{J}_\mathcal{A} - I \).

Nonexpansive properties of the resolvent are related to monotonicity properties of the corresponding set-valued map. For instance, as a consequence of a celebrated theorem due to Minty [115], the resolvent mapping is single-valued and firmly nonexpansive with full domain precisely when \( \mathcal{A} \) is maximally monotone. The following gives a concrete example of this correspondence which is of interest in our context.

**Example 2.3.11 (Projectors as a special case of resolvents).** Let \( C \subseteq \mathcal{H} \) be a nonempty closed convex sets. Then

\[ \mathcal{J}_{NC}(x) = (I + NC)^{-1}(x) = \{ p \in \mathcal{H} : x - p \in NC(p) \} = P_C(x), \]

where the last equality follows from Proposition 1.2.2.

We now state a convergence theorem for the Douglas–Rachford splitting method, due to Svaiter [138]. As a consequence of Examples 2.3.10 and 2.3.11, we see that the feasibility method is precisely the splitting method applied to the normal cones of the convex constraint sets with \( \gamma = \lambda = 1 \).

**Theorem 2.3.12 (Douglas–Rachford splitting method).** Let \( A, B : \mathcal{H} \rightrightarrows \mathcal{H} \) be maximally monotone operators with \( \text{zer}(A + B) \neq \emptyset \). Let \( \gamma > 0 \) and \( \lambda \in (0, 2) \). Given \( x_0 \in \mathcal{H} \), for all \( n \in \mathbb{N} \), set

\[ \begin{cases} 
  y_n := J_{\gamma B}x_n, \\
  z_n := J_{\gamma A}(2y_n - x_n), \\
  x_{n+1} := x_n + \lambda(z_n - y_n),
\end{cases} \]

Then \( x_n \xrightarrow{\text{w}} x \in \text{Fix}(\mathcal{R}_{\gamma A}\mathcal{R}_{\gamma B}) \) such that \( J_{\gamma B}x \in \text{zer}(A + B) \). Moreover, \( y_n \xrightarrow{\text{w}} J_{\gamma B}x \) and \( \mathcal{R}_{\gamma A}\mathcal{R}_{\gamma B}x_n - x_n \to 0 \).

**Proof.** See [138].

**Remark 2.3.13.** A number of variants and extensions of the Douglas–Rachford splitting method have been studied in the literature including those which allow relaxations or the inclusion of an error term in each resolvent evaluation. For further details see [49, 59, 60, 71, 72] and references therein.
2.4 The Cyclic Douglas–Rachford Method

Most projection algorithms can be extended in various natural ways to handle convex feasibility problems involving more than two sets. Prior to the work contained in this thesis, an exception was the Douglas–Rachford method, for which only the theory of two-set feasibility problems had been successfully investigated. For applications involving greater than two sets, the standard approach was to use the equivalent two-set feasibility problem, posed in a product space, as discussed in Section 1.4. In practice, this reformulation has the disadvantage of requiring the computation to be performed in a larger dimensional space. This section introduces and studies the cyclic Douglas–Rachford method, as introduced by the author and Borwein [42], which can be applied to many-set convex feasibility problems without recourse to a product space reformulation.

We begin with an illustration of some of the difficulties involved in extending the Douglas–Rachford method to more than two sets.

Example 2.4.1 (A failed generalisation). Let \( C_1, C_2, C_3 \subseteq \mathcal{H} \) be nonempty closed and convex. As a first attempt to generalise the Douglas–Rachford method to three set problems, it is natural to consider the sequence generated by the operator

\[
T_{C_1,C_2,C_3} := I + R_{C_3}R_{C_2}R_{C_1}.
\]

It can be easily verified that this mapping is firmly nonexpansive and has at least one fixed point whenever \( C_1 \cap C_2 \cap C_3 \neq \emptyset \). Opial’s theorem (Theorem 2.1.7) can thus be applied to deduce weak convergence to a fixed point of the corresponding iteration.

What is not clear is how to use \( \text{Fix}T_{C_1,C_2,C_3} \) to generate a point in the intersection. The following example, due to Brailey Sims [137], shows that the simply projecting onto the constraint sets fails to produce a solution.

Let \( \mathcal{H} := \mathbb{R}^2 \) and define the three subspace constraints

\[
C_1 := \mathbb{R}(0,-1), \quad C_2 := \mathbb{R}(\sqrt{3}, 1), \quad C_3 := \mathbb{R}(-\sqrt{3}, 1),
\]

with intersection \( C_1 \cap C_2 \cap C_3 = \{0\} \). The point \( x = (-\sqrt{3}, -1) \) is a fixed point of \( T_{C_1,C_2,C_3} \) but none of \( P_{C_1}x, P_{C_2}x \) or \( P_{C_3}x \) are contained in the intersection (see Figure 2.2).

We instead propose the following variant which exploits cyclic structure. Let \( C_1, C_2, \ldots, C_m \subseteq \mathcal{H} \) be nonempty closed convex sets. The \( m \)-set cyclic Douglas–Rachford operator, \( T_{[C_1,C_2,\ldots,C_m]} : \mathcal{H} \rightarrow \mathcal{H} \), is the mapping

\[
T_{[C_1,C_2,\ldots,C_m]} := T_{C_m,C_1}T_{C_{m-1},C_m} \cdots T_{C_2,C_3}T_{C_1,C_2},
\]

where each \( T_{C_i,C_j} \) is a two-set Douglas–Rachford operator, as defined in (2.8).
2.4. THE CYCLIC DOUGLAS–RACHFORD METHOD

Given an initial point \( x_0 \in \mathcal{H} \), the cyclic Douglas–Rachford method generates a sequence \((x_n)_{n=1}^{\infty}\) by setting

\[
x_{n+1} = T_{[C_1 C_2 \ldots C_m]}(x_n) \quad \forall n \in \mathbb{N}.
\]

**Remark 2.4.2 (The two-set cyclic Douglas–Rachford operator).** For the case of two set feasibility problems, the cyclic Douglas–Rachford operator simplifies to

\[
T_{[C_1 C_2]} = T_{C_2, C_1} T_{C_1, C_2} = \left( I + \frac{R_{C_1} R_{C_2}}{2} \right) \left( I + \frac{R_{C_2} R_{C_1}}{2} \right).
\]

In this case, it does not coincide with the classic Douglas–Rachford scheme of Section 2.3. We discuss this further in Section 2.5.

Throughout the remainder of this section, it is convenient to take indices modulo \( m \). In other words, we define

\[
C_0 := C_m, \quad C_{m+1} := C_1, \quad T_{C_0, C_1} := T_{C_m, C_1}, \quad T_{C_m, C_{m+1}} := T_{C_m, C_1}.
\]

The following proposition is concerned with nonexpansivity properties of Douglas–Rachford operators.

**Proposition 2.4.3 (Compositions and combinations of Douglas–Rachford operators).** Let \( T_1, T_2, \ldots, T_m : \mathcal{H} \rightarrow \mathcal{H} \) be two-set Douglas–Rachford operators for non-empty closed convex sets. Then
(a) The composition \((T_m \ldots T_2 T_1)\) is \(m/(m + 1)\)-averaged.
(b) Any convex combination of \(T_1, T_2, \ldots, T_m\) is \(1/2\)-averaged.

Proof. By Proposition 2.3.1, any two-set Douglas–Rachford operator is \(1/2\)-averaged. The result now follows from Proposition 2.1.9.

We now turn our attention to the structure of the Douglas–Rachford operator fixed point sets.

Lemma 2.4.4 (Fixed point of cyclic Douglas–Rachford compositions). Let \(C_1, C_2, \ldots, C_m \subseteq \mathcal{H}\) be non-empty closed convex sets. For any \(x \in \cap_{j=1}^m \text{Fix} T_{C_j, C_{j+1}}\), we have

\[
P_{C_1} x = P_{C_2} x = \cdots = P_{C_m} x.
\]

In particular, for each \(j \in J := \{1, 2, \ldots, m\}\),

\[
P_{C_1} \bigcap_{j=1}^m \text{Fix} T_{C_j, C_{j+1}} = \cdots = P_{C_m} \bigcap_{j=1}^m \text{Fix} T_{C_j, C_{j+1}} = \bigcap_{j=1}^m C_j \neq \emptyset. \tag{2.11}
\]

Proof. Let \(x \in \cap_{j=1}^m \text{Fix} T_{C_j, C_{j+1}}\). By Proposition 2.3.1, it follows that \(P_{C_j} x \in C_j \cap C_{j+1} \subseteq C_{j+1}\) for all \(j \in J\), and hence by Proposition 1.2.2,

\[
\langle x - P_{C_j} x, P_{C_{j-1}} x - P_{C_j} x \rangle \leq 0.
\]

We now compute

\[
\frac{1}{2} \sum_{j=1}^m \|P_{C_j} x - P_{C_{j-1}} x\|^2 = \langle x, 0 \rangle + \frac{1}{2} \sum_{j=1}^m (\|P_{C_j} x\|^2 - 2\langle P_{C_j} x, P_{C_{j-1}} x \rangle + \|P_{C_{j-1}} x\|^2)
\]

\[
= \left\langle x, \sum_{j=1}^m (P_{C_{j-1}} x - P_{C_j} x) \right\rangle - \sum_{j=1}^m \langle P_{C_j} x, P_{C_{j-1}} x \rangle + \sum_{j=1}^m \|P_{C_j} x\|^2
\]

\[
= \sum_{j=1}^m \langle x - P_{C_j} x, P_{C_{j-1}} x - P_{C_j} x \rangle \leq 0.
\]

Therefore \(P_{C_j} x = P_{C_{j-1}} x\) for all \(j \in J\), and the result follows.

We are now ready to deduce the main results of Borwein & Tam [42], regarding convergence of the cyclic Douglas–Rachford in the consistent setting.

Theorem 2.4.5 (Cyclic Douglas–Rachford method). Let \(C_1, C_2, \ldots, C_m \subseteq \mathcal{H}\) be closed convex sets with a nonempty intersection. Let \(x_0 \in \mathcal{H}\) and set \(x_{n+1} = T_{[C_1, C_2, \ldots, C_m]} x_n\). Then \(x_n \rightharpoonup x\) such that \(P_{C_1} x = P_{C_2} x = \cdots = P_{C_m} x\). In particular, for each \(j \in \{1, 2, \ldots, m\}\), the point \(P_{C_j} x \in \bigcap_{i=1}^m C_i\).
2.4. THE CYCLIC DOUGLAS–RACHFORD METHOD

Proof. Since the two-set Douglas–Rachford operator is $1/2$-averaged (Proposition 2.3.1) and $\bigcap_{j=1}^m \text{Fix } T_{C_j, C_{j+1}} \supseteq \bigcap_{j=1}^m C_j \neq \emptyset$, using Proposition 2.1.13 we deduce

$$\text{Fix } T_{[C_1, C_2, \ldots, C_m]} = \bigcap_{j=1}^m \text{Fix } T_{C_j, C_{j+1}} \neq \emptyset.$$  

By Propositions 2.4.3 and 2.1.4, the operator $T_{[C_1, C_2, \ldots, C_m]}$ is $m/(m + 1)$-averaged and asymptotically regular. By Theorem 2.1.7, we deduce that the sequence $x_n \xrightarrow{w} x \in \bigcap_{j=1}^m \text{Fix } T_{C_j, C_{j+1}}$. By Lemma 2.4.4, it follows that $P_{C_1} x = \cdots = P_{C_m} x$, and the proof is complete. \hfill \Box

As it turns out, the cyclic Douglas–Rachford and the alternating projection methods can sometimes coincide. The following observation gives a relationship between Douglas–Rachford operators and projectors.

Proposition 2.4.6. Let $C_1, C_2 \subseteq \mathcal{H}$ be nonempty closed convex sets. Then the restrictions of $P_{C_2}$ and $T_{C_1, C_2}$ to $C_1$ coincide, i.e., $P_{C_2} x = T_{C_1, C_2} x$ for all $x \in C_1$.

Proof. Let $x \in C_1$. Then $T_{C_1, C_2} x = \frac{1}{2} (x + R_{C_2} R_{C_1} x) = \frac{1}{2} (x + R_{C_2} x) = P_{C_2} x$. \hfill \Box

The following counter-example is a modification of the Hundal Example (Example 2.2.10) for the cyclic Douglas–Rachford method, and shows that, in general, the method can fail to converge strongly.

Example 2.4.7 (Failure of norm convergence). Let the sets $C_1$ and $C_2$, and the function $v$ be defined as in Example 2.2.10. Suppose that the cyclic Douglas–Rachford method starting at $x_0 := P_{C_1} v(1) \in C_1$ generates a sequence $(x_n)$ which is strongly convergent to a point $x$. By continuity of convex projections and Theorem 2.4.5, it is necessarily the case that $P_{C_2} x_n \to P_{C_2} x = 0$.

By Proposition 2.4.6,

$$T_{[C_1, C_2]} x_0 = T_{C_2, C_1} T_{C_1, C_2} x_0 = T_{C_2, C_1} P_{C_2} x_0 = P_{C_1} P_{C_2} x_0 = P_{C_1} P_{C_2} P_{C_1} v(1).$$

Inductively, for $n = 1, 2, \ldots$, we have

$$P_{C_2} x_{n+1} = P_{C_2} T_{[C_1, C_2]}^n x_0 = P_{C_2} (P_{C_1} P_{C_2})^n P_{C_1} v(1) = (P_{C_2} P_{C_1})^n v(1).$$

This implies $(P_{C_2} P_{C_1})^{n+1} v(1) \to 0$. This contradicts Example 2.2.10, and thus we conclude that $(x_n) \xrightarrow{\infty} x$ does not converge strongly. \hfill \Box

We now present an averaged version of the cyclic Douglas–Rachford iteration. The variant is suitable for parallel implementation using the same scheme outlined in Remark 2.2.12.
Theorem 2.4.8 (Averaged Cyclic Douglas–Rachford method). Let $C_1, C_2, \ldots, C_m \subseteq \mathcal{H}$ be closed convex sets with a nonempty intersection, and let $\alpha \in \mathbb{R}_m^+$ with $\sum_{j=1}^m \alpha_j = 1$. For any $x_0 \in \mathcal{H}$ and set
\[
x_{n+1} := \left( \sum_{j=1}^m \alpha_j T_{C_i, C_{j+1}} \right)x_n.
\]
Then $x_n \rightharpoonup x$ such that $P_{C_1}x = P_{C_2}x = \cdots = P_{C_m}x$. In particular, for each $j \in \{1, 2, \ldots, m\}$, the point $P_{C_j}x \in \bigcap_{i=1}^m C_i$.

Proof. Since the two-set Douglas–Rachford operator is $1/2$-averaged (Proposition 2.3.1) and $\bigcap_{j=1}^m \text{Fix } T_{C_j, C_{j+1}} \supseteq \bigcap_{j=1}^m C_j \neq \emptyset$, as a consequence of Proposition 2.1.12 we deduce
\[
\text{Fix } \sum_{i=1}^m \alpha_i T_{C_i, C_{i+1}} = \bigcap_{j=1}^m \text{Fix } T_{C_j, C_{j+1}} \neq \emptyset.
\]
By Propositions 2.4.3 and 2.1.4 the operator $\sum_{j=1}^m \alpha_j T_{C_i, C_{j+1}}$ is $1/2$-averaged and asymptotically regular. By Theorem 2.1.7, the sequence $x_n \rightharpoonup x \in \bigcap_{j=1}^m \text{Fix } T_{C_j, C_{j+1}}$. By Lemma 2.4.4, it follows that $P_{C_1}x = \cdots = P_{C_m}x$, and the proof is complete. \qed

2.5 Other Douglas–Rachford Variants

Following the discovery of the cyclic Douglas–Rachford method, a related Douglas–Rachford variant was proposed by Bauschke, Noll & Phan [30]. In this section, we present their variant, known as the cyclically anchored Douglas–Rachford method. Its proof follows the strategy used in the previous section. Furthermore, we give an averaged variant of the method which was not explicitly mentioned in [30].

Given sets $C_1, C_2, \ldots, C_m \subseteq \mathcal{H}$, the cyclically $C_1$-anchored Douglas–Rachford operator is defined by
\[
T_{C_1; C_2 \ldots C_m} := T_{C_1, C_m} T_{C_1, C_{m-1}} \cdots T_{C_1, C_2}.
\]

The following lemma is concerned with the structure of this operator’s fixed point set.

Lemma 2.5.1 (Common fixed points of anchored Douglas–Rachford operators). Let $C_1, C_2, \ldots, C_m \subseteq \mathcal{H}$ be nonempty closed convex sets. For any $x \in \bigcap_{j=2}^m \text{Fix } T_{C_1, C_j}$, $P_{C_1}x \in \bigcap_{j=1}^m C_j$. Moreover,
\[
P_{C_1} \bigcap_{j=2}^m \text{Fix } T_{C_1, C_j} = \bigcap_{j=1}^m C_j \neq \emptyset.
\]
2.5. OTHER DOUGLAS–RACHFORD VARIANTS

Proof. Let \( x \in \cap_{j=2}^{m} \text{Fix} T_{C_1,C_j} \) be arbitrary. By Proposition 1.2.2, it follows that \( P_{C_j} x \in C_1 \cap C_j \) for all \( j \in \{2, \ldots, m\} \), and hence \( P_{C_j} x \in \cap_{j=1}^{m} C_j \). This proves the first claim. The second claim follows since \( \text{Fix} T_{C_1,C_j} \supseteq \cap_{j=1}^{m} C_j \).

We are now in a position to prove our main result concerning the behaviour of the cyclically anchored Douglas–Rachford algorithm which appears as part of [30, Th. 8].

Theorem 2.5.2 (Cyclically anchored Douglas–Rachford algorithm). Let \( C_1, C_2, \ldots, C_m \subseteq \mathcal{H} \) be closed convex sets with nonempty intersection. Let \( x_0 \in \mathcal{H} \), and set \( x_{n+1} := T_{C_1,[C_2 \ldots C_m]}(x_n) \) \( \forall n \in \mathbb{N} \).

Then \( x_n \rightharpoonup x \in \text{Fix} T_{C_1,[C_2 \ldots C_m]} \) such that \( P_{C_1} x \in \cap_{j=2}^{m} \text{Fix} T_{C_1,C_j} \).

Proof. Since the two-set Douglas–Rachford operator is 1/2-averaged (Proposition 2.3.1) and \( \cap_{j=2}^{m} \text{Fix} T_{C_1,C_j} \supseteq \cap_{j=1}^{m} C_j \neq \emptyset \), Proposition 2.1.12 implies

\[
\text{Fix} T_{C_1,[C_2 \ldots C_m]} = \bigcap_{j=2}^{m} \text{Fix} T_{C_1,C_j+1} \neq \emptyset.
\]

By Propositions 2.4.3 and 2.1.4 the operator \( T_{C_1,[C_2 \ldots C_m]} \) is \( (m-1)/m \)-averaged and asymptotically regular. By Theorem 2.1.7, the sequence \( x_n \rightharpoonup x \in \cap_{j=2}^{m} \text{Fix} T_{C_1,C_j} \) and, by Lemma 2.5.1, \( P_{C_1} x \in \cap_{j=1}^{m} C_j \).

The following averaged variant of Theorem 2.5.2 is again of interest because it is suited to a parallel implementation.

Theorem 2.5.3 (Averaged anchored Douglas–Rachford algorithm). Let \( C_1, C_2, \ldots, C_m \subseteq \mathcal{H} \) be closed convex sets with nonempty intersection, and let \( \alpha \in \mathbb{R}^{m-1} \) with \( \sum_{j=2}^{m} \alpha_j = 1 \). Let \( x_0 \in \mathcal{H} \) and define

\[
x_{n+1} := Tx_n \text{ where } T := \sum_{j=2}^{m} \alpha_j T_{C_1,C_j}.
\]

Then \( x_n \rightharpoonup x \in \text{Fix} T \) such that \( P_{C_1} x \in \cap_{j=1}^{m} C_j \).

Proof. Since the two-set Douglas–Rachford operator is 1/2-averaged (Proposition 2.3.1), and \( \cap_{j=2}^{m} \text{Fix} T_{C_1,C_j+1} \supseteq \cap_{j=1}^{m} C_j \neq \emptyset \), Proposition 2.1.12 implies

\[
\text{Fix} T = \bigcap_{j=2}^{m} \text{Fix} T_{C_1,C_j+1} \neq \emptyset.
\]

By Propositions 2.4.3 and 2.1.4 the operator \( T \) is 1/2-averaged and asymptotically regular. By Theorem 2.1.7, the sequence \( x_n \rightharpoonup x \in \cap_{j=2}^{m} \text{Fix} T_{C_1,C_j} \) and, by Lemma 2.5.1, \( P_{C_1} x \in \cap_{j=1}^{m} C_j \).
Remark 2.5.4. Both the cyclically anchored and averaged anchored Douglas–Rachford algorithms reduce the classical Douglas–Rachford method of Section 2.3 in the case of two sets whereas the cyclic Douglas–Rachford method does not (see Remark 2.4.2). Further algorithmic structures which apply a potentially different combination of Douglas–Rachford operators during each iteration have very recently been investigated in [54, 127]. These algorithmic structures can be found in the literature as the string averaging, block iterative, and modular string averaging procedures, amongst other names. The key, and previously unknown, observations which enable to the possibility of all these results are Lemmas 2.5.1 and 2.4.4. These lemmas provide a useful characterisations of the corresponding operators’ fixed point sets.

2.6 Finer Behaviour of the Cyclic D.–R. Method

In our analysis of the cyclic Douglas–Rachford algorithm in Section 2.4, it was always assumed that a common fixed point of underlying two-set Douglas–Rachford operators existed; an assumption that trivially holds whenever the feasibility problem is consistent. When a common fixed point exists, the fixed point set of the cyclic Douglas–Rachford operator could be expressed as the intersection of the fixed point sets of the corresponding two-set operators. This section, based on a second paper of Borwein & Tam [44], examines the behaviour of the cyclic Douglas–Rachford method in closer detail, extending and complementing the results in Section 2.4. In particular, we characterise the behaviour of the method in the important case in which the underlying feasibility problem is potentially inconsistent (Theorem 2.6.7).

As was the case, in Section 2.2, it is necessary to introduce some additional notation.

Let $C_1, C_2, \ldots, C_m \subseteq \mathcal{H}$ and, for all $j \in \{1, 2, \ldots, m\}$, let $\sigma_j$ denote the cyclic permutation of $C_1, C_2, \ldots, C_N$ starting with $C_j$. Under this notation

$$T_{[\sigma_1]} := T_{[C_1, C_2, \ldots, C_{m-1}, C_m]}, \quad T_{[\sigma_2]} := T_{[C_2, C_3, \ldots, C_m, C_1]}, \quad \text{and so on.}$$

Again, following Section 2.2, throughout this section we take indices modulo $m$. In other words, we define

$$\sigma_0 := \sigma_m, \quad \sigma_{m+1} := \sigma_1, \quad C_0 := C_m, \quad C_{m+1} := C_1.$$ 

Let $x_0 \in \mathcal{H}$ be an initial point. The cyclic Douglas–Rachford method can thus be described as

$$x_1^1 := x_0, \quad x_{n+1} := T_{C_j, C_{j+1}} x_n^j, \quad x_{n+1}^1 := x_{n+1}^{m+1}. \quad (2.12)$$
2.6. FINER BEHAVIOUR OF THE CYCLIC DOUGLAS–RACHFORD METHOD

The sequences \((x^1_n)_{n=1}^\infty, (x^2_n)_{n=1}^\infty, \ldots, (x^m_n)_{n=1}^\infty\) are called the cyclic Douglas–Rachford sequences.

Note that, for each \(j \in \{1, 2, \ldots, m\}\), the sequence \((x^j_n)_{n=1}^\infty\) may also be expressed
\[
x^j_{n+1} := T_{[\sigma_j]} x^j_n.
\]
(2.13)

Remark 2.6.1. By Proposition 2.4.6, \(T_{C_j, C_{j+1}}\) and \(P_{C_{j+1}}\) coincide on \(C_j\). Hence, if \(x_0 \in C_1\), the cyclic projection sequences of Section 2.2 starting at \(x_0\), and the cyclic Douglas–Rachford sequences (2.1) starting at \(x_0\) coincide. This fact will be exploited throughout this section.

We shall require some preparatory lemmas. The first lemma shows that the cyclic Douglas–Rachford method has similar asymptotic behaviour to the method of cyclic projections.

Lemma 2.6.2. Let \((x^j_n)_{n=1}^\infty\) (resp. \((y^j_n)_{n=1}^\infty\)) be a cyclic Douglas–Rachford sequence, as defined in (2.12), starting at \(x_0 \in H\) (resp. \(y_0 \in H\)). For each index \(j \in \mathbb{J} := \{1, 2, \ldots, m\}\),
\[
\lim_{n \to \infty} [(x^j_n - x^{j+1}_n) - (y^j_n - y^{j+1}_n)] = 0.
\]

Proof. Since \(T_{[\sigma_1]}\) is nonexpansive,
\[
\|x^{1}_{n+1} - y^{1}_{n+1}\| \leq \|x^{1}_n - y^{1}_n\|,
\]
and hence \(\lim_{n \to \infty} \|x^{1}_n - y^{1}_n\|\) exists. Since \(T_{C_j, C_{j+1}}\) is firmly nonexpansive, for each \(j \in \mathbb{J}\),
\[
\|x^j_n - y^j_n\|^2 - \|x^j_{n+1} - y^j_{n+1}\|^2 \geq \sum_{j=1}^{m} \|x^j_n - x^{j+1}_n\| - \|y^j_n - y^{j+1}_n\|^2.
\]
The result follows by taking the limit as \(n \to \infty\). \(\square\)

Lemma 2.6.3. Let \(J := \{1, 2, \ldots, m\}\), and let \((x^j_n)_{n=1}^\infty\) (resp. \((y^j_n)_{n=1}^\infty\)) be a cyclic Douglas–Rachford sequence, as defined in (2.12), starting at \(x_0 \in H\) (resp. \(y_0 := P_{C_1} x_0\)). Then the following four properties are equivalent.

(a) There exists an index \(j \in \mathbb{J}\) such that \((x^j_n)_{n=1}^\infty\) is bounded.
(b) The sequences \((x^1_n)_{n=1}^\infty, (x^2_n)_{n=1}^\infty, \ldots, (x^m_n)_{n=1}^\infty\) are bounded.
(c) There exists an index \(j \in \mathbb{J}\) such that \((y^j_n)_{n=1}^\infty\) is bounded.
(d) The sequences \((y^1_n)_{n=1}^\infty, (y^2_n)_{n=1}^\infty, \ldots, (y^m_n)_{n=1}^\infty\) are bounded.
Furthermore, if \( (x_j^i)_{i=1}^\infty \) is unbounded then \( \|x_j^i\| \to +\infty \).

Proof. Fix an index \( j \in J \). Note that the sequences \( (y_n^1)_{n=1}^\infty, \ldots, (y_n^m)_{n=1}^\infty \) are also cyclic projection sequences (see Remark 2.6.1). Since \( T_{\sigma_j} \) is nonexpansive,

\[
\|x_n^j - y_n^j\| \leq \|x_{n-1}^j - y_{n-1}^j\| \leq \cdots \leq \|x_1^j - y_1^j\|.
\]

By the triangle inequality

\[
\|x_n^j\| \leq \|x_1^j - y_1^j\| + \|y_n^j\| \quad \text{and} \quad \|y_n^j\| \leq \|x_n^j\| + \|x_1^j - y_1^j\|.
\]

Thus \( (x_n^j)_{n=1}^\infty \) is bounded if and only if \( (y_n^j)_{n=1}^\infty \) is bounded. By Theorem 2.2.6, \( (y_n^j)_{n=1}^\infty \) is bounded if and only if \( (y_n^1)_{n=1}^\infty, (y_n^2)_{n=1}^\infty, \ldots, (y_n^m)_{n=1}^\infty \) are bounded, and if \( (y_n^j)_{n=1}^\infty \) is unbounded then \( \|x_n^j\| \to +\infty \). The result follows by combining these two statements.

Lemma 2.6.4. Let \( J := \{1,2,\ldots,m\} \), and let \( (x_j^i)_{i=1}^\infty \) be a cyclic Douglas–Rachford sequence, as defined in (2.12), starting at \( x_0 \in H \). For all \( j \in J \) and for all \( n \in \mathbb{N} \),

\[
\|x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}\|^2 \leq \langle x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}, x_n^j - P_{C_{j}}x_n^j \rangle. \tag{2.14}
\]

In particular, for all \( j \in J \) and all \( n \in \mathbb{N} \),

\[
\|x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}\| \leq \|x_n^j - P_{C_{j}}x_n^j\|. \tag{2.15}
\]

Proof. For each \( j \in J \), we observe

\[
x_n^{j+1} := T_{C_j}x_n^j = x_n^j + P_{C_{j+1}}R_{C_j}x_n^j - P_{C_j}x_n^j. \tag{2.16}
\]

By (2.16) and the variational characterization of convex projections,

\[
\|x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}\|^2 - \langle x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}, x_n^j - P_{C_{j}}x_n^j \rangle
\]

\[
= \langle x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}, (x_n^{j+1} - x_n^j + P_{C_{j}}x_n^j) - P_{C_{j+1}}x_n^{j+1} \rangle - \langle x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}, P_{C_{j+1}}R_{C_j}x_n^j - P_{C_{j+1}}x_n^{j+1} \rangle \leq 0,
\]

which proves (2.14). Equation (2.15) follows by an application of the Cauchy–Schwarz inequality.

Lemma 2.6.5. Let \( J := \{1,2,\ldots,m\} \), and let \( (x_n^i)_{n=1}^\infty \) be a cyclic Douglas–Rachford sequence, as defined in (2.12), starting at \( x_0 \in H \). For all \( m \in \mathbb{N} \),

\[
\sum_{j=1}^{m} \sum_{n=2}^{k} \|x_n^{j+1} - P_{C_{j+1}}x_n^{j+1} - P_{C_{j}}x_n^j \|^2
\]

\[
\leq \langle x_2^1 - P_{C_1}x_2^1, x_1^m - P_{C_m}x_1^m \rangle - \langle x_{k+1}^1 - P_{C_k}x_{k+1}^1, x_k^m - P_{C_m}x_k^m \rangle. \tag{2.17}
\]
where \((x_2^1 - P_{C_1}x_2^1, x_1^m - P_{C_m}x_k^m)\) and \((x_{k+1}^1 - P_{C_1}x_{k+1}^1, x_{k}^m - P_{C_m}x_k^m)\) are nonnegative. In particular, the double-sum in (2.17) is bounded, and hence, for each \(j \in \mathbb{J}\), as \(n \to \infty\),
\[
(x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}) - (x_n^j - P_{C_j}x_n^j) \to 0.
\]

Proof. Applying Lemma 2.6.4,
\[
\sum_{n=2}^{k} \sum_{j=1}^{m} \| (x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}) - (x_n^j - P_{C_j}x_n^j) \|^2
\]
\[
= \sum_{n=2}^{k} \sum_{j=1}^{m} \left( \|x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}\|^2 - 2\langle x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}, x_n^j - P_{C_j}x_n^j \rangle + \|x_n^j - P_{C_j}x_n^j\|^2 \right)
\]
\[
\leq \sum_{n=2}^{k} \sum_{j=1}^{m} \left( \langle x_n^j - P_{C_j}x_n^j, x_n^{j-1} - P_{C_j}x_n^{j-1} \rangle - \langle x_n^{j+1} - P_{C_{j+1}}x_n^{j+1}, x_n^j - P_{C_j}x_n^j \rangle \right).
\]

The nonnegativity of \(\langle x_2^1 - P_{C_1}x_2^1, x_1^m - P_{C_m}x_k^m \rangle\) and \(\langle x_{k+1}^1 - P_{C_1}x_{k+1}^1, x_{k}^m - P_{C_m}x_k^m \rangle\) is a consequence of (2.14).

We now prove the analogue of Lemma 2.6.4 for the limits (if they exist) of the cyclic Douglas–Rachford sequences.

Lemma 2.6.6. Let \(\mathbb{J} := \{1, 2, \ldots, m\}\), and suppose \((x^j)_{j=1}^{m}\) are points such that \(x^{j+1} = T_{C_j}C_{j+1}x^j\) for all \(j \in \mathbb{J}\). Then, for all \(j \in \mathbb{J}\),
\[
P_{C_{j+1}}R_{C_j}x^j - P_{C_{j+1}}x^{j+1} = (x^{j+1} - x^j) - (P_{C_{j+1}}x^{j+1} - P_{C_j}x^j) = 0.
\]

Proof. Consider the cyclic Douglas–Rachford sequences for initial point \(x_0 := x^1\), note that
\[
P_{C_{j+1}}R_{C_j}x^j = x^{j+1} - x^j + P_{C_j}x^j.
\]

Since \(x^j \in \text{Fix} T_{[\sigma_j]}\), for each \(j \in \mathbb{J}\), the result follows from (2.18) and Lemma 2.6.5.

We are now ready to prove a dichotomy theorem which is the cyclic Douglas–Rachford method analogue of Theorem 2.2.6.

Theorem 2.6.7 (Cyclic Douglas–Rachford dichotomy). Let \(\mathbb{J} := \{1, 2, \ldots, m\}\), and let \((x_n^j)_{n=1}^{\infty}\) be a cyclic Douglas–Rachford sequence, as defined in (2.12), starting at \(x_0 \in \mathcal{H}\). The following holds.
(a) For each \( j \in \mathcal{J} \),
\[
P_{C_{j+1}}R_{C_{j}}x_{n}^{j} - P_{C_{j+1}}x_{n}^{j} = (x_{n}^{j+1} - x_{n}^{j}) - (P_{C_{j+1}}x_{n}^{j+1} - P_{C_{j}}x_{n}^{j}) \to 0.
\]

(b) Exactly one of the following alternatives hold.

(i) Each \( \text{Fix} T_{[\sigma_{j}]} \) is empty. Then \( \|x_{n}^{j}\| \to +\infty \), for all \( j \in \mathcal{J} \).

(ii) Each \( \text{Fix} T_{[\sigma_{j}]} \) is nonempty. Then, for each \( j \in \mathcal{J} \),
\[
x_{n}^{j} \leftarrow x_{n}^{j} \in \text{Fix} T_{[\sigma_{j}]} \text{ with } x_{n}^{j+1} = T_{C_{j}, C_{j+1}}x_{n}^{j}.
\]

Furthermore, for each \( j \in \mathcal{J} \),
\[
x_{n}^{j+1} - x_{n}^{j} = P_{C_{j+1}}R_{C_{j}}x_{n}^{j} - P_{C_{j+1}}x_{n}^{j} \to d^{j}, \quad P_{C_{j+1}}x_{n}^{j+1} - P_{C_{j}}x_{n}^{j} \to d^{j},
\]
\[
x_{n}^{j+1} - x_{n}^{j} = P_{C_{j+1}}x_{n}^{j+1} - P_{C_{j}}x_{n}^{j} = d^{j}, \quad P_{C_{j+1}}R_{C_{j}}x_{n}^{j} = P_{C_{j+1}}x_{n}^{j+1},
\]
where \((d^{j})_{j=1}^{m}\) are the difference vectors, as defined in (2.2).

Proof. (a): Follows from Lemma 2.6.5. (b): By appealing to Theorem 2.1.16 and Lemma 2.6.3, we establish the two possible alternatives: either \( \text{Fix} T_{[\sigma_{j}]} = \emptyset \) and \( \|x_{n}^{j}\| \to +\infty \) for all \( j \in \mathcal{J} \), or \( \text{Fix} T_{[\sigma_{j}]} \neq \emptyset \) for all \( j \in \mathcal{J} \).

If each \( \text{Fix} T_{[\sigma_{j}]} \neq \emptyset \), Proposition 2.4.3 together with Theorem 2.1.15 imply that the sequence \((x_{n}^{j})_{n=1}^{\infty}\) converges weakly to a point \( x^{j} \in \text{Fix} T_{[\sigma_{j}]} \) with \( x_{n}^{j+1} = T_{C_{j}, C_{j+1}}x^{j} \). Lemma 2.6.2 with Theorem 2.2.6 implies \( x_{n}^{j+1} - x_{n}^{j} \to d^{j} \), which together with (a) yields \( P_{C_{j+1}}x_{n}^{j+1} - P_{C_{j}}x_{n}^{j} \to d^{j} \).

Lemma 2.6.6 together with Theorem 2.2.6 applied to the cyclic Douglas–Rachford sequences having initial point \( x^{1} \) yields \( x_{n}^{j+1} - x^{j} = P_{C_{j+1}}x_{n}^{j+1} - P_{C_{j}}x^{j} = d^{j} \) and \( P_{C_{j+1}}R_{C_{j}}x_{n}^{j} = P_{C_{j+1}}x_{n}^{j+1} \). \(\quad\Box\)

If \( \cap_{i=1}^{m}C_{i} \neq \emptyset \), it can be shown that the limits \((x^{j})_{j=1}^{m}\) coincide (see, for example, [42, Lem. 2.3]). In this case, we obtain [42, Th. 3.1] as a special case of Theorem 2.6.7. That is, we have the following remark.

