Theory and Applications of Convex and Non-convex Feasibility Problems

Laureate Prof. Jonathan Borwein with Matthew Tam
http://carma.newcastle.edu.au/DRmethods/paseky.html

THE UNIVERSITY OF NEWCASTLE AUSTRALIA
CARMA

Originally prepared for:

Spring School on Variational Analysis VI
Paseky nad Jizerou, April 19–25, 2015

Last Revised: May 6, 2016

1. Introduction and Outline

Spring School on Variational Analysis 2015

For Spring School on Function Spaces and Lineability 2015, click here

What am I if I will not participate?
- Antoine de Saint-Exupery

Dear Colleague,

Following a longstanding tradition, the Faculty of Mathematics and Physics, Charles University in Prague and the Academy of Sciences of the Czech Republic will organize the Spring School on Variational Analysis VI. The School will be held in Paseky nad Jizerou, in a chalet in the Krkonose Mountains, **April 19 - 25, 2015**.

The program will consist of series of lectures on

Variational Analysis

and its Applications

delivered by

Jonathan M. Borwein
The University of Newcastle, Australia
Theory and Applications of Convex and Non-convex Feasibility Problems

Marián Fabian
Academy of Sciences of the Czech Republic, Prague, Czech Republic
Separable Reductions and Rich Families in Theory of Fréchet Subdifferentials

Alexander Ioffe
Technion, Haifa, Israel
Variational Analysis and Optimization Theory

David Russell Luke
Georg-August-Universität Göttingen, Germany
Variational Methods in Numerical Analysis

The purpose of this meeting is to bring together researchers with common interest in the field. There will be opportunities for informal discussions. Graduate students and others beginning their mathematical career are encouraged to participate.

Introduction

A feasibility problem requests solution to the problem

$$\text{Find} \quad x \in \bigcap_{i=1}^{N} C_i,$$

where $C_1, C_2, \ldots C_N$ are closed sets lying in a Hilbert space $\mathcal{H}$.

We consider iterative methods based on the non-expansive properties of the metric projection operator

$$P_C(x) := \operatorname{argmin}_{c \in C} \|x - c\|$$

or reflection operator $R_C := 2P_C - I$ on a closed convex set C.

The two methods which we focus are on the method of alternating projections (MAP) and the Douglas–Rachford method (DR).

Introduction

A feasibility problem requests solution to the problem

$$\text{Find} \quad x \in \bigcap_{i=1}^{N} C_i,$$

where $C_1, C_2, \dots C_N$ are closed sets lying in a Hilbert space $\mathcal{H}$.

We consider iterative methods based on the non-expansive properties of the metric projection operator

$$P_C(x) := \operatorname{argmin}_{c \in C} \|x - c\|$$

or reflection operator $R_C := 2P_C - I$ on a closed convex set C.

The two methods which we focus are on the method of alternating projections (MAP) and the Douglas–Rachford method (DR).

Introduction

A feasibility problem requests solution to the problem

$$\text{Find} \quad x \in \bigcap_{i=1}^{N} C_i,$$

where $C_1, C_2, \ldots C_N$ are closed sets lying in a Hilbert space $\mathcal{H}$.

We consider iterative methods based on the non-expansive properties of the metric projection operator

$$P_C(x) := \operatorname{argmin}_{c \in C} \|x - c\|$$

or reflection operator $R_C := 2P_C - I$ on a closed convex set C.

The two methods which we focus are on the method of alternating projections (MAP) and the Douglas–Rachford method (DR).

Introduction

These methods work best when the projection on each set C_i is easy to describe or approximate. These methods are especially useful when the number of sets involved is large as the methods are fairly easy to parallelize.

The theory is pretty well understood when all sets are convex but much less clear in the non-convex case. But as we shall see application of this case has had may successes. So this is a fertile area for both pure and applied study.

The five hours of lectures will cover the following topics.

1. **Feasibility problems:** convex theory, nonexpansivity, Fejér monotonicity & convergence of MAP and variants.
2. **The Douglas–Rachford Method:** convex Douglas–Rachford iterations and variants.
3. **Non-convex Douglas Rachford iterations** and **iterative geometry**.
4. **Applications to completion problems**: an introduction & detailed case studies.

- Each lecture will contain closing commentary, open questions, and exercises.

Introduction

These methods work best when the projection on each set C_i is easy to describe or approximate. These methods are especially useful when the number of sets involved is large as the methods are fairly easy to parallelize.

The theory is pretty well understood when all sets are convex but much less clear in the non-convex case. But as we shall see application of this case has had may successes. So this is a fertile area for both pure and applied study.

The five hours of lectures will cover the following topics.

1. **Feasibility problems:** convex theory, nonexpansivity, Fejér monotonicity & convergence of MAP and variants.
2. **The Douglas–Rachford Method:** convex Douglas–Rachford iterations and variants.
3. **Non-convex Douglas Rachford iterations** and **iterative geometry**.
4. **Applications to completion problems**: an introduction & detailed case studies.

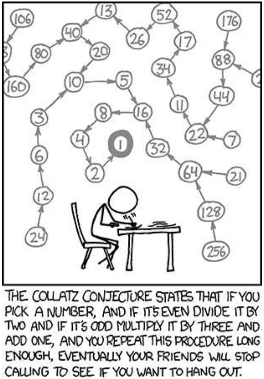
- Each lecture will contain closing commentary, open questions, and exercises.

Motivation

The need to integrate and iterate real theory with real models for real applications:

- Good theoretical understanding
  - you can not use what you do not know
  - you can work inductively
- Careful modelling of applications
  - the model matters especially in the nonconvex case
  - moving to application specific refinements
- Good implementations
  - starting with 'general purpose agents'
  - moving to application specific refinements

Introduction

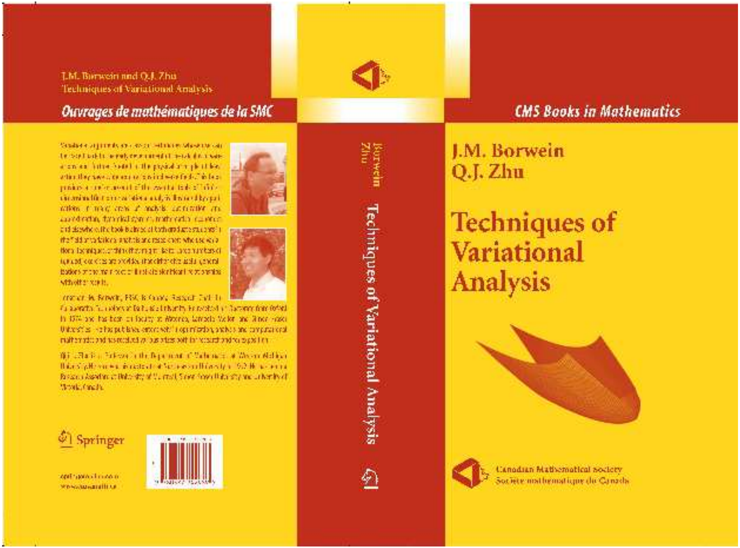


Lectures are available online at:

http://carma.newcastle.edu.au/DRmethods/paseky.html

2. Convex Feasibility Problems

Techniques of Variational Analysis



This lecture is based on Chapter 4.5: Convex Feasibility Problems

Abstract

Let X be a Hilbert space and let $C_n, n = 1, \ldots, N$ be convex closed subsets of X. The convex feasibility problem is to find some point

$$x \in \bigcap_{n=1}^{N} C_n,$$

when this intersection is non-empty.

In this talk we discuss projection algorithms for finding such a feasibility point. These algorithms have wide ranging applications including:

- solutions to convex inequalities,
- minimization of convex nonsmooth functions,
- medical imaging,
- computerized tomography, and
- electron microscopy

Abstract

Let X be a Hilbert space and let $C_n, n = 1, \ldots, N$ be convex closed subsets of X. The convex feasibility problem is to find some point

$$x \in \bigcap_{n=1}^{N} C_n,$$

when this intersection is non-empty.

In this talk we discuss projection algorithms for finding such a feasibility point. These algorithms have wide ranging applications including:

- solutions to convex inequalities,
- minimization of convex nonsmooth functions,
- medical imaging,
- computerized tomography, and
- electron microscopy

Projections

We start by defining projection to a closed convex set and its basic properties. This is based on the following theorem.

Theorem 4.5.1 (Existence and Uniqueness of Nearest Point)

Let X be a Hilbert space and let C be a closed convex subset of X. Then for any $x \in X$, there exists a unique element $\bar{x} \in C$ such that

$$\|x - \bar{x}\| = d(C; x).$$

Proof. If $x \in C$ then $\bar{x} = x$ satisfies the conclusion. Suppose that $x \notin C$. Then there exists a sequence $x_i \in C$ such that $d(C; x) = \lim_{i\to\infty} \|x - x_i\|$. Clearly, x_i is bounded and therefore has a subsequence weakly converging to some $\bar{x} \in X$.

Projections

We start by defining projection to a closed convex set and its basic properties. This is based on the following theorem.

Theorem 4.5.1 (Existence and Uniqueness of Nearest Point)

Let X be a Hilbert space and let C be a closed convex subset of X. Then for any $x \in X$, there exists a unique element $\bar{x} \in C$ such that

$$\|x - \bar{x}\| = d(C; x).$$

Proof. If $x \in C$ then $\bar{x} = x$ satisfies the conclusion. Suppose that $x \notin C$. Then there exists a sequence $x_i \in C$ such that $d(C; x) = \lim_{i \to \infty} \|x - x_i\|$. Clearly, x_i is bounded and therefore has a subsequence weakly converging to some $\bar{x} \in X$.

Since a closed convex set is weakly closed (Mazur's Theorem), we have $\bar{x} \in C$ and $d(C; x) = \|x - \bar{x}\|$. We show such $\bar{x}$ is unique. Suppose that $z \in C$ also has the property that $d(C; x) = \|x - z\|$. Then for any $t \in [0, 1]$ we have $t\bar{x} + (1 - t)z \in C$. It follows that

$$\begin{aligned} d(C; x) &\leq \|x - (t\bar{x} + (1 - t)z)\| = \|t(x - \bar{x}) + (1 - t)(x - z)\| \\ &\leq t\|x - \bar{x}\| + (1 - t)\|x - z\| = d(C; x). \end{aligned}$$

That is to say

$$t \to \|x - z - t(\bar{x} - z)\|^2 = \|x - z\|^2 - 2t\langle x - z, \bar{x} - z\rangle + t^2\|\bar{x} - z\|^2$$

is a constant mapping, which implies $\bar{x} = z$. ●

Projections

The nearest point can be characterized by the normal cone as follows.

Theorem 4.5.2 (Normal Cone Characterization of Nearest Point)

Let X be a Hilbert space and let C be a closed convex subset of X. Then for any $x \in X$, $\bar{x} \in C$ is a nearest point to x if and only if

$$x - \bar{x} \in N(C; \bar{x}).$$

Proof. Noting that the convex function $f(y) = \|y - x\|^2/2$ attains a minimum at $\bar{x}$ over set C, this directly follows from the Pshenichnii–Rockafellar condition (Theorem 4.3.6):

$$0 \in \partial f(\overline{x}) + N(C; \overline{x}).$$

Projections

The nearest point can be characterized by the normal cone as follows.