Remark 2.6.8. The proof of Theorem 2.4.5 for the consistent case given in Section 2.4 depended on the fact that \( \text{Fix} T_{[\sigma_{1}]} = \cap_{j=1}^{m} \text{Fix} T_{C_{j}, C_{j+1}} \neq \emptyset \). Since \( \text{Fix} T_{C_{j}, C_{j+1}} \neq \emptyset \) if and only if \( C_{j} \cap C_{j+1} \neq \emptyset \), in the inconsistent case one can only guarantee that \( \text{Fix} T_{[\sigma_{1}]} \supseteq \cap_{j=1}^{m} \text{Fix} T_{C_{j}, C_{j+1}} = \emptyset \). \(\quad\Diamond\)

Remark 2.6.9 (Approximating the difference vectors). Theorem 2.6.7 shows that the sequences
\[
(x_{n}^{j+1} - x_{n}^{j})_{n=1}^{\infty}, \quad (P_{C_{j+1}}R_{C_{j}}x_{n}^{j} - P_{C_{j}}x_{n}^{j})_{n=1}^{\infty}, \quad (P_{C_{j+1}}x_{n}^{j+1} - P_{C_{j}}x_{n}^{j})_{n=1}^{\infty},
\]
converge (in norm) to \( d^{j} \). The latter two are suitable for approximating \( d^{j} \) using a pair of points from \( C_{j} \) and \( C_{j+1} \). \(\quad\Diamond\)
2.6. FINER BEHAVIOUR OF THE CYCLIC DOUGLAS–RACHFORD METHOD

Remark 2.6.10 (Comments on the cyclic Douglas–Rachford method). Theorem 2.6.7 reveals that the behaviour of the cyclic Douglas–Rachford scheme has similarities with both the cyclic projection method and the classical Douglas–Rachford scheme. In this sense, it can be considered a compromise between the two schemes having some of the desirable (as well as undesirable) properties of both.

The cyclic Douglas–Rachford and classical Douglas–Rachford schemes both perform the reflections with respect to the constraint sets, rather than using just a projection as is the case of the method of cyclic projections. In some situations, this can be advantageous. For instance, in the case of Figure 2.3, the reflector produces a point in the interior of the set where as the projector would only produce a boundary point.

![Figure 2.3: \( R_Cx \) is contained in the interior of \( C \), while \( P_Cx \) is on the boundary of \( C \).](image)

Both the cyclic Douglas–Rachford and classical Douglas–Rachford iterations proceed by applying a two set Douglas–Rachford mapping (i.e., one of the form \( T_{C_j,C_{j+1}} \)). In the consistent case, the limit obtained by both these schemes, once projected onto an appropriate constraint set, produces a solution to a feasibility problem. In the inconsistent case, the Douglas–Rachford scheme always produces an unbounded sequence. The behaviour described in Theorem 2.6.7 is much closer to that of the cyclic projection method as described in Theorem 2.2.6. Figure 2.4 illustrates and compares the three algorithms in the case of an empty intersection.

Thus, when attempting to diagnose infeasibility, the classical Douglas–Rachford scheme can be preferable to the cyclic variant. However, if one desires an estimate, even in the infeasible case, then the cyclic variant is preferable. ♦
Figure 2.4: Behaviour of the three methods starting with the same initial point. In (a)–(f), $C_2 := \mathbb{R} \times \{0\}$. In (a)–(c), $C_1 := \text{epi}(1 + 1/x) \cap (\mathbb{R}_+ \times \mathbb{R}_+) \cap (\mathbb{R}_+ \times \mathbb{R}_+)$ and $E, F$ are empty. In (d)–(f), $C_1 := \text{epi}(1 + (\cdot)^2)$ and $E, F$ are nonempty.
Chapter 3

Nonconvex Theory

Without convexity, the convergence theory for projection and reflection algorithms suffers a number of difficulties. The various non-expansivity properties exploited throughout Chapter 2 no longer hold and, further, there are no known sets having everywhere single-valued projectors which are not convex (Section 1.2). Consequently, the convergence theory is significantly more complex and, at times, somewhat delicate. Nevertheless, there exist many important and established applications of the methods in which one or more of the underlying constraint sets are not convex (as we shall see in Chapter 4). The theoretical underpinnings are therefore in need of further improvement.

Until the relatively recent work of Lewis, Luke & Malick [108], virtually nothing was known regarding the convergence of projection methods in non-convex settings. Roughly speaking, their work showed the cyclic projection method to converge \emph{locally} whenever the local geometry of the constraint sets is sufficiently well behaved. Since Lewis, Luke & Malicks’ work, local convergence theory has continued to develop healthily. Global behaviour in non-convex settings, on the other hand, is still largely unexplained.

The current literature can be divided into two closely related categories based on approach. The first assumes that the local geometry satisfies appropriate \emph{regularity properties} [24,25,85,122]. In each application, it must then be checked that the constraint sets at hand satisfy said properties [26,86]. The second involves selecting more specific classes of non-convex problems and attempting an analysis of a chosen method applied to the class [4,29,40].

This chapter begins by reviewing the current state of local convergence theory with a particular emphasis on the cyclic projection and the Douglas–Rachford methods. We then study the regularity properties of sparse and low-rank non-negativity constraints sets. The tools developed in this section are later applied to justify a distance matrix reconstruction problem in Chapter 4. In the final section, we analyse the global behaviour of the Douglas–Rachford method and provide the
first known results which explain success in combinatorial optimisation problems. Due to the additional complexities of the non-convex setting, this chapter mainly focuses on finite dimensional Euclidean spaces, as does most of the literature.

3.1 Regularity Notions and Local Convergence

In the presence of convexity, the convex normal cone plays a central role in understanding the geometry of sets and the analysis of algorithms. Recall that the convex normal cone to a set \( \Omega \subseteq \mathbb{E} \) at the point \( \bar{x} \in \Omega \) is given by

\[
N_{\Omega}^{\text{conv}}(\bar{x}) := \{ y \in \mathbb{E} : \langle y, x - \bar{x} \rangle \leq 0, \forall x \in \Omega \}.
\]

However, in settings without convexity, the convex normal cone is often of little use, even when the constraints are relatively well-behaved. The following example gives a concrete case in which the convex normal cone does not adequately describe the (local) geometry of a non-convex constraint set.

**Example 3.1.1 (Failure of the convex normal cone).** Consider the non-convex set \( C := \mathbb{R} \times \{0\} \cup \{0\} \times \mathbb{R} \) which is the union of two orthogonal subspaces. For any \( \delta > 0 \), the convex normal cone to \( C \cap B_\delta(0) \), denoted \( N_{C}^{\text{conv}} \), at zero is given by

\[
N_{C \cap B_\delta(0)}^{\text{conv}}(0) = \{0\}.
\]

Thus we see that, in this case, \( N_{C}^{\text{conv}} \) does not provide meaningful information about the set \( C \) at 0. On the other hand, consider any non-zero point \( \bar{x} \in \mathbb{R} \setminus \{0\} \times \{0\} \subseteq C \). For sufficiently small \( \delta > 0 \), the convex normal cone to \( C \cap B_\delta(0) \) at \( \bar{x} \) is given by

\[
N_{C \cap B_\delta(0)}^{\text{conv}}(\bar{x}) = (\mathbb{R} \times \{0\})^\perp = \{0\} \times \mathbb{R}.
\]

Examples of this kind will be further investigated in Section 3.2. ◊

**Remark 3.1.2 (Notation for the convex normal cone).** In Chapter 2, the convex normal cone was denoted \( N_{\Omega} \) rather than \( N_{\Omega}^{\text{conv}} \). In this section, the former shall be used to denote the limiting normal cone. This abuse of notion is justified since, in the convex setting, the various notions of normal cones coincide (Proposition 3.1.3) and so there is no ambiguity. ◊

The convex normal cone is always convex. In the non-convex setting, this represents a fundamental shortcoming, for it is unreasonable to expect that non-convex problems can always be properly analysed using convex machinery. In order to proceed, we must therefore consider non-convex normal cone notions.

Let \( \Omega \subseteq \mathbb{E} \) be non-empty and let \( \bar{x} \in \Omega \). The proximal normal cone \( \Omega \) at a point \( \bar{x} \in \Omega \) is the set

\[
N_{\Omega}^{\text{prox}}(\bar{x}) := \mathbb{R}_{+} \left( P^{-1}_\Omega(\bar{x}) - \bar{x} \right).
\]
3.1. REGULARITY NOTIONS AND LOCAL CONVERGENCE

Unfortunately, the proximal normal cone lacks such stability properties. For instance, consider the setting of Example 3.1.1. It is easy to check that $N^\text{prox}_C(0) = \{0\}$, while $N^\text{prox}_C((1/n, 0)) = \{0\} \times \mathbb{R}$ for all $n \in \mathbb{N}$. In many variational arguments, it is necessary that the objects used possess some stability under small perturbations. We therefore consider the following stable normal cone which is constructed from limits of proximal normals.

The limiting normal cone to $\Omega$ at a point $x \in \Omega$ is given by

$$N_\Omega(x) = \{y \in E : \exists x_n \in \Omega, \exists y_n \in N^\text{prox}_\Omega(x_n) \text{ s.t. } x_n \to x, y_n \to y\} = \{y \in E : \exists x_n \in E, \exists y_n \in \mathbb{R}_+(x_n - P_\Omega(x_n)) \text{ s.t. } x_n \to x, y_n \to y\}. \quad (3.1)$$

The following proposition summarises relationships between the three aforementioned types of normal cones. It shows that, in particular, the three notions coincide for convex sets.

**Proposition 3.1.3** (Normal cone inclusions). Let $\Omega \subseteq E$ be non-empty and let $\bar{x} \in \Omega$. Then the following hold.

(a) $N^\text{conv}_\Omega(\bar{x}) \subseteq N^\text{prox}_\Omega(\bar{x}) \subseteq N_\Omega(\bar{x})$.

(b) If $\Omega$ is convex then $N^\text{conv}_\Omega(\bar{x}) = N^\text{prox}_\Omega(\bar{x}) = N_\Omega(\bar{x})$.

**Proof.** See, for example, [25, Lem. 2.4].

**Remark 3.1.4.** For the full definition of the limiting normal cone, valid in any Banach space, see [116, Def. 1.1]. What we take above to be the definition is an equivalent characterization which holds in the finite dimensional case [116, Th. 1.6]. For further details, regarding non-convex normal cones, the reader is referred to the monographs [58,116,128], any of [94,95,103], and the reference therein.

We now turn our attention to two regularity notions for individual sets. Let $\Omega \subseteq E$ be a closed non-empty set and let $\bar{x} \in \Omega$. The set $\Omega$ is **prox-regular** at $\bar{x}$ if for all $y \in N_\Omega(\bar{x})$ there exists an $\epsilon > 0$ and $\rho > 0$ such that whenever $\|x - \bar{x}\| < \epsilon$ and $y \in N_\Omega(x)$ with $\|y - y\| < \epsilon$ then $x$ is the unique nearest point of \{z \in \Omega : \|z - \bar{x}\| < \epsilon \} to $x + \rho y$.

For our purposes, it will often be convenient to use the following characterisation of prox-regularity.

**Proposition 3.1.5** (Characterisation of prox-regularity). Let $\Omega \subseteq E$ be non-empty and closed. Then $\Omega$ is prox-regular at $\bar{x}$ if and only if there exists a neighbourhood of $\bar{x}$ on which $P_\Omega$ is single-valued.

**Proof.** See [126, Th. 1.3].
CHAPTER 3. NONCONVEX THEORY

The set $\Omega \subseteq \mathbb{E}$ is **super-regular** at $x \in \Omega$ if, given any $\delta > 0$, there exists a neighbourhood $U$ of $x$ such that, for any two points $u, z \in U$ with $z \in \Omega$ and any point $y \in P_\Omega(u)$, we have

$$\langle u - y, z - y \rangle \leq \delta \|u - y\| \|z - y\|.$$  

**Remark 3.1.6 (Examples of super-regular sets).** Super-regularity was first considered as part of [108]. It follows immediately from the variational characterisation of convex projectors, that every convex set is super-regular (with $U = \mathbb{E}$). In fact, the notion is weaker than prox-regularity but stronger than Clarke regularity. For details, see [108]. Quantitative versions of super-regularity which relate the size of the neighbourhood $U$ and scalar $\delta > 0$ have also been fruitfully studied [85, 86, 122].

We now turn our attention to regularity properties for collections of sets. Let $C_1, \ldots, C_m \subseteq \mathbb{E}$ be nonempty and closed. The collection $\{C_1, \ldots, C_m\}$ is **strongly regular** at $\bar{x} \in \cap_{j=1}^m C_j$ if

$$v_j \in N_{C_j}(\bar{x}) \text{ and } \sum_{j=1}^m v_j = 0 \implies v_1 = \cdots = v_m = 0. \quad (3.2)$$

When $m = 2$, equation (3.2) can be succinctly expressed as

$$N_{C_1}(\bar{x}) \cap -N_{C_2}(\bar{x}) = \{0\}.$$  

Strong regularity of the collection $\{C_1, C_2, \ldots, C_m\}$ can alternatively be described in terms of **metric regularity** [96] of the set-valued map $\Phi : \mathbb{E} \Rightarrow \mathbb{E}^m$ defined by

$$\Phi(x) := (C_1 - x) \times (C_2 - x) \times \cdots \times (C_m - x),$$

as is summarised in the following theorem.

**Theorem 3.1.7 (Characterisation of strong regularity).** Let $C_1, \ldots, C_m \subseteq \mathbb{E}$ be closed sets and let $\bar{x} \in \cap_{j=1}^m C_j$. The collection $\{C_1, \ldots, C_m\}$ is strongly regular at $\bar{x}$ if and only if there exists $\kappa > 0$ and $\delta > 0$ such that, for all $x_j \in B_\delta(0)$,

$$d^2 \left( x, \bigcap_{j=1}^m (C_j - x_j) \right) \leq \kappa \sum_{j=1}^m d^2(x + x_j, C_j), \quad \forall x \in B_\delta(\bar{x}). \quad (3.3)$$

In particular, if $\{C_1, \ldots, C_m\}$ is strongly regularity then there exists $\rho > 0$ and $\delta > 0$ such that

$$d \left( x, \bigcap_{j=1}^m C_j \right) \leq \rho \max_{j \in \{1, \ldots, m\}} d(x, C_j), \quad \forall x \in B_\delta(\bar{x}). \quad (3.4)$$
Proof. The characterisation of (3.3) can be found, for instance, in [102, Th. 9]. Equation (3.4) immediately follows from (3.3) by setting $x_1 = \ldots = x_m = 0$. □

Remark 3.1.8. In the literature, the term “strong regularity” is used inconsistently. What we refer to as “strong regularity” can also be found, for instance, in [104] under the name “uniform regularity”. The latter is well motivated due its primal space characterisation, but, nevertheless, “strong regularity” is still widely used to refer to (3.2). A discussion can be found in [84, Remark 5.8]. ♦

In the literature, particular attention has been given to strong regularity for the case of two sets. As the following proposition shows, a quantitative characterisation of strong regularity can be given by the angle between the respective normal cones.

Let $N_1, N_2 \subseteq E$ be two cones. Following Phan [122], the cosine of the local angle between $N_1$ and $N_2$ is defined to be

$$\bar{\theta}(N_1, N_2) := \max\{\langle u, v \rangle : u \in N_1 \cap B, v \in -N_2 \cap B\}. \quad (3.5)$$

Note that because $B$ is compact, the maximum in (3.5) is well-defined.

**Proposition 3.1.9.** Let $C_1, C_2 \subseteq E$ be two closed sets and let $\bar{x} \in C_1 \cap C_2$. Then the collection $\{C_1, C_2\}$ is strongly regular at $\bar{x}$ if and only if $\bar{\theta}(N_{C_1}(\bar{x}), N_{C_2}(\bar{x})) < 1$.

**Proof.** Observe that $\bar{\theta}(N_{C_1}(\bar{x}), N_{C_2}(\bar{x})) = 1$ if and only if there exists a $u \in N_{C_1}(\bar{x})$ with $\|u\| = 1$ and a $v \in -N_{C_1}(\bar{x})$ with $\|v\| = 1$ such that $\langle u, v \rangle = 1$. By the Cauchy–Schwarz inequality, the latter case is equivalent to having $u = v$, and hence is equivalent to the existence of a non-zero element in $N_{C_1}(\bar{x}) \cap -N_{C_2}(\bar{x})$. □

We are now ready to state convergence theorems for non-convex projection methods. Before doing so, recall that a sequence $(x_n)_{n=1}^{\infty}$ converges to $x$ with $R$-linear rate $r \in [0, 1)$ if, there exists $M > 0$ such that

$$\|x_n - x\| \leq Mr^n \quad \forall n \in \mathbb{N}.$$

**Theorem 3.1.10 (Lewis–Luke–Malick [108, Th. 5.16 & Cor. 5.18]).** Let $C_1, C_2 \subseteq E$ be closed sets and let $\bar{x} \in C_1 \cap C_2$. Suppose $C_2$ is super-regular at $\bar{x}$ and the collection $\{C_1, C_2\}$ is strongly regular at $\bar{x}$. Let $c \in (\bar{\theta}(N_{C_1}(\bar{x}), N_{C_2}(\bar{x})), 1)$. There exists $\delta > 0$ such that for any $x_0 \in B_\delta(\bar{x})$ the alternating projection sequence defined by

$$x_{2n+1} \in P_{C_1}(x_{2n}), \quad x_{2n+2} \in P_{C_2}(x_{2n+1}),$$

converges to a point $C_1 \cap C_2$ with $R$-linear rate $\sqrt{c}$. Furthermore, if $C_1$ is also super-regular at $\bar{x}$, then the alternating projection sequence converges with $R$-linear rate $c$. 
Theorem 3.1.11 (Phan [122, Th. 4.3]). Let $C_1, C_2 \subseteq \mathbb{E}$ be closed sets and let $x \in C_1 \cap C_2$. Suppose that $C_1$ and $C_2$ are super-regular at $x$ and the collection \{\(C_1, C_2\)\} is strongly regular at $x$. There exists $\delta > 0$ such that for any $x_0 \in B_\delta(x)$ the Douglas–Rachford sequence defined by
\[
x_{n+1} \in T_{C_1,C_2}x_n,
\]
converges to a point $C_1 \cap C_2$ with $R$-linear rate.

We provide some remarks comparing Theorems 3.1.10 and 3.1.11.

Remark 3.1.12. The conclusions of Theorem 3.1.10 for the alternating projection method are better than Theorem 3.1.11 in the following sense. To guarantee convergence of the alternating projection method only one of the constraint sets is required to be super-regular and, furthermore, the $R$-linear rate of convergence can be given by the local angle. For the Douglas–Rachford method, both constraints are required to be super-regular. The corresponding $R$-linear rate of convergence can be given although its formula is significantly more complex (see [122, Re. 4.4]). Furthermore, Theorem 3.1.11 guarantees that the Douglas–Rachford algorithm converges to a point in $C_1 \cap C_2$, something that does not happen in the general case.

Strong regularity is rather restrictive and fails to hold, even in many straight-forward classical settings. An example along the lines of the following is discussed in [24].

Example 3.1.13 (Failure of strong regularity). Consider the sets $C_1, C_2 \subseteq \mathbb{R}^3$ given by $C_1 := \text{span}\{(1, 0, 0)\}$ and $C_2 := \text{span}\{(0, 1, 0)\}$ which clearly have $C_1 \cap C_2 = \{0\}$. The normal cones to $C_1$ and $C_2$ at zero are given by
\[
N_{C_1}(0) = \text{span}\{(0, 1, 0), (0, 0, 1)\}, \quad N_{C_2}(0) = \text{span}\{(1, 0, 0), (0, 0, 1)\},
\]
and thus
\[
N_{C_1}(0) \cap -N_{C_2}(0) = \text{span}\{(0, 0, 1)\} \neq \{0\}.
\]
We therefore deduce that the collection \(\{C_1, C_2\}\) is not strongly regular at $0$.

Nevertheless, for any initial point $x = (x_1, x_2, x_3) \in \mathbb{R}^3$,
\[
P_{C_2}P_{C_1}x = P_{C_2}(x_1, 0, 0) = (0, 0, 0).
\]
In other words, the cyclic projection method converges after one iteration independently of the initial point.

A closer inspection of Example 3.1.13 suggests that it may be possible to overcome this shortcoming of strong regularity by requiring that the property hold only on the relative interior of the union of the constraints. In order to formalise this
3.1. REGULARITY NOTIONS AND LOCAL CONVERGENCE

In general, if \( L \) is not an affine subspace containing \( \Omega \), the restricted limiting normal cone cannot be given as the intersection of the standard limiting normal cone with a naturally associated cone or subspace [25, Ex. 4.1].

Let \( C_1, \ldots, C_m \subseteq \mathbb{E} \) be nonempty closed sets and set \( L := \text{aff}(\bigcup_{j=1}^{m} C_j) \). The collection of sets \( \{C_i : i = 1, \ldots, N\} \) is affine-hull regular at \( \bar{x} \in \bigcap_{j=1}^{m} C_j \) if

\[
v_j \in N_{C_j}(\bar{x}) \quad \text{and} \quad \sum_{j=1}^{m} v_j = 0 \quad \Rightarrow \quad v_1 = \cdots = v_N.
\] (3.8)

When \( m = 2 \), (3.8) can, in light of (3.7), be expressed succinctly as

\[
N_{C_1}(\bar{x}) \cap -N_{C_2}(\bar{x}) \cap (L - \bar{x}) = \{0\}.
\]

The following characterisation is the affine-hull analogue of Proposition 3.1.9. Since the proof is essentially the same as Proposition 3.1.9 with appropriate relativisation, we choose to omit it, directing the reader to [122] for further details.

**Proposition 3.1.15.** Let \( C_1, C_2 \subseteq \mathbb{E} \) be two closed sets, let \( L := \text{aff}(C_1 \cup C_2) \) and let \( \bar{x} \in C_1 \cap C_2 \). Then \( \{C_1, C_2\} \) is affine-hull regular at \( \bar{x} \) if and only if \( \bar{\theta}(N_{L_1}(\bar{x}), N_{L_2}(\bar{x})) < 1 \).
We now state convergence theorems which hold under the weaker assumption of affine-hull regularity.

**Theorem 3.1.16** (Bauschke–Luke–Phan–Wang [24]). Let $C_1, C_2 \subseteq \mathbb{E}$ be closed sets, set $L := \text{aff}(C_1 \cup C_2)$, and let $\bar{x} \in C_1 \cap C_2$. Suppose $C_2$ is super-regular at $\bar{x}$ and the collection $\{C_1, C_2\}$ is affine-hull regular at $\bar{x}$. There exists $\delta > 0$ such that for any $x_0 \in B_{\delta}(\bar{x})$ the alternating projection sequence defined by

$$x_{2n+1} \in P_{C_1}(x_{2n}), \quad x_{2n+2} \in P_{C_2}(x_{2n+1}),$$

converges to a point $C_1 \cap C_2$ with $R$-linear rate $\sqrt{c}$. Furthermore, if $C_1$ is also super-regular at $\bar{x}$, then the alternating projection sequence converge with $R$-linear rate $c$.

**Proof.** This is a special case of [24, Th. 3.14 & Th. 3.17].

The following result addresses some of the issues discussed in Remark 3.1.12. In particular, it shows that under affine-hull regularity, the Douglas–Rachford algorithm converges locally to fixed points which need not be in the target intersection.

**Theorem 3.1.17** (Phan [122, Th. 4.7]). Let $C_1, C_2 \subseteq \mathbb{E}$ be closed sets, set $L := \text{aff}(C_1 \cup C_2)$, and let $\bar{x} \in C_1 \cap C_2$. Suppose that $C_1$ and $C_2$ are super-regular at $\bar{x}$ and the collection $\{C_1, C_2\}$ is affine-hull regular at $\bar{x}$. There exists $\delta > 0$ such that for any $x_0 \in \mathbb{E}$ with $P_L(x_0) \in B_{\delta}(\bar{x})$ the Douglas–Rachford sequence defined by

$$x_{n+1} = T_{C_1,C_2}x_n,$$

converges to a point $x \in \text{Fix}T_{C_1,C_2}$ with $R$-linear rate. Moreover, $P_{C_1}z$ and $P_{C_2}x$ are both singleton, and $P_{C_1}x = P_{C_2}x = x - (x_0 - P_Lx_0) \in A \cap B$.

To give better pictures of strong regularity and affine-hull regularity, the following proposition provides characterisations in terms of familiar interiority notions when the sets are convex.

**Proposition 3.1.18** (Characterisations of regularity notions for convex sets). Let $C_1, C_2 \subseteq \mathbb{E}$ be non-empty closed convex sets. The following assertions are equivalent.

(a) $\{C_1, C_2\}$ is strongly regular (resp. affine-hull regular) at some $\bar{x} \in C_1 \cap C_2$.

(b) $\{C_1, C_2\}$ is strongly regular (resp. affine-hull regular) at every $\bar{x} \in C_1 \cap C_2$.

(c) $0 \in \text{int}(C_1 - C_2)$ (resp. $0 \in \text{ri}(C_1 - C_2)$).
3.2. REGULARITY OF SPARSITY CONSTRAINTS

We give one further local convergence result, due to Drusvyatskiy, Ioffe, & Lewis [69], which does not directly use any of the already discussed regularity assumptions. This result assumes the following transversality notion.

Let $C_1, C_2 \subseteq \mathbb{E}$ be two closed sets, and let $\bar{x} \in C_1 \cap C_2$. Then the collection $\{C_1, C_2\}$ is intrinsically transversal at $\bar{x}$ with constant $\kappa \in (0, 1]$ if, there exists $\delta > 0$ such that for any $u \in C_1 \setminus C_2$ and $v \in C_2 \setminus C_1$ it holds that

$$\max\{d(u - v, N_{C_2}(v)), d(v - u, N_{C_1}(u))\} \geq \kappa\|u - v\|.$$

Theorem 3.1.19 (Drusvyatskiy–Ioffe–Lewis [69, Th. 6.1]). Let $C_1, C_2 \subseteq \mathbb{E}$ be closed sets and let $\bar{x} \in C_1 \cap C_2$. Suppose that $C_1$ and $C_2$ are intrinsically transversal at $\bar{x}$ with constant $\sqrt{\kappa} > 0$. Let $c \in (0, \sqrt{\kappa})$. Then there exists $\delta > 0$ such that for any $x_0 \in B_\delta(\bar{x})$ the alternating projection sequence defined by

$$x_{2n+1} \in P_{C_1}(x_{2n}), \quad x_{2n+2} \in P_{C_2}(x_{2n+1}),$$

converge to a point in $C_1 \cap C_2$ with R-linear rate $1 - c$.

To conclude this section, we offer some perspectives on the local convergence theory more broadly.

Remark 3.1.20 (Local convergence theory). At their heart, all of the results in this section assume properties which ensure that the angle between constraint sets is non-zero, and lead to linear rates of convergence. In the convex setting however, the rate of convergence of projection and reflection algorithms is often slower than linear. The development of a non-convex theory which is able to deal with such cases is therefore important. A natural starting point would be to consider the class of sets which satisfy so-called semi-algebraic or Hölder regularity-type properties. In this direction, recent work of Borwein, Li & Tam [38] showed that, in the presence of convexity, such assumptions lead to sublinear rates of convergence. We emphasise that despite being slower than linear, criteria for sublinear convergence are still of interest because there exist feasibility problems for which the rate of convergence is arbitrarily slow [22, 64, 65].

3.2 Regularity of Sparsity Constraints

In this section we turn our investigation to the regularity properties of sparsity constraints with have additional non-negativity constraints. Here the term sparse refers to those objects which are expressible as a linear combination of a relatively small number of basis elements. We focus on two such sets: non-negative sparse vectors, and low-rank positive semi-definite matrices. In order to study the regularity properties of Section 3.2.3, we first derive simple formulae for their limiting
normal cones which are then used to formulate sufficient conditions to ensure the various regularity properties hold. The machinery developed in this section will be of particular importance in Section 4.3 where it is used to justify a low-rank Euclidean distance matrix reconstruction application.

Before proceeding with our investigation, we introduce the necessary background and preliminaries.

Through this section, we consider two spaces. The first is $\mathbb{R}^m$ equipped with the standard inner product. The second is the set of all real symmetric $m \times m$ matrices denoted $\mathbb{S}^m$ equipped with inner product $\langle X, Y \rangle := \text{tr}(X^T Y)$, where $\text{tr}(\cdot)$ (resp. $(\cdot)^T$) denote the trace (resp. transpose) of matrix. The induced norm is the Frobenius norm which is given by

$$\|X\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^m X_{ij}^2}.$$ 

Whenever there is no ambiguity, we omit the subscript $F$.

The set of positive (resp. negative) semi-definite $m \times m$ matrices is denoted $\mathbb{S}^m_+$ (resp. $\mathbb{S}^m_-$) and we write $x \succeq 0$ (resp. $x \preceq 0$) to mean $x \in \mathbb{S}^m_+$ (resp. $x \in \mathbb{S}^m_-$). The set of $m \times m$ orthogonal (resp. permutation) matrices is denoted $\mathcal{O}_m$ (resp. $\mathcal{P}_m$).

Given $X \in \mathbb{S}^m$ denote by $\lambda_j(X)$ the $j$th largest eigenvalue of $X$. In this way,

$$\lambda_1(X) \geq \lambda_2(X) \geq \cdots \geq \lambda_m(X).$$

The eigenvalue map is the function $\lambda: \mathbb{S}^m \to \mathbb{R}^m$ which maps a symmetric matrix to the $m$-dimensional vector of its eigenvalues arranged in non-increasing order. That is,

$$\lambda(X) := (\lambda_1(X), \lambda_2(X), \ldots, \lambda_m(X)).$$

The indicator function of a set $\Omega \subseteq \mathbb{E}$ is the function $\iota_\Omega: \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ which takes the value 0 on $\Omega$, and $+\infty$ otherwise. A function $f: \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is called symmetric if $f(x) = f(\sigma x)$ for all $x \in \text{dom } f := \{x \in \mathbb{R}^m : f(x) < +\infty\}$ and $\sigma \in \mathbb{P}^m$. A function $F: \mathbb{S}^m \to \mathbb{R} \cup \{+\infty\}$ is called spectral if $F(U^T X U) = F(X)$ for all $X \in \text{dom } F$ and $U \in \mathcal{O}_m$. A subset of $\mathbb{R}^m$ (resp. $\mathbb{S}^m$) is said to be symmetric (resp. spectral) if and only if its indicator function is symmetric (resp. spectral).

Symmetric and spectral functions have a natural one-to-one correspondence. The relationship is given by

$$F(X) = (f \circ \lambda)(X), \quad f(x) = F(\text{diag } x), \quad (3.9)$$
3.2. REGULARITY OF SPARSITY CONSTRAINTS

where \( \text{diag}(x) \) denotes the \( m \times m \) diagonal matrix whose diagonal entries are given by the entries of the vector \( x \). Consequently, many important properties can be transferred between symmetric and spectral functions [62]. For instance, the following fact shows projections onto spectral sets are easily computed whenever the projection onto the corresponding symmetric set is accessible.

**Theorem 3.2.1** (Projectors onto spectral sets). Let \( K \subseteq \mathbb{R}^m \) be a symmetric set. For any \( X \in S^m \), the projection of \( X \) onto the spectral set \( \lambda^{-1}(K) \) is given by

\[
P_{\lambda^{-1}(K)}(X) = \left\{ U^T(\text{diag}(y))U : y \in P_K \lambda(X), U \in \mathbb{O}^m(X) \right\},
\]

where the set \( \mathbb{O}^m(X) := \left\{ U \in \mathbb{O}^m : X = U^T(\text{diag} \lambda(X))U \right\} \).

**Proof.** We prove inclusion only. A proof of containment can be found in [109, Th. 21]. Suppose \( Y \in P_{\lambda^{-1}(K)}(X) \) but \( X \) and \( Y \) do not have a simultaneously ordered spectral decomposition. Let \( X = U^T(\text{diag} \lambda(X))U \) be an ordered spectral decomposition of \( X \). Since \( Y \in \lambda^{-1}(K) \), the vector \( \lambda(Y) \in K \) and hence

\[
U^T(\text{diag} \lambda(Y))U \in \lambda^{-1}(K).
\]

By Fan’s inequality [37, Th. 1.2.1] and the orthogonality of \( U \),

\[
\|X - Y\| > \|\lambda(X) - \lambda(Y)\| = \|U^T(\text{diag} \lambda(X) - \text{diag} \lambda(Y))U\| = \|X - U^T(\text{diag} \lambda(Y))U\|.
\]

This implies \( Y \notin P_{\lambda^{-1}(K)}(X) \) which is a contradiction, and completes the proof. \( \square \)

The various sub-differentials of symmetric and spectral functions are also closely related. The following fact states the equivalence in the context of normal cones, which is the important setting for our purposes.

**Theorem 3.2.2** (Normals to spectral sets). Let \( K \subseteq \mathbb{R}^m \) be a symmetric set. For any \( X \in \lambda^{-1}(K) \), the limiting normal cone to the spectral set \( \lambda^{-1}(K) \) is given by

\[
N_{\lambda^{-1}(K)}(X) = \left\{ U^T(\text{diag}(y))U : y \in N_K(\lambda(X)), U \in \mathbb{O}^m(X) \right\}.
\]

The corresponding result for the proximal normal cone also holds.

**Proof.** See [62, Th. 10] and [133, Note 6.10.6]. \( \square \)

For further details on the interplay between symmetric and spectral functions, the reader is referred to works of Lewis and others [37, 62, 106, 107, 133].

The symmetric function of central interest in this thesis is \( \ell_0 \)-functional denoted \( \| \cdot \|_0 : \mathbb{R}^m \to \{0, 1, \ldots, m\} \), which gives the number of non-zero entries of a vector.
The corresponding spectral function is the matrix rank function. Equivalence (3.9) becomes
\[ \text{rank} = \| \cdot \|_0 \circ \lambda, \quad \| \cdot \|_0 = (\text{rank}) \circ (\text{diag}). \]

Let \( s \in \{0, 1, 2, \ldots, m\} \). The set of non-negative sparse vectors is denoted
\[ K_s := \{ x \in \mathbb{R}_+^m : \| x \|_0 \leq s \}. \]

These can be viewed as the lower-level sets of the function from \( \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\} \) defined by \( x \mapsto \| x \|_0 + \iota_{\mathbb{R}_+^m}(x) \). The set of low-rank positive semi-definite matrices is denoted
\[ S_s := \{ X \in \mathcal{S}_+^m : \text{rank}(X) \leq s \}. \]

Similarly, these can be viewed as the lower-level sets of the function from \( \mathcal{S}^m \to \mathbb{R} \cup \{+\infty\} \) defined by \( X \mapsto \text{rank}(X) + \iota_{\mathcal{S}_+^m}(X) \).

Our strategy is to first address the non-negative sparse vector settings, before lifting the results to symmetric matrices. We therefore begin with a study of the set \( K_s \).

### 3.2.1 Non-Negative Sparse Vector Sets

Given a vector \( x \in \mathbb{R}^m \) we denote \( I(x) := \{ j \in \{1, 2, \ldots, m \} : x_j \neq 0 \} \). For convenience, we write \( x^+ := P_{\mathbb{R}_+^m}(x) = \max\{0, x\} \) (in the pointwise sense) and \( x^- := -P_{\mathbb{R}_+^m}(x) = -\min\{0, x\} \) (see [16] for further details). The standard basis for \( \mathbb{R}^m \) shall be denoted \( e_1, e_2, \ldots, e_m \). The set of sparse vectors is denoted
\[ A_s := \{ x \in \mathbb{R}^m : \| x \|_0 \leq s \}. \]

It is worth noting that the set \( A_s \) is convex precisely when either \( s = 0 \) or \( s = m \). In these cases, we have \( A_s = \{0\} \) and \( A_m = \mathbb{R}^m \), respectively.

The following proposition states, in particular, that the projection onto the set \( K_s \) of a vector is given by simple thresholding of the vector keeping only its \( s \) largest non-negative entries.

**Proposition 3.2.3** (Projection onto \( K_s \) and its inverse). The following hold.

(a) \( \forall x \in \mathbb{R}^m \) and \( \forall y \in P_{K_s}(x), \quad I(y) \subseteq I(x^+) \).

(b) \( \forall x \in \mathbb{R}^m, \quad P_{K_s}(x) = P_{A_s}(x^+) \).

(c) \( \forall x \in \mathbb{R}_+^m, \quad P_{K_s}(x) = P_{A_s}(x) \).

(d) \( \forall x \in \mathbb{R}^m, \quad P_{K_s}(x) = P_{A_s}(x^+) \) and hence
\[ P_{K_s}(x) = \left\{ y \in \mathbb{R}^m : y_j = \begin{cases} x_j^+, & j \in J, \\ 0, & j \notin J \end{cases}; \text{ for some } J \in J_s(x) \right\}. \]
where
\[ \mathcal{J}_s(x) := \left\{ J \subseteq \{1, 2, \ldots, m\} : |J| = s, \min_{j \in J} x^+_j \geq \max_{j \notin J} x^+_j \right\}. \]

(e) If \( y \in \mathcal{K}_s \) and \( \|y\|_0 = s \), then
\[ P^{-1}_{\mathcal{K}_s}(y) = \left\{ x : y_j = x_j \text{ for all } j \in \mathcal{I}(y), \min_{j \notin \mathcal{I}(y)} y_j \geq \max_{j \notin \mathcal{I}(y)} x^+_j \right\}. \]

(f) If \( y \in \mathcal{K}_s \) and \( \|y\|_0 < s \) then \( P^{-1}_{\mathcal{K}_s}(y) = \{ x : x^+ = y \} = P^{-1}_{\mathbb{R}^+_s}(y) \).

Proof. (a): Let \( y \in P_{\mathcal{K}_s}(x) \) and suppose there exists an index \( j_0 \in \mathcal{I}(y) \setminus \mathcal{I}(x^+) \). Then \( y_{j_0} > 0 \) and \( x_{j_0} < 0 \). Letting \( z := y - y_{j_0}e_{j_0} \in \mathcal{K}_s \) we deduce
\[ \|x - y\|^2 > \sum_{j \neq j_0} |x_j - y_j|^2 + |x_{j_0} - 0|^2 = \|x - z\|^2, \]
which contradicts the assumption \( y \in P_{\mathcal{K}_s}(x) \).

(b): Let \( y \in P_{\mathcal{K}_s}(x) \cup P_{\mathcal{K}_s}(x^+) \) be arbitrary. By (i), \( \mathcal{I}(y) \subseteq \mathcal{I}(x^+) \), hence \( (x^+ - y, x^-) = 0 \) and
\[ \|x - y\|^2 = \|x^+ - y\|^2 + \|x^-\|^2. \]
This implies
\[ \arg \min_{y \in \mathcal{K}_s} \|x - y\| = \arg \min_{y \in \mathcal{K}_s} \|x^+ - y\|, \]
from which the result follows.

(c): Since \( \mathcal{A}_s \cap \mathcal{K}_s = \mathcal{K}_s \), it suffices to show that any \( y \in P_{\mathcal{A}_s}(x) \) is contained in \( \mathcal{K}_s \). Suppose \( y \in P_{\mathcal{A}_s}(x) \setminus \mathcal{K}_s \) and let \( j_0 \) be an index such that \( y_{j_0} < 0 \). Letting \( z := y - y_{j_0}e_{j_0} \in \mathcal{A}_s \setminus \mathcal{K}_s \), since \( x \in \mathbb{R}^+_s \), we have
\[ \|x - y\|^2 > \sum_{j \neq j_0} |x_j - y_j|^2 + |x_{j_0} - 0|^2 = \|x - z\|^2, \]
which contradicts the assumption that \( y \in P_{\mathcal{A}_s}(x) \).

(d): Follows from (b), (c) and [26, Prop. 3.6(ii)]. (e): Follows from (d) and [26, Prop. 3.6(vi)]. (f): Follows from (d) and [26, Prop. 3.6(vi)]. \( \square \)

Given a vector \( x \in \mathbb{R}^m \) denote by \([x]\) the vector in \( \mathbb{R}^m \) obtained by permuting the entries of \( x \) in non-increasing order. Under this notation, \([x]_j \in \mathbb{R} \) is the \( j \)th largest entry of a vector \( x \). For vectors \( x, y \in \mathbb{R}^m \), \( x \odot y \) denotes the Hadamard product given component-wise by \((x \odot y)_j := x_jy_j \) for all \( j \in \{1, 2, \ldots, m\} \). It is worth noting that for fixed \( \overline{x} \), the set \( \{ y \in \mathbb{R}^m : \overline{x} \odot y = 0 \} \), is simply the perpendicular subspace to the support of \( \overline{x} \).

The following lemma is a kind of (non-convex) analogue to Moreau’s decomposition theorem \([19, \text{Th. 6.29}]\), which applies to convex cones, for the set \( \mathcal{K}_s \).
Lemma 3.2.4 (Decomposition lemma for $K_s$). Let $x, y, z \in \mathbb{R}^m$ with $x = y + z$. Then $y \in P_{K_s}x$ if and only if $y \in K_s$, $y \odot z = 0$ and $[y]_s \geq [z]_1$.

Proof. Suppose $y \in P_{K_s}x \subseteq K_s$. By Proposition 3.2.3 there is an index set $J_0 \in J_s(x)$ such that

$$y_j = \begin{cases} x_j^+, & j \in J_0, \\ 0, & j \notin J_0; \end{cases} \quad \text{and} \quad \min_{j \in J_0} x_j^+ \geq \max_{j \notin J_0} x_j^+.$$

Thus $z$ is given component-wise by

$$z_j = x_j - y_j = \begin{cases} -x_j^-, & j \in J_0, \\ x_j, & j \notin J_0. \end{cases}$$

It follows that $[y]_s \geq [z]_1$ since

$$[y]_s = \min_{j \in J_0} x_j^+ \geq \max_{j \notin J_0} x_j^+ \geq \max \{ \{ x_j^+ : j \notin J_0 \} \cup \{ -x_j^- : j \in J_0 \} \} \geq \max \{ \{ x_j : j \notin J_0 \} \cup \{ -x_j^- : j \in J_0 \} \} = [z]_1.$$

To show $y \odot z = 0$, observe that $(y \odot z)_j = y_j z_j$ is either $x_j^+( -x_j^-)$ if $j \in J_0$, or $0 \cdot x_j$ if $j \notin J_0$, which is zero in either case.

Conversely, suppose $y \in K_s, y \odot z = 0$ and $[y]_s \geq [z]_1$. Since $x = y + z$ and $y \odot z = 0$, for each $j \in \{1, 2, \ldots, m\}$ either $y_j = x_j$ and $z_j = 0$; or $y_j = 0$ and $z_j = x_j$. Since $y \in K_s$ there is an index set $J_0$ with $|J_0| \leq s$ such that we may express

$$y_j = \begin{cases} x_j^+, & j \in J_0, \\ 0, & j \notin J_0; \end{cases} \quad z_j = \begin{cases} -x_j^-, & j \in J_0, \\ x_j, & j \notin J_0; \end{cases} \quad (3.10)$$

noting that $x_j = x_j^+$ and $x_j^- = 0$ for $j \in J_0$.

To show that $y \in P_{K_s}(x)$ it suffices to consider the case in which $|J_0| = s$. For if $|J_0| < s$ then $[y]_s = 0$, hence

$$0 = [y]_s \geq [z]_1 = \max \{ \{ -x_j^- : j \in J_0 \} \cup \{ x_j : j \notin J_0 \} \}.$$

In particular, $x_j \leq 0$ for $j \notin J_0$, or equivalently $x_j = -x_j^-$ for $j \notin J_0$. It is therefore possible to replace the index set $J_0$ with a superset having cardinality $s$ without changing (3.10).

Thus, suppose $|J_0| = s$ but $J_0 \notin J_s(x)$. Then there exist indices $j_1 \in J_0$ and $j_2 \notin J_0$ such that $x_{j_1}^+ < x_{j_2}^+$. In particular, $x_{j_2} = x_{j_2}^+$ and hence

$$[y]_s = \min_{j \in J_0} x_j^+ \leq x_{j_1}^+ < x_{j_2}^+ = x_{j_2} \leq \max \{ \{ x_j^- : j \in J_0 \} \cup \{ x_j : j \notin J_0 \} \} = [z]_1.$$

This contradicts the assumption that $[y]_s \geq [z]_1$. Proposition 3.2.3 implies $y \in P_{K_s}(x)$, and thus completes the proof. \qed
The following lemma is concerned with the behavior of sequences converging to points in $K_s$.

**Lemma 3.2.5.** If $(x_k) \subseteq \mathbb{R}^m$ converges to a point $x \in K_s$ and $(z_k) \subseteq \mathbb{R}^m$ is a sequence such that $z_k \in (I - P_{K_s})x_k$ then $z_k \to 0$.

**Proof.** By Proposition 3.2.3, for each $k \in \mathbb{N}$, the point $z_k$ is of the form

$$(z_k)_j = \begin{cases} -(x_k)_j, & j \in \mathbb{J}_k, \\ (x_k)_j, & j \notin \mathbb{J}_k; \end{cases} \tag{3.11}$$

for some index set $\mathbb{J}_k \in \mathcal{J}_s(x_k)$. We refer to $\mathbb{J}_k$ as the *active set* for $z_k$. The collection of all active sets $\{\mathbb{J}_k : k \in \mathbb{N}\}$ is finite as a subset of the power set of $\{1, 2, \ldots, m\}$. In particular, only a finite number of index sets from the collection are active infinitely often.

Let $\mathbb{J}$ be an index set from the collection which is active infinitely often, and let $(z_k_l)$ be the subsequence consisting of all terms for which $\mathbb{J}$ is the active set. Since $x \in K_s$ and $\mathbb{J} \in \mathcal{J}_s(x_k)$ for all $l \in \mathbb{N}$, the representation given in (3.11) implies $z_{k_l} \to 0$. Since $\mathbb{J}$ was chosen arbitrarily from the collection of index sets which are active infinitely often, it follows that $z_k \to 0$. \hfill $\square$

We now provide our first main result: a simple characterization of the limiting normal cone to the set of non-negative sparse vectors. Given $y \in \mathbb{R}^m$ and an index set $\mathbb{J} \subseteq \{1, \ldots, m\}$ the notation $y\mid_{\mathbb{J}} = 0$ means $y_j = 0$ for all $j \in \mathbb{J}$.

**Theorem 3.2.6** (Limiting normal cone to $K_s$). The limiting normal cone to the set $K_s$ at a point $x \in K_s$ is given by

$$N_{K_s}(x) = \{y \in \mathbb{R}^m : x \odot y = 0, y \leq 0\} \cup \{y \in \mathbb{R}^m : x \odot y = 0, \|y\|_0 \leq m - s\}.\]$$

**Proof.** ($\subseteq$) Suppose $y \in N_{K_s}(x)$. Then there exists sequences $(x_k), (y_k), (z_k) \subseteq \mathbb{R}^m$ such that

$$x_k \to x, \quad y_k = \alpha_k z_k \to y, \quad z_k = x_k - p_k,$$

where $\alpha_k \in \mathbb{R}^+$ and $p_k \in P_{K_s}(x_k)$. By Lemma 3.2.4, $p_k \odot z_k = 0$ and thus

$$x_k \odot y_k = (z_k + p_k) \odot (\alpha_k z_k) = \alpha_k (z_k \odot z_k) = z_k \odot y_k.$$

Together with Lemma 3.2.5 this yields

$$\bar{x} \odot y = \left( \lim_{k \to \infty} x_k \right) \odot \left( \lim_{k \to \infty} y_k \right) = \lim_{k \to \infty} (x_k \odot y_k) = \lim_{k \to \infty} (z_k \odot y_k) = \left( \lim_{k \to \infty} z_k \right) \odot \left( \lim_{k \to \infty} y_k \right) = 0 \odot y = 0.$$
By the Pigeonhole Principle there exists a subsequence \((p_{ki})\) and an index set \(J_0 \in J_s(x_{ki})\) such that
\[
(p_{ki})_j = \begin{cases} (x_{ki})_j^+, & j \in J_0; \\ 0, & j \not\in J_0; \end{cases} \quad (y_{ki})_j = \begin{cases} -(x_{ki})_j^-, & j \in J_0; \\ (x_{ki})_j, & j \not\in J_0. \end{cases}
\] (3.12)

Since \(J_0 \in J_s(x_{ki})\) for all \(l \in \mathbb{N}\), we have
\[
\min_{j \in J_0} x_j + j = \lim_{l \to \infty} \left( \min_{j \in J_0} (x_{ki})_j^+ \right) \geq \lim_{l \to \infty} \left( \max_{j \not\in J_0} (x_{ki})_j \right) = \max_{j \not\in J_0} x_j.
\]
and thus \(J_0 \in J_s(x)\). If \(y \not\leq 0\) then there is an index \(j_0 \in \{1, 2, \ldots, m\}\) such that \(y_{j_0} > 0\). Since \(y_k \to y\), we assume \(l\) to be sufficiently large so that \((y_{ki})_{j_0} > 0\).

For the \(y \leq 0\) case, define sequences \((x_k)\) and \((y_k)\) by
\[
x_k := \overline{x} + \frac{1}{k}y \to \overline{x}, \quad y_k \in k(x_k - P_{K_s}(x_k)).
\]
Then, for any \(k \in \mathbb{N}\), we have \(P_{K_s}(x_k) = \{\overline{x}\}\) thus
\[
y_k = k \left( \left( \overline{x} + \frac{y}{k} \right) - \overline{x} \right) = y.
\]

For the other case, suppose \(\|y\|_0 \leq m - s\). Then there is an index set \(J_0 \in J_s(x)\) such that \(y_{j_0} = 0\). Let \(w \in \mathbb{R}^m\) be the vector whose entries are 1 on \(J_0\), and 0 otherwise. Define sequences \((x_k)\) and \((y_k)\) by
\[
x_k := \overline{x} + \frac{1}{k}y + \frac{1}{\sqrt{k}}w \to \overline{x}, \quad y_k \in k(x_k - P_{K_s}(x_k)).
\]
Since \(1/k \to 0\) at a faster rate than \(1/\sqrt{k} \to 0\), there exists a sufficiently large \(K\) such that for \(k > K\),
\[
\min_{j \in J_0} \left\{ \overline{x}_j + \frac{1}{\sqrt{k}}w_j \right\} > \frac{1}{k} \max_{j \not\in J_0} \{y_j\}.
\]
3.2. REGULARITY OF SPARSITY CONSTRAINTS

Hence for \( k > K \) we have \( P_{K_s}(x_k) = \{ \overline{x} + w/\sqrt{k} \} \), and thus that

\[
y_k = k \left( \frac{1}{k} y + \frac{1}{\sqrt{k}} w \right) - \left( \frac{1}{\sqrt{k}} w \right) = y.
\]

This completes the proof.

We give an example illustrating the different possible limiting normals to \( K_s \).

Example 3.2.7 (Limiting normals to \( K_2 \) in \( \mathbb{R}^4 \)). Let \( \overline{x} := (4, 0, 0, 0) \in \mathbb{R}^m \) which is contained in the non-negative sparsity set

\[
K_2 = \{ y \in \mathbb{R}_+^4 : \|y\|_0 \leq 2 \}.
\]

For the sequence \((x_k)\) given in each of the examples below define the sequence \((y_k)\) by

\[
y_k \in k (x_k - P_{K_2}(x_k)).
\]

(a) Consider \((x_k)\) given by \( x_k := (4, \frac{3}{k}, \frac{2}{k}, \frac{1}{k}) \to \overline{x} \), then \( P_{K_2}(x_k) = \{ (4, \frac{3}{k}, 0, 0) \} \), thus \( y_k = (0, 0, 2, 1) \to y := (0, 0, 2, 1) \). In this case \( \|y\|_0 = 2 \), and \( \|x_k\|_0 = 4 \) for all \( k \in \mathbb{N} \).

(b) Consider \((x_k)\) given by \( x_k := (4, -\frac{3}{k}, -\frac{2}{k}, -\frac{1}{k}) \to \overline{x} \), then \( P_{K_2}(x_k) = \{ (4, 0, 0, 0) \} \), thus \( y_k = (0, -3, -2, -1) \to y := (0, -3, -2, -1) \). In this case, \( y \leq 0 \). Again \( \|x_k\|_0 = 4 \) for all \( k \in \mathbb{N} \). Note also that the possibility of having \( \|y\|_0 > 2 \) is a consequence of the fact that \( \|\overline{x}\|_0 < 2 \).

(c) Consider \((x_k)\) given by \( x_k := (4, -\frac{3}{k}, -\frac{2}{k}, 0) \to \overline{x} \), then \( P_{K_2}(x_k) = \{ (4, 0, 0, 0) \} \), thus \( y_k = (0, -3, -2, 0) \to y := (0, -3, -2, 0) \). In this case both \( \|y\|_0 = 2 \) and \( y \leq 0 \).

\( \diamond \)

For convex sets the convex and limiting normal cones coincide [116, Ch. 1]. As an immediate consequence of Theorem 3.2.6 we recover the following result.

**Corollary 3.2.8 (Normal cone to \( \mathbb{R}^m_+ \)).** The normal cone to the convex set \( \mathbb{R}^m_+ \) at the point \( \overline{x} \in \mathbb{R}^m_+ \) is given by

\[
N_{\mathbb{R}_+^m}(\overline{x}) = N_{\mathbb{R}_+^m}^{\text{conv}}(\overline{x}) = \{ y \in \mathbb{R}_+^m : \overline{x} \odot y = 0, y \leq 0 \}.
\]

**Proof.** Since \( m - s = 0 \) we have \( \{ y \in \mathbb{R}_+^m : \overline{x} \odot y = 0, \|y\|_0 \leq 0 \} = \{0\} \). The result now follows from Theorem 3.2.6. \( \square \)

The following remark sheds light on the two sets in the normal cone formula of Theorem 3.2.6.
Remark 3.2.9 \((N_{K_s} \text{ is the union of two normal cones})\). The limiting normal to sparsity set \(A_s\) is given by (see [26, Th. 3.9])

\[
N_{A_s} (\overline{x}) = \{ y \in \mathbb{R}^m : \overline{x} \odot y = 0, \|y\|_0 \leq m - s \}.
\]

(3.13)

Combining with Corollary 3.2.8, Theorem 3.2.6 can be expressed

\[
N_{K_s} (\overline{x}) = N_{\mathbb{R}^m_+} (\overline{x}) \cup N_{A_s} (\overline{x}).
\]

That is, \(N_{K_s}\) is the union of the convex normal cone to the non-negativity set \(\mathbb{R}^m_+\), and the limiting normal cone to the sparsity set \(A_s\).

\(\Diamond\)

Around points of maximal sparsity the set \(K_s\) is locally indistinguishable from the sparsity set \(A_s\). In this case, the formula for the normal cone simplifies accordingly.

Corollary 3.2.10 (Points of maximal sparsity). The limiting normal cone to the set \(K_s\) at a point \(\overline{x} \in K_s\) having \(\|x\|_0 = s\) is given by

\[
N_{K_s} (\overline{x}) = N_{A_s} (\overline{x}) = \{ y \in \mathbb{R}^m : \overline{x} \odot y = 0 \}.
\]

Proof. Since \(\|x\|_0 = s\) we have \(\{ y \in \mathbb{R}^m : \overline{x} \odot y = 0, \|y\|_0 \leq m - s \} = \{ y : \overline{x} \odot y = 0 \}\). Observe,

\[
\{ y \in \mathbb{R}^m : \overline{x} \odot y = 0, y \leq 0 \} \subseteq \{ y \in \mathbb{R}^m : \overline{x} \odot y = 0 \}.
\]

The result now follows from Theorem 3.2.6 and (3.13).

To conclude our study of the vector setting, we give a characterization of the proximal normal cone to \(K_s\), in terms of already introduced objects.

Theorem 3.2.11 (Proximal normal cone to \(K_s\)). The proximal normal cone to \(K_s\) at the point \(\overline{x} \in K_s\) is given by

\[
N_{K_s}^{\text{prox}} (\overline{x}) = \begin{cases} 
N_{\mathbb{R}^m_+} (\overline{x}) = N_{\mathbb{R}^m_+} (\overline{x}), & \|\overline{x}\|_0 < s, \\
N_{A_s}^{\text{prox}} (\overline{x}) = N_{A_s} (\overline{x}) = N_{K_s} (\overline{x}), & \|\overline{x}\|_0 = s.
\end{cases}
\]

Proof. On one hand, if \(\|\overline{x}\|_0 < s\) then, by Proposition 3.2.3, \(P_{K_s}^{-1} (\overline{x}) = P_{\mathbb{R}^m_+}^{-1} (\overline{x})\) implying that the corresponding proximal normal cones coincide. The claimed formula now follows by Proposition 3.1.3. On the other hand, if \(\|\overline{x}\|_0 = s\) the result follows from Corollary 3.2.10 and [26, Prop. 3.8].

Remark 3.2.12. Theorem 3.2.11 shows that the proximal normal cone to a non-negative sparsity set does not capture all important features of the set at points which do not have maximal sparsity. In particular, the proximal normal cone coincides with the normal cone to the non-negative orthant, where as the corresponding limiting cone need not (see Example 3.2.7). Similar behaviour is observed for the proximal normal cone to the sparsity set \(A_s\), which is equal to \(\{0\}\) at points not having maximal sparsity [26, Prop. 3.8].

\(\Diamond\)
3.2.2 Low-Rank Positive Semi-Definite Matrix Sets

Using the correspondence between symmetric and spectral functions, we now lift vector results to the space of symmetric matrices. For the convenience of the reader, we first recall the definition

\[ S_s := \{ X \in S^+_m : \text{rank } X \leq s \} \]

**Proposition 3.2.13** (Projection onto \( S_s \)). Let \( X \in S^m \) and define

\[ \lambda^+_s(X) := (\lambda^+_1(X), \ldots, \lambda^+_s(X), 0, \ldots, 0) \]

The projection of \( X \) onto \( S_s \) is given by

\[ P_{S_s}(X) = \{ Y \in S^m : Y = U^T(\text{diag } \lambda^+_s(X))U, U \in O^m(X) \} \]

**Proof.** Follows from Theorem 3.2.1 and Proposition 3.2.3.

Our next main result is a simple characterization of the limiting normal cone to the set \( S_s \).

**Theorem 3.2.14** (Limiting normal cone to \( S_s \)). The limiting normal cone to the set \( S_s \) at the point \( X \in S_s \) is given by

\[ N_{S_s}(X) = \{ Y \in S^m : XY = 0, Y \preceq 0 \} \cup \{ Y \in S^m : XY = 0, \text{rank}(Y) \leq m - s \} \]

**Proof.** By Theorem 3.2.6,

\[ N_{K_s}(\lambda(X)) = \{ y \in \mathbb{R}^m : \lambda(X) \odot y = 0, y \preceq 0 \} \cup \{ y \in \mathbb{R}^m : \lambda(X) \odot y = 0, \|y\|_0 \leq m - s \} \]

By Theorem 3.2.2,

\[ N_{S_s}(X) = \{ U^T \text{diag}(y)U : y \in N_{K_s}(\lambda(X)), U \in O^m(X) \} \]

The result follows by combining these two expressions.

As before, we deduce consequences of Theorem 3.2.14. The first is the normal cone to the set of positive semi-definite matrices. This can be found, for example, in [88].

**Corollary 3.2.15** (Normal cone to \( S^m_+ \)). The normal cone to the set \( S^m_+ \) at a point \( X \in S^m_+ \) is given by

\[ N_{S^m_+}(X) = N_{S^m_+}^{\text{conv}}(X) = \{ Y \in S^m_+ : XY = 0, Y \preceq 0 \} \]
Proof. Since \( m - s = 0 \) we have \( \{ Y \in S^m : \bar{X}Y = 0, \text{rank}(Y) \leq 0 \} = \{ 0 \} \). The result now follows from Theorem 3.2.14. \( \square \)

Denote the set of low-rank symmetric matrices by

\[ \mathcal{R}_s := \{ X \in S^m : \text{rank}(X) \leq s \} .\]

The following proposition is a characterization of the limiting normal cone to \( \mathcal{R}_s \).

**Proposition 3.2.16** (Limiting normal cone to \( \mathcal{R}_s \)). The limiting normal cone to the set \( \mathcal{R}_s \) at a \( \bar{X} \in \mathcal{R}_s \) having \( \text{rank}(\bar{X}) = s \) is given by

\[ N_{\mathcal{R}_s}(\bar{X}) = \{ Y \in S^m : \bar{X}Y = 0 \} .\]

**Proof.** Follows from (3.13) and Theorem 3.2.2. \( \square \)

**Remark 3.2.17.** A related formula for the limiting normal cone to the set of low-rank (possibly rectangular) real matrices is derived in [112]. This formula does not directly apply in our context even when specialized to the square case. The reason being that the present study is concerned with the space of symmetric matrices whereas [112] consider the non-symmetric setting. \( \diamond \)

As was the case in the vector setting, here we are also able to express the normal cone to \( S_s \) in terms of the normal cones of its two ‘building blocks’.

**Remark 3.2.18 (\( N_{S_s} \) is the union of two normal cones).** In light of Theorem 3.2.14, Corollary 3.2.15 and Proposition 3.2.16, the limiting normal cone to \( S_s \) can be expressed as

\[ N_{S_s}(\bar{X}) = N_{S_s}^{\text{conv}}(\bar{X}) \cup N_{\mathcal{R}_s}(\bar{X}) .\]

That is, \( N_{S_s} \) is the union of the convex normal cone to the set of all positive semi-definite matrices, and the limiting normal cone to the low-rank set \( \mathcal{R}_s \). \( \diamond \)

A characterization of the proximal normal cone to \( S_s \) can also be given. As was the case in the vector setting, it also does not adequately describe the geometry of \( S_s \).

**Theorem 3.2.19 (Proximal normal cone to \( S_s \)).** The proximal normal cone to \( S_s \) at the point \( \bar{X} \in K_s \) is given by

\[
N_{S_s}^{\text{prox}}(\bar{X}) = \begin{cases} 
N_{S_s}^{\text{conv}}(\bar{X}) = N_{S_s}(\bar{X}) , & \text{rank}(\bar{X}) < s , \\
N_{\mathcal{R}_s}^{\text{prox}}(\bar{X}) = N_{\mathcal{R}_s}(\bar{X}) , & \text{rank}(\bar{X}) = s .
\end{cases}
\]

**Proof.** Follows from Theorem 3.2.2 and Theorem 3.2.11. \( \square \)
3.2. REGULARITY OF SPARSITY CONSTRAINTS

3.2.3 Characterisations of Regularity Conditions

Using the results formulae from the previous two sections, we now investigate the regularity properties of non-negative sparsity sets, and collections of sets containing non-negative sparsity sets.

We begin with strong regularity. The following propositions use the normal cone formulae in Theorems 3.2.6 and 3.2.14 to provide conditions for strong regularity of intersections involving non-negative sparsity sets. These conditions, which we derive below, are analogous to those given by [112, Prop. 3.8] for rank constraints not requiring non-negativity.

**Proposition 3.2.20** (Strong regularity for intersections with \( K_s \)). Let \( \Omega \subseteq \mathbb{R}^m \) be closed and \( \pi \in \Omega \cap K_s \). Then \( \{\Omega, K_s\} \) is strongly regular at \( \pi \) if and only if for all non-zero \( y \in N_{\Omega}(\pi) \) either (a) \( \pi \odot y \neq 0 \), or (b) \( y \not\geq 0 \) and \( \|y\|_0 > m - s \).

**Proof.** By characterization of \( N_{K_s} \) in Theorem 3.2.6, we deduce that strong regularity of the sets \( \{\Omega, K_s\} \) is equivalent to

\[
\{0\} = N_{\Omega}(\pi) \cap \{y \in \mathbb{R}^m : \pi \odot y = 0, y \geq 0\}, \text{ and }
\{0\} = N_{\Omega}(\pi) \cap \{y \in \mathbb{R}^m : \pi \odot y = 0, \|y\|_0 \leq (m - s)\}.
\]

The result now follows. \( \square \)

The following result is the symmetric matrix analogue of Proposition 3.2.20.

**Proposition 3.2.21** (Strong regularity for intersections with \( S_s \)). Let \( \Omega \subseteq \mathbb{S}^m \) be closed and \( X \in \Omega \cap S_s \). Then \( \{\Omega, S_s\} \) is strongly regular at \( X \) if and only if for all non-zero \( Y \in N_{\Omega}(X) \) either (a) \( XY \neq 0 \), or (b) \( Y \not\succeq 0 \) and \( \text{rank}(Y) > m - s \).

**Proof.** Argue similarly to Proposition 3.2.20, using the characterization of \( N_{S_s} \) in Theorem 3.2.14. \( \square \)

In Chapter 4, we consider applications of Proposition 3.2.20 to solving sparse linear systems, and of Proposition 3.2.21 to low-rank semi-definite programming feasibility.

As we have seen in Section 3.1, when strong regularity fails, it is possible that the formally weaker notion of affine-hull regularity still holds true (see also [122, Remark 2.1]). Unfortunately, as we now show, when applied to sparsity or non-negative sparsity sets this restricted constraint qualification gives no additional information.

**Proposition 3.2.22** (Affine restricted limiting normal cones to sparsity sets). Suppose \( s \geq 1 \), and let \( \Omega \) be one of the sets \( K_s, A_s, S_s \) or \( R_s \). For any \( \pi \in \Omega \) and affine subspace \( L \) containing \( \Omega \), the \( L \)-restricted and standard limiting normal cones coincide (i.e., \( N_L^L(\pi) = N_{\Omega}(\pi) \)).
Proof. We first consider the $\Omega = K_s$ case. Let $E := \{x \in \mathbb{R}^m : \|x\|_0 = 1\}$. Since $E \subseteq K_s \subseteq L$ and $\text{aff}(E) = \mathbb{R}^m$, it follows that $L = \mathbb{R}^m$. Using (3.7) we deduce

$$N_{\Omega}^L(\overline{x}) = N_{\Omega}(\overline{x}) \cap (\mathbb{R}^m - \overline{x}) = N_{\Omega}(\overline{x}) \cap \mathbb{R}^m = N_{\Omega}(\overline{x}).$$

The proof of the $\Omega = A_s$ case is performed similarly. The proofs of the $\Omega = S_s$ and $\Omega = A_s$ cases are also similar with $E := \{X \in S^m_+ : \text{rank}(X) = 1\}$.

An immediate consequence of Proposition 3.2.22 is the following.

**Corollary 3.2.23** (Affine-hull regularity for sparsity sets). Suppose $s \geq 1$ and let $\Omega$ be a closed non-empty set. The notions of strong regularity and affine-hull regularity coincide for each of the pairs: (a) $\{\Omega, K_s\}$, (b) $\{\Omega, A_s\}$, (c) $\{\Omega, S_s\}$, and (d) $\{\Omega, R_s\}$.

When $s = m$, both $K_s$ and $S_s$ are convex, and hence everywhere prox-regular. The following two propositions show that, in the non-convex case, prox-regularity holds only at points of maximal sparsity.

**Proposition 3.2.24** (Prox-regularity of $K_s$). Let $\overline{x} \in K_s$ with $s \in \{1, \ldots, (m - 1)\}$ and $m \geq 2$. Then $K_s$ is prox-regular at $\overline{x}$ if and only if $\|x\|_0 = s$.

Proof. On one hand, suppose $\|\overline{x}\|_0 < s$. To show that $K_s$ is not prox-regular at $\overline{x}$, it suffices to produce a sequence $x_k \to \overline{x}$ such that $P_{K_s}(x_k)$ is not singleton for any $k \in \mathbb{N}$. To this end, since $\|\overline{x}\|_0 \leq s - 1$, we have

$$2 \leq (s - \|\overline{x}\|_0) + 1 = (s + 1) - \|\overline{x}\|_0 \leq m - \|\overline{x}\|_0.$$

Since $\overline{x}$ has $m - \|x\|_0$ entries which are zero, therefore there exists an index set $\mathbb{I}_0 \subseteq \{1, \ldots, m\}$ with $|\mathbb{I}_0| = (s - \|\overline{x}\|_0) + 1 \geq 2$ such that $\overline{x}_j = 0$ for all $j \in \mathbb{I}_0$. Define the vector $v \in \mathbb{R}^m$ and the sequence $(x_k)$ by

$$v := \sum_{j \in \mathbb{I}_0} e_j, \quad x_k := \overline{x} + \frac{1}{k} v.$$

Observe $\|x_k\|_0 = \|\overline{x}\|_0 + \|v\|_0 = s + 1$. Thus, for sufficiently large $k$, Proposition 3.2.3 yields

$$P_{K_s}(x_k) = \left\{ \overline{x} + \frac{1}{k}(v - e_j) : j \in \mathbb{I}_0 \right\} \text{ where } |P_{K_s}(x_k)| = |\mathbb{I}_0| \geq 2.$$

By taking $k$ sufficiently large, the distance between $x_k$ and $\overline{x}$ can be made arbitrary small, thus proving the that $K_s$ is not prox-regular at $\overline{x}$. 


On the other hand, suppose $\|x_0\|_0 = s$, and let $\delta := \frac{1}{2} \min \{x_j : x_j > 0\} > 0$. Since $x$ has maximal sparsity, the set $J_s(x)$ is a singleton, say $J_s(x) = \{0\}$. For any $x \in B_\delta(x)$, $j \in J_0$ and $i \notin J_0$ we have

$$x_j - x_j \leq |x_j - x_j| \leq \|x - x\| < \delta \leq \frac{1}{2} x_j, \quad \text{and} \quad x_i = |x_i - x_i| \leq \|x - x\| < \delta \leq \frac{1}{2} x_j.$$ 

Altogether, $0 < x_j/2 \leq x_j, x_i < x_j/2$, and therefore $J_s(x) = \{0\}$. By Proposition 3.2.3, $P_K_s(x)$ is single-valued, hence $K_s$ is prox-regular at $x$. □

**Proposition 3.2.25** (Prox-regularity of $S_s$). Let $\overline{X} \in S_s$ with $s \in \{1, \ldots, m-1\}$ and $m \geq 2$. Then $S_s$ is prox-regular at $\overline{X}$ if and only if $\text{rank}(\overline{X}) = s$.

**Proof.** By Proposition 3.2.24, the set $K_s$ is prox-regular at $\lambda(\overline{X})$ if and only if $\|\lambda(\overline{X})\|_0 = s$. The result now follows from [62, Th. 9]. □

We now turn our attention to global behaviour.

### 3.3 Global Convergence for Half-Spaces

In the non-convex feasibility problems for which Douglas–Rachford methods have been successful, it is the methods’ apparent global convergence properties which deserve greater attention, and hence on which we shall now focus. This is the case, for instance, in problems having discrete or combinatorial constraints where local convergence can be deduced from the general theory of convex sets as a consequence of the local convexity of the constraints. Furthermore, an implementation of a Douglas–Rachford method would always round the current iterate, and check if this rounded iterate is a solution to the problem. That is, in practice, the algorithm is never run locally.

The purpose of this section is to analyse global properties of the basic Douglas–Rachford method. Of the two approaches already discussed, ours bears more resemblance to the latter. We focus on a two set feasibility problem with the first set allowed to be quite general; it must satisfy either an assumption which encompasses all (weakly) compact sets, or a well-quasi-ordering property with respect to a quasi-order induced by the distance function. Both assumptions always hold for finite sets. The trade-off for this assumption is that a more constrained structure is required of the second set; it must be a closed half-space. The fact that this analysis is far from immediate, illustrates the subtleness of the method’s global behaviour, even for common problems.

Throughout this section we consider the two set feasibility problem

$$\text{find } x \in H \cap Q, \quad (3.14)$$
where \( Q \subseteq H \) is a proximal set, and \( H \subseteq H \) is a (closed) half-space. We will be concerned with the case in which \( Q \) has additional properties (see Assumptions 3.3.11 and 3.3.21), but is intended to be as general as possible; these assumptions are explicitly stated where needed. It is convenient to represent \( H \), and its dividing hyperplane \( L \), in the form

\[
H := \{ x \in H | \langle a, x \rangle \leq b \}, \quad L := \{ x \in H | \langle a, x \rangle = b \},
\]

where \( b \in \mathbb{R} \), and \( a \in H \) with \( \|a\| = 1 \).

Remark 3.3.1. In light of Proposition 1.2.1, the set \( Q \) being proximal is equivalent to it being closed in finite dimensions. In infinite dimensions, assuming \( Q \) to be proximal is the most general setting which allows a projection algorithm to be well-defined.

To solve (3.14) we shall employ the Douglas–Rachford method induced by the two-set Douglas–Rachford operator \( T_{Q,H} : H \rightrightarrows H \) given by

\[
T_{Q,H}(x) := \frac{x + R_H(R_Q(x))}{2}.
\]  

(3.15)

When \( x \) is a fixed point of the Douglas–Rachford operator (i.e., \( x \in T_{Q,H}(x) \)) there is an element of \( P_Q(x) \) which solves (3.14) as is easy to confirm. This suggests that iterating the Douglas–Rachford operator to find a fixed point is a potential method for solving (3.14).

Given an initial point \( x_0 \in H \), we say the sequence \( \{x_k\}_{k \in \mathbb{N}} \) is a Douglas–Rachford iteration if

\[
x_{k+1} \in T_{Q,H}(x_k) \quad \text{for} \quad k \in \mathbb{N}.
\]

We now make explicit the precise form of the Douglas–Rachford operator for our feasibility problem in (3.14).

**Proposition 3.3.2.** The projections onto \( L \) and \( H \) are given by

\[
P_L(x) = x - (\langle a, x \rangle - b)a, \quad P_H(x) = \begin{cases} x & \text{if } \langle a, x \rangle \leq b, \\ x - (\langle a, x \rangle - b)a & \text{if } \langle a, x \rangle > b. \end{cases}
\]

The corresponding reflections are given by

\[
R_L(x) = x - 2(\langle a, x \rangle - b)a, \quad R_H(x) = \begin{cases} x & \text{if } \langle a, x \rangle \leq b, \\ x - 2(\langle a, x \rangle - b)a & \text{if } \langle a, x \rangle > b. \end{cases}
\]

The Douglas–Rachford operator (3.15) may therefore be expressed as the union of single-valued Douglas–Rachford operators, each defined by a singleton and the half-space \( H \), given by

\[
T_{Q,H}(x) = \bigcup_{q \in P_Q(x)} T_{q,H}(x),
\]
where, in an abuse of notation we denoted $T_{q,H} = T_{(q),H}$, and

$$
T_{q,H}(x) := \begin{cases} 
q & \text{if } \langle a, 2q - x \rangle \leq b, \\
q + (\langle a, x \rangle + b - 2\langle a, q \rangle)a & \text{if } \langle a, 2q - x \rangle > b.
\end{cases}
$$

(3.16)

Under this notation, a sequence $\{x_k\}_{k \in \mathbb{N}}$ is a Douglas–Rachford iteration if

$$
x_{k+1} = T_{q_k,H}(x_k)
$$

for some $q_k \in P_Q(x_k)$.

We shall refer to the sequence $\{q_k\}_{k \in \mathbb{N}}$ as an auxiliary sequence for $\{x_k\}$.

An example of a Douglas–Rachford iteration in the plane when $Q$ is a set of four points is given in Figure 3.1. The iteration converges to a solution of (3.14) in eight steps. This behaviour is explained by Theorem 3.3.20.

Figure 3.1: A finitely convergent Douglas–Rachford iteration in $\mathbb{R}^2$ with $Q = \{q_1, q_2, q_3, q_4\}$.

Remark 3.3.3. In practice, an implementation of the Douglas–Rachford iteration would hope to terminate as soon as $q_k \in Q \cap H$ for some $k_0 \in \mathbb{N}$. Observe, however, that (3.16) does not necessarily ensure that the Douglas–Rachford sequence or its auxiliary sequence remain constant for $k \geq k_0$. It is therefore important to distinguish the Douglas–Rachford iteration from an algorithm arising from its implementation (see Algorithm 1).