Theorem 4.5.2 (Normal Cone Characterization of Nearest Point)

Let X be a Hilbert space and let C be a closed convex subset of X. Then for any $x \in X$, $\bar{x} \in C$ is a nearest point to x if and only if

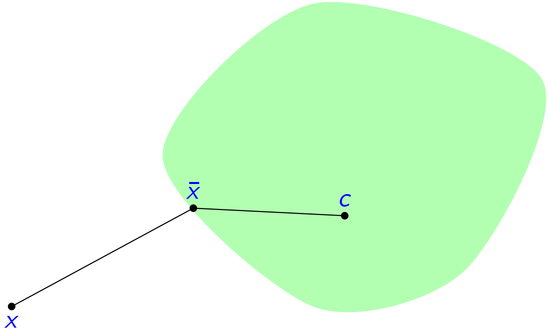
$$x - \bar{x} \in N(C; \bar{x}).$$

Proof. Noting that the convex function $f(y) = \|y - x\|^2/2$ attains a minimum at $\bar{x}$ over set C, this directly follows from the Pshenichnii–Rockafellar condition (Theorem 4.3.6):

$$0 \in \partial f(\overline{x}) + N(C; \overline{x}).$$

Projections

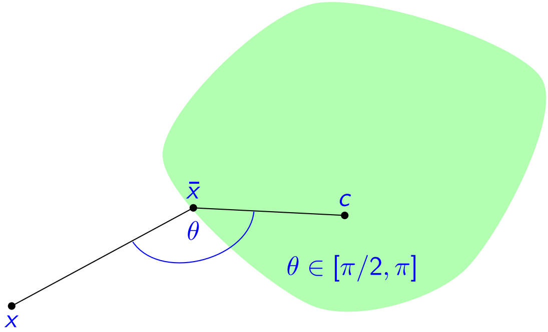
Geometrically, the normal cone characterization is:



$$x - \bar{x} \in N(C; \bar{x}) \iff \langle x - \bar{x}, c - \bar{x} \rangle \leq 0 \text{ for all } c \in C.$$

Projections

Geometrically, the normal cone characterization is:



$$x - \bar{x} \in N(C; \bar{x}) \iff \langle x - \bar{x}, c - \bar{x} \rangle \leq 0 \text{ for all } c \in C.$$

Projections

Definition 4.5.3 (Projection)

Let X be a Hilbert space and let C be a closed convex subset of X. For any $x \in X$ the unique nearest point $y \in C$ is called the projection of x on C and we define the projection mapping P_C by $P_C x = y$.

We summarize some useful properties of the projection mapping in the next proposition whose elementary proof is left as an exercise.

Proposition 4.5.4 (Properties of Projection)

Let X be a Hilbert space and let C be a closed convex subset of X. Then the projection mapping P_C has the following properties.

(i) for any $x \in C$, $P_C x = x$;

(ii) $P_C^2 = P_C$;

(iii) for any $x, y \in X$, $\|P_C y - P_C x\| \leq \|y - x\|$.

Projections

Definition 4.5.3 (Projection)

Let X be a Hilbert space and let C be a closed convex subset of X. For any $x \in X$ the unique nearest point $y \in C$ is called the projection of x on C and we define the projection mapping P_C by $P_C x = y$.

We summarize some useful properties of the projection mapping in the next proposition whose elementary proof is left as an exercise.

Proposition 4.5.4 (Properties of Projection)

Let X be a Hilbert space and let C be a closed convex subset of X. Then the projection mapping P_C has the following properties.

(i) for any $x \in C$, $P_C x = x$;

(ii) $P_C^2 = P_C$;

(iii) for any $x, y \in X$, $\|P_C y - P_C x\| \leq \|y - x\|$.

Projections

Theorem 4.5.5 (Potential Function of Projection)

Let X be a Hilbert space and let C be a closed convex subset of X. Define

$$f(x) = \sup\left\{ \langle x, y\rangle - \frac{\|y\|^2}{2} \;\middle|\; y \in C \right\}.$$

Then f is convex, $P_C(x) = f'(x)$, and therefore P_C is a monotone operator.

Proof. It is easy to check that f is convex and

$$f(x) = \frac{1}{2}(\|x\|^2 - \|x - P_C(x)\|^2).$$

We need only show $P_C(x) = f'(x)$.

Projections

Theorem 4.5.5 (Potential Function of Projection)

Let X be a Hilbert space and let C be a closed convex subset of X. Define

$$f(x) = \sup\left\{\langle x, y\rangle - \frac{\|y\|^2}{2} \;\middle|\; y \in C\right\}.$$

Then f is convex, $P_C(x) = f'(x)$, and therefore P_C is a monotone operator.

Proof. It is easy to check that f is convex and

$$f(x) = \frac{1}{2}(\|x\|^2 - \|x - P_C(x)\|^2).$$

We need only show $P_C(x) = f'(x)$.

Fix $x \in X$. For any $y \in X$ we have

$$\|(x+y) - P_C(x+y)\| \leq \|(x+y) - P_C(x)\|,$$

so

$$\begin{aligned} \|(x+y) - P_C(x+y)\|^2 &\leq \|x+y\|^2 - 2\langle x+y, P_C(x)\rangle + \|P_C(x)\|^2 \\ &= \|x+y\|^2 + \|x - P_C(x)\|^2 - \|x\|^2 \\ &- 2\langle y, P_C(x)\rangle, \end{aligned}$$

hence $f(x+y) - f(x) - \langle P_C(x), y\rangle \geq 0$. On the other hand, since $\|x - P_C(x)\| \leq \|x - P_C(x+y)\|$ we get

$$\begin{aligned} f(x+y) - f(x) - \langle P_C(x), y\rangle &\leq \langle y, P_C(x+y) - P_C(x)\rangle \\ &\leq \|y\| \times \|P_C(x+y) - P_C(x)\| \\ &\leq \|y\|^2, \end{aligned}$$

which implies $P_C(x) = f'(x)$.

Projection Algorithms as Minimization Problems

We start with the simple case of the intersection of two convex sets. Let X be a Hilbert space and let C and D be two closed convex subsets of X. Suppose that $C \cap D \neq \emptyset$. Define a function

$$f(c,d) := \frac{1}{2}\|c-d\|^2 + \iota_C(c) + \iota_D(d).$$

We see that f attains a minimum at $(\bar{c},\bar{d})$ if and only if $\bar{c} = \bar{d} \in C \cap D$. Thus, the problem of finding a point in $C \cap D$ becomes one of minimizing function f.

We consider a natural descending process for f by alternately minimizing f with respect to its two variables. More concretely, start with any $x_0 \in D$. Let x_1 be the solution of minimizing

$$x \to f(x, x_0).$$

Projection Algorithms as Minimization Problems

We start with the simple case of the intersection of two convex sets. Let X be a Hilbert space and let C and D be two closed convex subsets of X. Suppose that $C \cap D \neq \emptyset$. Define a function

$$f(c, d) := \frac{1}{2}\|c - d\|^2 + \iota_C(c) + \iota_D(d).$$

We see that f attains a minimum at $(\bar{c}, \bar{d})$ if and only if $\bar{c} = \bar{d} \in C \cap D$. Thus, the problem of finding a point in $C \cap D$ becomes one of minimizing function f.

We consider a natural descending process for f by alternately minimizing f with respect to its two variables. More concretely, start with any $x_0 \in D$. Let x_1 be the solution of minimizing

$$x \to f(x, x_0).$$

Projection Algorithms as Minimization Problems

We start with the simple case of the intersection of two convex sets. Let X be a Hilbert space and let C and D be two closed convex subsets of X. Suppose that $C \cap D \neq \emptyset$. Define a function

$$f(c,d) := \frac{1}{2}\|c-d\|^2 + \iota_C(c) + \iota_D(d).$$

We see that f attains a minimum at $(\bar{c}, \bar{d})$ if and only if $\bar{c} = \bar{d} \in C \cap D$. Thus, the problem of finding a point in $C \cap D$ becomes one of minimizing function f.

We consider a natural descending process for f by alternately minimizing f with respect to its two variables. More concretely, start with any $x_0 \in D$. Let x_1 be the solution of minimizing

$$x \to f(x, x_0).$$

Projection Algorithms as Minimization Problems

It follows from Theorem 4.5.2. that

$$x_0 - x_1 \in N(C; x_1).$$

That is to say $x_1 = P_C x_0$. We then let x_2 be the solution of minimizing

$$x \to f(x_1, x).$$

Similarly, $x_2 = P_D x_1$. In general, we define

$$x_{i+1} = \begin{cases} P_C x_i & i \text{ is even}, \\ P_D x_i & i \text{ is odd}. \end{cases} \tag{1}$$

This algorithm is a generalization of the classical von Neumann projection algorithm for finding points in the intersection of two subspaces.
We will show that in general x_i weakly converge to a point in $C \cap D$ and when $\text{int}(C \cap D) \neq \emptyset$ we have norm convergence.

Attracting Mappings and Fejér Sequences

We discuss two useful tools for proving the convergence.

Definition 4.5.6 (Nonexpansive Mapping)

Let X be a Hilbert space, let C be a closed convex nonempty subset of X and let $T\colon C \to X$. We say that T is nonexpansive provided that $\|Tx - Ty\| \leq \|x - y\|$.

Definition 4.5.7 (Attracting Mapping)

Let X be a Hilbert space, let C be a closed convex nonempty subset of X and let $T\colon C \to C$ be a nonexpansive mapping. Suppose that D is a closed nonempty subset of C. We say that T is attracting with respect to D if for every $x \in C \setminus D$ and $y \in D$,

$$\|Tx - y\| \leq \|x - y\|.$$

We say that T is k-attracting with respect to D if for every $x \in C \setminus D$ and $y \in D$,

$$k\|x - Tx\|^2 \leq \|x - y\|^2 - \|Tx - y\|^2.$$

Attracting Mappings and Fejér Sequences

We discuss two useful tools for proving the convergence.

Definition 4.5.6 (Nonexpansive Mapping)

Let X be a Hilbert space, let C be a closed convex nonempty subset of X and let $T\colon C \to X$. We say that T is nonexpansive provided that $\|Tx - Ty\| \leq \|x - y\|$.

Definition 4.5.7 (Attracting Mapping)

Let X be a Hilbert space, let C be a closed convex nonempty subset of X and let $T\colon C \to C$ be a nonexpansive mapping. Suppose that D is a closed nonempty subset of C. We say that T is attracting with respect to D if for every $x \in C \backslash D$ and $y \in D$,

$$\|Tx - y\| \leq \|x - y\|.$$

We say that T is k-attracting with respect to D if for every $x \in C \backslash D$ and $y \in D$,

$$k\|x - Tx\|^2 \leq \|x - y\|^2 - \|Tx - y\|^2.$$

Attracting Mappings and Fejér Sequences

Lemma 4.5.8 (Attractive Property of Projection)

Let X be a Hilbert space and let C be a convex closed subset of X. Then $P_C: X \to X$ is 1-attracting with respect to C.