It is worth emphasizing that, unlike the Douglas–Rachford algorithm, other projection algorithms can fail when applied to (3.14). An example is given in Example 3.3.4. The simplest method from this family is the alternating projection algorithm. It iterates by alternatively applying projectors onto $Q$ and $H$. Precisely, given an initial point $x_0 \in \mathcal{H}$, it generates a sequence $\{x_k\}$ given by $x_{k+1} \in P_H(P_Q(x_k))$. 
Example 3.3.4 (Failure of alternating projections). In general, von Neumann’s alternating projection is unable to find a point in the intersection of $H$ and $Q$ (and hence the same is true for the cyclic Douglas–Rachford algorithm [42, 44]). Figure 3.2 shows a simple example with a doubleton $Q = \{q_1, q_2\} \subset \mathbb{R}^2$. In this example, $P_Q(P_H(q_1)) = q_1$ and the algorithm cycles between $q_1$ and $P_H(q_1)$ for any starting point $x_0 \in P_Q^{-1}(q_1)$.

![Figure 3.2: Failure of the alternating projection algorithm for initial points in $P_Q^{-1}(q_1)$.]

3.3.1 Properties of the Douglas–Rachford Operator

In this section we investigate the behaviour of the Douglas–Rachford operator (3.15) and the corresponding iteration without imposing any additional assumptions on the closed set $Q$.

We begin by distinguishing two cases depending on whether the point is contained in the half-space. Our first proposition shows the image of $H$ under the Douglas–Rachford operator to be a subset of $H$.

**Proposition 3.3.5.** If $x \in H$ then $T_{Q,H}(x) \subset H$.

**Proof.** Choose any $q \in P_Q(x)$. We will distinguish two cases, depending on whether $q$ is contained in $H$. If $q \notin H$, then

$$\langle a, 2q - x \rangle = 2(\langle a, q \rangle - b) + (b - \langle a, x \rangle) + b > b,$$
whence, $T_{q,H}(x) = q + (\langle a, x \rangle + b - 2\langle a, q \rangle)a$. Thus,

$$\langle a, T_{q,H}(x) \rangle = \langle a, q \rangle + \langle a, x \rangle + b - 2\langle a, q \rangle \leq 2b - \langle a, q \rangle < b,$$

and we have $T_{q,H}(x) \in H$.

Suppose now that $q \in H$. If $\langle a, 2q - x \rangle \leq b$ we have $T_{q,H}(x) = q \in H$. Otherwise, if $\langle a, x \rangle + b < 2\langle a, q \rangle$, we have $T_{q,H}(x) = q + (\langle a, x \rangle + b - 2\langle a, q \rangle)a$ and

$$\langle a, T_{q,H}(x) \rangle = \langle a, x \rangle + b - \langle a, q \rangle < \langle a, q \rangle \leq b.$$

Thus, $T_{q,H}(x) \in H$.  

Our second proposition characterizes behaviour of the Douglas–Rachford operator for points which lie outside the half-space.

**Proposition 3.3.6.** Suppose that $x \notin H$ and $q \in P_Q(x)$. The following holds:

(a) if $q \in H$, then $T_{q,H}(x) = q$;

(b) if $q \notin H$, then

(i) if $d(x, H) \geq 2d(q, H)$, then $T_{q,H}(x) = q$ and $T_{q,H}(T_{q,H}(x)) = P_L(q)$;

(ii) if $d(x, H) < 2d(q, H)$, then $T_{q,H}(x) = q + (\langle a, x \rangle + b - 2\langle a, q \rangle)a$, and

(A) if $d(x, H) \leq d(q, H)$, then $T_{q,H}(x) \in H$;

(B) if $d(x, H) > d(q, H)$, then $d(T_{q,H}(x), H) = d(x, H) - d(q, H)$. Furthermore, if $q \in P_Q T_{q,H}(x)$, then $T_{q,H}^2(x) \in H$.

**Proof.** (a): If $q \in H$, then

$$\langle a, 2q - x \rangle = 2(\langle a, q \rangle - b) + b + (b - \langle a, x \rangle) < b;$$

whence, $T_{q,H}(x) = q$, as claimed.

(b)(i): We have $d(x, H) \geq 2d(q, H) \iff \langle a, 2q - x \rangle \leq b$. Thus, $T_{q,H}(x) = q$ and

$$T_{Q,H} T_{q,H}(x) = T_{q,H}(q) = q - (\langle a, q \rangle - b)a = P_L(q),$$

as required.

(b)(ii)(A): If $\langle a, x \rangle - b = d(x, H) \leq d(q, H) = \langle a, q \rangle - b$, then

$$\langle a, T_{q,H}(x) \rangle = \langle a, q \rangle + \langle a, x \rangle + b - 2\langle a, q \rangle = \langle a, x - q \rangle + b \leq b,$$

whence, $T_{q,H}(x) \in H$.

(b)(ii)(B): If $d(x, H) > d(q, H)$, then

$$\langle a, T_{q,H}(x) \rangle = \langle a, x - q \rangle + b > b,$$
and
\[ d(T_{q,H}(x), H) = \langle a, x - q \rangle = (\langle a, x \rangle - b) - (\langle a, q \rangle - b) = d(x, H) - d(q, H). \]

Further, suppose that \( q \in P_Q(T_{q,H}(x)) \). Then,
\[
\langle a, 2q - T_{q,H}(x) \rangle = \langle a, 2q - q - (\langle a, x \rangle + b - 2\langle a, q \rangle) a \rangle
= \langle a, q \rangle - (\langle a, x \rangle - b) + 2(\langle a, q \rangle - b)
= \langle a, q \rangle - d(x, H) + 2d(q, H) > b.
\]

Hence,
\[
T_{q,H}^2(x) = q + (\langle a, q + (\langle a, x \rangle + b - 2\langle a, q \rangle) a \rangle + b - 2\langle a, q \rangle) a
= q + (\langle a, x \rangle - 3\langle a, q \rangle + 2b)a.
\]

Finally, we have
\[
\langle a, T_{q,H}^2(x) \rangle = \langle a, x - b \rangle - 2(\langle a, q \rangle - b) + b
= d(x, H) - 2d(q, H) + b < b;
\]
whence, \( T_{q,H}^2(x) \in H. \]

By combining Propositions 3.3.5 and 3.3.6, we shall deduce the following lemma concerning the behaviour of Douglas–Rachford iterations which never enter the half-space.

**Lemma 3.3.7.** Let \( \{x_k\} \) be a Douglas–Rachford sequence with auxiliary sequence \( \{q_k\} \). If \( x_k \notin H \) for each \( k \in \mathbb{N} \), then \( q_k \notin H \) for all \( k \in \mathbb{N} \), the sequence \( \{d(x_k, L)\} \) is strictly decreasing, and
\[
\lim_{k \to \infty} d(x_k, L) = \lim_{k \to \infty} d(q_k, L) = 0.
\]

**Proof.** If \( x_k \notin H \) for all \( k \), by Propositions 3.3.5 and 3.3.6, it must be that \( d(q_k, H) < d(x_k, H) < 2d(q_k, H) \) and \( d(x_{k+1}, H) = d(x_k, H) - d(q_k, H) \). Thus the sequence \( \{d(x_k, H)\} \) is strictly decreasing and bounded below by zero. Since \( \lim_{k \to \infty} d(x_k, H) \) exists, we deduce
\[
\lim_{k \to \infty} d(q_k, H) = \lim_{k \to \infty} \left[ d(x_k, H) - d(x_{k+1}, H) \right] = 0.
\]
Thus \( \lim_{k \to \infty} d(q_k, L) = \lim_{k \to \infty} d(q_k, H) = 0 \), and, as \( d(x_k, H) < 2d(q_k, H) \), we have \( \lim_{k \to \infty} d(x_k, L) = \lim_{k \to \infty} d(x_k, H) = 0. \)
3.3. Global Convergence for Half-Spaces

We now turn our attention to the precise structure of the Douglas–Rachford operator at points which lie within the half-space. The following proposition gives a relationship between consecutive terms in a Douglas–Rachford sequence.

**Proposition 3.3.8.** For any \( x \in H \) and any \( q \in P_Q(x) \) such that \( q \notin H \) one has

\[
T_{q,H}(x) = q - (d(x, L) + 2d(q, L))a,
\]

\[
d(T_{q,H}(x), L) = d(q, L) + d(x, L).
\]

**Proof.** Since \( x \in H \) and \( q \notin H \) we have that \( \langle a, q \rangle > b \geq \langle a, x \rangle \). Thus,

\[
\langle a, 2q - x \rangle = \langle a, q \rangle + \langle a, q - x \rangle > b,
\]

and we have

\[
T_{q,H}(x) = q - (-\langle a, x \rangle - b + 2\langle a, q \rangle)a = q - (b - \langle a, x \rangle + 2(\langle a, q \rangle - b))
\]

\[
= q - (d(x, L) + 2d(q, L))a.
\]

Then,

\[
d(T_{q,H}(x), L) = |\langle a, q \rangle - b - d(x, L) - 2d(q, L)| = d(q, L) + d(x, L),
\]

which completes the proof. \( \square \)

When the Douglas–Rachford point \( x \) lies in \( H \), our next proposition relates \( x \) to auxiliary points which lie in \( Q \setminus H \).

**Proposition 3.3.9.** Let \( x \in H \) and \( q \in P_Q(x) \setminus H \). Then, if \( p \in P_Q(T_{q,H}(x)) \setminus H \), with \( p \neq q \), one has

\[
d(p, H) + \|T_{q,H}(x) - q\| \leq d(q, H) + d(T_{q,H}(x), Q). \tag{3.17}
\]

Furthermore, we have \( d(p, H) < d(q, H) \).

**Proof.** Let \( z := T_{q,H}(x) \in H \). Since \( p \in P_Q(z) \), we have \( \|z - q\| \geq \|z - p\| \). Observe that

\[
0 < d(p, H) + d(z, L) = \langle a, p \rangle - b + b - \langle a, z \rangle = \langle a, p - z \rangle \leq \|p - z\|. \tag{3.18}
\]

If the inequality in (3.18) is not strict, there is some \( \lambda \in \mathbb{R} \) such that \( \lambda a = p - z \). Then,

\[
\langle a, p - z \rangle = \|p - z\| \implies \lambda = \lambda \|a\|^2 = \|p - z\| > 0.
\]

If \( \|z - q\| = \|z - p\| = \lambda \), using Proposition 3.3.8, we obtain a contradiction with the assumption that \( p \neq q \):

\[
\|p - q\| = \|\lambda a + z - q\| = \|\lambda a - (d(x, L) + 2d(q, L))a\| = |\lambda - \|z - q\|| = 0.
\]
Thus, \( \| z - q \| > \| z - p \| = \lambda \), and by Proposition 3.3.8 we have
\[
d(p, H) - d(q, H) = \langle a, p - q \rangle = \langle a, \lambda a + z - q \rangle
\]
\[
= \lambda + \langle a, z - q \rangle = \lambda - (d(x, \mathcal{L}) + 2d(q, \mathcal{L}))
\]
\[
= \lambda - \| z - q \| < 0,
\]
whence,
\[
d(p, H) + \| z - q \| = d(q, H) + \| z - p \| = d(q, H) + d(z, Q),
\]
and we are done.

Otherwise, suppose that the inequality in (3.18) is strict, i.e.,
\[
d(p, H) + d(z, \mathcal{L}) < \| p - z \|.
\]
By Proposition 3.3.8 we know that
\[
\| z - q \| = d(x, \mathcal{L}) + 2d(q, \mathcal{L}) \quad \text{and} \quad d(z, \mathcal{L}) = d(q, \mathcal{L}) + d(x, \mathcal{L});
\]
whence,
\[
\| z - q \| = d(q, \mathcal{L}) + d(z, \mathcal{L}).
\]
Thus, by (3.20),
\[
d(q, H) = d(q, \mathcal{L}) = \| z - q \| - d(z, \mathcal{L}) > \| z - q \| - \| z - p \| + d(p, H) \geq d(p, H),
\]
which proves (3.17). The last assertion in the statement follows from (3.19) and the inequality above.

Remark 3.3.10. In particular, Proposition 3.3.9 shows that once a Douglas–Rachford sequence enters the half-space, it is not possible for its auxiliary sequence to cycle within points from \( Q \setminus H \). Indeed, once an element \( q_k \in Q \setminus H \) appears as the \( k_0 \)-th term of the auxiliary sequence either the sequence remains constant with \( q_k = q_{k_0} \) for all \( k \geq k_0 \), or there exists \( k_1 \geq k_0 \) such that \( q_k \not\in \mathcal{P}(x_k) \) for all \( k \geq k_1 \).

### 3.3.2 Convergence of the Douglas–Rachford Algorithm

In this section we establish our main results which analyse the global convergence properties of the Douglas–Rachford method assuming some additional structure on the closed set \( Q \). These assumptions encompass, but are not limited to, the setting in which \( Q \) is a finite set, compact set, or a weakly compact set.

Consider the set \( Q \setminus H \) equipped with the binary relation \( \lesssim \) defined, for all \( p, p' \in Q \setminus H \), by
\[
p \lesssim p' \iff d(p, H) \leq d(p', H).
\]
Since this relation is reflexive and transitive, it defines a quasi-ordering on the set \( Q \setminus H \).

We first consider the case in which the following assumption holds.
3.3. GLOBAL CONVERGENCE FOR HALF-SPACES

Assumption 3.3.11. The relation $\preceq$ is a well-quasi-ordering on $Q \setminus H$. That is, any sequence $\{z_k\}$ contained in $Q \setminus H$ contains a pair of terms such that $z_i \preceq z_j$ and $i < j$.

Remark 3.3.12. Assumption 3.3.11 is equivalent to requiring that there exists no sequence $\{p_k\}$ with $p_k \in Q \setminus H$, for all $k \in \mathbb{N}$, such that $\{d(p_k, H)\}$ is a strictly decreasing sequence.

The following lemma shows that, under this assumption, any Douglas–Rachford sequence eventually enters the half-space.

Lemma 3.3.13. Suppose Assumption 3.3.11 holds. Then any Douglas–Rachford sequence $\{x_k\}$ enters and remains in $H$ after a finite number of steps. That is, there exists $k_0 \in \mathbb{N}$ such that $x_k \in H$ for all $k \geq k_0$.

Proof. Let $\{x_k\}$ be a Douglas–Rachford sequence with auxiliary sequence $\{q_k\}$ such that $x_k \notin H$ for all $k \in \mathbb{N}$. By Propositions 3.3.5 and 3.3.6, $q_k \notin H$ for all $k \in \mathbb{N}$, and

$$d(x_{k+1}, H) = d(x_k, H) - d(q_k, H) \quad \text{for all } k \in \mathbb{N}.$$  

By telescoping we obtain

$$0 < d(x_{k+1}, H) = d(x_1, H) - \sum_{j=1}^{k} d(q_j, H).$$  

We therefore deduce that $\sum_{j=1}^{k} d(q_j, H) < d(x_1, H)$ for all $k \in \mathbb{N}$, hence the series is convergent, and in particular $d(q_k, H) \to 0$. For all $k \in \mathbb{N}$, $q_k \notin H$ hence $d(q_k, H) > 0$ for all $k \in \mathbb{N}$. Thus there exists a subsequence $\{q_{k_j}\}$ such that $\{d(q_{k_j}, H)\}$ is strictly decreasing. This contradicts Assumption 3.3.11 and completes the proof.

The following example shows that the conclusions of Lemma 3.3.13 need not hold without something akin to Assumption 3.3.11.

Example 3.3.14. Consider the following subsets of the real line:

$$Q := \left\{ \frac{2}{3^k} : k = 0, 1, 2, \ldots \right\} \cup \{0\}, \quad H := \{ x \in \mathbb{R} : x \leq 0 \}.$$  

For initial point $x_0 = 1$, the Douglas–Rachford iteration and auxiliary sequence are

$$x_k = \frac{1}{3^k}, \quad q_k := P_Q(x_k) = \frac{2}{3^{k+1}}.$$  

Both $\{x_n\}$ and $\{q_n\}$ are positive real numbers, and hence never enter $H$.

We shall also require to following result regarding Assumption 3.3.11.
Suppose Assumption 3.3.11 holds. Let \( \{x_k\} \) be a Douglas–Rachford sequence with auxiliary sequence \( \{q_k\} \) such that \( q_k \notin H \) for all \( k \in \mathbb{N} \). Then the auxiliary sequence is eventually constant. That is, there exists \( k_1 \in \mathbb{N} \) such that \( q_k = q_{k_1} \) for all \( k \geq k_1 \).

**Proof.** Suppose that the auxiliary sequence \( \{q_k\} \) is not eventually constant. By Lemma 3.3.13, we may assume, without loss of generality, that \( x_k \in H \) for all \( k \in \mathbb{N} \). Using Proposition 3.3.9 and the fact that \( \{q_k\} \) is not eventually constant, we deduce the existence of a subsequence \( \{q_{k_j}\} \) such that \( \{d(q_{k_j}, H)\} \) is strictly decreasing. This contradicts Assumption 3.3.11 and completes the proof. \[ \square \]

We formulate now our first main result regarding convergence of the method under Assumption 3.3.11.

**Theorem 3.3.16.** Suppose Assumption 3.3.11 holds. Let \( \{x_k\} \) be a Douglas–Rachford sequence with auxiliary sequence \( \{q_k\} \). Then either: (a) there exists \( k_0 \in \mathbb{N} \) such that \( q_{k_0} \in Q \cap H \neq \emptyset \), or (b) \( H \cap Q = \emptyset \). Moreover, in the latter case, \( \|x_k\| \to +\infty \).

**Proof.** Suppose \( q_k \notin H \) for all \( k \in \mathbb{N} \). By Lemma 3.3.13, there is some \( k_0 \in \mathbb{N} \) such that \( x_k \in H \) for all \( k \geq k_0 \). By Lemma 3.3.15, there is some \( k_1 \in \mathbb{N} \) with \( k_1 \geq k_0 \) and some \( q \in Q \setminus H \) such that \( P_Q(x_k) = q \) for all \( k \geq k_1 \). Then, by Proposition 3.3.8,

\[
x_{k+1} = q - (d(x_k, L) + 2d(q, L))a = q - (d(x_{k-1}, L) + 3d(q, L))a
\]

for all \( k \geq k_1 \). Hence,

\[
\|x_{k+1}\| \geq d(x_{k_1}, L) + (2 + k - k_1)d(q, L) - \|q\|,
\]

whence, \( \|x_k\| \to \infty \). Further, suppose by contradiction that there is \( p \in H \cap Q \). For any \( k \geq k_1 \), by (3.21), one has

\[
\|x_{k+1} - p\|^2 = \|(x_{k+1} - q) + (q - p)\|^2
= \|x_{k+1} - q\|^2 + \|q - p\|^2 + 2\langle x_{k+1} - q, q - p \rangle
= \|x_{k+1} - q\|^2 + \|q - p\|^2 - 2\langle a, q - p \rangle(d(x_{k_1}, L) + (2 + k - k_1)d(q, L)).
\]

Observe that, since \( q \notin H \) and \( p \in H \), we have

\[
\langle a, q - p \rangle = \langle a, q - b \rangle + \langle b - \langle a, p \rangle \rangle > 0.
\]

Thus, there is some \( k_2 \in \mathbb{N} \), with \( k_2 \geq k_1 \), such that

\[
\|q - p\|^2 - 2\langle a, q - p \rangle(d(x_{k_1}, L) + (2 + k - k_1)d(q, L)) < 0, \quad \forall k \geq k_2.
\]
3.3. GLOBAL CONVERGENCE FOR HALF-SPACES

Hence,

\[ \| x_{k+1} - p \|^2 < \| x_{k+1} - q \|^2, \quad \forall k \geq k_2, \]

that is, we obtain a contradiction with the fact that \( P_Q(x_{k+1}) = q \).

Theorem 3.3.16 suggests the Algorithm 1 for finding a point \( Q \cap H \) which can be justified by Corollary 3.3.17.

**Algorithm 1:** The DR algorithm for solving (3.14) under Assumption 3.3.11.

\[
\begin{align*}
\text{input} & : x_0 \in H \\
& \text{Choose any } q_0 \in P_Q(x_0); \\
& \text{Set } k := 0; \\
\text{while } q_k \not\in H \text{ do} & \\
& \quad x_{k+1} := T_{q_k,H}(x_k); \\
& \quad \text{Choose any } q_{k+1} \in P_Q(x_{k+1}); \\
& \quad k := k + 1;
\end{align*}
\]

**output:** \( q_k \in Q \cap H \)

**Corollary 3.3.17.** Suppose Assumption 3.3.11 holds. Then Algorithm 1 either:

(a) terminates finitely to a point in \( H \cap Q \), or (b) \( Q \cap H = \emptyset \) and \( \| x_k \| \to +\infty \).

**Proof.** Follows directly from Theorem 3.3.16.

It is easily seen that any finite set satisfies Assumption 3.3.11, hence Theorem 3.3.16 applies. However, as we now show, a stronger result holds in this special case. We first require the following two lemmas.

**Lemma 3.3.18.** Let \( \{x_k\} \) be a Douglas–Rachford sequence with auxiliary sequence \( \{q_k\} \). Suppose there exists \( k_0 \in \mathbb{N} \) such that \( x_{k_0} \in H \) and \( q_{k_0} \in H \). Then \( q_k \in H \) and \( d(q_{k+1}, L) \geq d(q_k, L) \) for all \( k \geq k_0 \). Furthermore, for any \( k \geq k_0 \), one has \( d(q_{k+1}, L) = d(q_k, L) \) if and only if \( q_{k+1} = q_k \).

**Proof.** We distinguish two cases. First suppose \( \langle a, 2q_k - x_k \rangle \leq b \). By (3.16) we have \( x_{k_0+1} = q_{k_0} \) and hence \( q_{k_0+1} = q_{k_0} \).

Now suppose \( \langle a, 2q_k - x_k \rangle > b \). By (3.16),

\[
\begin{align*}
x_{k_0+1} &= q_{k_0} - (\langle a, 2q_k - x_k \rangle - b)a = q_{k_0} - d(2q_k - x_k, H)a. \quad (3.23)
\end{align*}
\]

Then, as \( q_{k_0+1} \in P_Q(x_{k_0+1}) \), we have

\[
\begin{align*}
\langle a, q_{k_0+1} \rangle &= \langle a, q_{k_0+1} - x_{k_0+1} \rangle + \langle a, x_{k_0+1} \rangle \\
&\leq ||q_{k_0+1} - x_{k_0+1}|| + \langle a, q_{k_0} \rangle - d(2q_k - x_k, H) \quad (3.24) \\
&\leq ||q_{k_0} - x_{k_0+1}|| + \langle a, q_{k_0} \rangle - d(2q_k - x_k, H) \quad (3.25) \\
&= \langle a, q_{k_0} \rangle \leq b,
\end{align*}
\]
whence \( q_{k_0+1} \in H \) and \( d(q_{k_0+1}, L) \geq d(q_{k_0}, L) \).

Further, let us assume that \( d(q_{k_0+1}, L) = d(q_{k_0}, L) \). Then, we have \( \langle a, q_{k_0+1} \rangle = \langle a, q_{k_0} \rangle \). Therefore the inequalities in (3.24) and (3.25) must be equalities, from where we deduce

\[
\langle a, q_{k_0+1} - x_{k_0+1} \rangle = \|q_{k_0+1} - x_{k_0+1}\| = \|q_{k_0} - x_{k_0+1}\|.
\]  

(3.26)

Hence, by (3.26) and (3.23), we have

\[
\langle q_{k_0+1} - x_{k_0+1}, x_{k_0+1} - q_{k_0} \rangle = \langle q_{k_0+1} - x_{k_0+1}, -d(2q_{k_0} - x_{k_0}, H)a \rangle
\]

\[
= -\|x_{k_0+1} - q_{k_0}\| \langle q_{k_0+1} - x_{k_0+1}, a \rangle
\]

\[
= -\|x_{k_0+1} - q_{k_0}\|^2.
\]

Thus, using again (3.26), we obtain

\[
\|q_{k_0+1} - q_{k_0}\|^2 = \|q_{k_0+1} - x_{k_0+1}\|^2 + \|x_{k_0+1} - q_{k_0}\|^2 + 2 \langle q_{k_0+1} - x_{k_0+1}, x_{k_0+1} - q_{k_0} \rangle
\]

\[
= \|q_{k_0+1} - x_{k_0+1}\|^2 - \|x_{k_0+1} - q_{k_0}\|^2 = 0;
\]

that is, \( q_{k_0+1} = q_{k_0} \). The result now follows by induction.

\[\Box\]

**Lemma 3.3.19.** Let \( \{x_k\} \) be a Douglas–Rachford sequence with auxiliary sequence \( \{q_k\} \). Suppose there exists \( k_0 \in \mathbb{N} \) such that \( x_k \in H \) and \( q_k = q \in H \) for all \( k \geq k_0 \). Then there exists \( k_1 \in \mathbb{N} \) with \( k_1 \geq k_0 \) such that \( x_k = x_{k_1} \) for all \( k \geq k_1 \).

**Proof.** If \( \langle a, 2q - x_j \rangle \leq b \) for some \( j \geq k_0 \), then by (3.16) we deduce \( x_k = q \) for all \( k \geq j + 1 \) and we are done.

Otherwise, by (3.16), we have \( x_{k+1} = q + (\langle a, x_k \rangle + b - 2\langle a, q \rangle)a \) for all \( k \geq k_0 \).

Thus,

\[
\langle a, 2q - x_{k+1} \rangle = \langle a, 2q - (q + (\langle a, x_k \rangle + b - 2\langle a, q \rangle)a) \rangle
\]

\[
= \langle a, 2q - x_k \rangle - (b - \langle a, q \rangle)
\]

\[
= \langle a, 2q - x_k \rangle - d(q, L),
\]

whence,

\[
b < \langle a, 2q - x_k \rangle = \langle a, 2q - x_{k_0} \rangle - (k - k_0)d(q, L).
\]

This implies \( d(q, L) = 0 \) and then,

\[
x_{k+1} = q + ((\langle a, x_k \rangle + b - 2\langle a, q \rangle)a = q + ((\langle a, x_k \rangle - b)a,
\]

from which we deduce \( x_k = q + ((\langle a, x_k \rangle - b)a \) for all \( k \geq k_0 + 1 \). This completes the proof. \[\Box\]
3.3. GLOBAL CONVERGENCE FOR HALF-SPACES

The following theorem is a refinement of Theorem 3.3.16 when $Q$ is assumed to be finite.

**Theorem 3.3.20.** Suppose $Q$ is finite. Let $\{x_k\}$ be a Douglas–Rachford sequence with auxiliary sequence $\{q_k\}$. Then either: (a) $\{x_k\}$ and $\{q_k\}$ are eventually constant and the limit of $\{q_k\}$ is contained in $H \cap Q \neq \emptyset$, or (b) $H \cap Q = \emptyset$ and $\|x_k\| \to +\infty$.

**Proof.** By Theorem 3.3.16, it suffices to show that the sequences $\{x_k\}$ and $\{q_k\}$ are eventually constant and the limit of $\{q_k\}$ is contained in $H \cap Q \neq \emptyset$, assuming there exists a $k_0 \in \mathbb{N}$ such that $x_{k_0} \in H$ and $q_{k_0} \in H$.

To this end, suppose there is some $k_0 \in \mathbb{N}$ such that $x_{k_0} \in H$ and $q_{k_0} \in H$. By Proposition 3.3.5 and Lemma 3.3.18 we have $x_k \in H, q_k \in H$ and $d(q_{k+1}, L) \geq d(q_k, L)$ with equality if and only if $q_{k+1} = q_k$, for all $k \geq k_0$. Since $Q$ is finite, the latter implies that there exists $k_1 \geq k_0$ such that $q_k = q_{k_1}$ for all $k \geq k_1$. By Lemma 3.3.19, there exists a $k_2 \geq k_1$ such $x_k = x_{k_2}$ for all $k \geq k_2$. This completes the proof. □

Define the mapping

$$Q(\cdot) := \{p \in Q \mid d(p, H) \leq d(\cdot, H)\}.$$ 

We now consider the case in which one of the following assumptions hold. In particular, they include the cases in which $Q$ is (weakly) compact.

**Assumption 3.3.21.** The function $\iota_Q + d(\cdot, H)$ has compact lower-level sets. In particular, for every $q \in Q$, the set $Q(q)$ is compact. ◊

**Assumption 3.3.22.** The function $\iota_Q + d(\cdot, H)$ has weakly compact lower-level sets. In particular, for every $q \in Q$, the set $Q(q)$ is weakly compact. ◊

In finite dimension, Assumptions 3.3.21 and 3.3.22 may also be written in the form

$$\lim_{x \in Q, \|x\| \to \infty} d(x, H) = +\infty.$$ 

The following is our final main result. It characterizes behavior of the algorithm for $Q$ satisfying Assumption 3.3.21 (respectively Assumption 3.3.22). In particular, it shows that the Douglas–Rachford algorithm can be used to determine consistency of the feasibility problem (3.14) and to find a solution when it exists. Since the proofs are similar, we prove the result under both assumptions simultaneously.

**Theorem 3.3.23.** Suppose Assumption 3.3.21 holds (or Assumption 3.3.22 holds). Let $\{x_k\}$ be a Douglas–Rachford sequence with auxiliary sequence $\{q_k\}$. Then either: (a) $d(q_k, H) \to 0$ and the set of (weak) cluster points of the auxiliary sequence is non-empty and contained in $Q \cap H$, or (b) $d(q_k, H) \to \beta$ for some $\beta > 0$ and $H \cap Q = \emptyset$. Moreover, in the latter case, $\|x_k\| \to +\infty$. 

CHAPTER 3. NONCONVEX THEORY

Proof. Let \( \{x_k\} \) be a Douglas–Rachford sequence with auxiliary sequence \( \{q_k\} \). We distinguish two cases.

First suppose that \( x_k \notin H \) for all \( k \). By Proposition 3.3.6, \( q_k \notin H \) for all \( k \) and, by Lemma 3.3.7, \( d(q_k, H) \to 0 \). Hence there exists \( k_0 \in \mathbb{N} \) such \( d(q_k, H) \leq d(q_{k_0}, H) \) for all \( k \geq k_0 \), whence \( q_k \in Q(q_{k_0}) \) for all \( k \geq k_0 \). Since \( Q(q_{k_0}) \) is (weakly) compact and \( d(\cdot, H) \) is (weakly) continuous, the set of (weak) cluster points of the auxiliary sequence \( \{q_k\} \) is non-empty and contained in \( Q \cap H \).

Next suppose there is some \( k_0 \in \mathbb{N} \) such that \( x_{k_0} \in H \). Then, by Proposition 3.3.5, we have \( x_k \in H \) for all \( k \geq k_0 \). On one hand, if there exists \( k_1 \geq k_0 \) such that \( q_{k_1} \in H \) then, by Lemma 3.3.18, \( q_k \in H \) for all \( k \geq k_1 \). Since \( Q(q_{k_1}) = Q \cap H \) is (weakly) compact, it follows that the set of (weak) cluster points of the auxiliary sequence \( \{q_k\} \) is non-empty and contained in \( Q \cap H \), and we are done.

On the other hand, suppose \( q_k \notin H \) for all \( k \geq k_0 \). Then by Proposition 3.3.9, \( q_k \in Q(q_{k_0}) \) for all \( k \geq k_0 \), and \( \{d(q_k, H)\}_{k=k_0}^{\infty} \) is decreasing and bounded below by zero, hence

\[
\beta := \inf_{k \geq k_0} d(q_k, H) = \lim_{k \to \infty} d(q_k, H) \geq 0.
\]

By Proposition 3.3.8, for all \( k \geq k_0 \),

\[
x_{k+1} = q_k - (d(x_k, L) + 2d(q_k, L))a \\
= q_k - (d(x_{k-1}, L) + d(q_{k-1}, L) + 2d(q_k, L))a \\
= \ldots = q_k - \left( d(x_{k_0}, L) + d(q_k, L) + \sum_{j=k_0}^{k} d(q_j, L) \right) a.
\]

For all \( k \geq k_0 \), we may therefore express \( x_{k+1} = q_k - \lambda_k a \) where

\[
\lambda_k := d(x_{k_0}, L) + d(q_k, L) + \sum_{j=k_0}^{k} d(q_j, L) \geq (k - k_0) \beta \geq 0. \tag{3.27}
\]

If \( \beta = 0 \) then, by the (weak) compactness of \( Q(q_{k_0}) \) and (weak) continuity of \( d(\cdot, H) \), the set of (weak) cluster points of the auxiliary sequence \( \{q_k\} \) is non-empty and contained in \( Q \cap H \).

Conversely, assume \( \beta > 0 \). Since it is (weakly) compact, the set \( Q(q_{k_0}) \) is bounded, hence there exists \( K > 0 \) such that \( \|q\| \leq K \) for all \( q \in Q(q_{k_0}) \), and thus

\[
\|x_{k+1}\| \geq \lambda_k - \|q_k\| \geq \lambda_k - K \geq (k - k_0) \beta - K \to +\infty. \tag{3.28}
\]

To complete the proof, we must show \( Q \cap H = \emptyset \). To this end, suppose there is a \( p \in Q \cap H \). We claim that, for all \( k \geq k_0 \),

\[
\|x_k - q_k\| > \left\| x_k - P_L(q_k) - \frac{\beta}{2} a \right\|.
\]
3.3. GLOBAL CONVERGENCE FOR HALF-SPACES

To prove this claim, first observe

\[
\left(\langle a, q_k \rangle - b - \frac{\beta}{2}\right) + 2\langle a, x_k - q_k \rangle = -\left(\langle a, q_k \rangle - b\right) - 2\left(b - \langle a, x_k \rangle\right)
= -d(q_k, L) - \frac{\beta}{2} - 2d(x_k, L) < 0.
\]

Since \(\langle a, q_k \rangle - b - \frac{\beta}{2} = d(q_k, L) - \frac{\beta}{2} \geq \beta - \frac{\beta}{2} = \beta - \frac{\beta}{2} > 0\), we deduce the claimed result

\[
\left\| x_k - P_L(q_k) - \frac{\beta}{2}a \right\|^2 = \left\| x_k - q_k + \left(\langle a, q_k \rangle - b - \frac{\beta}{2}\right)a\right\|^2
= \|x_k - q_k\|^2 + \left(\langle a, q_k \rangle - b - \frac{\beta}{2}\right)^2
+ 2\left(\langle a, q_k \rangle - b - \frac{\beta}{2}\right)\langle a, x_k - q_k \rangle
< \|x_k - q_k\|^2.
\]

Further, for all \(k \geq k_0\), we have

\[
\|x_{k+1} - q_{k+1}\|^2 > \left\| x_{k+1} - P_L(q_{k+1}) - \frac{\beta}{2}a \right\|^2
= \|x_{k+1} - p\|^2 + \left\| p - P_L(q_{k+1}) - \frac{\beta}{2}a \right\|^2
+ 2\left( x_{k+1} - p, p - P_L(q_{k+1}) - \frac{\beta}{2}a \right).
\]

Since

\[
\left\langle a, P_L(q_{k+1}) + \frac{\beta}{2}a - p \right\rangle = \left\langle a, q_{k+1} - (\langle a, q_{k+1} \rangle - b) a + \frac{\beta}{2}a - p \right\rangle
= b - \langle a, p \rangle + \frac{\beta}{2} = d(p, L) + \frac{\beta}{2},
\]

we obtain

\[
\left\langle x_{k+1} - p, p - P_L(q_{k+1}) - \frac{\beta}{2}a \right\rangle
= \left\langle q_{k} - \lambda_k a - p, p - P_L(q_{k+1}) - \frac{\beta}{2}a \right\rangle
= \left\langle q_{k} - p, p - P_L(q_{k+1}) - \frac{\beta}{2}a \right\rangle + \lambda_k \left\langle a, P_L(q_{k+1}) + \frac{\beta}{2}a - p \right\rangle
= \left\langle q_{k} - p, p - P_L(q_{k+1}) - \frac{\beta}{2}a \right\rangle + \lambda_k \left( d(p, L) + \frac{\beta}{2} \right).
\]
Then,
\[
\|x_{k+1} - q_{k+1}\|^2 > \|x_{k+1} - p\|^2 + \left\| p - P_L(q_{k+1}) - \frac{\beta}{2}a \right\|^2 \\
+ 2 \left\langle q_k - p, p - P_L(q_{k+1}) - \frac{\beta}{2}a \right\rangle + 2\lambda_k \left( d(p, L) + \frac{\beta}{2} \right).
\]

By the boundedness of \(Q(q_{k_0})\) and since \(H \cap Q \subseteq Q(q_{k_0})\), we have
\[
\eta := \min_{w, z \in Q(q_{k_0})} \left\{ \left\| p - P_L(z) - \frac{\beta}{2}a \right\|^2 + 2 \left\langle w - p, p - P_L(z) - \frac{\beta}{2}a \right\rangle \right\} \in \mathbb{R}.
\]

Therefore,
\[
\|x_{k+1} - q_{k+1}\|^2 > \|x_{k+1} - p\|^2 + \eta + 2\lambda_k \left( d(p, L) + \frac{\beta}{2} \right).
\]

Finally, since \(\lambda_k \to +\infty\) (by (3.27)) and \(d(p, L) + \frac{\beta}{2} > 0\), there is some \(k_1 \geq k_0\) such that \(\eta + 2\lambda_k \left( d(p, L) + \frac{\beta}{2} \right) > 0\) for all \(k \geq k_1\). Then
\[
\|x_{k+1} - q_{k+1}\|^2 > \|x_{k+1} - p\|^2, \quad \forall k \geq k_1,
\]
which contradicts the fact that \(q_{k+1} \in P_Q(x_{k+1})\). We therefore have that \(H \cap Q = \emptyset\) and the proof is complete.

Every compact set satisfies Assumption 3.3.21, and thus we deduce the following important corollary.

**Corollary 3.3.24.** Suppose \(Q\) is a compact set (or a weakly compact set). Let \(\{x_k\}\) be a Douglas–Rachford sequence with auxiliary sequence \(\{q_k\}\). Then either:

(a) \(d(q_k, H) \to 0\) and the set of (weak) cluster points of the auxiliary sequence is non-empty and contained in \(Q \cap H\), or (b) \(d(q_k, H) \to \beta\) for some \(\beta > 0\) and \(H \cap Q = \emptyset\). Moreover, in the latter case, \(\|x_k\| \to +\infty\).

**Proof.** Follows immediately from Theorem 3.3.23.

**Remark 3.3.25** (Rate of divergence). A closer look at the proofs of Theorem 3.3.23 (resp. Theorem 3.3.16) in the case that \(H \cap Q = \emptyset\), shows that (3.28) (resp. (3.22)) gives information regarding the rate of divergence of the Douglas–Rachford sequence. More precisely, the sequence diverges with at least linear rate in the sense that \(\|x_{k+1}\| \geq kM + K\) for some \(M > 0\) and \(K \in \mathbb{R}\).

**Remark 3.3.26.** In general, the Douglas–Rachford sequence need not converge finitely for compact \(Q\) (see Theorem 3.3.20). Example 3.3.14 serves as a counter-example.
Remark 3.3.27 (The auxiliary sequence). During each iteration of the Douglas–Rachford method, the next term in the auxiliary sequence must be selected from the set $P_Q(x_k)$. We note that all the results in this section hold regardless of how these points are chosen.

The relationships amongst our assumptions are summarized in Figure 3.3.

![Diagram showing relationships between assumptions](image-url)

Figure 3.3: The relationships amongst our assumptions.

### 3.3.3 Examples and Counter-Examples

In this section, we give a number of examples which highlight the role of the hyperplane in Theorems 3.3.16 and 3.3.23. These examples are interesting in light of results such as [21, 85] which exploit linear structure to analyse the Douglas–Rachford method.

**Example 3.3.28 (Failure for two half-spaces).** The algorithm no longer remains globally convergent on replacing the half-space by a cone resulting from the intersection of two half-spaces, as is shown in Figure 3.4.

**Example 3.3.29 (Failure for hyperplane).** The Douglas–Rachford operator for $Q$ and $L$ (rather than $H$) is given by

$$T_{Q,L}(x) = \bigcup_{q \in P_Q(x)} T_{q,L}(x) = \bigcup_{q \in P_Q(x)} q + (\langle a, x \rangle + b - 2 \langle a, q \rangle a).$$

In this case, Proposition 3.3.9 no longer holds, hence the algorithm need not converge. An example with cycling behaviour is given in Figure 3.5. Moreover, cycles are still possible, even in the product formulation in terms of the diagonal space (see also Example 3.3.31). Let $C_1 = C_2 := \{0, 1\}$. Consider

$$C' := C_1 \times C_2 = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$$
Figure 3.4: A 2-cycle of the Douglas–Rachford algorithm when $H$ is a cone.

Figure 3.5: A 4-cycle of the Douglas–Rachford algorithm with hyperplane constraint.

and

$$D := \{(x, y) \in \mathbb{R}^2 \mid x = y\}.$$  

If $x_0 = (-1/2, 1)^T$, then $x_1 = (1/4, 3/4)^T$, $x_2 = (3/4, 1/4)^T$ and $x_3 = x_1$.  \hfill \lozenge

Example 3.3.30 (Failure with slab constraints). The algorithm no longer remains
3.3. GLOBAL CONVERGENCE FOR HALF-SPACES

globally convergent on replacing the half-space by a slab constraint. That is,

\[ H := \{ x : c \geq \langle a, x \rangle \geq d \}, \]

with \( d < c \), both finite. An example is given in Figure 3.6.

![Figure 3.6: A 4-cycle of the Douglas–Rachford algorithm with a slab constraint.](image)

**Example 3.3.31 (Failure of the product reformulation).** Pierra’s product space reformulation casts any feasibility problem with finitely many constraints as an equivalent two set problem in a larger product space. As the Douglas–Rachford method can only be directly applied to two set feasibility problems, this reformulation is crucial in many applications.

Consider the constraint sets

\[ H := \{ x \in \mathbb{R}^2 \mid x_2 \leq 1 \}, \quad Q := \{ 0,1 \} \times \{ 0,1 \}. \]

Applied to this problem, the global convergence of the Douglas–Rachford method is covered by Theorem 3.3.20. However, as we will show, the product space reformulation destroys convergence.

Consider the product space \( \mathbb{R}^2 \times \mathbb{R}^2 \) reformulation with constraints

\[ C := H \times Q = \{ (x, y) \mid x \in H, y \in Q \}, \quad D = \{ (x, y) \mid x = y \}. \]

Observe that \( x \in Q \cap H \) if and only if \((x, x) \in C \cap D\). Furthermore, note that neither of the sets \( C \) or \( D \) is a half-space, hence our results no longer apply.

Consider the Douglas–Rachford iteration with the reflection first performed with respect to the diagonal space \( D \) for initial point \((x_0, y_0) = ((0, 2/5), (0, 4/5))\). Then \((x_1, y_1) = ((0, 3/5), (0, 1/5))\) and \((x_2, y_2) = (x_0, y_0)\). That is, a 2-cycle is obtained.

If the reflection was instead performed first with respect to \( H \times Q \) for initial point \((x_0, y_0) = ((0, 4/5), (0, 2/5))\), a 2-cycle is still obtained. In this case, we have \((x_1, y_1) = ((0, 1/5), (0, 3/5))\) and \((x_2, y_2) = (x_0, y_0)\).
Chapter 4

Applications

This chapter considers various applications of projection and reflection methods. Particular attention is paid to those problems in which one or more of the underlying constraint sets lack the required convexity properties for the classical theory to apply but in which the methods, nevertheless, still perform well in practice. While projection and reflection methods have long been successfully used in non-convex imaging applications, it is only relatively recently that their usefulness when applied to problems of a combinatorial nature has been realised. The benefits of our investigation are two-fold. Firstly, the empirical experience gained is useful in understanding the behaviour of the methods viewed as heuristics; an endeavour which may be of interest in its own right. Secondly, a knowledge of the characteristics of non-convex problems which can be successfully solved can be used to guide theoretical developments. The overarching philosophy of this approach draws heavily from the experimental mathematics paradigm [8,9,36].

The novelty in using projection and reflection methods as algorithms for non-convex problems is that they can frequently be directly applied without the use of surrogate convexifications. This approach avoids the issue of recovering a valid solution to the original problem altogether. Furthermore, as we shall soon see, when modelled appropriately the corresponding projectors can be simply and efficiently computed, usually with no more than standard software libraries.

The chapter is organised as follows. In Section 4.1, we consider the problem of reconstructing an unknown signal from a finite number of non-negative moments. For this moment problem, two directions of research are considered. The first investigates strong convergence of algorithms in infinite dimensional Hilbert lattice, and the second, the compressive sensing application in which the unknown signal is known to be sparse. In Section 4.2, various problems arising in combinatorial optimisation are discussed including Sudoku and nonogram puzzles as well as searching for Hadamard matrices. In Section 4.3, we investigate low-dimensional distance matrix reconstruction with detailed case studies of protein conformation...
and ionic liquid bulk structure determination. These sections reveal that many of the successful non-convex applications are, in fact, particular instances of so-called real matrix completion problems. For this reason, Section 4.4 proposes a unified framework to model matrix completion problems as feasibility problems with additional examples. Throughout this chapter, aspects of modelling and efficient implementation are discussed as well as their connections to the theory of the previous chapters.

## 4.1 Reconstruction from Non-Negative Moments

The moment problem involves reconstructing an unknown signal from a finite number of its moments and occurs in various applications [16]. In this section, we consider some special cases of the moment problem in which the unknown signal is known a priori to lie within a convex cone.

In the first case, the setting is a Hilbert lattice, \( \mathcal{H} \), with lattice cone \( \mathcal{H}_+ \). The problem of reconstructing a non-negative signal \( x \in \mathcal{H} \) from a finite number of its moments can be stated abstractly as

\[
\text{find } x \in \{ x \in \mathcal{H}_+ : Ax = b \},
\]

(4.1)

where \( A : \mathcal{H} \to \mathbb{R}^n \) is a linear operator and \( b \in \mathbb{R}^n \) is a vector containing the \( n \)-moments.

It can be easily seen that Problem 4.1 can be formulated as a feasibility problem with two convex sets: an affine subspace and a lattice cone. In this section, we will be interested in the case in which \( \mathcal{H} \) is infinite dimensional (e.g., the signal to be reconstructed is a square-summable function on \( \mathbb{R} \)). As we have seen in Chapter 2, the Hundal counterexample shows that applied to general affine spaces and convex cones, projection and reflection methods can fail to converge strongly. In light of this example, it is natural to ask what assumptions might be required to ensure strong convergence. Questions of this kind have important implications, for instance, for robustness of discretisation schemes. Some partial answers to this question are provided in Section 4.1.1.

In the second case, we return back to the setting of a finite dimensional space, where we consider a further refinement of the non-negative moment problem arising in compressive sensing applications [98]. Roughly speaking, the compressive sensing paradigm considers the case in which the a priori information used to characterise the solutions involves sparse objects. For our purposes, an object is considered sparse when it is expressible in terms of relatively few basis elements. As a concrete example of the paradigm, it is sometimes possible to uniquely solve under-determined linear systems under additional assumptions of sparsity [67]. The difficulty arising in such formulations involving sparsity is in dealing with
poorly behaved sparsity functionals. Two important examples of such functionals are the \( \ell_0 \)-functional for vectors,\(^1\) and the rank function for matrices. The reason for such difficulties is that sparsity functionals lead to problems involving non-convexity and NP-hard complexity [77,117]. The following concrete example highlights the fundamental difficulty.

**Example 4.1.1 (Under-determined linear systems).** Consider the problem of finding the sparsest solution to an under-determined linear system. That is,

\[
\min_{x \in \mathbb{R}^n} \{ \| x \|_0 : Ax = b \},
\]

where \( A \in \mathbb{R}^{m \times n} \) with \( m \ll n \), \( b \in \mathbb{R}^m \), and \( \| \cdot \|_0 : \mathbb{R}^n \to \{0, 1, \ldots, n\} \) denotes the \( \ell_0 \)-functional which maps a vector to the number of non-zero entries it contains.

Problem (4.2) can be reduced to exact cover by 3 sets and hence is NP-hard. Details can be found in [77, Th. 2.17].

A popular approach to addressing the aforementioned difficulty is to employ convex relaxations [39,50,142], thus allowing for application of industrial strength non-linear solvers. The \( \ell_1 \)-norm promotes sparsity and has consequently been used as a surrogate for its \( \ell_0 \) counterpart. Such relaxations come with varying strengths and theoretical guarantees. For an introduction to the topic, we refer the reader to [77, Ch. 4]. Whilst one may be able to exactly solve a relaxation, it is not always the case that this translates into a satisfactory sparse solution of the original problem.

An alternative approach involves attempting to deal with the original problem’s non-convexity directly [5,26,86], and thus avoiding the potential complication of recovering a sparse solution from a convex relaxation. This is the approach we consider here. Moreover, the combinatorial nature of Example 4.1.1 suggests, more broadly, that combinatorial problems are also of interest. Indeed, this situation shall later be considered in Section 4.2. In Section 4.1.2 we apply the results of Section 3.2 to derive regularity conditions for two compressive sensing problems: sparse non-negative and low-rank positive semi-definite solutions to linear systems.

### 4.1.1 Strongly Convergent Algorithms

We recall some definitions and notation regarding cones and lattices. A cone is a set \( K \subseteq \mathcal{H} \) such that \( \mathbb{R}_+ K \subseteq K \). A cone \( K \) is pointed if \( K \cap (-K) = \{0\} \), generating if \( K - K = \mathcal{H} \), and (norm) normal if there exist a (norm) neighbourhood basis, \( \mathcal{V} \), of 0 such that

\[
V = (V + K) \cap (V - K) \text{ for all } V \in \mathcal{V}.
\]

\(^1\)In the literature, the \( \ell_0 \)-functional is also known as the \( \ell_0 \)-norm. This is a misnomer since it fails positive homogeneity.
CHAPTER 4. APPLICATIONS

Given a set $S \subseteq \mathcal{H}$, its negative polar cone is the convex cone

$$S^\ominus := \{ x \in \mathcal{H} : \langle x, S \rangle \leq 0 \}.$$ 

If $S$ is nonempty, $(S^\ominus)^\ominus = \text{cl conv}(\mathbb{R}_+S)$ (see, for example, [34]). In particular, if $K$ is a closed convex cone then $(K^\ominus)^\ominus = K$. The positive polar cone to $S$ is defined similarly and $S^\oplus := -S^\ominus$.

Given a convex pointed cone $K \subseteq \mathcal{H}$, the partial ordering $\leq_K$ on $\mathcal{H}$ induced by $K$ is defined by

$$x \leq_K y \iff y - x \in K.$$  \hspace{1cm} (4.3)

A partially ordered Hilbert space is a Hilbert space, $\mathcal{H}$, ordered by a convex pointed cone denoted $K$ according to (4.3). With respect to such an ordering, the supremum (resp. infimum) of the doubleton $\{x, y\} \subseteq \mathcal{H}$ is denoted $x \lor y$ (resp. $x \land y$). The positive part, negative part and modulus of a point $x \in \mathcal{H}$ are given by

$$x^+ := x \lor 0, x^- := (-x) \lor 0 \text{ and } |x| := x \lor (-x),$$

respectively. Furthermore, $x = x^+ - x^-$ and $|x| = x^+ + x^-$. A Hilbert lattice is a partially ordered Hilbert space such that the supremum of every doubleton exists and the induced norm is Reisz with respect to the ordering. The latter means that

$$|x| \leq_K |y| \Rightarrow \|x\| \leq \|y\|.$$ 

In a Hilbert lattice $\mathcal{H}$, the lattice cone $K$ is characterised by (see, for example, [46, Th. 8])

$$K = K^\oplus = (-K^\ominus) = \{ x \in \mathcal{H} : \langle x, y \rangle \geq 0, \forall y \in K \}.$$  \hspace{1cm} (4.4)

Where there is no ambiguity, we will say that $\mathcal{H}$ is a Hilbert lattice (i.e., without reference to the cone) and denote the order cone by $\mathcal{H}_+$. It is well known that every Hilbert lattice is isomorphic to a space of square integrable functions over an appropriate measure space [45, p. 322].

Remark 4.1.2. In texts on ordered topological vector spaces, it is common (but not uniformly so) to define a “cone” to be both pointed and convex, in addition to being closed under positive scalar multiplication.

The following fact allows one to exploit the order structure induced by a closed convex pointed cone.

**Fact 4.1.3** (Characterisations of (norm) normal cones). Let $K \subseteq \mathcal{H}$ be a closed convex pointed cone, and consider $\mathcal{H}$ equipped with the order $\leq_K$. The following are equivalent.

(a) Every order non-decreasing, norm bounded sequence is strongly convergent.

(b) Every order non-decreasing, order bounded sequence is strongly convergent.
4.1. RECONSTRUCTION FROM NON-NEGATIVE MOMENTS

(c) $K$ is (norm) normal.

Proof. See, for example, [3, Th. 2.45].

A Hilbert space can be expressed as the direct sum of any closed subspace and its orthogonal complement. The following theorem is an analogue for closed convex cones.

Theorem 4.1.4 (Moreau decomposition theorem). Suppose $K \subseteq \mathcal{H}$ is a nonempty closed convex cone. For any $x \in \mathcal{H}$,

(a) $x = P_K x + P_{K^\ominus} x$.
(b) $\langle P_K x, P_{K^\ominus} x \rangle = 0$.
(c) $\|x\|^2 = d^2(x, K) + d^2(x, K^\ominus)$.

Proof. See, for example, [19, Th. 6.29]. For extensions in Banach space see [61].

Suppose we have two sequences $(\lambda_n)_{n=1}^\infty \subseteq \mathcal{H}$, and $(\kappa_n)_{n=1}^\infty \subseteq K \subseteq \mathcal{H}$, for some closed convex cone $K$. Given an initial point $x_0 \in \mathcal{H}$, iteratively define the sequence $(x_n)_{n=1}^\infty$ by

$$x_n := x_{n-1} - \kappa_n + Q \lambda_n,$$

where $Q: \mathcal{H} \to M$ is a linear mapping, and $M$ is a finite dimensional subspace of $\mathcal{H}$. Using the linearity of $Q$, (4.5) implies

$$x_n - x_0 = -\sigma_n + Q \alpha_n,$$

where

$$\sigma_n = \sum_{k=1}^n \kappa_k \in K, \quad \alpha_n := \sum_{k=1}^n \lambda_k.$$

The following examples show that two important projection algorithms fit the aforementioned specification. In both, examples we suppose that $S$ is a closed convex cone, and that $A$ is a closed affine subspace of finite codimension. with $Q$ denoting the projection onto the (finite dimensional) orthogonal complement of the subspace parallel to $A$. That is,

$$P_A x = x + Q(\mathbf{x} - x), \text{ for any } \mathbf{x} \in A.$$

Example 4.1.5 (von Neumann sequences). For any $x_0 \in \mathcal{H}$, the von Neumann sequence is defined by

$$x_{n+1} := P_A P_S x_n,$$

which, for any $\mathbf{x}_n \in A$, is expressible as

$$x_{n+1} = P_S x_n + Q(\mathbf{x}_n - P_S x_n),$$

$$= x_n - P_{S^\ominus} x_n + Q(\mathbf{x}_n - P_S x_n).$$

So here, again $K := S^\ominus$ and $\kappa_{n+1} = P_{S^\ominus} x_n$ while $\lambda_{n+1} = \mathbf{x}_n - P_S x_n$. 

\cqfd
CHAPTER 4. APPLICATIONS

Example 4.1.6 (Douglas–Rachford sequences). For any \( x_0 \in \mathcal{H} \) the Douglas–Rachford sequence is defined by

\[
x_{n+1} := T_{S,A} x_n \text{ where } T_{S,A} := \frac{I + R_A R_S}{2},
\]

which, for any \( x_n \in A \), is expressible as

\[
x_{n+1} = P_S x_n + Q(\pi_n - R_S x_n) = x_n - P_S x_n + Q(\pi_n - R_S x_n).
\]

(4.7)

So in this case, \( K := S^\ominus, \kappa_{n+1} = P_S x_n \) and \( \lambda_{n+1} = \pi_n - R_S x_n \).

We also give a concrete example of the linear operator \( Q \).

Example 4.1.7 (Computation of \( Q \)). As in [16, Sec. 5], let \( \mathcal{H} = \ell_2(\mathbb{N}) \) and define

\[
K := \ell_2^+(\mathbb{N}), \quad A := T^{-1}\lambda,
\]

where \( T : \ell_2 \to \mathbb{R}^N \) is linear, continuous and given by \( x \mapsto (\langle t_i, x \rangle)_{i=1}^N \) for given linearly independent vectors \( t_i \in \ell_2 \). Letting \( Q := T^*(TT^*)^{-1} \), we have as above

\[
P_A x = x + Q(\pi - x), \quad \text{for any } \pi \in A.
\]

Whence,

\[
R_A x = x + 2Q(\pi - x), \quad R_K x = 2x^+ - x = |x|.
\]

For the remainder of this section, unless explicitly stated otherwise, we shall use \( (x_n)_{n=1}^\infty \) to denote a sequence of the form given in (4.5).

To begin our analysis, the following two lemmas give some insight into what might cause the sequence \( (x_n)_{n=1}^\infty \) to fail to converge in norm.

Lemma 4.1.8 (Recession directions). Let \( K \subseteq \mathcal{H} \) be a nonempty closed convex pointed norm normal cone. Suppose \( (x_n)_{n=1}^\infty \) is a bounded sequence of the form given in (4.5). Then, either \( (x_n)_{n=1}^\infty \) contains a norm convergent subsequence or the set of norm cluster points of \( (Q\alpha_n/\|Q\alpha_n\|)_{\{n:Q\alpha_n \neq 0\}} \) is nonempty and contained in \( K \). In particular, the latter implies \( \text{range}(Q) \cap K \neq \{0\} \).

Proof. Since \( (x_n)_{n=1}^\infty \) is bounded, by (4.6), we see that \( (\sigma_n)_{n=1}^\infty \) is bounded if and only if \( (Q\alpha_n)_{n=1}^\infty \) is bounded. We distinguish two cases: (i) \( (Q\alpha_n)_{n=1}^\infty \) contains a bounded subsequence,say \( (Q\alpha_{n_k})_{k=1}^\infty \), or (ii) no subsequence of \( (Q\alpha_n)_{n=1}^\infty \) is bounded.

(i): In this case, by passing to a further subsequence if necessary, we may assume that \( (Q\alpha_{n_k})_{k=1}^\infty \) converges weakly and hence in norm since it is contained within
4.1. RECONSTRUCTION FROM NON-NEGATIVE MOMENTS

a finite dimensional subspace. Further, \((\sigma_n)_{k=1}^\infty\) is bounded, and, along with \(\sigma_n\) itself, increasing with respect to the partial order induced by \(K\), so it converges in norm (by Fact 4.1.3). Equation (4.6) now implies that \((x_n)_{k=1}^\infty\) converges in norm.

(ii): Let \(q_n := Q\alpha_n/\|Q\alpha_n\|\) when \(\|Q\alpha_n\| \neq 0\). And, let \(q\) be an arbitrary norm cluster point of \((q_n)_{n: Q\alpha_n \neq 0}\), which exists because \((q_n)_{n: Q\alpha_n \neq 0}\) is bounded and contained within a finite dimensional subspace. Let \((q_{nk})_{k=1}^\infty\) be a subsequence convergent to \(q\), which by passing to a further subsequence if necessary, we may assume has \(0 < \|Q\alpha_{nk}\| \to +\infty\). Then,

\[
\frac{x_{nk} - x_0}{\|Q\alpha_{nk}\|} = \frac{-\sigma_{nk}}{\|Q\alpha_{nk}\|} + q_{nk} \implies q = \lim_{k \to \infty} \frac{\sigma_{nk}}{\|Q\alpha_{nk}\|}.
\]

This completes the proof. \(\Box\)

Lemma 4.1.9 (Iteration for a hyperplane). Let \(\mathcal{H}\) be a Hilbert lattice with Hilbert cone \(S := \mathcal{H}_+\), \(\kappa_{n+1} := -x_n^-\), and \(Q\) be the projection onto a 1-dimensional subspace. If \((x_n)_{n=1}^\infty\) is bounded and \((x_n)_{n=1}^\infty\) fails to converge in norm, then \(Q(S) \subseteq S\) and range(\(Q\)) \(\subseteq S \cup (-S)\).

Proof. Since the range of \(Q\) has dimension 1, we may write \(Q = \langle a, \cdot \rangle a\) for some \(a\) with \(\|a\| = 1\). Since, for any \(\alpha_n\) with \(Q\alpha_n \neq 0\),

\[
\frac{\langle a, \alpha_n \rangle a}{\|\langle a, \alpha_n \rangle a\|} = \frac{\langle a, \alpha_n \rangle a}{|\langle a, \alpha_n \rangle|} \in \{\pm a\},
\]

we see that the only possible cluster points of \((Q\alpha_n/\|Q\alpha_n\|)_{n \in \{n: Q\alpha_n \neq 0\}}\) are \(\pm a\). Hence, by Lemma 4.1.8, if \((x_n)_{n=1}^\infty\) fails to converge in norm then \(a \in S \cup (-S)\). Since \(\mathcal{H}\) is a Hilbert lattice, it follows that \(Q(S) \subseteq S\) and that, for any \(x \in \mathcal{H}\), \(Qx = \langle a, x \rangle a \in \mathbb{R}a \subseteq S \cup (-S)\). \(\Box\)

The following lemma shows that the sequence \((x_n)_{n=1}^\infty\) converges in norm under additional ‘compatibility’ assumptions.

Lemma 4.1.10. Let \(\mathcal{H}\) be a Hilbert lattice with lattice cone \(S := \mathcal{H}_+\), \(\lambda_{n+1} := -x_n^-\). Suppose \((\lambda_n)_{n=1}^\infty \subseteq \Lambda\) for some set \(\Lambda\) such that \(Q(\Lambda) \subseteq S \cup (-S)\), and at least one of following assumptions hold.

(a) \(Q\lambda_{n+1} \in S\) whenever \(x_n \in S\),

(b) If \(x_{n_0} \in S\) for some \(n_0\) then \((Q\lambda_n)_{n=1}^\infty\) is eventually zero.

Then \((x_n^+)_{n=1}^\infty\) converges in norm as soon as \((x_n^-)_{n=1}^\infty\) remains bounded.
Proof. In this setting equation (4.5) becomes
\[ x_{n+1} = x_n^+ + Q \lambda_{n+1}. \] (4.8)
We consider the two possible cases: (i) \( Q \lambda_n \in (-S) \) for all \( n \geq 1 \), or (ii) \( Q \lambda_{n_0} \in S \) for some \( n_0 \geq 1 \).

(i): For all \( n \geq 1 \),
\[ x_{n+1}^+ = (x_n^+ + Q \lambda_{n+1})^+ \leq x_n^+ + (Q \lambda_{n+1})^+ = x_n^+. \]
Since the sequence \( (x_n^+)_{n=1}^\infty \) is bounded and decreasing, by Fact 4.1.3 (a) it converges in norm.

(ii): By (4.8) \( Q \lambda_{n_0} \in S \) implies that \( x_{n_0} \in S \). So, if (a) holds we have \( Q \lambda_{n_0+1} \in S \) and inductively \( x_n \) and \( Q \lambda_n \in S \) for \( n \geq n_0 \). In which case, for \( n \geq n_0 \),
\[ x_{n+1}^+ = x_{n+1} = x_n^+ + Q \lambda_{n+1} \geq x_n^+. \]
So, \( (x_n^+)_{n=n_0}^\infty \) is increasing, and by assumption bounded, hence norm convergent by Fact 4.1.3(a).

On the other hand, if (b) holds then there exists \( k_0 \) such that \( Q \lambda_n = 0 \) for all \( n \geq k_0 \). This implies that \( (x_n)_{n=1}^\infty \) is positive and constant from \( n = k_0 - 1 \) onwards. A fortiori, \( (x_n^+)_{n=1}^\infty \) converges in norm. \( \square \)

Remark 4.1.11. Condition (b) of Lemma 4.1.10 is satisfied, for example, by the von Neumann sequence of Example 4.1.5, and under an additional assumption, by the Douglas–Rachford sequence of Example 4.1.6

Consider the sequence \( (Q \lambda_n)_{n=1}^\infty \) in the von Neumann sequence of Example 4.1.5. A useful observation of [16] is that as \( x_n \in A \), one has
\[ Q \lambda_{n+1} = Q(x_n - P_S x_n) = Q(P_{S\oplus} x_n). \]
Hence if \( x_{n_0} \in S \) then \( P_{S\oplus} x_{n_0} = 0 \), so \( Q \lambda_{n_0+1} = 0 \). Thus, inductively we see that condition (b) of Lemma 4.1.10 is satisfied provided \( x_{n_0} \in S \) for some \( n_0 \). In which case the Von Neumann sequence is eventually constant.

For the Douglas–Rachford sequence it is not as straightforward to select a point in \( A \). Nevertheless, a similar argument can be performed, under an additional assumption, using the point given in the following lemma (see also, Figure 4.1).

Lemma 4.1.12. Let \( (x_n)_{n=1}^\infty \) be the Douglas–Rachford sequence defined by \( x_{n+1} := T_{S,A} x_n \) where \( S \subseteq H \) is a nonempty closed convex cone, and \( A \subseteq H \) is a closed affine subspace with finite codimension. (i.e., \( \kappa_{n+1} := P_{S\oplus} x_n \) and \( \lambda_{n+1} := \pi_n - R_S x_n \) where \( \pi_n \in A \).) Then
\[ x_{n+1} - P_{S\oplus} x_n \in A. \]
Furthermore, if \( Q(-S^\circ) \subseteq S \) and \( x_{n_0} \in S \), for some \( n_0 \geq 1 \), then \( Q \lambda_k = 0 \) for all \( k \geq n_0 + 1 \), and hence the Douglas–Rachford sequence is eventually constant.
4.1. RECONSTRUCTION FROM NON-NEGATIVE MOMENTS

Figure 4.1: A Douglas–Rachford sequence \((x_n)_{n=1}^{\infty}\) converges in five iterations, as described in Lemma 4.1.12. For the same initial point, the von Neumann sequence \((y_n)_{n=1}^{\infty}\) does not terminate finitely.

**Proof.** Apply \(Q\) to both sides of (4.7) and use Theorem 4.1.4 to obtain

\[ Q(x_{n+1} - P_S x_n) = Q\overline{x}_n \text{ for } \overline{x}_n \in A. \]

Suppose further that \(Q(-S^\ominus) \subseteq S\) and \(x_{n_0} \in S\) for some \(n_0 \geq 1\). Then \(R_S x_{n_0} = x_{n_0}\) and \(x_{n_0+1} = P_A x_{n_0} \in A\). Since \(x_{n_0} - P_S x_{n_0-1} \in A\), we have

\[ Q\lambda_{n_0+1} = Q((x_{n_0} - P_S x_{n_0-1}) - x_{n_0}) = Q(-P_S x_{n_0-1}) \in S, \]

and therefore \(x_{n_0+1} = x_{n_0} + Q\lambda_{n_0+1} \in S\). That is, \(x_{n_0+1} \in S \cap A \subseteq \text{Fix } T_{S,A}\) and \(Q\lambda_k = 0\) for all \(k \geq n_0 + 1\).

We now consider a framework which include the von Neumann and Douglas–Rachford methods as within the same one-paramter family. Within the framework, we also recover the corresponding results relating to the von Neumann sequences originally derived in [16]. More precisely, for any \(x_0 \in \mathcal{H}\), we consider the sequence given by

\[ x_{n+1} := T_{S,A}^c x_n, \quad (4.9) \]

where

\[ T_{S,A}^c := cI + (1 - c)R_A^b R_S^a, \quad R_S^a := aI + (1 - a)R_S, \quad R_A^b := bI + (1 - b)R_A, \]
for some $a, b \in [0, 1]$ and $c \in [0, 1]$. That is, we replace each of $T_{S,A}, R_S$ and $R_A$ in the Douglas–Rachford method, with a convex combination of itself and the identity.

The following proposition collects properties of the operator $T_{S,A}^c$ and its fixed point iteration.

**Proposition 4.1.13.** Let $T_{S,A}^c : \mathcal{H} \to \mathcal{H}$ be the relaxed operator defined by (4.9) where $S \subseteq \mathcal{H}$ is a closed convex cone and $A$ is a closed affine subspace. Let $(x_n)_{n=1}^\infty$ a sequence defined by (4.9). Then the following statements hold.

(a) The sequence $(x_n)_{n=1}^\infty$ is of form prescribed by (4.5). More precisely,

$$x_{n+1} = x_n - \kappa_{n+1} + Q\lambda_{n+1},$$

where, for any $\bar{x} \in A$, the sequences $(\kappa_n)_{n=1}^\infty$ and $(\lambda_n)_{n=1}^\infty$ are given by

$$\kappa_{n+1} = 2(1 - a)(1 - c)P_Sx_n, \quad \lambda_{n+1} = 2(1 - b)(1 - c)(\bar{x} - R_S^ax_n).$$

(b) Further suppose $(1 - a)(1 - c) = (1 - b)(1 - c) = 1/2$. Then:

(i) The operator $T_{S,A}^c$ is $\alpha$-averaged for some $\alpha \in [0, 1]$.

(ii) If $\text{Fix} T_{S,A}^c \neq \emptyset$ then $(x_n)_{n=1}^\infty$ converges weakly to an element of $\text{Fix} T_{S,A}^c$.

(iii) If $\text{Fix} T_{S,A}^c \neq \emptyset$ then $(x_n)_{n=1}^\infty$ converges in norm to an element of $\text{Fix} T_{S,A}^c$ as soon as it contains a norm convergent subsequence.

**Proof.** (a): Observe that, for any $\bar{x} \in A$,

$$x_{n+1} = x_n + (1 - c)(-x_n + R_A^bR_S^cx_n)$$

$$= x_n + (1 - c)(-x_n + bR_S^cx_n + (1 - b)[2P_AR_S^cx_n - R_S^cx_n])$$

$$= x_n + (1 - c)(-x_n + bR_S^cx_n - (1 - b)R_S^cx_n + 2(1 - b)P_AR_S^cx_n)$$

$$= x_n + (1 - c)(-x_n + (2b - 1)R_S^cx_n + 2(1 - b)[R_S^cx_n + Q(\bar{x} - R_S^ax_n)])$$

$$= x_n + (1 - c)(-x_n + R_S^cx_n + 2(1 - b)Q(\bar{x} - R_S^ax_n))$$

$$= x_n + (1 - c)(-x_n + ax_n + (1 - a)R_Sx_n + 2(1 - b)Q(\bar{x} - R_S^ax_n))$$

$$= x_n + (1 - c)(-2(1 - a)P_Sx_n + 2(1 - b)Q(\bar{x} - R_S^ax_n))$$

$$= x_n - 2(1 - a)(1 - c)P_Sx_n + 2(1 - b)(1 - c)Q(\bar{x} - R_S^ax_n).$$

(b)(i): Since the reflectors $R_S$ and $R_A$ are nonexpansive, the relaxed reflectors $R_S^c$ and $R_A^b$, as well as their composition $R_A^bR_S^c$, are also nonexpansive. On one hand, if $c \in [0, 1]$ then $T_{S,A}^c$ is $(1 - c)$-averaged. On the other hand, if $c = 0$ then $a = b = 1/2$ and $T_{S,A}^c = R_A^bR_S^c = P_AP_S$. By Proposition 2.1.9, $T_{S,A}^c$ is 1/3-averaged. (ii): Follows from Theorem 2.1.15. (iii): Suppose that the subsequence $(x_{n_k})_{k=1}^\infty$ converges strongly to $x$. By (ii), $(x_n)_{n=1}^\infty$ converges weakly to an element of $\text{Fix} T_{S,A}^c$, and since weak limits are unique, it follows that $x \in \text{Fix} T_{S,A}^c$. Strong convergence of $(x_n)_{n=1}^\infty$ now follows by virtue of $T_{S,A}^c$ being nonexpansive. \qed
4.1. RECONSTRUCTION FROM NON-NEGATIVE MOMENTS

The following lemma is the generalisation of Lemma 4.1.12 for $T^c_{S,A}$.

**Lemma 4.1.14.** Let $(x_n)_{n=1}^\infty$ be the relaxed sequence defined by $x_{n+1} := T^c_{S,A} x_n$ where $S \subseteq \mathcal{H}$ is a nonempty closed convex cone, and $A \subseteq \mathcal{H}$ is an affine subspace with finite codimension. That is,

\[ \kappa_{n+1} = 2(1-a)(1-c)P_{S^\ominus} x_n, \quad \lambda_{n+1} = 2(1-b)(1-c)(\overline{x} - R^a_S x_n), \]

where $\overline{x}$ is any point in $A$. Then

\[ \frac{x_{n+1} + (\tau - 1) x_n - (1-a)\tau P_{S^\ominus} x_n}{\tau} \in A, \]

where $\tau := 2(1-b)(1-c)$. Further suppose that $(1-a)(1-c) = (1-b)(1-c) = 1/2$. If $x_n \in S$ then $Q\lambda_k = 0$ for all $k \geq n + 2$, and hence the relaxed sequence is eventually constant.

**Proof.** The proof is similar to Lemma 4.1.12. To prove the first claim, apply $Q$ to both sides of (4.9) and use Theorem 4.1.4. If $x_n \in S$ then $P_{S^\ominus} x_n = 0$, so that the affine combination of $x_{n+1}$ and $x_n$ is contained in $A$. In this case,

\[ Q\lambda_{n+2} = Q (x_{n+1} + [2(1-b)(1-c) - 1] x_n - R^a_S x_{n+1}) \]
\[ = Q (x_{n+1} + [2(1-b)(1-c) - 1] x_n - x_{n+1}) \]
\[ = [2(1-b)(1-c) - 1] Qx_n. \]

If $(1-b)(1-c) = 1/2$ we have $Q\lambda_{n+2} = 0$. As before, this implies $x_{n+2} = x_{n+1} = x_n$, and hence $x_{n+2} \in S \cap A \subseteq \text{Fix} T^c_{S,A}$. \qed

We are now ready to prove our main result which gives conditions for strong convergence.

**Theorem 4.1.15** (Strong convergence). Let $\mathcal{H}$ be a Hilbert lattice, $S := \mathcal{H}_+$, let $A$ be a closed affine subspace with finite codimension, and suppose $A \cap S \neq \emptyset$. For any $x_0 \in \mathcal{H}$, define $x_{n+1} := T^c_{S,A} x_n$ as in (4.9). Then $(x_n)_{n=1}^\infty$ converges in norm whenever one of the following conditions holds:

(a) \( \text{range}(Q) \cap S = \{0\} \).