Proof. Let $y \in C$. We have

$$\begin{aligned} \|x-y\|^2 - \|P_Cx - y\|^2 &= \langle x - P_Cx, x + P_Cx - 2y\rangle \\ &= \langle x - P_Cx, x - P_Cx + 2(P_Cx - y)\rangle \\ &= \|x - P_Cx\|^2 + 2\langle x - P_Cx, P_Cx - y\rangle \\ &\geq \|x - P_Cx\|^2. \end{aligned}$$

Note that if T is attracting (k-attracting) with respect to a set D, then it is attracting (k-attracting) with respect to any subset of D.

Attracting Mappings and Fejér Sequences

Lemma 4.5.8 (Attractive Property of Projection)

Let X be a Hilbert space and let C be a convex closed subset of X. Then $P_C\colon X \to X$ is 1-attracting with respect to C.

Proof. Let $y \in C$. We have

$$\begin{aligned}
\|x-y\|^2 - \|P_Cx - y\|^2 &= \langle x - P_Cx, x + P_Cx - 2y\rangle \\
&= \langle x - P_Cx, x - P_Cx + 2(P_Cx - y)\rangle \\
&= \|x - P_Cx\|^2 + 2\langle x - P_Cx, P_Cx - y\rangle \\
&\geq \|x - P_Cx\|^2.
\end{aligned}$$

Note that if T is attracting (k-attracting) with respect to a set D, then it is attracting (k-attracting) with respect to any subset of D.

Attracting Mappings and Fejér Sequences

Lemma 4.5.8 (Attractive Property of Projection)

Let X be a Hilbert space and let C be a convex closed subset of X. Then $P_C\colon X \to X$ is 1-attracting with respect to C.

Proof. Let $y \in C$. We have

$$\begin{aligned} \|x-y\|^2 - \|P_C x - y\|^2 &= \langle x - P_C x, x + P_C x - 2y\rangle \\ &= \langle x - P_C x, x - P_C x + 2(P_C x - y)\rangle \\ &= \|x - P_C x\|^2 + 2\langle x - P_C x, P_C x - y\rangle \\ &\geq \|x - P_C x\|^2. \end{aligned}$$

∎

Note that if T is attracting (k-attracting) with respect to a set D, then it is attracting (k-attracting) with respect to any subset of D.

Attracting Mappings and Fejér Sequences

Definition 4.5.9 (Fejér Monotone Sequence)

Let X be a Hilbert space, let C be closed convex set and let (x_i) be a sequence in X. We say that (x_i) is Fejér monotone with respect to C if

$$\|x_{i+1} - c\| \leq \|x_i - c\|, \quad \text{for all } c \in C \text{ and } i = 1, 2, \ldots$$

Next we summarize properties of Fejér monotone sequences.

Theorem 4.5.10 (Properties of Fejér Monotone Sequences)

Let X be a Hilbert space, let C be a closed convex set and let (x_i) be a Fejér monotone sequence with respect to C. Then

(i) (x_i) is bounded and $d(C; x_{i+1}) \leq d(C; x_i)$.

(ii) (x_i) has at most one weak cluster point in C.

(iii) If the interior of C is nonempty then (x_i) converges in norm.

(iv) $(P_C x_i)$ converges in norm.

Attracting Mappings and Fejér Sequences

Definition 4.5.9 (Fejér Monotone Sequence)

Let X be a Hilbert space, let C be closed convex set and let (x_i) be a sequence in X. We say that (x_i) is Fejér monotone with respect to C if

$$\|x_{i+1} - c\| \leq \|x_i - c\|, \quad \text{for all } c \in C \text{ and } i = 1, 2, \ldots$$

Next we summarize properties of Fejér monotone sequences.

Theorem 4.5.10 (Properties of Fejér Monotone Sequences)

Let X be a Hilbert space, let C be a closed convex set and let (x_i) be a Fejér monotone sequence with respect to C. Then

(i) (x_i) is bounded and $d(C; x_{i+1}) \leq d(C; x_i)$.

(ii) (x_i) has at most one weak cluster point in C.

(iii) If the interior of C is nonempty then (x_i) converges in norm.

(iv) $(P_C x_i)$ converges in norm.

Proof. (i) is obvious.
Observe that, for any $c \in C$ the sequence $(\|x_i - c\|^2)$ converges and so does

$$(\|x_i\|^2 - 2\langle x_i, c\rangle). \tag{2}$$

Now suppose $c_1, c_2 \in C$ are two weak cluster points of (x_i). Letting c in (2) be c_1 and c_2, respectively, and taking limits of the difference, yields $\langle c_1, c_1 - c_2\rangle = \langle c_2, c_1 - c_2\rangle$ so that $c_1 = c_2$, which proves (ii).
To prove (iii) suppose that $B_r(c) \subset C$. For any $x_{i+1} \neq x_i$, simplifying

$$\|x_{i+1} - \left(c - h\frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}\right)\|^2 \leq \|x_i - \left(c - h\frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}\right)\|^2$$

we have

$$2h\|x_{i+1} - x_i\| \leq \|x_i - c\|^2 - \|x_{i+1} - c\|^2.$$

Proof. (i) is obvious.
Observe that, for any $c \in C$ the sequence $(\|x_i - c\|^2)$ converges and so does

$$(\|x_i\|^2 - 2\langle x_i, c\rangle). \tag{2}$$

Now suppose $c_1, c_2 \in C$ are two weak cluster points of (x_i). Letting c in (2) be c_1 and c_2, respectively, and taking limits of the difference, yields $\langle c_1, c_1 - c_2\rangle = \langle c_2, c_1 - c_2\rangle$ so that $c_1 = c_2$, which proves (ii).
To prove (iii) suppose that $B_r(c) \subset C$. For any $x_{i+1} \neq x_i$, simplifying

$$\|x_{i+1} - \left(c - h\frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}\right)\|^2 \leq \|x_i - \left(c - h\frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}\right)\|^2$$

we have

$$2h\|x_{i+1} - x_i\| \leq \|x_i - c\|^2 - \|x_{i+1} - c\|^2.$$

Proof. (i) is obvious.
Observe that, for any $c \in C$ the sequence $(\|x_i - c\|^2)$ converges and so does

$$(\|x_i\|^2 - 2\langle x_i, c\rangle). \tag{2}$$

Now suppose $c_1, c_2 \in C$ are two weak cluster points of (x_i). Letting c in (2) be c_1 and c_2, respectively, and taking limits of the difference, yields $\langle c_1, c_1 - c_2\rangle = \langle c_2, c_1 - c_2\rangle$ so that $c_1 = c_2$, which proves (ii).
To prove (iii) suppose that $B_r(c) \subset C$. For any $x_{i+1} \neq x_i$, simplifying

$$\|x_{i+1} - \big(c - h\frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}\big)\|^2 \leq \|x_i - \big(c - h\frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|}\big)\|^2$$

we have

$$2h\|x_{i+1} - x_i\| \leq \|x_i - c\|^2 - \|x_{i+1} - c\|^2.$$

For any $j > i$, adding the above inequality from i to $j-1$ yields

$$2h\|x_j - x_i\| \leq \|x_i - c\|^2 - \|x_j - c\|^2.$$

Since $(\|x_i - c\|^2)$ is a convergent sequence we conclude that (x_i) is a Cauchy sequence.

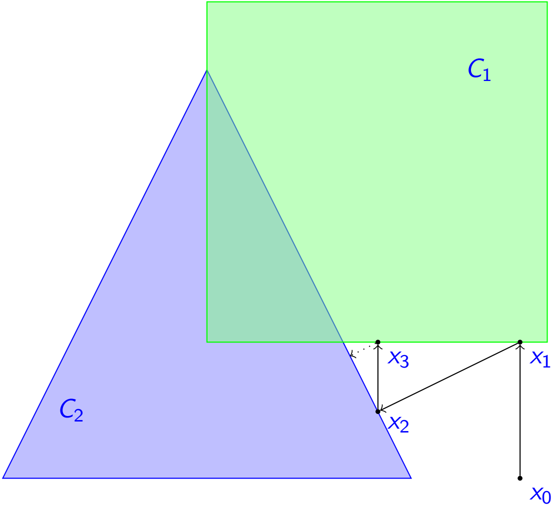
Finally, for natural numbers i, j with $j > i$, apply the parallelogram law $\|a-b\|^2 = 2\|a\|^2 + 2\|b\|^2 - \|a+b\|^2$ to $a := P_C x_j - x_j$ and $b := P_C x_i - x_j$ we obtain

$$\begin{aligned}
\|P_C x_j - P_C x_i\|^2 &= 2\|P_C x_j - x_j\|^2 + 2\|P_C x_i - x_j\|^2 \\
&\qquad - 4\left\|\frac{P_C x_j + P_C x_i}{2} - x_j\right\|^2 \\
&\leq 2\|P_C x_j - x_j\|^2 + 2\|P_C x_i - x_j\|^2 \\
&\quad - 4\|P_C x_j - x_j\|^2 \\
&\leq 2\|P_C x_i - x_j\|^2 - 2\|P_C x_j - x_j\|^2 \\
&\leq 2\|P_C x_i - x_i\|^2 - 2\|P_C x_j - x_j\|^2.
\end{aligned}$$

We identify $(P_C x_i)$ as a Cauchy sequence, because $(\|x_i - P_C x_i\|)$ converges by (i). ●

Attracting Mappings and Fejér Sequences

The following example shows the first few terms of a sequence $\{x_n\}$ which is Fejér monotone with respect to $C = C_1 \cap C_2$.



Convergence of Projection Algorithms

Let X be a Hilbert space. We say a sequence (x_i) in X is *asymptotically regular* if

$$\lim_{i\to\infty} \|x_i - x_{i+1}\| = 0.$$

Lemma 4.5.11 (Asymptotical Regularity of Projection Algorithm)

Let X be a Hilbert space and let C and D be closed convex subsets of X. Suppose $C \cap D \neq \emptyset$. Then the sequence (x_i) defined by the projection algorithm

$$x_{i+1} = \begin{cases} P_C x_i & i \text{ is even,} \\ P_D x_i & i \text{ is odd.} \end{cases}$$

is asymptotically regular.

Convergence of Projection Algorithms

Let X be a Hilbert space. We say a sequence (x_i) in X is *asymptotically regular* if

$$\lim_{i\to\infty} \|x_i - x_{i+1}\| = 0.$$

Lemma 4.5.11 (Asymptotical Regularity of Projection Algorithm)

Let X be a Hilbert space and let C and D be closed convex subsets of X. Suppose $C \cap D \neq \emptyset$. Then the sequence (x_i) defined by the projection algorithm

$$x_{i+1} = \begin{cases} P_C x_i & i \text{ is even,} \\ P_D x_i & i \text{ is odd.} \end{cases}$$

is asymptotically regular.

Proof. By Lemma 4.5.8 both P_C and P_D are 1-attracting with respect to $C \cap D$. Let $y \in C \cap D$. Since x_{i+1} is either $P_C x_i$ or $P_D x_i$ it follows that

$$\|x_{i+1} - x_i\|^2 \leq \|x_i - y\|^2 - \|x_{i+1} - y\|^2.$$

Since $(\|x_i - y\|^2)$ is a monotone decreasing sequence, therefore the right-hand side of the inequality converges to 0 and the result follows. ●

Convergence of Projection Algorithms

Now, we are ready to prove the convergence of the projection algorithm.

Theorem 4.5.12 (Convergence for Two Sets)

Let X be a Hilbert space and let C and D be closed convex subsets of X. Suppose $C \cap D \neq \emptyset$ ($\mathrm{int}(C \cap D) \neq \emptyset$). Then the projection algorithm

$$x_{i+1} = \begin{cases} P_C x_i & i \text{ is even,} \\ P_D x_i & i \text{ is odd.} \end{cases}$$

converges weakly (in norm) to a point in $C \cap D$.

Proof. Let $y \in C \cap D$. Then, for any $x \in X$, we have

$$\|P_C x - y\| = \|P_C x - P_C y\| \leq \|x - y\|, \quad \text{and}$$
$$\|P_D x - y\| = \|P_D x - P_D y\| \leq \|x - y\|.$$

Since x_{i+1} is either $P_C x_i$ or $P_D x_i$ we have that

$$\|x_{i+1} - y\| \leq \|x_i - y\|.$$

That is to say (x_i) is a Fejér monotone sequence with respect to $C \cap D$. By item (i) of Theorem 4.5.10 the sequence (x_i) is bounded, and therefore has a weakly convergent subsequence. We show that all weak cluster points of (x_i) belong to $C \cap D$. In fact, let (x_{i_k}) be a subsequence of (x_i) converging to x weakly.

Taking a subsequence again if necessary we may assume that (x_{i_k}) is a subset of either C or D. For the sake of argument let us assume that it is a subset of C and, thus, the weak limit x is also in C. On the other hand by the asymptotical regularity of (x_i) in Lemma 4.5.11 $(P_D x_{i_k}) = (x_{i_k+1})$ also weakly converges to x. Since $(P_D x_{i_k})$ is a subset of D we conclude that $x \in D$, and therefore $x \in C \cap D$. By item (ii) of Theorem 4.5.10 (x_i) has at most one weak cluster point in $C \cap D$, and we conclude that (x_i) weakly converges to a point in $C \cap D$. When $\text{int}(C \cap D) \neq \emptyset$ it follows from item (iii) of Theorem 4.5.10 that (x_i) converges in norm. ●

Convergence of Projection Algorithms

Whether the alternating projection algorithm converged in norm without the assumption that

$$\mathrm{int}(C \cap D) \neq \emptyset,$$

or more generally of metric regularity, was a long-standing open problem.

Recently Hundal constructed an example showing that the answer is negative [5].

The proof of Hundal's example is self-contained and elementary. However, it is quite long and delicate, therefore we will be satisfied in stating the example.

Convergence of Projection Algorithms

Whether the alternating projection algorithm converged in norm without the assumption that

$$\mathrm{int}(C \cap D) \neq \emptyset,$$

or more generally of metric regularity, was a long-standing open problem.

Recently Hundal constructed an example showing that the answer is negative [5].

The proof of Hundal's example is self-contained and elementary. However, it is quite long and delicate, therefore we will be satisfied in stating the example.

Convergence of Projection Algorithms

Example 4.5.13 (Hundal)

Let $X = \ell_2$ and let $\{e_i \mid i = 1, 2, \dots\}$ be the standard basis of X. Define $v\colon [0, +\infty) \to X$ by

$$v(r) := \exp(-100r^3)e_1 + \cos\left((r - [r])\pi/2\right)e_{[r]+2} + \sin\left((r - [r])\pi/2\right)e_{[r]+3},$$

where $[r]$ signifies the integer part of r and further define

$$C = \{e_1\}^\perp \text{ and } D = \operatorname{conv}\{v(r) \mid r \geq 0\}.$$

Then the hyperplane C and cone D satisfies $C \cap D = \{0\}$.
However, Hundal's sequence of alternating projections x_i given by

$$x_{i+1} = P_D P_C x_i$$

starting from $x_0 = v(1)$ (necessarily) converges weakly to 0, but not in norm.

Convergence of Projection Algorithms

A related useful example is the moment problem.

Example 4.5.14 (Moment Problem)

Let X be a Hilbert lattice[1] with lattice cone $D = X^+$. Consider a linear continuous mapping A from X onto $\mathbb{R}^N$. The moment problem seeks the solution of $A(x) = y \in \mathbb{R}^N, x \in D$.

Define $C = A^{-1}(y)$. Then the moment problem is feasible iff

$$C \cap D \neq \emptyset.$$

A natural question is whether the projection algorithm converges in norm.

This problem is answered affirmatively in [1] for $N = 1$ yet remains open in general when $N > 1$.

[1] All Hilbert lattices are realized as $L_2(\Omega, \mu)$ in the natural ordering for some measure space.

Projection Algorithms for Multiple Sets

We now turn to the general problem of finding some points in

$$\bigcap_{n=1}^{N} C_n,$$

where C_n, $n = 1, \ldots, N$ are closed convex sets in a Hilbert space X.

Let a_n, $n = 1, \ldots, N$ be positive numbers. Denote

$$X^N := \{x = (x_1, x_2, \ldots, x_N) \mid x_n \in X, n = 1, \ldots, N\}$$

the product space of N copies of X with inner product

$$\langle x, y \rangle = \sum_{n=1}^{N} a_n \langle x_n, y_n \rangle.$$

Then X^N is a Hilbert space.

Projection Algorithms for Multiple Sets

Define

$$C := C_1 \times C_2 \times \cdots \times C_N, \quad \text{and}$$
$$D := \{(x_1, \dots, x_N) \in X^N : x_1 = x_2 = \cdots = x_N\}.$$

Then C and D are closed convex sets in X^N and

$$x \in \bigcap_{n=1}^{N} C_n \iff (x, x, \dots, x) \in C \cap D.$$

Applying the projection algorithm (1) to the convex sets C and D defined above we have the following generalized projection algorithm for finding some points in

$$\bigcap_{n=1}^{N} C_n,$$

as we shall now explain.

Projection Algorithms for Multiple Sets

Define

$$C := C_1 \times C_2 \times \cdots \times C_N, \quad \text{and}$$
$$D := \{(x_1, \ldots, x_N) \in X^N : x_1 = x_2 = \cdots = x_N\}.$$

Then C and D are closed convex sets in X^N and

$$x \in \bigcap_{n=1}^{N} C_n \iff (x, x, \ldots, x) \in C \cap D.$$

Applying the projection algorithm (1) to the convex sets C and D defined above we have the following generalized projection algorithm for finding some points in

$$\bigcap_{n=1}^{N} C_n,$$

as we shall now explain.

Projection Algorithms for Multiple Sets

Denote $P_n = P_{C_n}$. The algorithm can be expressed by

$$x_{i+1} = \left(\sum_{n=1}^{N} \lambda_n P_n \right) x_i, \tag{3}$$

where $\lambda_n = a_n / \sum_{m=1}^{N} a_m$.

In other words, each new approximation is the convex combination of the projections of the previous step to all the sets $C_n, n = 1, \ldots, N$. It follows from the convergence theorem in the previous subsection that the algorithm (3) converges weakly to some point in $\bigcap_{n=1}^{N} C_n$ when this intersection is nonempty.

Projection Algorithms for Multiple Sets

Denote $P_n = P_{C_n}$. The algorithm can be expressed by

$$x_{i+1} = \Big(\sum_{n=1}^{N} \lambda_n P_n\Big) x_i, \tag{3}$$

where $\lambda_n = a_n / \sum_{m=1}^{N} a_m$.

In other words, each new approximation is the convex combination of the projections of the previous step to all the sets $C_n, n = 1, \dots, N$. It follows from the convergence theorem in the previous subsection that the algorithm (3) converges weakly to some point in $\bigcap_{n=1}^{N} C_n$ when this intersection is nonempty.

Projection Algorithms for Multiple Sets

Theorem 4.5.15 (Weak Convergence for N Sets)

Let X be a Hilbert space and let $C_n, n = 1, \ldots, N$ be closed convex subsets of X. Suppose that $\bigcap_{n=1}^N C_n \neq \emptyset$ and $\lambda_n \geq 0$ satisfies $\sum_{n=1}^N \lambda_n = 1$. Then the projection algorithm

$$x_{i+1} = \left(\sum_{n=1}^N \lambda_n P_n\right) x_i,$$

converges weakly to a point in $\bigcap_{n=1}^N C_n$.

Proof. This follows directly from Theorem 4.5.12.

Projection Algorithms for Multiple Sets

When the interior of $\bigcap_{n=1}^{N} C_n$ is nonempty we also have that the algorithm (3) converges in norm. However, since D does not have interior this conclusion cannot be derived from Theorem 4.5.12. Rather it has to be proved by directly showing that the approximation sequence is Fejér monotone w.r.t. $\bigcap_{n=1}^{N} C_n$.

Theorem 4.5.16 (Strong Convergence for N Sets)

Let X be a Hilbert space and let $C_n, n = 1, \ldots, N$ be closed convex subsets of X. Suppose that $\operatorname{int} \bigcap_{n=1}^{N} C_n \neq \emptyset$ and $\lambda_n \geq 0$ satisfies $\sum_{n=1}^{N} \lambda_n = 1$. Then the projection algorithm

$$x_{i+1} = \Big(\sum_{n=1}^{N} \lambda_n P_n \Big) x_i,$$

converges to a point in $\bigcap_{n=1}^{N} C_n$ in norm.

Projection Algorithms for Multiple Sets

When the interior of $\bigcap_{n=1}^{N} C_n$ is nonempty we also have that the algorithm (3) converges in norm. However, since D does not have interior this conclusion cannot be derived from Theorem 4.5.12. Rather it has to be proved by directly showing that the approximation sequence is Fejér monotone w.r.t. $\bigcap_{n=1}^{N} C_n$.

Theorem 4.5.16 (Strong Convergence for N Sets)

Let X be a Hilbert space and let $C_n, n = 1, \ldots, N$ be closed convex subsets of X. Suppose that $\operatorname{int} \bigcap_{n=1}^{N} C_n \neq \emptyset$ and $\lambda_n \geq 0$ satisfies $\sum_{n=1}^{N} \lambda_n = 1$. Then the projection algorithm

$$x_{i+1} = \left(\sum_{n=1}^{N} \lambda_n P_n \right) x_i,$$

converges to a point in $\bigcap_{n=1}^{N} C_n$ in norm.