(b) \((1-a)(1-c) = (1-b)(1-c) = 1/2\) and either

(i) \( Q(A - S) \subseteq S \cup (-S) \), \( Q(S) \subseteq S \) and \( a \in [0,1/2] \), or

(ii) \( A \) has codimension 1.
Proof. (a): Follows immediately from Lemma 4.1.8. (b)(i): Since $0 \leq a \leq 1/2$ and 
\[ R_S^a x_n = ax_n + (1 - a)|x_n| = x_n^+ + (1 - 2a)x_n^-, \]
we have $R_S^a x_n \in S$ and $\lambda_{n+1} \in \Lambda := A - S$ for all $n \geq 1$. By Lemma 4.1.14, we may express 
\[ \lambda_{n+1} = (x_n + x_{n-1}^-) - (x_n^+ + (1 - 2a)x_n^-) = x_{n-1}^- - 2(1 - a)x_n^- . \]
Thus if $x_n \in S$, for some $n \geq 1$, then $Q\lambda_{n+1} = Qx_{n-1}^- \in Q(S) \subseteq S$. We therefore have that Lemma 4.1.10(a) holds, and thus $(x_n^\infty)_{n=1}^\infty$ converges in norm. We note that since we are in a Hilbert lattice, $\kappa_{n+1} = -x_n^-$ and hence may express 
\[ x_{n+1} = T_{S,A}^c x_n = x_n + x_n^- + Q\lambda_{n+1} = x_n^+ + Q\lambda_{n+1}. \] (4.10)
Since $(x_n)_{n=1}^\infty$ is bounded (being weakly convergent by Proposition 4.1.13), using (4.7), we see that $(Q\lambda_n)_{n=1}^\infty$ is also bounded. As it is contained in a finite dimensional subspace, it contains a norm convergent subsequence $(Q\lambda_n)_{k=1}^\infty$. Observing (4.10) we see that $(x_n)_{n=1}^\infty$ converges in norm. The result now follows from Proposition 4.1.13. (b)(ii): If $(x_n)_{n=1}^\infty$ fails to converge in norm, then Lemma 4.1.9 implies $Q(S) \subseteq S$ and $Q(A - S) \subseteq S \cup (-S)$. But then (b)(i) implies that $(x_n)_{n=1}^\infty$ was actually norm convergent, which is a contradiction. \[\square\]

Taking $a = b = 0$ and $c = 1/2$ in Theorem 4.1.15 we recover the Douglas-Rachford iteration, and taking $a = b = 1/2$ and $c = 0$ we obtain the von Neumann iteration.

### 4.1.2 Linear Systems with Sparsity Bounds

In this section we give the two basic examples of problems to which the derived normal cone formulae and regularity results of Section 3.2 apply. Both problems are instances of the feasibility problems given by 
\[ \text{find } x \in C_1 \cap C_2 \subseteq \mathbb{E}, \] (4.11)
for constraints sets $C_1$ and $C_2$ which shall be specified in a moment.

Consider the problem of finding the sparsest non-negative solution to a linear system. That is, 
\[ \min_{x \in \mathbb{R}^n} \{ \|x\|_0 : Ax = b, \ x \geq 0 \}, \] (4.12)
where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^n$.

When a desired sparsity bound, $s$, can be given a priori, the problem can be reformulated as 
\[ \text{find } x \in \mathbb{R}^m \text{ such that } Ax = b, \ x \geq 0, \ \|x\|_0 \leq s. \] (4.13)
As we have already mentioned, problems of precisely this kind arise when a signal \( x \), which is known \textit{a priori} to be non-negative, purely real and sparse, is to be recovered from the under-determined linear system specified by the measurement matrix \( A \) and observation vector \( b \). Another example is given by \textit{mixed-integer linear programs}.

\textbf{Example 4.1.16 (Mixed-binary linear program).} Consider the mixed-binary linear program with feasible region

\[
\left\{ (x, y) \in \mathbb{R}^m \times \{0, 1\}^m : Ax = b, \ 0 \leq x \leq My, \ \sum_{j=1}^m y_j \leq s \right\}.
\]

This problem can be formulated as (4.13), provided the real number \( M > 0 \) is sufficiently large. Mixed-integer linear programs are frequently solved using branch-and-bound techniques. Such techniques require feasible solutions during the \textit{pruning steps}.

Returning our attention to the general problem, we observe (4.13) to be equivalent to the feasibly problem (4.11) with constraint sets

\[
C_1 := \{ x \in \mathbb{R}^m : Ax = b \}, \\
C_2 := \mathcal{K}_s = \{ x \in \mathbb{R}^n : x \geq 0, \| x \|_0 \leq s \}.
\] (4.14)

We may now give a characterisation of strong regularity of the collection \( \{C_1, C_2\} \) using the results of Section 3.2. This characterisation is of importance in applying the local convergence results of Section 3.1.

\textbf{Proposition 4.1.17.} Suppose \( \bar{x} \in C_1 \cap C_2 \) where the constraint sets \( C_1 \) and \( C_2 \) are as defined in (4.14). Then \( \{C_1, C_2\} \) is strongly regular at \( \bar{x} \) if and only if, for all \( y \in \text{range} \ A^T \{0\} \), either \( x \odot y = 0 \), or \( b \) \( y \) \( \geq 0 \) and \( \| y \|_0 > m - s \).

\textit{Proof.} Suppose \( \bar{x} \in C_1 \cap C_2 \). The normal cone to \( C_1 \) at \( \bar{x} \) is given by (see [37, Sec. 2.1, Exer. 4])

\[
N_{C_1}(\bar{x}) = \text{range} A^T.
\]

The result now follows from Proposition 3.2.20 with \( \Omega := C_1 \).

We now turn our attention to the symmetric matrix analogue. Consider the problem of finding a minimum rank positive semi-definite matrix solution to a linear system. That is,

\[
\min_{X \in \mathbb{S}^n} \{ \text{rank } X : (\langle A_j, X \rangle)_{j=1}^p = b, \ X \succeq 0 \}.
\] (4.15)
As before, when a desired bound on the rank, $s$, can be given \textit{a priori}, the problem can be reformulated as

$$\text{find } X \in \mathbb{S}^m \text{ such that } ((A_j, X))_{j=1}^p = b, \quad X \succeq 0, \quad \text{rank}(X) \leq s,$$

(4.16)

where $A_j \in \mathbb{S}^m$ for each $j \in \{1, \ldots, p\}$ and $b \in \mathbb{R}^p$.

Problems of this kind arise in rank-constrained semi-definite programming [78, Exer. 2.4], an NP-hard problem, as well as low-dimensional distance matrix reconstruction (Section 4.3).

We observe (4.16) to be equivalent to the two set feasibility problem (4.11) with constraint sets

$$C_1 := \{ X \in \mathbb{S}^m : ((A_j, X))_{j=1}^p = b \},$$

$$C_2 := \mathcal{S}_s = \{ X \in \mathbb{S}^m : X \succeq 0, \text{rank}(X) \leq s \}.$$

(4.17)

We may now give a characterisation of strong regularity of the collection $\{C_1, C_2\}$ using the results of Section 3.2.

**Proposition 4.1.18.** Suppose $\bar{X} \in C_1 \cap C_2$ where the constraint sets $C_1$ and $C_2$ are as defined in (4.17). Then $\{C_1, C_2\}$ is strongly regular at $\bar{X}$ if and only if, for all $Y \in \text{span}\{A_1, A_2, \ldots, A_p\} \setminus \{0\}$, either (a) $\bar{X}Y = 0$, or (b) $Y \not\succeq 0$ and $\text{rank}(Y) > m - s$.

**Proof.** Suppose $\bar{X} \in C_1 \cap C_2$. Then the normal cone to $C_1$ at $\bar{X}$ is given by (see [37, Sec 2.1, Exer 4] & [16, Sec. 5])

$$N_{C_1}(\bar{X}) = \text{range} \left( ((A_j, \cdot))_{j=1}^p \right)^* = \text{span}\{A_1, A_2, \ldots, A_p\}.$$

The result now follows from Proposition 3.2.21 with $\Omega := C_1$. \qed

**Remark 4.1.19 (Sparsity upper bounds).** In order to apply formulations (4.13) and (4.16), it is necessary that the sparsity/rank parameter $s$ is known \textit{a priori}. While this is not always possible, we emphasise that it is usually not the case that the \textit{optimal} sparsity parameter (in the sense of (4.12) or (4.15)) is needed. When projection or reflection methods are applied to a formulation only prescribing an upper bounded for the parameter, there is nothing to prevent sparser/lower-rank solutions from being returned. For further details, see [86].

\section*{4.2 Combinatorial Optimisation}

In this section we discuss application of the Douglas–Rachford method to various combinatorial optimisation problems. In many such situations, in order to
apply a projection algorithm, one needs to compute the projection of a point 
x = (x_1, \ldots, x_m) \in \mathbb{R}^m
onto the set of vectors whose entries are a permutation of 
m given real numbers c_1, \ldots, c_m \in \mathbb{R}. The set of all such vector will be denoted by 
\mathcal{C}. In what follows, we show that a projector onto \mathcal{C} can be easily and efficiently 
computed.

Given \( y \in \mathbb{R}^m \), we denote by \([y]\) the vector with the same components permuted 
in non-increasing order. We recall a classical rearrangement inequality.

**Proposition 4.2.1** (Rearrangement inequality). Any \( x, y \in \mathbb{R}^m \) satisfy

\[ x^T y \leq [x]^T [y]. \]

**Proof.** See, for instance, [82, Th. 368]. \( \square \)

Fix a vector \( x \in \mathbb{R}^m \). Denote by \([\mathcal{C}]_x\) the set of vectors in \( \mathcal{C} \) (which therefore 
have the same components but perhaps permuted) such that \( y \in [\mathcal{C}]_x \) if the \( i \)th 
largest entry of \( y \) has the same index in \( y \) as the \( i \)th largest entry of \( x \). Since 
the projection onto \( \mathcal{C} \) is, in general, not unique, we shall be content with finding 
any element of the projector \( P_\mathcal{C} \). In what follows, for convenience, we denote the 
mapping which sends a matrix to its \( j \)th row by \( P_j : \mathbb{R}^{m \times n} \to \mathbb{R}^n \).

An efficient algorithms for computing a projection is given in Algorithm 2, and 
its theoretical justification in Proposition 4.2.2.

**Algorithm 2:** Computing a projection onto the set of permutation points.

**input**: \( x = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m \) and \( c = (c_1, c_2, \ldots, c_m) \in \mathbb{R}^m \)

By relabelling if necessary, assume that \( c_1 \leq c_2 \leq \cdots \leq c_m \);

Set

\[ y = \begin{pmatrix} 1 & 2 & \cdots & m \\ x_1 & x_2 & \cdots & x_m \end{pmatrix} \in \mathbb{R}^{2 \times m}, \]

Let \( \tilde{y} \) denote a matrix obtained from \( y \) by rearranging its columns such that 
the vector \( P_2(\tilde{y}) \) is in non-increasing order;

Set

\[ z = \begin{pmatrix} \tilde{y} \\ c \end{pmatrix} \in \mathbb{R}^{3 \times m}; \]

Let \( \tilde{z} \) be a matrix obtained from \( z \) by rearranging its columns such that the 
vector \( P_1(\tilde{z}) \) is in non-increasing order;

**output**: \( P_3(\tilde{z}) \)

**Proposition 4.2.2** (Projections on permutations). Denote by \( \mathcal{C} \subset \mathbb{R}^m \) the set 
of all vectors whose entries are permutations of \( c_1, c_2, \ldots, c_m \in \mathbb{R} \). Then for any 
\( x \in \mathbb{R}^m \),

\[ P_\mathcal{C}(x) = [\mathcal{C}]_x. \quad (4.18) \]
Furthermore, using Algorithm 2, an element of $P_C x$ can be computed in order $O(m \log m)$ operations.

**Proof.** For any $c \in C$ and any $y \in [C]_x$,
\[
\|x - c\|^2 = \|x\|^2 + \|c\|^2 - 2x^T c \\
= \|[x]\|^2 + \|[c]\|^2 - 2[x]^T [c] \\
\geq \|[x]\|^2 + \|[c]\|^2 - 2[x]^T [c] \\
= \|[x - [c]]\|^2 \\
= \|x - y\|^2.
\]
This proves (4.18). Since an array of length $m$ can be sorted in $O(m \log m)$ operations (see, for example, [119]), the complexity of Algorithm 2 follows. \hfill \Box

The usefulness of Proposition 4.2.2 is, in part, due to the fact that every finite set of real numbers can be reformed as a binary vector. To see this, consider the constraint
\[
x \in \{c_1, c_2, \ldots, c_m\} \subseteq \mathbb{R}. \tag{4.19}
\]
Reformulating $x$ as a vector $y \in \mathbb{R}^m$, the real number $x = c_i$ is equivalent to the vector $y = (y_1, \ldots, y_m)$ defined by
\[
y_j = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{otherwise.} \end{cases}
\]
With this interpretation it is clear that (4.19) is equivalent to the constraint
\[
y \in \{e_1, e_2, \ldots, e_m\} \subseteq \mathbb{R}^m, \tag{4.20}
\]
with $x = c_i$ if and only if $y = e_i$.

The projection onto the reformulated set in (4.20) can be computed as follows, and is of particular importance in Section 4.2.1.

**Corollary 4.2.3.** Let $C = \{e_1, e_2, \ldots, e_m\} \subseteq \mathbb{R}^m$ where $(e_i)_{i=1}^m$ denotes the standard orthonormal basis. Then
\[
P_C(x) = \{e_i : x_i = \max\{x_1, x_2, \ldots, x_m\}\}. \tag{4.21}
\]
In this case, an element of $P_C(x)$ can be computed in order $O(m)$ operators.

**Proof.** Equation (4.21) is a consequence of Proposition 4.2.2 with $c_1 = 1$ and $c_2 = c_3 = \cdots = c_m = 0$. Since finding the maximum element of an array of length $m$ is $O(m)$, the claimed complexity of computing an element of $P_C(x)$ follows. \hfill \Box

**Remark 4.2.4.** A direct proof of Corollary 4.2.3 appears in [131, Sec. 5.9]. \hfill \Diamond
4.2. COMBINATORIAL OPTIMISATION

4.2.1 Sudoku Puzzles

This section details a case study of the Douglas–Rachford algorithm as a Sudoku solver. In order to better understand the Douglas–Rachford method’s behaviour in non-convex settings it is important to collect a catalogue of problems which can be successfully solved by the method. Sudoku represents both a genuinely combinatorial problem and an accessible starting point. In this direction, we acknowledge the fundamental contributions made by Elser [74, 75] who was the first to realise the usefulness of Douglas–Rachford-type methods as Sudoku solvers.

Let $m \in \mathbb{N}$ and let $I = \{1, 2, \ldots, m\}$. An $m^2 \times m^2$ Sudoku puzzle is a partial matrix $S \in I^{m^2 \times m^2}$. That is, a matrix for which some of the entries are missing. The matrix $S$ is viewed as consisting of $m^2$ sub-blocks, each of which is an $m \times m$ sub-partial matrix. The goal is to fill in the missing entries of the partial such that each row, column and sub-block contains each of the elements in $I$ exactly once.

A typical Sudoku is of size $9 \times 9$. Figure 4.2 shows an example of such a puzzle and its unique solution. As a decision problem, Sudoku is NP-complete [139].

Figure 4.2: (Left) An incomplete Sudoku and (right) its unique solution.

We introduce some further notations. Let $A[i, j]$ denote the $ij$-th entry of a matrix $A$. The submatrix of $A$ formed by taking rows $i$ through $i'$ and columns $j$ through $j'$ (inclusive) is denoted $A[i : i', j : j']$. When $i$ and $i'$ are the indices of the first and last rows, this former is abbreviate $A[: i', j : j']$. The column indices are abbreviated similarly. The vectorisation of the matrix $A$ by columns, is denoted by $\text{vec} A$. The notation extends to multidimensional arrays in the obvious way.

For the purpose of clarity of exposition, we shall formulate the Sudoku problem only for $9 \times 9$ Sudoku. The same principles can, however, be transparently applied to larger Sudoku puzzles. From herein, let $S$ denote the partially filled $9 \times 9$ integer
matrix representing the incomplete Sudoku, let \( I = \{1, 2, \ldots, 9\} \) and let \( J \subseteq I^2 \) be the set of indices for which \( S \) is filled.

**Binary and Integer Programs**

Sudoku can be formulated as an integer feasibility problem in the obvious way. To do so, denote the set of vectors which contain all permutations of the real numbers \( 1, 2, \ldots, 9 \) by \( \mathcal{E} \). Given the incomplete Sudoku \( S \), the matrix \( A \in \mathbb{R}^{9 \times 9} \) is a solution to \( S \) if and only if

\[
A \in C_1 \cap C_2 \cap C_3 \cap C_4,
\]  

(4.22)

where the constraints are given by

\[
C_1 = \{ A : A[i, j] = S[i, j] \text{ for each } (i, j) \in J \},
\]

\[
C_2 = \{ A : A[i, :] \in \mathcal{E} \text{ for each } i \in I \},
\]

\[
C_3 = \{ A : A[:, j] \in \mathcal{E} \text{ for each } j \in I \},
\]

\[
C_4 = \{ A : \text{vec} A[3i + 1 : 3(i + 1), 3j + 1 : 3(j + 1)] \in \mathcal{E} \text{ for } i, j = 0, 1, 2 \}.
\]

The first constraint, \( C_1 \), ensures that the entries of the solution match those in the incomplete puzzle. The remaining three constraints enforce the row, column and sub-block constraint, respectively. Since the feasibility problem (4.22) has four sets, it is necessary to use the product reformulation of Section 1.4. In doing so, the reflection is first performed with respect to the diagonal set \( i.e., \) the operator \( T_{D,C} \) is used. See also, Section 4.4 for further discussion.

Since the constraint \( C_1 \) applies independently to each matrix entry, its projection with respect to the Frobenius norm also applies entry-wise. It is given by

\[
(P_{C_1}A)[i, j] = \begin{cases} 
S[i, j] & \text{if } (i, j) \in J, \\
A[i, j] & \text{otherwise};
\end{cases}
\]

for each \((i, j) \in I^2\). Formulae for the projectors onto \( C_2, C_3 \) and \( C_4 \) are Proposition 4.2.2, and their efficient implementation can be performed using Algorithm 2.

Unfortunately, this integer formulation is ineffective, except for small \( 4 \times 4 \) Sudoku. Instead, we consider a binary reformulation.

Denote by \( \mathcal{C} \), the set of all \( n \)-dimensional standard basis vectors. The integer matrix \( X \in I^{9 \times 9} \) can be seen equivalent to the three dimensional \( B \in \mathbb{R}^{9 \times 9 \times 9} \) given by

\[
B[i, j, k] = \begin{cases} 
1 & \text{if } A[i, j] = k, \\
0 & \text{otherwise.}
\end{cases}
\]

Let \( S' \) denote the partially filled \( 9 \times 9 \times 9 \) zero-one array representing the incomplete Sudoku, \( S \), under the reformulation, and let \( J' \subseteq I^3 \) be the set of indices for which \( S' \) is filled.
The four constraints of the previous section become

\[ C_1 = \{ B : B[i, j, k] = 1 \text{ for each } (i, j, k) \in J' \}, \]
\[ C_2 = \{ B : B[i, :, k] \in C \text{ for each } i, k \in I \}, \]
\[ C_3 = \{ B : B[:, j, k] \in C \text{ for each } j, k \in I \}, \]
\[ C_4 = \{ B : \text{vec } B[3i + 1 : 3(i + 1), 3j + 1 : 3(j + 1), k] \in C \]
\[ \quad \text{for } i, j = 0, 1, 2 \text{ and } k \in I \}, \]

In addition, since each Sudoku square has precisely one entry, we require

\[ C_5 = \{ B : B[i, j, :] \in C \text{ for each } i, j \in I \}. \]

A visualisation of the constraints is provided in Figure 4.3.

![Figure 4.3: A visualisation of the binary array B showing its constraints used in Sudoku modelled as a zero-one program. With the shaded blocks, all except a single of the entry is zero.](image)

Clearly there is a one-to-one correspondence between completed integer Sudokus, and zero-one arrays contained in the intersection of the five constraint sets. Moreover, \( B \) is a completion of \( S' \) if and only if

\[ B \in C_1 \cap C_2 \cap C_3 \cap C_4 \cap C_5. \] \hfill (4.23)

As before, the projectors onto \( C_1, C_2, C_3, C_5 \) are given by Corollary 4.2.3, and the projection onto \( C_4 \) is given pointwise by

\[ (P_{C_4}B)[i, j, k] = \begin{cases} 1 & \text{if } (i, j, k) \in J', \\ B[i, j, k] & \text{otherwise}; \end{cases} \]

for each \( (i, j, k) \in I^3 \).
Numerical Experiments

In this section the Douglas–Rachford Sudoku solver is tested on various suites of Sudoku puzzles. The binary implementation described in Algorithm 3 is used. In what follows, the function \( \text{round}(\cdot) \) performs point-wise rounding to the nearest integer.

**Algorithm 3:** The Douglas–Rachford Sudoku solver \((m \times m\) Sudoku).

1. Select a random \( y \in [0, 1]^{m \times m \times m} \);
2. Set \( x_0 := (y, y, y, y, y) \in D \);
3. Set \( p_0 := P_Dx_0 \);
4. Set \( n := 0 \);
5. **while** \( n \leq 10000 \) and \( \text{round}(p_n) \notin D \cap C \) **do**
   1. Select any \( c_n \in P_C(2p_n - x_n) \);
   2. \( x_{n+1} := x_n + c_n - p_n \);
   3. \( p_{n+1} := P_Dx_{n+1} \);
   4. Set \( n := n + 1 \);
6. **end**

**Remark 4.2.5** (Termination criterion). The implementation described in Algorithm 3 uses the rounded value of \( p_n \) in the termination criterion rather than \( p_n \) itself. This improvement is justified due to the following observation: If \( p_n = P_Dx_n \) is a solution to (4.23) then all entries of \( p_n \) are either 0 or 1. Similar improvements are possible for any discrete problem.

Since the Douglas–Rachford method produces a point whose projection onto \( D \) is a solution to (4.23), we also consider a variant which iterates according to

\[
x_{n+1} := \begin{cases} 
P_DT_{D,C}x_n & \text{if } n \in \{400, 800, 1600, 3200, 6400\}; \\ 
T_{D,C}x_n, & \text{otherwise.}
\end{cases}
\]

The implementation of this variant, which we refer to as \( DR+Proj \), is described in Algorithm 4.

We now turn our attention to the test libraries examined, and some other methods used in the comparison. The following five libraries were examined.

1. **top95** and **1465** [1]: Collections of \( 9 \times 9 \) instances containing 95 and 1465 test problems, respectively. They are frequently used by programmers to test their solvers.

2. Gordon Royle’s minimal Sudoku [129]: A collection of \( 9 \times 9 \) instances containing around 50,000 distinct Sudokus with 17 entries (the best known
4.2. COMBINATORIAL OPTIMISATION

Algorithm 4: The DR+Proj variant ($m \times m$ Sudoku).

Select a random $y \in [0, 1]^{m \times m}$;
Set $x_0 := (y, y, y, y, y) \in D$;
Set $p_0 := P_D x_0$;
Set $n := 0$;

while $n \leq 10000$ and round($p_n$) $\notin D \cap C$ do

Select any $c_n \in P_C (2p_n - x_n)$;
$x_{n+1} := x_n + c_n - p_n$;
$p_{n+1} := P_D x_{n+1}$;
if $n \in \{400, 800, 1600, 3200, 6400\}$ then
$x_{n+1} := p_{n+1}$;
end
Set $n := n + 1$;
end

lower bound on the number of entries required for a unique solution). We considered the first 1000 puzzles from the collection, which we refer to as minimal1000.

3. reglib-1.3 [90]: A collection of $9 \times 9$ instances each containing various problems suited to human-style solving techniques.

4. ksudoku16 and ksudoku25: Collections containing around 30 Sudokus each, of various difficulties, as generated by KSudoku. Contains $16 \times 16$ and $25 \times 25$ instances, respectively.

Table 4.1: Summary of Sudoku test libraries.

<table>
<thead>
<tr>
<th>Library</th>
<th>Size</th>
<th># of Puzzles</th>
</tr>
</thead>
<tbody>
<tr>
<td>top95</td>
<td>$9 \times 9$</td>
<td>95</td>
</tr>
<tr>
<td>top1465</td>
<td>$9 \times 9$</td>
<td>1465</td>
</tr>
<tr>
<td>minimal1000</td>
<td>$9 \times 9$</td>
<td>1000</td>
</tr>
<tr>
<td>reglib-1.3</td>
<td>$9 \times 9$</td>
<td>983</td>
</tr>
<tr>
<td>ksudoku16</td>
<td>$16 \times 16$</td>
<td>30</td>
</tr>
<tr>
<td>ksudoku25</td>
<td>$25 \times 25$</td>
<td>30</td>
</tr>
</tbody>
</table>

A brief description of the methods used in the comparison follows

1. Douglas–Rachford solver in C++: Our implementation as described in Algorithm 3. Our experiments were performed using both the normal Douglas–Rachford method (DR) and the DR+Proj variant.
2. **Gurobi binary program** [80]: Solves a binary integer program formulation using *Gurobi Optimizer 5.5*. The formulation is the same $n \times n \times n$ binary array model used in the Douglas–Rachford implementation. Our experiments were performed using the default settings (Gurobi), and the default settings with the pre-solver off (Gurobi−PS).

3. **YASS** (Yet Another Sudoku Solver) [13]: Solves the Sudoku problem in two phases. In the first phase, a *reasoning algorithm* determines the possible candidates for each of the empty Sudoku squares. If the Sudoku is not completely solved, the second phase uses a deterministic recursive algorithm.

4. **DLX** [55]: Solves an exact cover formulation using the *Dancing Links* implementation of Knuth’s *Algorithm X* (a non-deterministic, depth-first, backtracking algorithm).

Since YASS and DLX were only designed to be applied to $9 \times 9$ instances, their performances on ksudoku16 and ksudoku25 are not included in the comparison.

Table 4.2: Mean (maximum) time in seconds over all instances.

<table>
<thead>
<tr>
<th></th>
<th>top95</th>
<th>top1465</th>
<th>reglib-1.3</th>
<th>minimal1000</th>
<th>ksudoku16</th>
<th>ksudoku25</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DR</strong></td>
<td>1.432 (6.056)</td>
<td>0.929 (6.038)</td>
<td>0.279 (5.925)</td>
<td>0.509 (5.934)</td>
<td>5.064 (30.079)</td>
<td>4.011 (24.627)</td>
</tr>
<tr>
<td><strong>DR+Proj</strong></td>
<td>1.894 (6.038)</td>
<td>1.261 (12.646)</td>
<td>0.363 (6.395)</td>
<td>0.953 (5.901)</td>
<td>6.757 (31.949)</td>
<td>8.608 (84.190)</td>
</tr>
<tr>
<td><strong>Gurobi</strong></td>
<td>0.063 (0.095)</td>
<td>0.063 (0.171)</td>
<td>0.059 (0.123)</td>
<td>0.063 (0.091)</td>
<td>0.168 (0.327)</td>
<td>0.401 (0.490)</td>
</tr>
<tr>
<td><strong>Gurobi−PS</strong></td>
<td>0.077 (0.322)</td>
<td>0.076 (0.405)</td>
<td>0.058 (0.103)</td>
<td>0.064 (0.104)</td>
<td>0.635 (4.621)</td>
<td>0.414 (0.496)</td>
</tr>
<tr>
<td><strong>YASS</strong></td>
<td>2.256 (58.822)</td>
<td>1.440 (113.195)</td>
<td>0.039 (3.796)</td>
<td>0.654 (61.405)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>DLX</strong></td>
<td>1.386 (38.466)</td>
<td>0.310 (34.179)</td>
<td>0.105 (5.500)</td>
<td>3.871 (60.541)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4.2 shows a comparison of the time taken by each of the examined methods when applied to the test libraries of Table 4.1. Computations were performed on an Intel Core i5-3210 @ 2.50GHz running 64-bit Ubuntu 12.10. For each Sudoku puzzle, 10 replications were performed. We make some general comments about the results.

All methods easily solve instances from reglib-1.3 (the test library consisting of puzzles suited to human-style techniques). Since human-style technique usually avoid excessive use of ‘trial-and-error’, less backtracking is required to solve puzzles aimed at human players. Since all of the algorithms, except the Douglas–Rachford method, utilise some form of backtracking, this may explain the observed good performance.

The Gurobi binary program performs best amongst the methods, regardless of the test library. Of the methods tested, the Gurobi Optimizer is the most sophisticated. Whether or not the pre-solver was used did not significantly effect computational time.

Our Douglas–Rachford implementation outperformed YASS on top95, top1465 and DLX on minimal1000. For all other algorithm/test library combinations, the
Douglas–Rachford was competitive. The performance of the normal Douglas–Rachford method appears slightly better than the variant including the additional projection step.

The Douglas–Rachford method solves Sudoku puzzles with a high success rate; no lower than 84% for any of the test libraries. For most test libraries the success rate observed was much higher (see Table 4.3). Puzzles solved by the method were typically done so in the first 2000 iterations (see Figure 4.4).

### Further Explorations

The unjustified success of binary Douglas–Rachford algorithm in the previous section demands further attention. This is made even more curious by the failure of the integer formulation of the problem. A question which naturally arises involves how to understand the model characteristics which lead to success and those which lead to failure. Further, it would also be desirable to have the tools diagnose inconsistent feasibility problems (cf. Theorem 2.3.3). For the case of Sudoku, such an ability would give a full treatment of the corresponding NP-complete decision problem. In this direction, we consider, in detail, a particularly troublesome Sudoku puzzle.

Recall the Sudoku puzzle and its unique solution in Figure 4.2. This puzzle has proven intractable for the Douglas–Rachford method. As given, it can not be solved by Schaad’s Douglas–Rachford based Sudoku solver [130], nor can it be solved reliably by our implementation. This ‘nasty’ Sudoku is a modified version of an example due to Elser [73], who found the first puzzle which could not be solved using Douglas–Rachford-type methods.

In an attempt to better understand the characteristics of this puzzle which make it hard, we consider the 22 Sudoku puzzles obtained by removing a single entry from Figure 4.2. For each of these new puzzles, one hundred random initialisations of Algorithm 3 are performed. The puzzle obtained by removing the top-left entry, a “7”, is still difficult for the Douglas–Rachford algorithm. It was solved with 24% success rate; a rate comparable to the ‘nasty’ Sudoku without any entries removed. If any other single entry was removed, the problem was solved with a 99% success rate.

For each of the puzzles with an entry removed, the number of distinct solutions
Figure 4.4: Frequency histograms showing the distribution of puzzles solved by number of iterations for the Douglas–Rachford method.
was determined using *SudokuSolver* [136], and are reported in Table 4.4. On one hand, of the puzzles with an entry removed that could be reliably solved, all have many solutions; anywhere from a few hundred to a few thousand. On the other hand, the puzzle with the top-left entry removed has relatively few; only five.\(^2\) We conjecture it is this structure that makes the ‘nasty’ Sudoku difficult to solve due to the Douglas–Rachford algorithm being hindered by an abundance of ‘near’ solutions.

In the opposite direction, the effect of adding entries to the incomplete ‘nasty’ Sudoku was considered. As before, one hundred random initialisations of Algorithm 3 are performed to each of the obtained puzzles. If any single entry was added, the corresponding Sudoku puzzles could be solved more often but not reliably. More precisely, they could be solved with a 54% success rate.

While Elser’s ‘nasty’ Sudoku is difficult for the Douglas–Rachford solver, this does not necessarily imply that it will be equally as difficult for other solvers. Similarly, puzzles which are difficult for other specialised Sudoku solvers need not pose any problem for the Douglas–Rachford solver. To explore this avenue, the behaviour of the Douglas–Rachford solver applied to the ‘nasty’ Sudoku was compared relative to its behaviour on other hard problems. Specially, we considered *AI escargot*, a Sudoku purposely designed by Arto Inkala to be difficult. According to Inkala, “Solving it is like an intellectual culinary pleasure.” The results are shown in Table 4.5.

\(^2\)The five solutions can be found online at [http://carma.newcastle.edu.au/DRmethods/comb-opt/nasty_nonunique.txt](http://carma.newcastle.edu.au/DRmethods/comb-opt/nasty_nonunique.txt)
Table 4.5: Number of instances solved from 1000 replications.

<table>
<thead>
<tr>
<th></th>
<th>AI escargot</th>
<th>‘Nasty’</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>985</td>
<td>202</td>
</tr>
<tr>
<td>DR+Proj</td>
<td>975</td>
<td>172</td>
</tr>
</tbody>
</table>

The Douglas–Rachford solver could solve AI escargot fairly reliably; it had a success rate of 99%. It is interesting to note, that in contrast to the ‘nasty’ Sudoku, the number of solutions to AI escargot with one entry removed was no more than a few hundred and typically much less.

Table 4.6: Mean (Max) time in second from 1000 replications.

<table>
<thead>
<tr>
<th></th>
<th>AI escargot</th>
<th>‘Nasty’</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>1.232 (6.243)</td>
<td>4.840 (6.629)</td>
</tr>
<tr>
<td>DR+Proj</td>
<td>1.623 (6.074)</td>
<td>5.312 (7.689)</td>
</tr>
<tr>
<td>Gurobi</td>
<td>0.157 (0.845)</td>
<td>0.111 (0.125)</td>
</tr>
<tr>
<td>Gurobi–PS</td>
<td>0.094 (0.153)</td>
<td>0.253 (0.365)</td>
</tr>
<tr>
<td>YASS</td>
<td>0.162 (0.255)</td>
<td>12.370 (13.612)</td>
</tr>
<tr>
<td>DLX</td>
<td>0.020 (0.032)</td>
<td>0.110 (0.126)</td>
</tr>
</tbody>
</table>

Each of the methods used in the Sudoku comparison were also applied to the two difficult Sudoku puzzles. Computational times are reported in Table 4.6. While all solved AI escargot easily, applied to the ‘nasty’ Sudoku, YASS was significantly slower and the Douglas–Rachford solver was not the only method to find the puzzle difficult.

The distances to the solution as a function of the number of iterations performed for the two difficult puzzles is shown in Figure 4.5. More precisely, if \((x_n)\) the sequence of iterates obtained from the Douglas–Rachford solver as in Algorithm 3, and \(x^*\) denote the corresponding Sudoku solution obtained from \((x_n)\), then the normalised distance

\[
\frac{\|x_n - x^*\|}{\max_{n \in \mathbb{N}} \|x_n - x^*\|},
\]

is plotted as a function of \(n\).

In contrast to the convex setting, Figure 4.5 shows that the sequence \((\|x_n - x^*\|)\) need not be decreasing. Recall that, in the convex setting, \((x_n)\) is Fejér monotone with respect to \(\text{Fix} T_{D,C}\), and hence, in particular, the sequence \((\|x_n - x^*\|)\) is
4.2. COMBINATORIAL OPTIMISATION

Figure 4.5: The normalised distance to the solution of typical runs of the Douglas–Rachford solver applied to (left) the ‘nasty’ Sudoku and (right) AI escargot.

decreasing. Returning our attention to Figure 4.5, we observe that when \((x_n)\) converged to a solution, \(\|x_n - x^*\|\) decreased rapidly just before the solution was found. This seemed to occur regardless of the behaviour of earlier iterations. We conjecture that this happens when the Douglas–Rachford iterates entering an appropriate local basin of attraction.

4.2.2 Nonogram Puzzles

Recall that a nonogram puzzle consists of a blank \(m \times n\) grid of pixels (the canvas) together with \((m+n)\) cluster-size sequences, one for each row and each column [47]. The goal is to paint the canvas with a picture such that:

- Each pixel must be black or white.
- If a row (resp. column) has cluster-size sequence \(s_1, s_2, \ldots, s_k\) then it must contain \(k\) clusters of black pixels, separated by at least one white pixel, such that the \(i\)th leftmost (resp. uppermost) cluster contains \(s_i\) black pixels.

An example of a nonogram puzzle is given in Figure 4.6 and its solution, found by the Douglas–Rachford algorithm, is shown in Figure 4.8. As a decision problem, nonogram puzzles are NP-complete [143,144].

Nonogram puzzles can be modelled as a binary feasibility problem. The \(m \times n\) grid is represented as a matrix \(A \in \mathbb{R}^{m \times n}\) with

\[
A[i,j] = \begin{cases} 
0 & \text{the } (i,j)-\text{th entry of the grid is white}, \\
1 & \text{the } (i,j)-\text{th entry of the grid is black}. 
\end{cases}
\]
Let $\mathcal{R}_i \subset \mathbb{R}^m$ (resp. $\mathcal{C}_j \subset \mathbb{R}^n$) denote the set of vectors having cluster-size sequences matching row $i$ (resp. column $j$). A binary matrix $A$ is a solution to the corresponding nonogram puzzle if and only if

$$A \in \mathcal{C}_1 \cap \mathcal{C}_2,$$

where

$$\mathcal{C}_1 = \{ A : A[i, :] \in \mathcal{R}_i \text{ for } i = 1, \ldots, m \},$$
$$\mathcal{C}_2 = \{ A : A[:, j] \in \mathcal{C}_j \text{ for } j = 1, \ldots, n \}.$$

The viability of the Douglas–Rachford method as a nonogram solver was examined by testing seven puzzles. More precisely, the puzzle in Figure 4.6 and the six puzzles shown in Figure 4.7. Since formulation (4.24) contains only two sets, the algorithms could be directly applied without a product reformulation. For each of the examined puzzles the Douglas–Rachford solver was highly successful. From 1000 random initializations, all puzzles considered were solved with a 100% success rate. While this is promising, we have not yet discussed computation of the projectors onto $\mathcal{C}_1$ and $\mathcal{C}_2$.

Within (4.24), a difficulty is that the projectors onto $\mathcal{C}_1$ and $\mathcal{C}_2$ have no simple form. To date, our attempts to find an efficient algorithm to do so have been unsuccessful. The aforementioned implementation uses a brute-force approach.
4.2. **COMBINATORIAL OPTIMISATION**

Figure 4.7: Solutions to six nonograms found by the Douglas–Rachford algorithm.

which pre-computes $\mathcal{R}_i$ and $\mathcal{C}_j$, for all indices $i, j$, and at each iteration chooses the nearest point by computing the distance to each point in the appropriate set. For nonograms with large canvases, the enumeration of $\mathcal{R}_i$ and $\mathcal{C}_j$ becomes intractable. However once the pre-computation has been performed, the Douglas–Rachford iterations themselves are relatively fast.