Proof. Let $y \in \bigcap_{n=1}^{N} C_n$. Then

$$\begin{aligned}
\|x_{i+1} - y\| &= \left\| \left(\sum_{n=1}^{N} \lambda_n P_n \right) x_i - y \right\| = \left\| \sum_{n=1}^{N} \lambda_n (P_n x_i - P_n y) \right\| \\
&\leq \sum_{n=1}^{N} \lambda_n \|P_n x_i - P_n y\| \leq \sum_{n=1}^{N} \lambda_n \|x_i - y\| = \|x_i - y\|.
\end{aligned}$$

That is to say (x_i) is a Fejér monotone sequence with respect to $\bigcap_{n=1}^{N} C_n$. The norm convergence of (x_i) then follows directly from Theorems 4.5.10 and 4.5.15. ●

Commentary and Open Questions

- We have proven convergence of the projection algorithm. It can be traced to von Neumann, Weiner and before, and has been studied extensively.
- We emphasize the relationship between the projection algorithm and variational methods in Hilbert spaces:
  - While projection operators can be defined outside of the setting of Hilbert space, they are not necessarily non-expansive.
  - In fact, non-expansivity of the projection operator characterizes Hilbert space in two more dimensions.
- The Hundal example clarifies many other related problems regarding convergence. Simplifications of the example have since been published.
  - What happens if we only allow “nice” cones?
- Bregman distance provides an alternative perspective into many generalizations of the projection algorithm.

Exercises

1. Let $T : \mathcal{H} \to \mathcal{H}$ be nonexpansive and let $\alpha \in [-,1,1]$. Show that $(I + \alpha T)$ is a *maximally monotone* continuous operator.
2. (Common projections) Prove formula for the projection onto each of the following sets:
   1. Half-space: $H := \{x \in \mathcal{H} : \langle a, x\rangle = b\}$, $0 \neq a \in \mathcal{H}$, $b \in \mathbb{R}$.
   2. Line: $L := x + \mathbb{R}y$ where $x, y \in \mathcal{H}$.
   3. Ball: $B := \{x \in \mathcal{H} : \|x\| \leq r\}$ where $r > 0$.
   4. Ellipse in $\mathbb{R}^2$: $E := \{(x, y) \in \mathbb{R}^2 : x^2/a^2 + y^2/b^2 = 1\}$.
      Hint: $P_E(u, v) = \left(\frac{a^2 u}{a^2 - t}, \frac{b^2 v}{b^2 - t}\right)$ where t solves
      $$\frac{a^2u^2}{(a^2 - t)^2} + \frac{b^2v^2}{(b^2 - t)^2} = 1.$$
3. (Non-existence of best approximations) Let $\{e_n\}_{n\in\mathbb{N}}$ be an orthonormal basis of an infinite dimensional Hilbert space. Define the set $A := \{e_1/n + e_n : n \in \mathbb{N}\}$. Show that A is norm closed and bounded but $d_A(0) = 1$ is not attained. Is A weakly closed?

References

- H.H. Bauschke & and J.M. Borwein. **On the convergence of von Neumann's alternating projection algorithm for two sets**. *Set-Valued Analysis* 1(2):185–212 (1993).
- J.M. Borwein & Q.J. Zhu. **Techniques of Variational Analysis**. Springer (2013).
- Y. Censor & A. Cegielski. **Projection methods: an annotated bibliography of books and reviews**. *Optimization*, accepted for publication. Preprint PDF.
- R. Escalante & M. Raydan, **Alternating projection methods**. SIAM Fundamentals of Algorithms (2011)
- H.S. Hundal. **An alternating projection that does not converge in norm**. *Nonlinear Analysis: Theory, Methods & Applications* 57(1):35–61 (2004).

Many resources (and definitions) available at:

http://www.carma.newcastle.edu.au/jon/ToVA/

3. Convex Douglas–Rachford

Feasibility Problem

Given closed sets $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ the feasibility problem asks

$$\text{find } x \in \bigcap_{j=1}^{N} C_j.$$

Many problems can be cast is this form. Three examples:

1. Linear systems "$Ax = b$": $C_j = \{x : \langle a_j, x\rangle = b_j\}$.
2. Phase retrieval: $C_1 = \{f : |\hat{f}| = m \text{ a.e.}\}$ and $C_2 = \{f : f = 0 \text{ on } D\}$.
3. **Matrix completion problems:** more on this later!

Projection algorithms are a popular approach to solving feasibility problems. They work on the following principle:

1. While the intersection might be difficult to deal with directly, the individual constraint sets are sufficiently "simple".
2. "Simple" means we can efficiently compute nearest points.
3. Use an iterative scheme which employs nearest points to individual constraint sets at each stage, and obtain a solution in the limit.

Feasibility Problem

Given closed sets $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ the feasibility problem asks

$$\text{find } x \in \bigcap_{j=1}^{N} C_j.$$

Many problems can be cast is this form. Three examples:

1. Linear systems "$Ax = b$": $C_j = \{x : \langle a_j, x\rangle = b_j\}$.
2. Phase retrieval: $C_1 = \{f : |\hat{f}| = m \text{ a.e.}\}$ and $C_2 = \{f : f = 0 \text{ on } D\}$.
3. **Matrix completion problems:** more on this later!

Projection algorithms are a popular approach to solving feasibility problems. They work on the following principle:

1. While the intersection might be difficult to deal with directly, the individual constraint sets are sufficiently "simple".
2. "Simple" means we can efficiently compute nearest points.
3. Use an iterative scheme which employs nearest points to individual constraint sets at each stage, and obtain a solution in the limit.

Feasibility Problem

Given closed sets $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ the feasibility problem asks

$$\text{find } x \in \bigcap_{j=1}^{N} C_j.$$

Many problems can be cast is this form. Three examples:

1. Linear systems "$Ax = b$": $C_j = \{x : \langle a_j, x \rangle = b_j\}$.
2. Phase retrieval: $C_1 = \{f : |\hat{f}| = m \text{ a.e.}\}$ and $C_2 = \{f : f = 0 \text{ on } D\}$.
3. **Matrix completion problems:** more on this later!

Projection algorithms are a popular approach to solving feasibility problems. They work on the following principle:

1. While the intersection might be difficult to deal with directly, the individual constraint sets are sufficiently "simple".
2. "Simple" means we can efficiently compute nearest points.
3. Use an iterative scheme which employs nearest points to individual constraint sets at each stage, and obtain a solution in the limit.

Douglas, Rachford & Peaceman


Jim Douglas Jnr (1927 –)


Henry Rachford


Donald Peaceman

Algorithmic Building Blocks

Let $S \subseteq \mathcal{H}$ be non-empty. The (nearest point) projection onto S is the (set-valued) mapping,

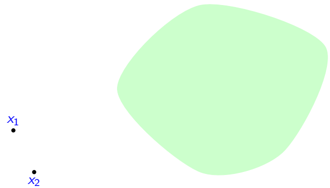
$$P_S x := \left\{ s \in S : \|x - s\| \leq \inf_{s \in S} \|x - s\| \right\}.$$

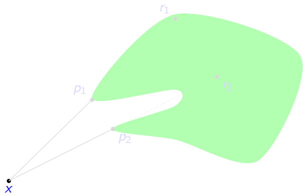
If S is closed and convex then projections exists uniquely with

$$P_S(x) = p \iff \langle x - p, s - p \rangle \leq 0 \text{ for all } s \in S.$$

The reflection w.r.t. S is the (set-valued) mapping,

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





Algorithmic Building Blocks

Let $S \subseteq \mathcal{H}$ be non-empty. The (nearest point) projection onto S is the (set-valued) mapping,

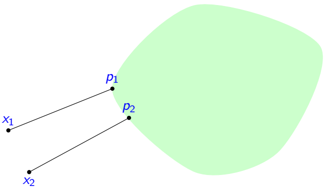
$$P_S x := \left\{ s \in S : \|x - s\| \leq \inf_{s \in S} \|x - s\| \right\}.$$

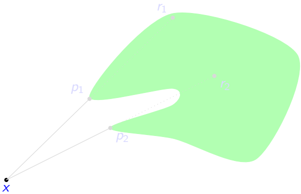
If S is closed and convex then projections exists uniquely with

$$P_S(x) = p \iff \langle x - p, s - p \rangle \leq 0 \text{ for all } s \in S.$$

The reflection w.r.t. S is the (set-valued) mapping,

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





Algorithmic Building Blocks

Let $S \subseteq \mathcal{H}$ be non-empty. The (nearest point) projection onto S is the (set-valued) mapping,

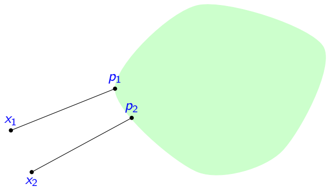
$$P_S x := \left\{ s \in S : \|x - s\| \leq \inf_{s \in S} \|x - s\| \right\}.$$

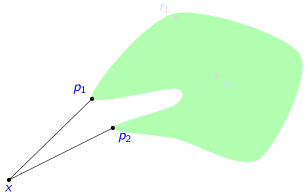
If S is closed and convex then projections exists uniquely with

$$P_S(x) = p \iff \langle x - p, s - p \rangle \leq 0 \text{ for all } s \in S.$$

The reflection w.r.t. S is the (set-valued) mapping,

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





Algorithmic Building Blocks

Let $S \subseteq \mathcal{H}$ be non-empty. The (nearest point) projection onto S is the (set-valued) mapping,

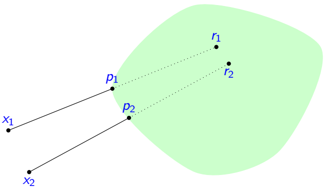
$$P_S x := \left\{ s \in S : \|x - s\| \leq \inf_{s \in S} \|x - s\| \right\}.$$

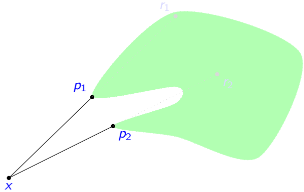
If S is closed and convex then projections exists uniquely with

$$P_S(x) = p \iff \langle x - p, s - p \rangle \leq 0 \text{ for all } s \in S.$$

The reflection w.r.t. S is the (set-valued) mapping,

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





Algorithmic Building Blocks

Let $S \subseteq \mathcal{H}$ be non-empty. The (nearest point) projection onto S is the (set-valued) mapping,

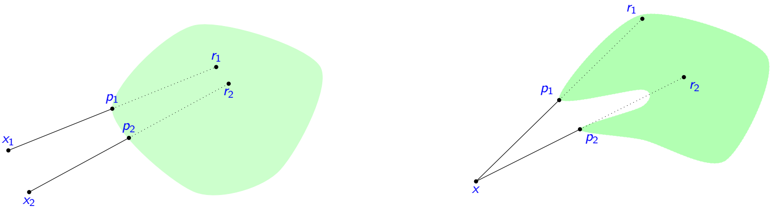
$$P_S x := \left\{ s \in S : \|x - s\| \leq \inf_{s \in S} \|x - s\| \right\}.$$

If S is closed and convex then projections exists uniquely with

$$P_S(x) = p \iff \langle x - p, s - p \rangle \leq 0 \text{ for all } s \in S.$$

The reflection w.r.t. S is the (set-valued) mapping,

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

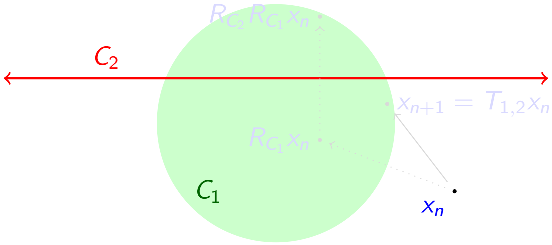


The Douglas–Rachford Algorithm

Given an initial point $x_0 \in \mathcal{H}$, the Douglas–Rachford method is the fixed-point iteration given by

$$x_{n+1} \in T_{C_1,C_2} x_n \quad \text{where} \quad T_{C_1,C_2} := \frac{Id + R_{C_2} R_{C_1}}{2}.$$

We hope that (x_n) converges to a fixed point of of the operator T_{C_1,C_2}.



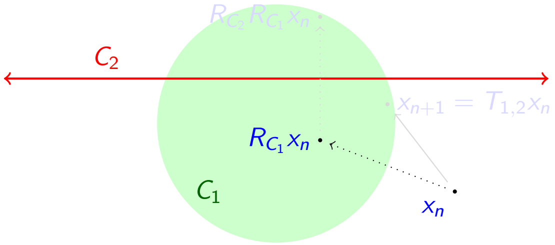
$$C_1 = \{x \in \mathcal{H} : \|x\| \leq 1\}, \quad C_2 = \{x \in \mathcal{H} : \langle a, x \rangle = b\}.$$

The Douglas–Rachford Algorithm

Given an initial point $x_0 \in \mathcal{H}$, the Douglas–Rachford method is the fixed-point iteration given by

$$x_{n+1} \in T_{C_1,C_2} x_n \quad \text{where} \quad T_{C_1,C_2} := \frac{Id + R_{C_2} R_{C_1}}{2}.$$

We hope that (x_n) converges to a fixed point of of the operator T_{C_1,C_2}.



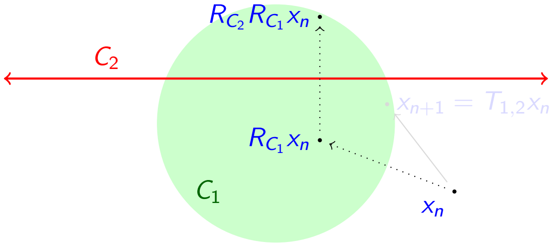
$$C_1 = \{x \in \mathcal{H} : \|x\| \leq 1\}, \quad C_2 = \{x \in \mathcal{H} : \langle a, x \rangle = b\}.$$

The Douglas–Rachford Algorithm

Given an initial point $x_0 \in \mathcal{H}$, the Douglas–Rachford method is the fixed-point iteration given by

$$x_{n+1} \in T_{C_1,C_2} x_n \quad \text{where} \quad T_{C_1,C_2} := \frac{Id + R_{C_2} R_{C_1}}{2}.$$

We hope that (x_n) converges to a fixed point of of the operator T_{C_1,C_2}.



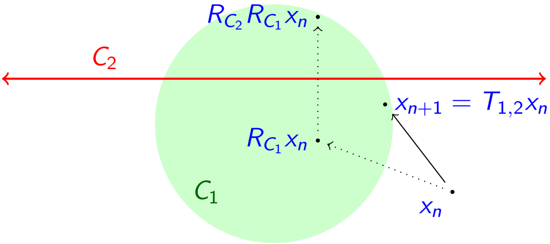
$$C_1 = \{x \in \mathcal{H} : \|x\| \leq 1\}, \quad C_2 = \{x \in \mathcal{H} : \langle a, x \rangle = b\}.$$

The Douglas–Rachford Algorithm

Given an initial point $x_0 \in \mathcal{H}$, the Douglas–Rachford method is the fixed-point iteration given by

$$x_{n+1} \in T_{C_1,C_2} x_n \quad \text{where} \quad T_{C_1,C_2} := \frac{Id + R_{C_2} R_{C_1}}{2}.$$

We hope that (x_n) converges to a fixed point of of the operator T_{C_1,C_2}.



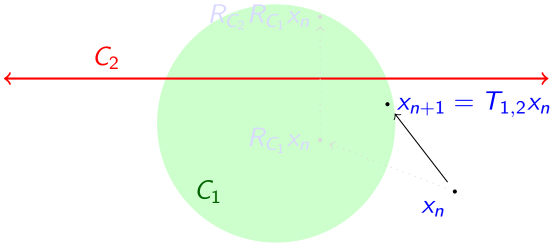
$$C_1 = \{x \in \mathcal{H} : \|x\| \le 1\}, \quad C_2 = \{x \in \mathcal{H} : \langle a, x \rangle = b\}.$$

The Douglas–Rachford Algorithm

Given an initial point $x_0 \in \mathcal{H}$, the Douglas–Rachford method is the fixed-point iteration given by

$$x_{n+1} \in T_{C_1,C_2} x_n \quad \text{where} \quad T_{C_1,C_2} := \frac{Id + R_{C_2} R_{C_1}}{2}.$$

We hope that (x_n) converges to a fixed point of of the operator T_{C_1,C_2}.



$$C_1 = \{x \in \mathcal{H} : \|x\| \leq 1\}, \quad C_2 = \{x \in \mathcal{H} : \langle a, x \rangle = b\}.$$

Douglas–Rachford Fixed Points

Why $\mathrm{Fix}\, T_{C_1,C_2}$**?** Assuming single-valueness of R_{C_1} and R_{C_2} we have:

$$x \in \mathrm{Fix}\, T_{C_1,C_2} \qquad \iff \qquad x = \frac{x + R_{C_2} R_{C_1} x}{2}$$

The same argument for the set-valued case yields:

- If $x \in T_{C_1,C_2} x$ then there is an element of $P_{C_1} x$ contained in $C_1 \cap C_2$.

Douglas–Rachford Fixed Points

Why $\mathrm{Fix}\, T_{C_1,C_2}$**?** Assuming single-valueness of R_{C_1} and R_{C_2} we have:

$$x \in \mathrm{Fix}\, T_{C_1,C_2} \iff x = \frac{x + R_{C_2} R_{C_1} x}{2}$$
$$\iff x = R_{C_2} R_{C_1} x$$

The same argument for the set-valued case yields:

- If $x \in T_{C_1,C_2} x$ then there is an element of $P_{C_1} x$ contained in $C_1 \cap C_2$.

Douglas–Rachford Fixed Points

Why Fix T_{C_1,C_2}**?** Assuming single-valueness of R_{C_1} and R_{C_2} we have:

$$
\begin{aligned}
x \in \operatorname{Fix} T_{C_1,C_2} \quad &\iff \quad x = \frac{x + R_{C_2}R_{C_1}x}{2} \\
&\iff \quad x = R_{C_2}R_{C_1}x \\
&\iff \quad x = 2P_{C_2}R_{C_1}x - R_{C_1}x
\end{aligned}
$$

The same argument for the set-valued case yields:

- If $x \in T_{C_1,C_2}x$ then there is an element of $P_{C_1}x$ contained in $C_1 \cap C_2$.

Douglas–Rachford Fixed Points

Why Fix T_{C_1,C_2}**?** Assuming single-valueness of R_{C_1} and R_{C_2} we have:

$$
\begin{aligned}
x \in \operatorname{Fix} T_{C_1,C_2} \quad &\iff \quad x = \frac{x + R_{C_2}R_{C_1}x}{2} \\
&\iff \quad x = R_{C_2}R_{C_1}x \\
&\iff \quad x = 2P_{C_2}R_{C_1}x - R_{C_1}x \\
&\iff \quad x = 2P_{C_2}R_{C_1}x - 2P_{C_1}x + x
\end{aligned}
$$

The same argument for the set-valued case yields:

- If $x \in T_{C_1,C_2}x$ then there is an element of $P_{C_1}x$ contained in $C_1 \cap C_2$.

Douglas–Rachford Fixed Points

Why $\operatorname{Fix} T_{C_1,C_2}$**?** Assuming single-valueness of R_{C_1} and R_{C_2} we have:

$$
\begin{aligned}
x \in \operatorname{Fix} T_{C_1,C_2} \quad &\iff \quad x = \frac{x + R_{C_2}R_{C_1}x}{2} \\
&\iff \quad x = R_{C_2}R_{C_1}x \\
&\iff \quad x = 2P_{C_2}R_{C_1}x - R_{C_1}x \\
&\iff \quad x = 2P_{C_2}R_{C_1}x - 2P_{C_1}x + x \\
&\iff \quad P_{C_1}x = P_{C_2}R_{C_1}x
\end{aligned}
$$

The same argument for the set-valued case yields:

- If $x \in T_{C_1,C_2}x$ then there is an element of $P_{C_1}x$ contained in $C_1 \cap C_2$.

Douglas–Rachford Fixed Points

Why Fix T_{C_1,C_2}**?** Assuming single-valueness of R_{C_1} and R_{C_2} we have:

$$\begin{aligned}
x \in \operatorname{Fix} T_{C_1,C_2} \quad &\iff \quad x = \frac{x + R_{C_2}R_{C_1}x}{2} \\
&\iff \quad x = R_{C_2}R_{C_1}x \\
&\iff \quad x = 2P_{C_2}R_{C_1}x - R_{C_1}x \\
&\iff \quad x = 2P_{C_2}R_{C_1}x - 2P_{C_1}x + x \\
&\iff \quad P_{C_1}x = P_{C_2}R_{C_1}x \\
&\implies \quad P_{C_1}x \in C_1 \cap C_2.
\end{aligned}$$

The same argument for the set-valued case yields:

- If $x \in T_{C_1,C_2}x$ then there is an element of $P_{C_1}x$ contained in $C_1 \cap C_2$.

Tools from Nonexpansive Mapping Theory

Let $T:\mathcal{H}\to\mathcal{H}$. Then T is:

- nonexpansive if

$$\|Tx-Ty\|\le\|x-y\|,\quad \text{for all } x,y\in\mathcal{H}.$$

- firmly nonexpansive if

$$\|Tx-Ty\|^2+\|(I-T)x-(I-T)y\|^2\le\|x-y\|^2,\quad \text{for all } x,y\in\mathcal{H}.$$

Tools from Nonexpansive Mapping Theory

Let $T : \mathcal{H} \to \mathcal{H}$. Then T is:

- nonexpansive if
$$\|Tx - Ty\| \leq \|x - y\|, \quad \text{for all } x, y \in \mathcal{H}.$$

- firmly nonexpansive if
$$\|Tx - Ty\|^2 + \|(I - T)x - (I - T)y\|^2 \leq \|x - y\|^2, \quad \text{for all } x, y \in \mathcal{H}.$$

Proposition (Nonexpansive properties)

The following are equivalent.

- T is firmly nonexpansive.
- $I - T$ is firmly nonexpansive.
- $2T - I$ is nonexpansive.
- $T = \alpha I + (1 - \alpha)R$, for $\alpha \in (0, 1/2]$ and some nonexpansive R.
- $\langle x - y, Tx - Ty \rangle \geq \|Tx - Ty\|^2$ for all $x, y \in \mathcal{H}$.
- Other characterisations.