**Remark 4.2.6 (Perspectives on combinatorial problems).** We note that for Sudoku puzzles in Section 4.2.1, the computation of projections is easy but the typical number of (easy) iterative steps is large. In contrast for nonogram puzzles, the number of iterative steps is very small but an exponential amount of work is presumably buried in computing the projections.
4.2.3 Hadamard matrices

A matrix $H = (H_{ij}) \in \{-1, 1\}^{n \times n}$ is said to be a **Hadamard matrix of order** $n$ if

$$H^T H = n I.$$  \hfill (4.25)

There are many equivalent characterizations of Hadamard matrices. For instance, (4.25) is equivalent to asserting that $H$ has maximal determinant (i.e., $|\det H| = n^{n/2}$) [91, Ch. 2]. A classical result of Hadamard shows that Hadamard matrices exist only if the order $n$ is equal to 1, 2 or a multiple of 4. For orders 1 and 2, such matrices can be easily found. For multiples of 4, the question of existence of Hadamard matrices remains in general, and is known as the **Hadamard conjecture**. For further background on Hadamard matrices, we refer the reader to the monograph [91].

Fix an order $n \in \mathbb{N}$, and consider the problem of finding a Hadamard matrix of order $n$. It is easily seen that a matrix $X \in \mathbb{R}^{n \times n}$ is a Hadamard matrix of order $n$ if and only if

$$X \in C_1 \cap C_2,$$  \hfill (4.26)

where the constraints are defined by

$$C_1 = \{X \in \mathbb{R}^{n \times n} : X_{ij} = \pm 1 \text{ for } i, j = 1, \ldots, n\},$$  \hfill (4.27)

$$C_2 = \{X \in \mathbb{R}^{n \times n} : X^T X = n I\}.$$  \hfill (4.28)

In order to apply the Douglas–Rachford algorithm to (4.26), we must describe the projectors onto $C_1$ and $C_2$.

The first constraint, $C_1$, whilst being discrete and non-convex, nevertheless, has a simple projector. Since the constraint applies entry-wise, its projector is given entry-wise by

$$P_{C_1}(X)_{ij} = \begin{cases} -1 & \text{if } X_{ij} < -1, \\ \pm 1 & \text{if } X_{ij} = 0, \\ 0 & \text{otherwise}. \end{cases}$$  \hfill (4.29)

The second constraint, $C_2$, is also non-convex as the following example shows.
Example 4.2.7 ($C_2$ is not convex). A direct verification shows that the matrices
\[
\begin{pmatrix}
\sqrt{2} & 0 \\
0 & \sqrt{2}
\end{pmatrix}, \quad \begin{pmatrix}
0 & \sqrt{2} \\
\sqrt{2} & 0
\end{pmatrix},
\]
are in $C_2$ but their mid-point is not. ♦

A projection onto $C_2$ can be found by solving the equivalent problem of finding a projection onto the set of orthogonal matrices which is, in turn, a special case of the Procrustes problem.

**Proposition 4.2.8** (A projection onto $C_2$). Let $X = USV^T$ be a singular value decomposition of $X$. Then
\[
\sqrt{n}UV^T \in P_{C_2}(X).
\]

**Proof.** Let $Y = X/\sqrt{n}$. Then
\[
\min_{X \in \mathbb{R}^{n \times n}} \{\|X - A\|_F : A^TA = nI\} = \sqrt{n} \min_{Y \in \mathbb{R}^{n \times n}} \{\|Y - B\|_F : B^TB = I\}.
\]
Any solution to the latter is a projection of $Y$ onto the set of orthogonal matrices. One such matrix can be obtained by replacing all singular values of $Y$ by ‘one’ [132]. Since
\[
Y = U\hat{S}V^T \text{ where } \hat{S} = S/\sqrt{n},
\]
is a singular value decomposition, it follows that $UV$ is a projection of $Y$ onto the set of orthogonal matrices. The result now follows. ☐

**Remark 4.2.9** (Reconstruction of partial Hadamard matrices). Consider instead the matrix completion problem of finding a Hadamard matrix with some known entries. This can be cast within the above framework by appropriate modification of $C_1$. The projection onto $C_1$ only differs by leaving the known entries unchanged. ♦

We next give a second formulation for the Hadamard matrix problem which is also useful albeit less obvious. Consider the feasibility problem
\[
\text{find } X \in C_1 \cap C_3 \quad (4.30)
\]
where $C_1$ is as in (4.25) and $C_3$ is defined to be
\[
C_3 = \{X \in \mathbb{R}^{n \times n} : X^TX = \|X\|_F I\}. \quad (4.31)
\]
The following theorem shows that $X$ is a Hadamard matrix if and only if it solves the feasibility problem (4.30).
Theorem 4.2.10 (Equivalence of Hadamard matrix formulations). Consider the three constraint sets $C_1, C_2$ and $C_3$ defined as in (4.27), (4.28) and (4.31). Then $C_1 \cap C_2 = C_1 \cap C_3$. In other words, the solutions of the feasibility problems (4.26) and (4.30) coincide.

Proof. For any matrix $X = (X_{ij}) \in C_1$, since $X_{ij} \in \{\pm 1\}$, we have $X_{ij}^2 = 1$. It follows that

$$\|X\|_F^2 = \sum_{j=1}^n \sum_{i=1}^n X_{ij}^2 = \sum_{j=1}^n \sum_{i=1}^n 1 = n^2.$$ 

The result now follows from the definition of $C_2$ and $C_3$. \hfill \Box

A projection onto $C_3$ can be computed using a similar approach to the projection onto $C_2$.

Proposition 4.2.11 (A projection onto $C_3$). Let $X = USV^T$ be a singular value decomposition of $X$. Then

$$\sqrt{\|X\|_F} UV^T \in P_{C_3}(X).$$

Proof. This is a straightforward modification of Proposition 4.2.8. \hfill \Box

Remark 4.2.12 (Complex Hadamard matrices). It would also be possible to consider complex Hadamard matrices. In this case, the constraint set $C_1$ in (4.27) would be replaced with

$$C_1 = \{X \in \mathbb{C}^{n \times n} : |X_{ij}| = 1 \text{ for } i, j = 1, \ldots, n\}.$$ 

Its projection is also straightforward. It is given by

$$P_{C_1}(X)_{ij} = \begin{cases} 
X_{ij}/|X_{ij}| & \text{if } X_{ij} \neq 0, \\
\{z \in \mathbb{C} : |z| = 1\} & \text{otherwise.}
\end{cases}$$

Note that if we restricted ourselves to real solutions of $|X_{ij}| = 1$ then $X_{ij} = \pm 1$ and recover the real setting. \hfill \Diamond

Having described all the necessary algorithmic building blocks, we now turn our attention to computational experiments. Before doing so, we require some further properties of Hadamard matrices.

Let $H_1$ and $H_2$ be Hadamard matrices of the same order. We say $H_1$ are $H_2$ are distinct if $H_1 \neq H_2$. We say $H_1$ and $H_2$ are equivalent if $H_2$ can be obtained from $H_1$ by performing a sequence of row/column permutations, and/or multiplying row/columns by $-1$. The number of distinct (resp. inequivalent) Hadamard matrices of order $4n$ is given in the On-Line Encyclopedia of Integer Sequences (OEIS)
4.2. COMBINATORIAL OPTIMISATION

sequence A206712: $768, 4954521600, 2025150935014912000, \ldots$ (resp. A007299: $1, 1, 1, 1, 5, 3, 60, 487, 13710027, \ldots$). It is worth noting that with increasing order, the number of Hadamard matrices is a faster than exponentially decreasing proportion of the total number of $\{\pm 1\}$-matrices (of which there are $2^{n^2}$ for order $n$).

Table 4.7 and Fig. 4.9 summarise computational results for the Douglas–Rachford algorithm applied to finding Hadamard matrices. Since formulations (4.26) & (4.30) have only two sets, the method can be applied without a product reformulation. For each order, 1000 randomly initialised replications were performed. To determine if two Hadamard matrices are equivalent, the graph isomorphism approach due to McKay [114] was employed using Sage’s graph theory library.

Table 4.7: Results for Hadamard matrices found from 1000 instances.

<table>
<thead>
<tr>
<th>Order</th>
<th>Formulation (4.26)</th>
<th>Formulation (4.30)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave. Time (s)</td>
<td>Solved</td>
</tr>
<tr>
<td>2</td>
<td>1.1371</td>
<td>534</td>
</tr>
<tr>
<td>4</td>
<td>1.0791</td>
<td>627</td>
</tr>
<tr>
<td>8</td>
<td>0.7368</td>
<td>996</td>
</tr>
<tr>
<td>12</td>
<td>7.1298</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>9.4228</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>20.6674</td>
<td>0</td>
</tr>
</tbody>
</table>

We make some brief comments about the results. Formulation (4.30) was found to be faster and more successful than formulation eqrefeq:hadamard 1, especially for orders 8 and 12 where it was successful in every replication. For orders less than or equal to 12, the Douglas–Rachford schema was able to find the unique inequivalent Hadamard matrix under both formulations (except for order 12 with formulation (4.26)). Furthermore, formulation (4.30) was able to find four of the five inequivalent Hadamard matrices of order 16.3

From Figure 4.9 we observe that if a Hadamard matrix was found, it was usually found within the first few thousand iterations. The frequency histogram for order 16, shown in Figure 4.9(f), varied significantly from the corresponding histograms for lower orders.

For orders 20 and above, it is possible that another formulation might be more fruitful. It is likely that better and more problem-specific heuristics will be needed.

**Remark 4.2.13 (Selections from multi-valued projectors).** Since $C_2$ and $C_3$ are non-convex, when computing their projections it will sometimes be the case that a selection must be made from the set of nearest points. In our computational experiments, we always selected the projection in the same way (using Propositions 4.2.8 and 4.2.11). It may certainly be possible to benefit from making the selection according to some other or more sophisticated criterion. However, since we have no compelling reason to think some other criterion might be better, we leave this as a problem for future work.

We now turn our attention to special classes of Hadamard matrices, although we first recall some definitions.

Recall that a matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is skew-symmetric if $A^T = -A$. A **skew-Hadamard matrix** is a Hadamard matrix, $H$, such that the matrix $(I - H)$ is skew-symmetric. In order words,

$$H + H^T = 2I.$$

Skew-Hadamard matrices are of interest, for example, in the construction of combinatorial designs [101]. The number of inequivalent skew-Hadamard matrices of order $4n$ is given in OEIS sequence A001119: $1, 1, 2, 2, 16, 54, \ldots$ (for $n = 2, 3, \ldots$).

In addition to the constraints $C_1$ and $C_2$ from the previous section, we define the affine constraint

$$C_4 = \{X \in \mathbb{R}^{n \times n} : X + X^T = 2I\}.$$

A projection onto $C_1 \cap C_4$ is given by

$$P_{C_1 \cap C_4}(X)_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } X_{ij} \geq X_{ji}, \\ -1 & \text{if } i \neq j \text{ and } X_{ij} < X_{ji}, \\ 1 & \text{otherwise}. \end{cases}$$

Then $X$ is a skew-Hadamard matrix if and only if $X \in (C_1 \cap C_4) \cap C_2$.

Table 4.8 shows the results of the same experiment as in Section 4.2.3, but with the skew constraint incorporated.

**Remark 4.2.14 (Limiting the search space).** Comparing the results of Table 4.8 with those of Table 4.7, it is notable that by placing additional constraints on the
Figure 4.9: Frequency histograms showing the number of iterations required to find a Hadamard matrix (solved instances only).
Table 4.8: Result for skew-Hadamard matrices found from 1000 instances.

<table>
<thead>
<tr>
<th>Order</th>
<th>Formulation (4.26)</th>
<th>Formulation (4.30)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave. Time (s)</td>
<td>Solved</td>
</tr>
<tr>
<td>2</td>
<td>0.0003</td>
<td>1000</td>
</tr>
<tr>
<td>4</td>
<td>1.1095</td>
<td>719</td>
</tr>
<tr>
<td>8</td>
<td>0.7039</td>
<td>902</td>
</tr>
<tr>
<td>12</td>
<td>14.1835</td>
<td>43</td>
</tr>
<tr>
<td>16</td>
<td>19.3462</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>29.0383</td>
<td>0</td>
</tr>
</tbody>
</table>

problem, both methods now succeed at higher orders. In addition, method two is faster than before and can successfully find all inequivalent skew matrices of order 20 or less.

In contrast, the three-set feasibility problem $C_1 \cap C_2 \cap C_4$ was unsuccessful, except for order 2. This is despite the projection onto the affine set $C_4$ having the simple formula

$$P_{C_4}(X) = I + \frac{X - X^T}{2},$$

which allows for its efficient computation.

We now turn our attention to another special class of Hadamard matrices. Recall that a matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is circulant if it can be expressed as

$$A = \begin{pmatrix} 
\lambda_1 & \lambda_2 & \ldots & \lambda_n \\
\lambda_n & \lambda_1 & \ldots & \lambda_{n-1} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_2 & \lambda_3 & \ldots & \lambda_1 
\end{pmatrix}$$

for some vector $\lambda \in \mathbb{R}^n$.

The set of circulant matrices form a subspace of $\mathbb{R}^{n \times n}$. The set $\{P^k : i =$
1, 2, \ldots, n\}, where \( P \) is the cyclic permutation matrix

\[
P = \begin{pmatrix}
0 & 0 & \ldots & 0 & 1 \\
1 & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & 0 \\
\end{pmatrix},
\]

forms a basis. Consequently, any circulant matrix, \( A \), can be expressed as the linear combination of the form

\[
A = \sum_{k=1}^{n} \lambda_k P^k.
\]

A **circulant Hadamard matrix** is a Hadamard matrix which is also circulant.

**Remark 4.2.15.** Right (resp. left) multiplication by \( P \) results in a cyclic permutation of rows (resp. columns). Hence \( P^2, P^3, \ldots, P^n \) represent all cyclic permutations of the rows (resp. columns) of \( P \). In particular, \( P^n \) is the identity matrix.

The projection onto the set of circulant matrices is given by the following proposition.

**Proposition 4.2.16 (Nearest circulant matrix).** For \( X \in \mathbb{R}^{n \times n} \), the nearest circulant matrix is given by

\[
\sum_{k=1}^{n} \lambda_k P^k \text{ where } \lambda_k = \frac{1}{n} \sum_{i,j} P^k_{ij} X_{ij}.
\]

**Proof.** See, for instance, [89, Exer. 6.7].

The circulant Hadamard conjecture asserts: No circulant Hadamard matrix of order larger than 4 exists. For recent progress on the conjecture, we refer the reader to a paper of Leung and Schmidt [105]. Consistent with this conjecture, our Douglas–Rachford implementation can successfully find circulant matrices of order 4, but fails for higher orders.

### 4.3 Euclidean Distance Matrix Reconstruction

A square matrix \( D = (D_{ij}) \in \mathbb{R}^{m \times m} \) is a **Euclidean distance matrix (EDM)** if there exists points \( z_1, z_2, \ldots, z_n \in \mathbb{R}^m \) such that

\[
D_{ij} = \|z_i - z_j\|^2 \quad \forall i, j \in \{1, 2, \ldots, m\}.
\]
From the definition, it immediately follows that any EDM is symmetric, non-negative, and hollow (i.e., contains only zeros along its main diagonal). When (4.33) holds for a set of points in $\mathbb{R}^q$, we say $D$ is embeddable in $\mathbb{R}^q$. If $D$ is embeddable in $\mathbb{R}^q$ but not in $\mathbb{R}^{q-1}$, then we say that $D$ is irreducibly embeddable in $\mathbb{R}^q$.

**Remark 4.3.1 (EDMs in the literature).** There are at least a few equivalent definitions of EDMs which appear in the literature having different normalisations. For instance, [79] considers EDMs whose $ij$th entry in (4.33) is given by

\[ D_{ij} = -\frac{1}{2} \| z_i - z_j \|^2. \]

We now recall a useful characterization of EDMs, due to Hayden and Wells [83].

**Theorem 4.3.2 (EDM characterization [83, Th. 3.3]).** A non-negative, symmetric, hollow matrix $X \in \mathbb{R}^{m \times m}$ is a Euclidean distance matrix if and only if the block

\[ \hat{X} \in \mathbb{R}^{(m-1) \times (m-1)} \]

in

\[ Q(-X)Q = \begin{bmatrix} \hat{X} & d \\ d^T & \delta \end{bmatrix} \]

(4.34)

is positive semi-definite. In this case, $X$ is irreducibly embeddable in $\mathbb{R}^q$ where $q = \text{rank}(\hat{X}) \leq m - 1$.

The problem of low-dimensional Euclidean distance matrix completion can now be formulated. Let $D$ denote a partial Euclidean distance matrix, with entry $D_{ij}$ known whenever $(i,j) \in \Omega$ for some index set $\Omega$, which is embeddable in $\mathbb{R}^q$. Without loss of generality, we make the following three simplifying assumptions on the partial matrix $D$ and index set $\Omega$.

1. (non-negative) $D \geq 0$ (i.e., $D_{ij} \geq 0$ for all $i, j = 1, 2, \ldots, m$);
2. (hollow) $D_{ii} = 0$ and $(i,i) \in \Omega$ for $i = 1, 2, \ldots, m$;
3. (symmetric) $(i, j) \in \Omega \iff (j, i) \in \Omega$, and $D_{ij} = D_{ji}$ for all $(i, j) \in \Omega$.

We define two constraint sets

\[ C_1 = \{ X \in S^m : X \geq 0, X_{ij} = D_{ij} \text{ for all } (i,j) \in \Omega \}, \]

\[ C_2 = \left\{ X \in S^m : Q(-X)Q = \begin{bmatrix} \hat{X} & d \\ d^T & \delta \end{bmatrix}, \hat{X} \in S^{m-1}_{+}, d \in \mathbb{R}^{m-1}, \text{rank} \hat{X} \leq q, \delta \in \mathbb{R} \right\}. \]

(4.35)

In light of Theorem 4.3.2, the problem of low-dimensional Euclidean distance matrix completion can be cast as the two-set feasibility problem

\[ \text{find } X \in C_1 \cap C_2. \]
That is, a matrix $X$ is a low-rank Euclidean distance matrix which completes $D$ if and only if $X \in C_1 \cap C_2$. Some comments regarding the constraint sets in (4.35) are in order.

The set $C_1$ encodes the experimental data obtained from NMR, and the *a priori* knowledge that the matrix is non-negative, symmetric and hollow. Its projection has a simple formula, as we now show.  

**Proposition 4.3.3 (Projection onto $C_1$).** Let $X \in S^m$. Then $P_{C_1}X$ is given element-wise, for all $i, j \in \{1, 2, \ldots, m\}$, by 

$$
(P_{C_1}X)_{ij} = \begin{cases} D_{ij} & (i, j) \in \Omega, \\ \max\{0, X_{ij}\} & (i, j) \notin \Omega. \end{cases} (4.36)
$$

**Proof.** Let $Y$ be any matrix in $C_1$. We have 

$$
\|X - Y\|_F^2 = \sum_{(i,j) \in \Omega} (X_{ij} - Y_{ij})^2 + \sum_{(i,j) \notin \Omega \text{ s.t. } X_{ij} < 0} (X_{ij} - Y_{ij})^2 + \sum_{(i,j) \notin \Omega \text{ s.t. } X_{ij} \geq 0} (X_{ij} - Y_{ij})^2
$$

by combining (4.37) and (4.38) we see that 

$$
\|X - Y\|_F^2 \geq \|X - P\|_F^2 \text{ for all } Y \in C_1.
$$

Since $C_1$ is closed and convex, $P$ is the unique nearest point to $X$ in $C_1$. 

Using the conditions given by Theorem 4.3.2, the non-convex set $C_2$ encodes the *a priori* knowledge that the matrix of interest is an EDM together with the dimension of the space in which the corresponding points generating the matrix are contained. We now derive the projector onto $C_2$. 

**Theorem 4.3.4 (Nearest low-dimensional EDMs [5]).** Let $X \in S^m$ be a non-negative, hollow matrix. Then 

$$
P_{C_2}(X) = \left\{ -Q \begin{bmatrix} \hat{Y}^T & d \\ d & \delta \end{bmatrix} Q(-X)Q \begin{bmatrix} \hat{X}^T & d \\ d & \delta \end{bmatrix} : \hat{X} \in \mathbb{R}^{(m-1) \times (m-1)}, \hat{Y} \in P_{S_q} \hat{X}, \delta \in \mathbb{R} \right\},
$$

where $S_q$ is the set of positive semi-definite matrices with rank $q$ or less. In particular, $P_{C_2}(X)$ is a singleton if and only if $P_{S_q}(\hat{X})$ is a singleton.
Proof. Let $Y$ be any matrix in $C_2$. That is, 
$$Y = \begin{bmatrix} \hat{Y} & c \\ c^T & \beta \end{bmatrix},$$
for some $c \in \mathbb{R}^{m-1}$, $\beta \in \mathbb{R}$, $\hat{Y} \in \mathbb{S}_+^{m-1}$.

Using the orthogonality of $Q$, we compute
$$\|X - Y\|_F^2 = \|Q(X - Y)Q\|_F^2 = \|Q(-X)Q - Q(-Y)Q\|_F^2.$$

Using the orthogonality of $Q$, we compute
$$\|X - Y\|_F^2 = \|Q(X - Y)Q\|_F^2$$
$$= \left\| \begin{bmatrix} \hat{X} & d \\ d^T & \delta \end{bmatrix} - \begin{bmatrix} \hat{Y} & c \\ c^T & \beta \end{bmatrix} \right\|_F^2$$
$$= \left\| \begin{bmatrix} \hat{X} - \hat{Y} & (d - c) \\ (d - c)^T & (\delta - \beta) \end{bmatrix} \right\|_F^2$$
$$= \|\hat{X} - \hat{Y}\|_F^2 + 2\|d - c\|^2 + |\gamma - \beta|^2. \quad (4.39)$$

To complete the proof we observe that (4.39) is minimal if and only if $c = d$, $\gamma = \beta$ and $\hat{Y} \in P_{S_q}(\hat{X})$. $\Box$

The set $S_q$ in Theorem 4.3.4 is a set of low-rank positive semi-definite matrices. One method to compute its projection, which is justified by Proposition 3.2.13, can be given using a spectral-decomposition of $\hat{X}$. More precisely, let $\hat{X} = U \text{diag}(\lambda)U^T$ be a spectral-decomposition of $\hat{X}$ such that
$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q^+ \geq \cdots \geq \lambda_m.$$

Then a projection onto the $S_q$ is given by
$$U \text{diag}((\lambda_1^+, \lambda_2^+, \ldots, \lambda_q^+, 0 \ldots, 0, 0))U^T,$$
where the notation $x^+$ denotes $\max\{0, x\}$.

4.3.1 Protein Conformation Determination

Proteins are large biomolecules comprised of multiple amino acid residues, each of which typically consists of between 10 and 25 atoms. Proteins participate in virtually every cellular process, and knowledge of their structural conformation gives insight into the mechanisms by which they perform.

One of many techniques that can be used to determine conformation is nuclear magnetic resonance (NMR). Currently NMR is only able to non-destructively resolve relatively short distances (i.e., those less than $\approx 6\text{Å}$). In a typical protein, this corresponds to less than 9% of all non-zero inter-atom distances. This problem is naturally described as a 3-dimensional Euclidean distance matrix reconstruction problem specified by (4.35) where the partial EDM $D$ describes the data collected.

---

4When two amino acids form a peptide bond, a water molecule is formed. An amino acid residue is what remains of each amino acid after this reaction.
from NMR, and the rank upper-bound is given as $q = 3$. Algorithm 5 described the implementation of the Douglas–Rachford algorithm for this problem. Once the EDM has been successfully reconstructed, Algorithm 6 is used to produce the corresponding points in $\mathbb{R}^q$.

### Algorithm 5: The basic DR algorithm for EDM reconstruction.

| input | $x_0 \in \mathbb{E}$ and $\epsilon > 0$
| $n = 0$
| $p_0 \in P_{C_1}(x_0)$
| while $n = 0$ or $\|r_n - p_n\| > \epsilon \|p_n\|$ do
| $r_n \in P_{C_2}(2p_n - x_n)$
| $x_{n+1} = x_n + r_n - p_n$
| $p_{n+1} \in P_{C_1}(x_{n+1})$
| $n = n + 1$
| end
| output: $p_n \in \mathbb{E}$

### Algorithm 6: Conversion of an EDM to points in $\mathbb{R}^q$.

| input | A Euclidean distance matrix $D \in X$
| Set $L = I - ee^T/n$ where $e = (1, 1, \ldots, 1)^T$
| Set $\tau = -LDL/2$
| Let $\tau = USV^T$ be a singular value decomposition of $\tau$
| $Z =$ first $q$ columns of $U\sqrt{S}$
| $z_i =$ $i$th row of $Z$ for each $i = 1, 2, \ldots, m$
| output: Points $z_1, z_2, \ldots, z_q \in \mathbb{R}^q$ corresponding to $D$

It is worth noting that there are many other techniques for solving (variants of) the protein conformation problem (see for instance [134]). Such a discussion, however, is beyond the scope of this chapter.

In order to investigate the potential of this formulation, we consider six proteins, shown in Table 4.9, obtained from the RCSB Protein Data Bank [32]. For each of these proteins, NMR data was simulated by constructing the partial matrix containing only inter-atomic distances less than 6Å. Further, any inter-atom distance between atoms within the same residue was also assumed to be known, since the structure of given residue is usually known. Table 4.9 gives further information regarding the proportion of known distances.

Experiments were implemented in Cython and performed on a machine having an Intel Xeon E5540 @ 2.83GHz running Red Hat Enterprise Linux 6.5. The
reconstructed EDM, $D$, was converted to points $z_1, z_2, \ldots, z_m \in \mathbb{R}^3$ using Algorithm 6.

Two error metrics are reported. Denote the actual EDM by $x^*$. The first error metric is a measure of the error in the reconstructed EDM, and is given by

$$\text{EDM-error} = \|x^* - x\|_F = \sqrt{\sum_{i,j=1}^{m} |x^*_{ij} - x_{ij}|^2}.$$ 

Denote the actual atom positions by $z_1^*, z_2^*, \ldots, z_m^* \in \mathbb{R}^3$. The second error metric measures the error in the reconstructed atom positions $z_1, z_2, \ldots, z_m \in \mathbb{R}^3$. Since EDMs are invariant under translation, reflection, and rotation of the points by which they are induced, we first perform a Procrustes analysis [63] to obtain $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_m \in \mathbb{R}^3$. These points are a best fit of the reconstructed points when the aforementioned transformations are allowed. The second error metric is given by

$$\text{Position-error} = \sqrt{\sum_{k=1}^{m} \|z_k^* - \tilde{z}_k\|^2}.$$ 

When comparing the relative size of these two errors, it is worth noting that the summation in the EDM-error contains $m^2$ terms whereas the summation in the position-error contains only $3m$.

**Remark 4.3.5 (Previous work).** As part of previous work of Aragón Artacho, Borwein & Tam [5], reconstructions of the same six proteins were attempted using the Douglas–Rachford algorithm. Unlike Algorithm 5, the implementation of the Douglas–Rachford algorithm in [5] simply performed 5,000 iterations and returned the result. Whilst they were able to successfully reconstruct the smaller proteins in Table 4.9, their implementation failed for the larger proteins. Figure 4.10 shows two of their reconstructions. The first protein, 1HOE, containing 581 atoms was
well reconstructed, while the second, 1POA, containing 1067 atoms was poorly reconstructed.

Figure 4.10: Two examples of the actual conformation (left in each subfigure) and its Douglas–Rachford reconstruction (right in each subfigure) presented in [5].

In the computational experiments that follow, the stopping tolerance is taken to be $\epsilon = 10^{-5}$. In the following remark, we now provide some justification for this choice.

Remark 4.3.6 (Stopping criterion and tolerance). For each of the six proteins in Table 4.9, Figure 4.11 shows the relative error as a function of the number of iterations starting from a given initial point for the Douglas–Rachford method. We make some comments regarding the figure.

- When the number of iteration is less than 5000 the relative error exhibits non-monotone oscillatory behaviour which seems to provide much of the potency of the method. Heuristically it may allow the reflection method to sample regions and avoid settling at an inferior local minimum of the configuration space. In [5] it was observed that the alternating projection method fails to produce good reconstructions.

- When the relative error is between $10^{-3}$ and $10^{-4}$, it decreases sharply after which a period of more predictable decrease is observed.

- Beyond this point slower progress is made. The stopping tolerance is therefore chosen to be $\epsilon = 10^{-5}$, thus ensuring that the algorithm terminates in this region.

The change in successive iterates was found to also exhibit similar behaviour (not shown), and so provides another suitable candidate for a stopping criterion.

Remark 4.3.7. In the literature, it is not uncommon to report the relative error in decibels (dB) [5]. In other words,

$$\text{Relative error (dB)} = 10 \log_{10} \left( \frac{||P_B R_A x - P_A x||_F^2}{||P_A x||_F^2} \right).$$
Figure 4.11: The relative error as a function of iterations.

Note that the stopping criterion described by Algorithm 5 and Remark 4.3.6 is equivalent to asking that the decibel error be less than $10 \log_{10}(\epsilon^2)$. Requiring that $\epsilon = 10^{-5}$ corresponds to aiming at a relative error of $-100 \text{dB}$.

Table 4.10 gives results for the basic Douglas–Rachford algorithm presented in Algorithm 5. Images of the reconstructions are given in Figure 4.12. We make some comments regarding these results.

The EDM-error increases with increasing problem size; yet the same trend is not observed for the position-error for which 1PHT reported the largest error. For all of the proteins studied, the differences between the average and worst case results for the position-errors were small. This strongly suggests that the method can consistently produce a EDM which gives the desired atomic positions.

The second column of Figure 4.12 shows the conformation of the basic Douglas–Rachford reconstructions, which are visually indistinguishable from the actual conformation shown in the first column. This is an improvement from what was reported in [5] whose Douglas–Rachford reconstructions of two of the larger proteins, 1POA and 1AX8, gave unrealistic conformations consisting of disjoint blocks of atoms. In light of Remark 4.3.6 it is likely that this was due to premature algorithm termination.
### 4.3. EUCLIDEAN DISTANCE MATRIX RECONSTRUCTION

Table 4.10: Average (worst) results from five random replications of the basic Douglas–Rachford algorithm with $\epsilon = 10^{-5}$.

<table>
<thead>
<tr>
<th>Protein</th>
<th>EDM-Error</th>
<th>Position-Error</th>
<th>Iterations</th>
<th>Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1PTQ</td>
<td>3.6816 (4.0938)</td>
<td>0.1307 (0.1457)</td>
<td>4339.6 (4686)</td>
<td>0.28 (0.30)</td>
</tr>
<tr>
<td>1HOE</td>
<td>9.7475 (13.8503)</td>
<td>0.1781 (0.2636)</td>
<td>20 794.4 (21 776)</td>
<td>3.50 (3.67)</td>
</tr>
<tr>
<td>1LFB</td>
<td>9.8728 (17.2860)</td>
<td>1.1388 (2.1177)</td>
<td>22 346.2 (23 295)</td>
<td>4.64 (4.85)</td>
</tr>
<tr>
<td>1PHT</td>
<td>10.3709 (12.9557)</td>
<td>12.8782 (13.0056)</td>
<td>20 103.0 (20 251)</td>
<td>13.90 (14.00)</td>
</tr>
<tr>
<td>1POA</td>
<td>25.4225 (46.5804)</td>
<td>0.5844 (1.1639)</td>
<td>28 426.0 (29 766)</td>
<td>23.33 (24.47)</td>
</tr>
<tr>
<td>1AX8</td>
<td>25.7369 (39.4586)</td>
<td>0.6592 (0.9160)</td>
<td>17 969.8 (19 059)</td>
<td>15.04 (15.95)</td>
</tr>
</tbody>
</table>

Figure 4.12: Conformations of the six proteins, and their three reconstructions.
CHAPTER 4. APPLICATIONS

4.3.2 Performance Improving Heuristics

In our formulation of the protein confirmation problem, the most expensive step is the computation of the projection onto the rank constraint $C_2$. This requires the spectral-decomposition of an $(m - 1) \times (m - 1)$ symmetric matrix. In this section we propose problem specific heuristics which allow for this computation to sometimes be avoided.

To avoid performing a spectral-decomposition at every intersection, we propose a scheme in which the sequence $(r_n)_{n=1}^\infty$ in Algorithm 5 is only updated periodically. This approach is described in Algorithm 7, and computation results, with updates performed at every third iteration, in Table 4.11.

We now compare the results of this section to those of Section 4.3.1. For the results of this section, a small increase in the position-errors, and a larger increase in the EDM-errors was observed. The number of iterations required also increased, with this number almost doubling for 1PTQ. For all six test proteins, the total time required was less. The biggest improvement was 1POA whose total time was more than halved. The quality of the reconstructed conformations seem not to be adversely effected by the use of periodic rank projections, as can be seen in Figure 4.12.

Algorithm 7: The DR algorithm with $\rho$-periodic projections onto $C_2$.

input : $x_0 \in X, \rho \in \mathbb{N}$ and $\epsilon > 0$

$n = 0$;
$p_0 \in P_{C_1}(x_0)$;
while $n = 0$ or $\|r_n - p_n\| > \epsilon\|p_n\|$ do

if $n \mod \rho = 0$ then

$r_n \in P_{C_2}(2p_n - x_n)$;

else

$r_n = r_{n-1}$;
end

$x_{n+1} = x_n + r_n - p_n$;
$p_{n+1} \in P_{C_1}(x_{n+1})$;

$n = n + 1$;
end

output: $p_n \in X$

4.3.3 Additional Distance Data

Sections 4.3.1 and 4.3.2 considered the physically realistic setting in which distances below the threshold of 6Å were known. As noted, when the number of
Table 4.11: Average (worst) results from five random replications of the Douglas–Rachford algorithm with periodic rank projections with $\rho = 3$ and $\epsilon = 10^{-5}$.

<table>
<thead>
<tr>
<th>Protein</th>
<th>EDM-Error</th>
<th>Position-Error</th>
<th>Iterations</th>
<th>Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1PTQ</td>
<td>4.3709 (4.7200)</td>
<td>0.1919 (0.2240)</td>
<td>7160.6 (7595)</td>
<td>0.16 (0.17)</td>
</tr>
<tr>
<td>1HOE</td>
<td>10.1790 (12.1089)</td>
<td>0.2603 (0.2933)</td>
<td>20305.4 (22550)</td>
<td>1.21 (1.35)</td>
</tr>
<tr>
<td>1LFB</td>
<td>17.6532 (19.0984)</td>
<td>1.2709 (1.7243)</td>
<td>28983.8 (31211)</td>
<td>2.15 (2.31)</td>
</tr>
<tr>
<td>1PHT</td>
<td>23.8594 (25.9794)</td>
<td>13.1358 (13.2805)</td>
<td>20559.2 (20981)</td>
<td>5.03 (5.13)</td>
</tr>
<tr>
<td>1POA</td>
<td>49.8406 (51.3411)</td>
<td>1.0948 (1.2084)</td>
<td>33150.8 (39083)</td>
<td>9.55 (11.25)</td>
</tr>
<tr>
<td>1AX8</td>
<td>45.5203 (49.1866)</td>
<td>1.1696 (1.4482)</td>
<td>27080.6 (31250)</td>
<td>7.96 (9.20)</td>
</tr>
</tbody>
</table>

Atoms in a protein increases, the proportion of inter-atomic distances below this threshold compared to the total number of (non-zero) distances decreases.

To better understand the Douglas–Rachford method applied to larger problem instances, reconstruction of the six test proteins was attempted with the percentage of known non-zero distances constant. More precisely, we assumed the smallest 10% of inter-atomic distances to be known. Computation results for this experiment are given in Table 4.12.

Table 4.12: Average (worst) results from five random replications of the basic DR algorithm from the smallest 10% of inter-atomic distances with $\epsilon = 10^{-5}$.

<table>
<thead>
<tr>
<th>Protein</th>
<th>EDM-Error</th>
<th>Position-Error</th>
<th>Iterations</th>
<th>Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1PTQ</td>
<td>3.1924 (3.5936)</td>
<td>0.0963 (0.1213)</td>
<td>4014.4 (4184)</td>
<td>0.26 (0.27)</td>
</tr>
<tr>
<td>1HOE</td>
<td>8.0905 (10.4357)</td>
<td>0.0960 (0.1265)</td>
<td>15110.4 (15709)</td>
<td>2.54 (2.64)</td>
</tr>
<tr>
<td>1LFB</td>
<td>7.2941 (13.9893)</td>
<td>0.4647 (0.9182)</td>
<td>11060.6 (11912)</td>
<td>2.29 (2.46)</td>
</tr>
<tr>
<td>1PHT</td>
<td>14.1302 (20.2476)</td>
<td>0.3542 (0.4326)</td>
<td>6071.0 (6512)</td>
<td>4.19 (4.49)</td>
</tr>
<tr>
<td>1POA</td>
<td>19.5619 (31.1987)</td>
<td>0.1624 (0.2665)</td>
<td>11555.8 (13244)</td>
<td>9.44 (10.81)</td>
</tr>
<tr>
<td>1AX8</td>
<td>14.0747 (29.7259)</td>
<td>0.0940 (0.1922)</td>
<td>10099.2 (11125)</td>
<td>8.38 (9.23)</td>
</tr>
</tbody>
</table>

As could perhaps be predicted, when more distance information is incorporated the error metrics, and the number of iterations decrease. Problem size and EDM-error do not correlate as strongly compared to the results in Table 4.10. However, the general trend that larger problem sizes give larger EDM-errors is still observed. The most notable improvement, when compared to Table 4.10, is the position-error for 1PHT. This suggests that perhaps, in the realistic setting of Section 4.3.1, the underlying protein’s conformation (e.g., a compact or a dispersed conformation) is an important factor in the difficulty of the reconstruction problem.
4.3.4 Ionic Liquid Bulk Structure Determination

Ionic liquids (ILs) are salts (i.e., they are comprised of positively and negatively charged ions) having low melting points and typically occupying the liquid state at room temperature. An analogous reconstruction problem arising in the context of ionic liquid chemistry is to determine a given ionic liquid’s bulk structure. That is, the configuration of its ions with respect to each other (the structure of the individual ions is known).