Tools from Nonexpansive Mapping Theory

Let $T : \mathcal{H} \to \mathcal{H}$. Then T is:

- nonexpansive if

$$\|Tx - Ty\| \leq \|x - y\|, \quad \text{for all } x, y \in \mathcal{H}.$$

- firmly nonexpansive if

$$\|Tx - Ty\|^2 + \|(I - T)x - (I - T)y\|^2 \leq \|x - y\|^2, \quad \text{for all } x, y \in \mathcal{H}.$$

Nonexpansive properties of projections

Let $C_1, C_2 \subseteq \mathcal{H}$ be closed and convex. Then

- $P_{C_1} := \arg\min_{c \in C_1} \|\cdot - c\|$ is firmly nonexpansive.
- $R_{C_1} := 2P_{C_1} - I$ is nonexpansive.
- $T_{C_1,C_2} := \frac{1}{2}(I + R_{C_2}R_{C_1})$ is firmly nonexpansive.

Nonexpansive maps are closed under composition, convex combinations, etc. Firmly nonexpansive maps need not be. E.g., Composition of two projections onto subspace in $\mathbb{R}^2$ (Bauschke–Borwein–Lewis, 1997).

Tools from Nonexpansive Mapping Theory (cont.)

- asymptotically regular if, for all $x \in \mathcal{H}$,

$$\|T^{n+1}x - T^n x\| \to 0.$$

Lemma (Asymptotic regularity)

Every firmly nonexpansive mapping with at least one fixed point is asymptotically regular.

Proof. Let $z \in \operatorname{Fix} T$ then, for any $x \in \mathcal{H}$, we have

$$\|T^{n+1}x - z\|^2 + \|(I-T)(T^n x)\|^2 = \|T(T^n x) - Tz\|^2 + \|(I-T)(T^n x) - (I-T)z\|^2 \le \|T^n x - z\|^2.$$

Hence $\lim_{n\to\infty} \|T^n x - z\|$ exists, and thus $\|(I-T)(T^n x)\| \to 0$. ∎

A useful Theorem for building iterative schemes:

Theorem (Opial, 1967)

Let $T : \mathcal{H} \to \mathcal{H}$ be nonexpansive and asymptotically regular with $\operatorname{Fix} T \neq \emptyset$. Set $x_{n+1} = Tx_n$. Then $x_n \xrightarrow{w} x$ such that $x \in \operatorname{Fix} T$.

$\to$ Design a non-expansive operator with a useful fixed point set.

Tools from Nonexpansive Mapping Theory (cont.)

- asymptotically regular if, for all $x \in \mathcal{H}$,

$$\|T^{n+1}x - T^n x\| \to 0.$$

Lemma (Asymptotic regularity)

Every firmly nonexpansive mapping with at least one fixed point is asymptotically regular.

Proof. Let $z \in \operatorname{Fix} T$ then, for any $x \in \mathcal{H}$, we have

$$\|T^{n+1}x - z\|^2 + \|(I-T)(T^n x)\|^2$$
$$= \|T(T^n x) - Tz\|^2 + \|(I-T)(T^n x) - (I-T)z\|^2 \le \|T^n x - z\|^2.$$

Hence $\lim_{n\to\infty} \|T^n x - z\|$ exists, and thus $\|(I-T)(T^n x)\| \to 0$. ●

A useful Theorem for building iterative schemes:

Theorem (Opial, 1967)

Let $T : \mathcal{H} \to \mathcal{H}$ be nonexpansive and asymptotically regular with $\operatorname{Fix} T \neq \emptyset$. Set $x_{n+1} = Tx_n$. Then $x_n \xrightarrow{w} x$ such that $x \in \operatorname{Fix} T$.

$\to$ Design a non-expansive operator with a useful fixed point set.

Tools from Nonexpansive Mapping Theory (cont.)

- asymptotically regular if, for all $x \in \mathcal{H}$,
$$\|T^{n+1}x - T^n x\| \to 0.$$

Lemma (Asymptotic regularity)

Every firmly nonexpansive mapping with at least one fixed point is asymptotically regular.

Proof. Let $z \in \operatorname{Fix} T$ then, for any $x \in \mathcal{H}$, we have

$$\|T^{n+1}x - z\|^2 + \|(I-T)(T^n x)\|^2$$
$$= \|T(T^n x) - Tz\|^2 + \|(I-T)(T^n x) - (I-T)z\|^2 \leq \|T^n x - z\|^2.$$

Hence $\lim_{n\to\infty} \|T^n x - z\|$ exists, and thus $\|(I-T)(T^n x)\| \to 0$. ●

A useful Theorem for building iterative schemes:

Theorem (Opial, 1967)

Let $T : \mathcal{H} \to \mathcal{H}$ be nonexpansive and asymptotically regular with $\operatorname{Fix} T \neq \emptyset$. Set $x_{n+1} = Tx_n$. Then $x_n \xrightarrow{w} x$ such that $x \in \operatorname{Fix} T$.

→ Design a non-expansive operator with a useful fixed point set.

Proof of Opial's Theorem

Before proving this theorem, we require the following lemma.

Lemma (Demiclosedness)

Let $T : \mathcal{H} \to \mathcal{H}$ be nonexpansive and denote $x_n := T^n x_0$ for some initial point $x_0 \in \mathcal{H}$. Suppose $x_n \overset{w}{\rightharpoonup} x$ and $x_n - Tx_n \to 0$. Then $x \in \operatorname{Fix} T$.

Proof. Since T is nonexpansive,

$$\begin{aligned}\|x - Tx\|^2 &= \|x_n - Tx\|^2 - \|x_n - x\|^2 - 2\langle x_n - x, x - Tx\rangle \\ &= \|x_n - Tx_n\|^2 + 2\langle x_n - Tx_n, Tx_n - Tx\rangle + \|Tx_n - Tx\|^2 \\ &\qquad - \|x_n - x\|^2 - 2\langle x_n - x, x - Tx\rangle \\ &\leq \|x_n - Tx_n\|^2 + 2\langle x_n - Tx_n, \underbrace{Tx_n}_{x_{n+1}} - Tx\rangle - 2\langle x_n - x, x - Tx\rangle.\end{aligned}$$

Since $x_n \overset{w}{\rightharpoonup} x$ and $x_n - Tx_n \to 0$, it follows that each term tends to 0. ∎

Proof of Opial's Theorem

Before proving this theorem, we require the following lemma.

Lemma (Demiclosedness)

Let $T : \mathcal{H} \to \mathcal{H}$ be nonexpansive and denote $x_n := T^n x_0$ for some initial point $x_0 \in \mathcal{H}$. Suppose $x_n \overset{w}{\rightharpoonup} x$ and $x_n - Tx_n \to 0$. Then $x \in \operatorname{Fix} T$.

Proof. Since T is nonexpansive,

$$\begin{aligned}
\|x - Tx\|^2 &= \|x_n - Tx\|^2 - \|x_n - x\|^2 - 2\langle x_n - x, x - Tx\rangle \\
&= \|x_n - Tx_n\|^2 + 2\langle x_n - Tx_n, Tx_n - Tx\rangle + \|Tx_n - Tx\|^2 \\
&\qquad - \|x_n - x\|^2 - 2\langle x_n - x, x - Tx\rangle \\
&\le \|x_n - Tx_n\|^2 + 2\langle x_n - Tx_n, \underbrace{Tx_n}_{x_{n+1}} - Tx\rangle - 2\langle x_n - x, x - Tx\rangle.
\end{aligned}$$

Since $x_n \overset{w}{\rightharpoonup} x$ and $x_n - Tx_n \to 0$, it follows that each term tends to 0. ●

Proof of Opial's Theorem

Proof (Opial's Theorem)**.** Since T is non-expansive, for any $y \in \operatorname{Fix} T$, we have

$$\|T^{n+1}x - y\| \leq \|T^n x - y\| \leq \cdots \leq \|x - y\|.$$

Whence the sequence $\{x_n\}_{n\in\mathbb{N}}$ is Fejér monotone w.r.t the closed convex set $\operatorname{Fix} T$. By Th. 4.5.10(iii) of Lect. I (Properties of Fejér monotone sequences) the sequence $\{x_n\}_{n\in\mathbb{N}}$ has at most one weak cluster point in $\operatorname{Fix} T$. To complete the proof it suffices to show: (i) $\{x_n\}_{n\in\mathbb{N}}$ has at least one cluster point; and (ii) that every cluster point of $\{x_n\}_{n\in\mathbb{N}}$ is contained in $\operatorname{Fix} T$.

Indeed, as $\{x_n\}$ is bounded, it contains at least one weak cluster point. Let z be any weak cluster point and denote by $\{x_{n_k}\}_{k\in\mathbb{N}}$ a subsequence weakly convergent to z. Since T is asymptotically regular,

$$\|x_{n_k} - Tx_{n_k}\| \to 0.$$

By the Demiclosedness Lemma, $z \in \operatorname{Fix} T$. This completes the proof. ∎

Proof of Opial's Theorem

Proof (Opial's Theorem). Since T is non-expansive, for any $y \in \operatorname{Fix} T$, we have

$$\|T^{n+1}x - y\| \leq \|T^n x - y\| \leq \cdots \leq \|x - y\|.$$

Whence the sequence $\{x_n\}_{n\in\mathbb{N}}$ is Fejér monotone w.r.t the closed convex set $\operatorname{Fix} T$. By Th. 4.5.10(iii) of Lect. I (Properties of Fejér monotone sequences) the sequence $\{x_n\}_{n\in\mathbb{N}}$ has at most one weak cluster point in $\operatorname{Fix} T$. To complete the proof it suffices to show: (i) $\{x_n\}_{n\in\mathbb{N}}$ has at least one cluster point; and (ii) that every cluster point of $\{x_n\}_{n\in\mathbb{N}}$ is contained in $\operatorname{Fix} T$.

Indeed, as $\{x_n\}$ is bounded, it contains at least one weak cluster point. Let z be any weak cluster point and denote by $\{x_{n_k}\}_{k\in\mathbb{N}}$ a subsequence weakly convergent to z. Since T is asymptotically regular,

$$\|x_{n_k} - Tx_{n_k}\| \to 0.$$

By the Demiclosedness Lemma, $z \in \operatorname{Fix} T$. This completes the proof. ●

Proof of Opial's Theorem

Proof (Opial's Theorem). Since T is non-expansive, for any $y \in \operatorname{Fix} T$, we have

$$\|T^{n+1}x - y\| \leq \|T^n x - y\| \leq \cdots \leq \|x - y\|.$$

Whence the sequence $\{x_n\}_{n\in\mathbb{N}}$ is Fejér monotone w.r.t the closed convex set $\operatorname{Fix} T$. By Th. 4.5.10(iii) of Lect. I (Properties of Fejér monotone sequences) the sequence $\{x_n\}_{n\in\mathbb{N}}$ has at most one weak cluster point in $\operatorname{Fix} T$. To complete the proof it suffices to show: (i) $\{x_n\}_{n\in\mathbb{N}}$ has at least one cluster point; and (ii) that every cluster point of $\{x_n\}_{n\in\mathbb{N}}$ is contained in $\operatorname{Fix} T$.

Indeed, as $\{x_n\}$ is bounded, it contains at least one weak cluster point. Let z be any weak cluster point and denote by $\{x_{n_k}\}_{k\in\mathbb{N}}$ a subsequence weakly convergent to z. Since T is asymptotically regular,

$$\|x_{n_k} - Tx_{n_k}\| \to 0.$$

By the Demiclosedness Lemma, $z \in \operatorname{Fix} T$. This completes the proof. ●

The Douglas–Rachford Algorithm

The basic result which we have proven is the following.