In this section, we apply the Douglas–Rachford method to a simplified version of this problem. The EDM reconstruction problem (4.35) is used with the entries of the partial EDM assumed to be known whenever two atoms are bonded (i.e., when their Van der Waals radii, taken from [33], overlap).

Table 4.13 reports results for a propylammonium nitrate (PAN) data set consisting of 180 atoms. The corresponding rank-3 EDM completion problem has a total of 32,220 non-zero inter-atomic distances of which 5.95% form the partial EDM.

Table 4.13: Average (worst) results from five random replications of the basic DR algorithm, applied to ionic liquid bulk structure determination, with \( \epsilon = 10^{-5} \).

<table>
<thead>
<tr>
<th>EDM-Error</th>
<th>Position-Error</th>
<th>Iterations</th>
<th>Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6323 (0.6918)</td>
<td>2.0374 (2.5039)</td>
<td>41 553.2 (82 062)</td>
<td>0.22 (0.43)</td>
</tr>
</tbody>
</table>

As was the case in the protein conformation application, the difference between the average and worst case results for the two error metrics is observed to be small. The actual conformation of PAN, and its Douglas–Rachford reconstruction are shown in Figure 4.13. A high degree of visual coincidence is observed, although a small amount of the finer detail is missing.

4.3.5 Regularity Properties of Constraints

The goal of this section is to provide conditions under which the local convergence results in Section 3.1 apply. To do so, we must investigate supper-regularity and strong regularity of the sets \( C_1 \) and \( C_2 \) as defined in (4.35). For the convenience of the reader, we repeat the definition of these sets below.

\[
C_1 = \{X \in S^m : X \geq 0, X_{ij} = D_{ij} \text{ for all } (i, j) \in \Omega\},
\]

\[
C_2 = \left\{X \in S^m : Q(-X)Q = \begin{bmatrix} \hat{X} & d \\ d^T & \delta \end{bmatrix}, \hat{X} \in S^{m-1}_+, d \in \mathbb{R}^{m-1}, \text{ rank } \hat{X} \leq q, \delta \in \mathbb{R} \right\},
\]
4.3. **EUCLIDEAN DISTANCE MATRIX RECONSTRUCTION**

Figure 4.13: (Left) Actual conformation of PAN and (right) its DR reconstruction.

In what follows, we denote by $G: S^{m+1} \rightarrow S^{m+1}$ the linear isometry $X \mapsto Q(-X)Q$. Further note that $G^2 = I$.

Let $\bar{X}$ be a solution to the feasibility problem given by (4.35), then the corresponding $\hat{X}$ has rank $\hat{X} = s$. By Proposition 3.2.25, the set $S_s$, and hence $C_2$, is prox-regular at $\bar{X}$. Since the set $C_1$ is convex, hence also prox-regular. To appeal to the above theorem we need only show strong regularity of $\{C_1, C_2\}$ at $\bar{X}$.

We briefly summarise what follow. Corollary 4.3.11 shows that for physically realistic EDM reconstruction problems, checking strong regularity reduces to the non-existence of a non-zero solution to a linear system. For special instances of the problem, Example 4.3.13 gives a probabilistic argument showing strong regularity almost surely holds.

The next two Propositions give the normal cones to the sets in (4.35).

**Proposition 4.3.8** (Convex normal cone to $C_1$). Let $\bar{X} \in C_1$. Then

$$N_{C_1}(\bar{X}) = \left\{ Y + Z : \bar{X} \odot Z = 0, Z \leq 0, Y_{ij} = 0 \text{ for all } (i,j) \notin \mathcal{I} \right\}.$$ 

In particular, if $\bar{X}$ has zeros on the main diagonal, has strictly positive entries elsewhere, and $(j,j) \in \mathcal{I}$ for $j \in \{1, 2, \ldots, m\}$ then

$$N_{C_1}(\bar{X}) = \left\{ Y : Y_{ij} = 0 \text{ for all } (i,j) \notin \mathcal{I} \right\}.$$ 

**Proof.** Observe that $C_1$ is the intersection of the affine set $A$ and convex cone $K$ given by

$$A := \{ X \in S^{m+1} : X_{ij} = D_{ij} \text{ for all } (i,j) \in \mathcal{I} \}, \quad K := \{ X \in S^{m+1} : X \geq 0 \}.$$ 

These have normal cones given by

$$N_A(\bar{X}) = \{ Y \in S^{m+1} : Y_{ij} = 0 \text{ for all } (i,j) \notin \mathcal{I} \},$$

$$N_K(\bar{X}) = \{ Z \in S^{m+1} : \bar{X} \odot Z = 0, Z \leq 0 \},$$
hence by the sub-differential sum-rule, $N_{C_1}(\overline{X}) = N_A(\overline{X}) + N_K(\overline{X})$.

Further suppose $\overline{X}$ has zeros on the main diagonal and strictly positive entries elsewhere, and that $(j, j) \in I$ for $j \in \{1, 2, \ldots, m\}$. Let $Y + Z \in N_{C_1}(\overline{X})$. Since $\overline{X} \odot Z = 0$ and $Z \leq 0$ it follows that $Z$ is non-zero only on the main diagonal (where is non-positive). Thus $(Z + Y)_{ij} = 0$ for all $(i, j) \notin I$, and the result follows.

**Proposition 4.3.9** (Limiting normal cone to $C_2$). Let $\overline{X} \in C_2$ and $\hat{X} \in S_s$. Then

$$N_{G(C_2)} \left( G(\overline{X}) \equiv \begin{bmatrix} \hat{X} & d \\ d^T & \delta \end{bmatrix} \right) = \left\{ \begin{bmatrix} \hat{Y} & 0 \\ 0 & 0 \end{bmatrix} : \hat{Y} \in N_{S_s}(\hat{X}) \right\}. $$

In particular, if $\text{rank } \hat{X} = s$ then

$$N_{G(C_2)} \left( G(\overline{X}) \equiv \begin{bmatrix} \hat{X} & d \\ d^T & \delta \end{bmatrix} \right) = \left\{ \begin{bmatrix} \hat{Y} & 0 \\ 0 & 0 \end{bmatrix} : \hat{X}\hat{Y} = 0 \right\}. $$

**Proof.** The first formula follows from Theorem 3.2.14 and the fact that the normal cone of a Cartesian product is the Cartesian product of the normal cones [116, Prop. 1.2]. The second formula follows from Proposition 3.2.16.

The following proposition gives a formulation of strong regularity in terms of the normal cone formulae in Proposition 4.3.8 & 4.3.9. That is, in terms of objects of known form.

**Proposition 4.3.10** (Strong regularity of $\{C_1, C_2\}$). Let $\overline{X} \in C_1 \cap C_2$. Then strong regularity of $\{C_1, C_2\}$ is equivalent to

$$G(N_{C_1}(\overline{X})) \cap -G(N_{G(C_2)}(\overline{X})) = \{0\}. $$

**Proof.** Since $G$ is self-inverse (hence self-adjoint) and linear, applying [116, Th. 1.17] yields

$$\{0\} = N_{C_1}(\overline{X}) \cap -N_{C_2}(\overline{X}) \iff \{0\} = G(N_{C_1}(\overline{X})) \cap -G(N_{C_2}(\overline{X})) \iff \{0\} = G(N_{C_1}(\overline{X})) \cap -N_{G(C_2)}(G(\overline{X})).$$

The proof is complete.

The following corollary shows strong regularity of $\{C_1, C_2\}$ can be expressed in terms of a checkable condition (at least in principle).
Corollary 4.3.11. Suppose $X \in C_1 \cap C_2$ is irreducibly embeddable in $\mathbb{R}^s$, has zeros on the main diagonal and has strictly positive entries elsewhere. Then $\{C_1, C_2\}$ is strongly regular at $X$ if and only if the linear system

$$(Y, \hat{Y}) \in S^{m+1} \times S^m \text{ s.t. } G(Y) = \begin{bmatrix} \hat{Y} & 0 \\ 0 & 0 \end{bmatrix}, \hat{X}\hat{Y} = 0, Y_{ij} = 0 \text{ for all } (i, j) \notin I,$$  

(4.40)

has a non-zero solution.

Proof. Follows from Propositions 4.3.8, 4.3.9 & 4.3.10. \qed

We may thus deduce the following local convergence result.

Corollary 4.3.12 (Local convergence for EDM reconstruction). Suppose $X \in C_1 \cap C_2$ is a EDM which is irreducibly embeddable in $\mathbb{R}^s$ such that (4.40) holds. Then there exists $\delta > 0$ such that for any initial point $x_0 \in B_\delta(X)$, the Douglas–Rachford and cyclic projection methods convergence to a point in $C_1 \cap C_2$ with $R$-linear rate.

Proof. The sets $C_1$ and $C_2$ are super-regular at $X$ for the reasons outlined at the beginning of this section. Strong regularity of $\{C_1, C_2\}$ at $X$ follows from Corollary 4.3.11. The result now follows from Theorems 3.1.10 and 3.1.11. \qed

The matrix $\hat{X}$ and index set $I$ are specific to each instance of the problem. In practice, this collected data will vary in each experiment. Therefore as a template for application of the method to this problem, we give a probabilistic argument showing that for a special case of Corollary 4.3.11, strong regularity holds almost surely.

Example 4.3.13 (Probabilistic strong regularity). Under the assumptions of Corollary 4.3.11, suppose there exists a non-zero solution $(Y, \hat{Y})$ to (4.40). Since $\hat{X}\hat{Y} = 0$ it follows that range $\hat{Y} \subseteq \ker \hat{X}$, hence

$$\text{range } \begin{bmatrix} \hat{Y} & 0 \\ 0 & 0 \end{bmatrix} = \text{range } \hat{Y} \times \{0\} \subseteq \ker \hat{X} \times \{0\},$$

and hence

$$\text{range } Y = \text{range } G \left( \begin{bmatrix} \hat{Y} & 0 \\ 0 & 0 \end{bmatrix} \right) = Q \left( \text{range } \hat{Y} \times \{0\} \right) \subseteq Q \left( \ker \hat{X} \times \{0\} \right).$$

Since $X$ is irreducibly embeddable in $\mathbb{R}^s$, rank $\hat{X} = s$ hence dim ker $\hat{X} = m - s$.

We now focus on the special case in which $(m - s) = 1$ and $m \geq 2$. Under this assumption, we may express

$$\ker \hat{X} = \text{span}\{u\},$$
where without loss of generality \( u \in \mathbb{R}^m \) is assumed to be a unit vector. Further suppose there is a pair of indices \((i_0, j_0) \notin \mathcal{I}\) (i.e., there is at least one unspecified entry in the EDM reconstruction problem).

To proceed we make a probabilistic argument, which we can be summarized as follows: For a randomly chosen \( \ker \hat{X} \), the only solution to the linear system (4.40) is almost surely the zero matrix.

To this end, suppose \( u \) is chosen uniformly at random from the unit sphere in \( \mathbb{R}^m \), denoted \( S_{\mathbb{R}^m} \). This is equivalent to assuming \( u \times \{ 0 \} \) be chosen uniformly at random from \( S_{\mathbb{R}^m} \times \{ 0 \} \), which because \( Q \) is a linear isometry is equivalent to assuming \( v := Q(u \times \{ 0 \}) \) to be chosen uniformly at random from \( Q(S_{\mathbb{R}^m} \times \{ 0 \}) = S_{\mathbb{R}^{m+1}} \cap \{ e \}^\perp \) where \( e = (1, 1, \ldots, 1) \in \mathbb{R}^{m+1} \).

Since \( Y \neq 0 \) it follows that it is of the form \( Y = \lambda vv^T \) for some non-zero \( \lambda \in \mathbb{R} \). A computation shows

\[
Y = \lambda vv^T = \lambda \begin{bmatrix}
v_1^2 & v_1v_2 & \cdots & v_1v_{m+1} \\
v_1v_2 & v_2^2 & \cdots & v_2v_{m+1} \\
\vdots & \vdots & \ddots & \vdots \\
v_{m+1}v_1 & v_{m+1}v_2 & \cdots & v_{m+1}^2
\end{bmatrix}.
\]

The set \( \{ x \in S_{\mathbb{R}^{m+1}} \cap \{ e \}^\perp : x_{i_0} = 0 \text{ or } x_{j_0} = 0 \} \) is a null subset of \( S_{\mathbb{R}^{m+1}} \cap \{ e \}^\perp \), hence \( \Pr(v_{i_0}v_{j_0} = 0) = 0 \). But (4.40) implies that \( Y_{i_0j_0} = \lambda v_{i_0}v_{j_0} = 0 \). ♦

Remark 4.3.14. While the assumption \( m - s = 1 \) in Example 4.3.13 is not immediately useful to completely justify Section 4.3, nevertheless it still provides a useful template. More general arguments for probabilistic strong regularly are left for a future study. ♦

### 4.4 Matrix Completion: A Unified Framework

Many of the applications presented in this chapter, particularly those in Sections 4.2 and 4.3, are special cases of so-called matrix completion problems; a topic well-studied in the literature [97]. The purpose of this section is to place the already discussed applications within a unified matrix completion framework as well as discuss modelling considerations. Some further matrix completion applications with convexity are also discussed. We begin by introducing the matrix completion framework.

A partial (real) matrix is an \( m \times n \) array for which only entries in certain locations are known. A completion of the partial matrix \( A = (a_{ij}) \in \mathbb{R}^{m \times n} \), is a matrix \( B = (b_{ij}) \in \mathbb{R}^{m \times n} \) such that if \( a_{ij} \) is specified then \( b_{ij} = a_{ij} \). The problem of (real) matrix completion is as follows: Given a partial matrix, find a completion which possess certain properties of interest.
Recall the $N$-set feasibility problem

$$\text{find } X \in \bigcap_{i=1}^{N} C_i \subseteq \mathbb{R}^{m \times n}. \quad (4.41)$$

Let $A$ be the partial matrix to be completed. Matrix completion can now be modelled as a feasibility problem by choosing the sets $C_1, C_2, \ldots, C_N$ such that their intersection contains all completions of $A$ which possess the property of interest. For example, the simplest, and often useful case, is given by choosing $C_1$ to be the set of all completions of $A$, and to choose $C_2, \ldots, C_N$ to be such that their intersection is the set of all matrices having the properties of interest.

**Example 4.4.1 (Sudoku as matrix completion).** In the binary Sudoku formulation of Section 4.2.1, the set $C_1$ is taken to be all binary completions of the three dimensional array. The sets $C_2, \ldots, C_5$ are chosen such their intersection is the collection of all legal Sudoku puzzles; the latter being the properties of interest.

As illustrated in the previous sections, there are typically many ways to formulate the constraint set for a given matrix completion problem. To apply the Douglas–Rachford method, it is important that the chosen sets have nearest point projections which are succinctly simple to compute, and ideally in closed form. It is frequently the case that there is a trade-off between the number of sets in the intersection, and the simplicity of their projections (compare Sections 4.2.1 and 4.2.2). For example, one extreme involves encoding the property of interest in a single constraint set. In this case, it is likely that its projection is difficult to compute.

To illustrate this philosophy, consider the following example which we shortly revisit in Section 4.4.3.

**Example 4.4.2 (Modelling philosophy).** Suppose the property of interest is the constraint

$$\left\{ X \in \mathbb{R}^{m \times n} : X_{ij} \geq 0, \sum_{k=1}^{m} X_{kj} = 1 \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n \right\}. \quad (4.42)$$

This set is equal to the intersection of $C_2$ and $C_3$ where

$$C_2 = \left\{ X \in \mathbb{R}^{m \times n} : X_{ij} \geq 0 \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n \right\},$$

$$C_3 = \left\{ X \in \mathbb{R}^{m \times n} : \sum_{i=1}^{m} X_{ij} = 1 \text{ for } j = 1, \ldots, n \right\}.$$

Here the projections onto the cone $C_2$ and the affine space $C_3$ can be easily computed (see Section 4.4.3). In contrast, the projection directly onto $C_2 \cap C_3$ amounts
to finding the nearest point in the convex hull of the set of matrices having a one in each row and remaining entries zero. This projection is less straightforward, and has no explicit form [56].

The order of the constraint sets in (4.41) also requires consideration. Within the framework of feasibility problems, there can be numerous ways to model a given type of problem. The product space reformulation of Section 1.4 gives one example, even without assuming any additional knowledge of the underlying problem; there is some art involved.

For matrix completion problems with two constraints, we directly apply the Douglas–Rachford method to $C_1 \cap C_2$, with the reflection first performed with respect to the set $C_1$. For matrix completion problems with more than two constraints, it is preferable to use the Douglas–Rachford method on the product formulation of Section 1.4 with the reflection with respect to $D$ performed first. In this case, the solution is obtained by projecting onto $D$ and thus can be monitored by considering only a single product coordinate.

As a heuristic for problems involving one or more non-convex sets, the sensitivity of the Douglas–Rachford method to the formulation used must be emphasised. In contrast, for convex applications, the formulation influences performance of the algorithm, while in the non-convex setting, the formulation determines whether or not the algorithm can successfully and reliably solve the problem at hand. We also note that direct applications to feasibility problems with integer constraints have been largely unsuccessful. For further details, see [6].

### 4.4.1 Positive semi-definite matrices

Recall that $S^n$ denotes the set of all real $n \times n$ symmetric matrices, and that the set of positive semi-definite matrices is denoted

$$S_+^n := \{ A \in S^n : x^T A x \geq 0 \text{ for all } x \in \mathbb{R}^n \}. \quad (4.42)$$

The Löwner partial order is defined on $S^n$ by taking $A \succeq B$ if $A - B \in S_+^n$. An alternative characterisation of positive semi-definite matrix is that they are precisely the symmetric matrices having all non-negative eigenvalues. For further details, the reader is referred back to Section 3.2.

Consider the matrix completion problem of reconstructing a positive semi-definite matrix $A$ where only some of its entries are known. Let $\Omega$ denote the indices for which the entries of $A$ are known (i.e., $(i,j) \in \Omega$ if $A_{ij}$ is known). Without loss of generality, we may assume that $\Omega$ is symmetric in the sense that $(i,j) \in \Omega$ if and only if $(j,i) \in \Omega$.

A matrix $X$ is a positive semi-definite matrix that completes $A$ if and only if

$$X \in C_1 \cap C_2, \quad (4.43)$$
where the constraint sets are defined by

\[ C_1 = \{ X \in \mathbb{R}^{n \times n} : X_{ij} = A_{ij} \text{ for all } (i,j) \in \Omega \}, \quad C_2 = S^n. \quad (4.44) \]

The set \( C_1 \) is a closed affine subspace with an entry-wise constraint. As we have already seen, its projection is straightforward, and given entry-wise by

\[ P_{C_1}(X)_{ij} = \begin{cases} A_{ij} & \text{if } (i,j) \in \Omega, \\ X_{ij} & \text{otherwise}; \end{cases} \quad (4.45) \]

for all \( i, j = 1, \ldots, n \).

The projection onto the set of positive semi-definite matrices can be computed using a polar decomposition; something which is implemented in any standard linear algebra library.

**Theorem 4.4.3** (Nearest positive semi-definite matrix). Let \( X \in \mathbb{R}^{n \times n} \). Define \( Y = (A + A^T)/2 \) and let \( Y = UP \) be a polar decomposition of \( Y \). Then

\[ P_{C_2}(X) = \frac{Y + P}{2}. \quad (4.46) \]

**Proof.** See, for instance, [87, Th. 2.1]. \( \square \)

**Remark 4.4.4.** For \( X \in S^n \), \( Y = X \) in the statement of Theorem 4.4.3. If this is the case, the computation of \( P_{C_2} \) is also simplified. If the initial matrix is symmetric, then the corresponding Douglas–Rachford iterates are symmetric too. This condition can be easily satisfied. For instance, if \( X \in \mathbb{R}^{n \times n} \) then the iterates can instead be computed starting at \( P_{S_n}(X) = (X + X^T)/2 \) or \( XX^T \in S^n \). This restriction was already considered in Section 4.3. \( \diamond \)

**Remark 4.4.5** (Positive definite matrices). Recall that a real symmetric \( n \times n \) matrix is said to be positive definite if the inequality in (4.42) holds strictly whenever \( x \neq 0 \). Denote the set of all such matrices by \( S^n_{++} \). Since \( S^n_{++} \) is not closed, the problem of positive definite matrix completion cannot be directly cast within this framework by setting \( C_2 = S^n_{++} \).

In practice, one might wish to consider a closed convex subset of \( S^n_{++} \). For example, one could instead define

\[ C_2 = \{ X \in \mathbb{R}^{n \times n} : X^T = X, x^TXx \geq \epsilon \| x \|^2 \text{ for all } x \in \mathbb{R}^n \}, \quad (4.47) \]

for some small \( \epsilon > 0 \). Then (4.47) is equivalent to requiring that the eigenvalues be not less than \( \epsilon \). \( \diamond \)
4.4.2 Correlation matrices

An important class of positive semi-definite matrices is the class of correlation matrices. In this section we describe an appropriate modification of (4.43) to handle the problem of reconstructing a partial correlation matrix.

Given \( n \) random variables, the associated correlation matrix is an element of \([-1, 1]^{n \times n}\) whose \( ij \)th entry is the correlation between the \( i \)th and \( j \)th variables. Since, any random variable perfectly correlates with itself, the entries along the main diagonal of any correlation matrix are all ones. As a consequence, without loss of generality, we may and do assume

\[
\{(i, i) : i = 1, \ldots, n\} \subseteq \Omega, \text{ and } A_{ii} = 1 \text{ for } i = 1, \ldots, n. \tag{4.48}
\]

On the surface, it would appear that formulation (4.43) requires further modification to ensure that its entries are contained in \([-1, 1]\). However, the following proposition shows that whenever (4.48) holds, \( A \) is necessarily contained in \([-1, 1]^{n \times n}\).

**Proposition 4.4.6 (Bounds on PSD matrix entries).** Let \( A = (a_{ij}) \in S_+^n \). Then \( a_{ii}a_{jj} \geq a_{ij}^2 \) for all \( i, j \in \{1, 2, \ldots, m\} \).

**Proof.** See [92, p. 398]. \( \Box \)

Thus, if \( A \) is an incomplete correlation matrix, \( X \) is correlation matrix that completes \( A \) if and only if \( X \in C_1 \cap C_2 \) where the constraints are as defined in (4.44).

To illustrate the formulation, we consider the problem of generating a random sample of correlation matrices. This is the case, for example, when one uses simulation to determine an unknown probability distribution [123]. The Douglas–Rachford method provides a heuristic for generating such a sample by applying the method to initial points chosen according to some probability distribution. In this case, the set of known indices, and the corresponding values of the known entries, are given by

\[
\Omega = \{(i, i) : i = 1, \ldots, n\}, \text{ and } A_{ii} = 1 \text{ for } i = 1, \ldots, n.
\]

The distribution of the entries in 100,000 matrices of size \( 5 \times 5 \) obtained from three different sets of choices of initial point distribution is shown in Figure 4.14.

4.4.3 Stochastic matrices

A matrix \( A = (A_{ij}) \in \mathbb{R}^{m \times n} \) is said to be doubly stochastic if

\[
\sum_{i=1}^{m} A_{ij} = \sum_{j=1}^{n} A_{ij} = 1, \ A_{ij} \geq 0 \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n. \tag{4.49}
\]
4.4. MATRIX COMPLETION: A UNIFIED FRAMEWORK

The set of all doubly stochastic matrices are known as the Birkhoff polytope, and can be realised as the convex hull of the set of permutation matrices [37, Th. 1.25].

Consider the matrix completion problem in which a doubly stochastic matrix is to be reconstructed when only some of its entries are known. Denote by \( \Omega \) the location of these entries (i.e., \((i, j) \in \Omega \) if \( A_{ij} \) is known). The set of all such candidates is given by

\[
C_1 = \{ X \in \mathbb{R}^{m \times n} : X_{ij} = A_{ij} \text{ for all } (i, j) \in \Omega \},
\]

which is similar to (4.44). The Birkhoff polytope may be expressed as the intersection of the three convex sets

\[
C_2 = \left\{ X \in \mathbb{R}^{m \times n} : \sum_{i=1}^{m} X_{ij} = 1 \text{ for } j = 1, \ldots, n \right\},
\]

\[
C_3 = \left\{ X \in \mathbb{R}^{m \times n} : \sum_{j=1}^{n} X_{ij} = 1 \text{ for } i = 1, \ldots, m \right\},
\]

\[
C_4 = \left\{ X \in \mathbb{R}^{m \times n} : X_{ij} \geq 0 \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n \right\}.
\]

The matrix \( X \) is a double stochastic matrix that completes \( A \) if and only if

\[
X \in C_1 \cap C_2 \cap C_3 \cap C_4.
\]

As in (4.45), the set \( C_1 \) is a closed affine subspace whose projection is given pointwise by

\[
P_{C_1}(X)_{ij} = \begin{cases} A_{ij} & \text{if } (i, j) \in \Omega, \\ X_{ij} & \text{otherwise}; \end{cases}
\]
for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

The projection onto $C_2$ (resp. $C_3$) can be easily computed by applying the following proposition row-wise (resp. column-wise).

**Proposition 4.4.7.** Let $S = \{ x \in \mathbb{R}^m : \sum_{i=1}^{m} x_i = 1 \}$. For any $x \in \mathbb{R}^m$,

$$P_S(x) = x + \frac{1}{m} \left( 1 - \sum_{i=1}^{m} x_i \right) e,$$

where $e = [1, 1, \ldots, 1]^T$.

**Proof.** Since $S = \{ x \in \mathbb{R}^n : \langle e, x \rangle = 1 \}$, the result follows from the standard formula for the projection onto a hyperplane (see, for instance, [76, Sec. 4.2.1]). $\square$

The projection of $A$ onto $C_4$ also has a simple formula. It is given entry-wise by

$$P_{C_4}(A)_{ij} = \max\{0, A_{ij}\},$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

**Remark 4.4.8 (Singly stochastic matrices).** The problem of singly-stochastic matrix completion can also be addressed. The problem of row (resp. column) stochastic matrix completion is formulated by dropping constraint $C_3$ (resp. $C_2$). $\Diamond$

**Remark 4.4.9 (An alternative two-set formulation).** Finally, let us mention a related work [140], where Takouda applies Dykstra’s algorithm [70] to find the closest square doubly-stochastic matrix to a given one in $\mathbb{R}^{n \times n}$ by considering the intersection of two sets: $C_2 \cap C_3$ and $C_4$. In particular, he shows that $P_{C_2\cap C_3}(X) = WXW + J$, where $W = I - J$ and $J = \frac{1}{n}ee^T$ [140, Prop. 4.4]. $\Diamond$
Chapter 5

Conclusions and Open Questions

Despite their inception over half a century ago, projection and reflection methods continue to receive significant attention. From the view of the practitioner, the methods are appealing due to their relative simplicity, ease-of-implementation, and empirical good performance. From a theoretical perspective, the lack of a sound theoretical foundation to justify, and to explain, many of these observed successes begs for further investigation.

Whilst projection methods such as the alternating projection method have been traditionally the most popular members of the family, in recent times, those methods which can be realised as reflection methods have emerged as having promising potential. Mostly striking is their good success when applied to highly non-convex problems including those arising in imaging applications, combinatorial optimisation, and low-dimensional distance matrix reconstruction.

The contributions of this dissertation can be concisely summarised as follows.

1. The discovery and analysis, in the convex setting, of the cyclic Douglas–Rachford method; the first many-set extension of the classical Douglas–Rachford method not requiring a product space reformulation.

2. A thorough study of the local regularity properties of sparse non-negative vector and low-rank positive semi-definite matrix constraints. A knowledge of these properties is of fundamental importance, for instance, in the justification of non-convex applications including those in Chapter 4.

3. The first global convergence results for the classical Douglas–Rachford method which apply to problems of a combinatorial nature. These results complement the emerging local theory as well as being of interest in their own right.

4. An extensive empirical investigation of non-convex problems which the Douglas–Rachford method can successfully solve. This growing library of problems is important for both the refinement of applications and to guide theoretical developments.
To conclude this work, we indicate three open questions for further work.

Chapter 2 introduced a number of variants of the Douglas–Rachford method. While the behaviour of the cyclic projection method, the classical Douglas–Rachford method, and the cyclic Douglas–Rachford method with convexity is described in Theorems 2.2.6 and 2.3.7 and Section 2.6, the behaviour of the aforementioned variants have not been fully investigated.

**Problem 5.1 (Behaviour of Douglas–Rachford variants).** How do the various Douglas–Rachford variants behave when applied to inconsistent convex feasibility problems and to non-convex feasibility problems?

Chapter 3 reviewed the emerging local non-convex theory. As outlined in Remark 3.1.20, the existing local theory is often too restrictive to be applied where it is desired.

**Problem 5.2 (Non-convex convergence theory).** Is it possible to develop a local non-convex convergence theory which is more broadly applicable?

Beyond the directions outlined in Remark 3.1.20, another avenue of investigation for Problem 5.2 may be to look for results which hold generically.

The fixed point sets of the alternating projection and Douglas–Rachford operators are, in general, of quite different forms. The former can be described in terms of (local) best approximation pairs and, in the convex setting, the latter can be described using normal cones (Proposition 2.3.1). A better understanding of the Douglas–Rachford operator’s fixed point sets would be of use in the development of non-convex theory.

**Problem 5.3 (Non-convex fixed point sets).** How do the fixed point sets of the underlying operators in the alternating projection method, the Douglas–Rachford method and its variants compare in non-convex settings?
Bibliography


[73] V. Elser. Personal communication, 08 2012.


[137] B. Sims. Personal communication, April 2013.


## Index

- $\ell_0$-functional, 87
- affine-hull regularity, 49, 50, 63
- asymptotic regularity, 13, 14, 17
- averaged mapping, 3
- best approximation pair, 7, 22, 144
- Chebyshev set, 2
- Clarke regularity, 46
- codimension, 8, 89
- compressive sensing, 86, 95
- cone, 88
- convex relaxation, 87
- cyclic Douglas–Rachford
  - averaged variant, 33
  - fixed point set, 32
  - method, 29, 32
  - operator, 31
- cyclic projection
  - averaged variant, 24
  - limit cycles, 22
  - method, 9, 18, 19, 21, 33, 47, 50, 51, 89, 94
- distance function, 2, 72
  - differentiability, 3
- distance matrix, 5, 51, 120
- Douglas–Rachford
  - averaged anchored variant, 35
  - cyclically anchored variant, 35
  - fixed point set, 10
  - method, 9, 24, 26, 27, 48, 50, 90, 94, 99, 110, 113, 120
  - operator, 25, 31, 34
  - splitting, 28
- experimental mathematics, 85
- feasibility problem, 4, 9
  - consistent, 7
  - inconsistent, 7
- Hadamard matrix, 112
  - circulant, 119
  - complex, 115
  - distinct, 115
  - equivalent, 115
  - skew, 116
- Hilbert space, 1, 5, 7
  - induced norm, 1
  - inner product, 1
  - lattice, 5, 86, 88
- Hundal counter-example, 23, 26, 33, 86
- interior, 8, 50
- intrinsically transversal, 51
- knapsack problem, 6
- limiting normal cone, 132
- local angle, 47
- matrix, 135
  - orthogonal, 52
  - permutation, 52

157
INDEX

positive semi-definite, 6, 52, 60  
symmetric, 5, 52
matrix completion, 6, 86, 101, 114, 121, 135
matrix:positive semi-definite, 54
metric regularity, 46
moment problem, 5, 86, 90
monotone operator, 27
  maximal monotone operator, 28
Moreau theorem, 55, 89
nonexpansive mapping, 12
  averaged mapping, 12, 16
  composition, 15
  contraction, 12
  convex combination, 15
  firmly nonexpansive mapping, 12
  fixed point set, 15
nonogram puzzle, 110
normal cone, 45
  convex, 25, 28, 44, 144
  limiting, 45, 53, 57, 62, 63, 96
  proximal, 45, 53, 60, 62
  restricted limiting, 49, 63
  restricted proximal, 49
NP (complexity), 7, 87, 96, 99
operator splitting, 27
Opial’s theorem, 14, 16
Pazy’s trichotomy theorem, 17
polar cone, 44, 88
product space formulation, 7, 29, 100, 137
projection, see projector
projector, 2, 4, 29, 116
  affine set, 5
  circulant matrix, 120
  convex, 2, 18
  diagonal space, 8
  distance matrix, 121, 122
  Hadamard matrix sets, 113, 114, 118
hyperplane, 5
low-rank, 61
Motzkin–Bunt theorem, 2
nonogram sets, 112
permutation set, 97
product set, 8
sparse, 54
Sudoku sets, 101
translation and dilation, 4
  weak continuity, 26
prox-regularity, 45, 46, 64, 132
proximal set, 2, 65
rank, 6, 53, 96, 133
rearrangement inequality, 97
reflector, 3
  convex characterisation, 4
  reflection, 4
  translation and dilation, 4
resolvent, 28
spectral decomposition, 6, 123
spectral set, 52
strong convergence, 33, 87
strong regularity, 46–48, 50, 63, 96, 132
Sudoku, 99
super-regularity, 46–48, 50, 132
symmetric set, 52
trace, 5, 52
zeros, 28
Notation and Symbols

\[ x_n \rightarrow x \]  the sequence \((x_n)_{n=1}^{\infty}\) converges strongly to the point \(x\)

\[ x_n \overset{w}{\rightharpoonup} x \]  the sequence \((x_n)_{n=1}^{\infty}\) converges weakly to the point \(x\)

\[ f : D \to C \]  \(f\) is a mapping from the set \(C\) to \(D\)

\[ f : D \rightharpoonup C \]  \(f\) is a set-valued mapping from the set \(C\) to \(D\)

\[ \| \cdot \| \]  inner-product induced by the Hilbert norm

\[ \| x \|_0 \]  \(\ell_0\)-functional of evaluated at the vector \(x\)

\[ \| X \|_F \]  Frobenius norm of the matrix \(X\)

\[ \langle x, y \rangle \]  inner-product of the vectors \(x\) and \(y\)

\[ \mathcal{B}_\delta(x) \]  closure ball of radius \(\delta > 0\) centred at \(x\)

\[ \text{cone}(C) \]  cone generated by the set \(C\)

\[ \text{conv}(C) \]  convex hull of the set \(C\)

\[ C \]  product constraint, see p. 7

\[ \text{cl}(C) \]  (strong) closure of the set \(C\)

\[ d(x, C) \]  distance between the point \(x\) and the set \(C\)

\[ \mathbf{D} \]  diagonal subspace, see p. 7

\[ \text{diag}(x) \]  diagonal matrix with diagonal entries given by the vector \(x\)

\[ \text{dom}(f) \]  domain of the function \(f\)

\[ \mathbb{E} \]  finite dimensional Hilbert space

\[ \text{Fix}(T) \]  fixed point set of the operator \(T\)

\[ \text{gra} \]  graph of a function

\[ \mathcal{H} \]  real Hilbert space

\[ \mathcal{H}_+ \]  the lattice cone of a Hilbert lattice \(\mathcal{H}\)

\[ I \]  identity map

\[ \text{int}(C) \]  interior of the set \(C\)

\[ C_1 + C_2 \]  Minkowski sum of the sets \(C_1\) and \(C_2\)
\(C_1 - C_2\)  Minkowski difference of the sets \(C_1\) and \(C_2\)
\(N_{C}\)  limiting normal cone to \(C\)
\(N_{C}\) \(^{\text{conv}}\)  (convex) normal cone to \(C\)
\(N_{C}\) \(^{\text{prox}}\)  proximal normal cone to \(C\)
\(N_{C}\) \(^{L-\text{prox}}\)  \(L\)-restricted proximal normal cone to \(C\)
\(O\)  Landau notation for asymptotics
\(O^m\)  set of \(m \times m\) orthogonal matrices
\(P^m\)  set of \(m \times m\) permutation matrices
\(P_C(x)\)  nearest point projector of the point \(x\) onto the set \(C\)
\(R_C(x)\)  reflector of the point \(x\) with respect to the set \(C\)
\(\text{range}(f)\)  range of the function \(f\)
\(\text{rank}(X)\)  rank of the matrix \(X\)
\(J_A\)  resolvent of the set-valued map \(A\)
\(\mathbb{R}\)  set of real numbers
\(\mathbb{R}_+\)  set of non-negative real numbers
\(\mathbb{R}_+^+\)  set of positive real numbers
\(\mathbb{R}^{m \times n}\)  set of \(m \times n\) real matrices
\(\mathbb{R}_+^{m \times n}\)  set of \(m \times n\) real matrices with non-negative entries
\(\mathbb{R}_+^{m \times n}\)  set of \(m \times n\) real matrices with positive entries
\(\text{ri}(C)\)  relative interior of the set \(C\)
\(\text{span}\)  span of a set of vectors
\(\mathbb{S}^n\)  set of symmetric real \(n \times n\) matrices
\(\mathbb{S}_+^n\)  set of positive semi-definite \(n \times n\) matrices
\(\mathbb{S}_+^n\)  set of positive definite \(n \times n\) matrices
\(T_{C_1,C_2}\)  the Douglas–Rachford operator for the sets \(C_1\) and \(C_2\)
\(T_{[C_1,\ldots,C_N]}\)  the cyclic Douglas–Rachford operator for the sets \(C_1,\ldots,C_N\)
\(\text{tr}(X)\)  trace of the matrix \(X\)
\(X^T\)  transpose of the matrix \(X\)
\(\text{zer}\)  zeros of a function