Theorem (Douglas–Rachford '56, Lions–Mercier '79, Eckstein–Bertsekas '92, ...)

Suppose $C_1, C_2 \subseteq \mathcal{H}$ are closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := T_{C_1,C_2} x_n \quad \text{where} \quad T_{C_1,C_2} := \frac{I + R_{C_2} R_{C_1}}{2}.$$

Then (x_n) converges weakly to some $x \in \operatorname{Fix} T_{C_1,C_2}$ with $P_{C_1} x \in C_1 \cap C_2$.

Proof. Since $C_1 \cap C_2 \subseteq \operatorname{Fix} T_{C_1,C_2}$, the latter is non-empty. Thus T_{C_1,C_2} is (firmly) nonexpansive with a fixed point, hence asymptotically regular by the previous lemma. The result follows from Opial's Theorem.

- If the intersection is empty the iterates diverge: $\|x_n\| \to \infty$.
- Bauschke–Combettes–Luke (2004): Thorough analysis of convex case.
- Hesse *et al.* & Bauschke *et al.* (2014): Convergence is strong for subspaces, and the rate is linear whenever their sum is closed.
- Phan (arXiv:1401.6509v3): If $\dim \mathcal{H} < \infty$ and $\operatorname{ri} C_1 \cap \operatorname{ri} C_2 \neq \emptyset$ then convergence in linear.

The Douglas–Rachford Algorithm

The basic result which we have proven is the following.

Theorem (Douglas–Rachford '56, Lions–Mercier '79, Eckstein–Bertsekas '92, . . .)

Suppose $C_1, C_2 \subseteq \mathcal{H}$ are closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := T_{C_1,C_2} x_n \quad \text{where} \quad T_{C_1,C_2} := \frac{I + R_{C_2} R_{C_1}}{2}.$$

Then (x_n) converges weakly to some $x \in \operatorname{Fix} T_{C_1,C_2}$ with $P_{C_1} x \in C_1 \cap C_2$.

Proof. Since $C_1 \cap C_2 \subseteq \operatorname{Fix} T_{C_1,C_2}$, the latter is non-empty. Thus T_{C_1,C_2} is (firmly) nonexpansive with a fixed point, hence asymptotically regular by the previous lemma. The result follows from Opial's Theorem. ●

- If the intersection is empty the iterates diverge: $\|x_n\| \to \infty$.
- Bauschke–Combettes–Luke (2004): Thorough analysis of convex case.
- Hesse *et al.* & Bauschke *et al.* (2014): Convergence is strong for subspaces, and the rate is linear whenever their sum is closed.
- Phan (arXiv:1401.6509v3): If $\dim \mathcal{H} < \infty$ and $\operatorname{ri} C_1 \cap \operatorname{ri} C_2 \neq \emptyset$ then convergence in linear.

The Douglas–Rachford Algorithm

The basic result which we have proven is the following.

Theorem (Douglas–Rachford '56, Lions–Mercier '79, Eckstein–Bertsekas '92, ...)

Suppose $C_1, C_2 \subseteq \mathcal{H}$ are closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := T_{C_1,C_2} x_n \quad \text{where} \quad T_{C_1,C_2} := \frac{I + R_{C_2} R_{C_1}}{2}.$$

Then (x_n) converges weakly to some $x \in \operatorname{Fix} T_{C_1,C_2}$ with $P_{C_1} x \in C_1 \cap C_2$.

Proof. Since $C_1 \cap C_2 \subseteq \operatorname{Fix} T_{C_1,C_2}$, the latter is non-empty. Thus T_{C_1,C_2} is (firmly) nonexpansive with a fixed point, hence asymptotically regular by the previous lemma. The result follows from Opial's Theorem. ∎

- If the intersection is empty the iterates diverge: $\|x_n\| \to \infty$.
- Bauschke–Combettes–Luke (2004): Thorough analysis of convex case.
- Hesse *et al.* & Bauschke *et al.* (2014): Convergence is strong for subspaces, and the rate is linear whenever their sum is closed.
- Phan (arXiv:1401.6509v3): If $\dim \mathcal{H} < \infty$ and $\operatorname{ri} C_1 \cap \operatorname{ri} C_2 \neq \emptyset$ then convergence in linear.

The Douglas–Rachford Algorithm

The following generalization include potentially empty intersections. Let

$$V := \overline{C_1 - C_2}, \qquad v := P_V(0), \qquad F := C_1 \cap (C_2 + v).$$

Theorem (Bauschke–Combettes–Luke 2004)

Suppose $C_1, C_2 \subseteq \mathcal{H}$ are closed and convex. Given $x_0 \in \mathcal{H}$ define $x_{n+1} := T_{C_2,C_1} x_n$. Then the following hold.

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

(b) If $C_1 \cap C_2 \neq \emptyset$ then (x_n) converges weakly to a point in

$$\operatorname{Fix} T_{C_1,C_2} = C_1 \cap C_2 + N_V(0);$$

otherwise, $\|x_n\| \to +\infty$.

(c) Exactly one of the following alternatives holds:

(i) $F = \emptyset$, $\|P_{C_1} x_n\| \to +\infty$ and $\|P_{C_2} P_{C_1} x_n\| \to +\infty$.

(ii) $F \neq \emptyset$, the sequence $(P_{C_1} x_n)$ and $(P_{C_2} P_{C_1} x_n)$ are bounded and their weak cluster points are best approximation pairs relative to (C_1, C_2).

The Douglas–Rachford Algorithm: Moment Problem

Recall the moment problem from Lecture I for linear map $A : X \to \mathbb{R}^M$ and a point $y \in \mathbb{R}^M$ has constraints:

$$C_1 := \mathcal{H}^+, \qquad C_2 := \{x \in \mathcal{H} : A(x) = y\}.$$

The following theorem gives conditions for norm convergence.

Theorem (Borwein–Sims–Tam 2015)

Let $\mathcal{H}$ be a Hilbert lattice, $C_1 := \mathcal{H}^+$, C_2 be a closed affine subspace with finite codimensions, and $C_1 \cap C_2 \neq \emptyset$. For $x_0 \in \mathcal{H}$ define $x_{n+1} = T_{C_1,C_2}x_n$. Let Q denote the projection onto the subspace parallel to C_2. Then (x_n) converges in norm whenever:

(a) $C_1 \cap \text{range}(Q) = \{0\}$,

(b) $Q(C_2 - C_1) \subseteq C_1 \cup (-C_1)$ and $Q(C_1) \subseteq C_1$.

(c) C_2 has codimension 1.

For codimension greater than 1?

The Douglas–Rachford Algorithm: Moment Problem

Recall the moment problem from Lecture I for linear map $A : X \to \mathbb{R}^M$ and a point $y \in \mathbb{R}^M$ has constraints:

$$C_1 := \mathcal{H}^+, \qquad C_2 := \{x \in \mathcal{H} : A(x) = y\}.$$

The following theorem gives conditions for norm convergence.

Theorem (Borwein–Sims–Tam 2015)

Let $\mathcal{H}$ be a Hilbert lattice, $C_1 := \mathcal{H}^+$, C_2 be a closed affine subspace with finite codimensions, and $C_1 \cap C_2 \neq \emptyset$. For $x_0 \in \mathcal{H}$ define $x_{n+1} = T_{C_1,C_2} x_n$. Let Q denote the projection onto the subspace parallel to C_2. Then (x_n) converges in norm whenever:

(a) $C_1 \cap \operatorname{range}(Q) = \{0\}$,

(b) $Q(C_2 - C_1) \subseteq C_1 \cup (-C_1)$ and $Q(C_1) \subseteq C_1$.

(c) C_2 has codimension 1.

For codimension greater than 1?

Pierra's Product Space Reformulation

For our constraint sets $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ we define

$$\mathbf{D} := \{(x, x, \ldots, x) \in \mathcal{H}^N : x \in \mathcal{H}\}, \quad \mathbf{C} := \prod_{j=1}^{N} C_j.$$

We now have an equivalent two set feasibility problem since

$$x \in \bigcap_{j=1}^{N} C_j \subseteq \mathcal{H} \iff (x, x, \ldots, x) \in \mathbf{D} \cap \mathbf{C} \subseteq \mathcal{H}^N.$$

Moreover the projections onto the new sets can be computed whenever $P_{C_1}, P_{C_2}, \ldots, P_{C_N}$. Denote $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ they are given by

$$P_{\mathbf{D}}\mathbf{x} = \left(\frac{1}{N}\sum_{j=1}^{N} x_i\right)^N \quad \text{and} \quad P_{\mathbf{C}}\mathbf{x} = \prod_{j=1}^{N} P_{C_j} x_j.$$

Pierra's Product Space Reformulation

For our constraint sets $C_1, C_2, \dots, C_N \subseteq \mathcal{H}$ we define

$$\mathbf{D} := \{(x, x, \dots, x) \in \mathcal{H}^N : x \in \mathcal{H}\}, \quad \mathbf{C} := \prod_{j=1}^{N} C_j.$$

We now have an equivalent two set feasibility problem since

$$x \in \bigcap_{j=1}^{N} C_j \subseteq \mathcal{H} \iff (x, x, \dots, x) \in \mathbf{D} \cap \mathbf{C} \subseteq \mathcal{H}^N.$$

Moreover the projections onto the new sets can be computed whenever $P_{C_1}, P_{C_2}, \dots, P_{C_N}$. Denote $\mathbf{x} = (x_1, x_2, \dots, x_N)$ they are given by

$$P_{\mathbf{D}}\mathbf{x} = \left(\frac{1}{N}\sum_{j=1}^{N} x_i\right)^N \quad \text{and} \quad P_{\mathbf{C}}\mathbf{x} = \prod_{j=1}^{N} P_{C_j} x_j.$$

A Many-set Douglas–Rachford Scheme?

A Many-set Douglas–Rachford Scheme?

Is there a Douglas–Rachford variant which can be used to solve the problem in the original space? *i.e.*, Without recourse to a product space formulation?

An obvious candidate is the following: Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

A similar argument shows:

- (x_n) converges weakly to a point $x \in \operatorname{Fix} T_{A,B,C}$.
- Unfortunately, it is possible that $P_A x, P_B x, P_C x \notin A \cap B \cap C$.

A Many-set Douglas–Rachford Scheme?

A Many-set Douglas–Rachford Scheme?

Is there a Douglas–Rachford variant which can be used to solve the problem in the original space? *i.e.,* Without recourse to a product space formulation?

An obvious candidate is the following: Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

A similar argument shows:

- (x_n) converges weakly to a point $x \in \operatorname{Fix} T_{A,B,C}$.
- Unfortunately, it is possible that $P_A x, P_B x, P_C x \notin A \cap B \cap C$.

A Many-set Douglas–Rachford Scheme?

A Many-set Douglas–Rachford Scheme?

Is there a Douglas–Rachford variant which can be used to solve the problem in the original space? *i.e.,* Without recourse to a product space formulation?

An obvious candidate is the following: Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

A similar argument shows:

- (x_n) converges weakly to a point $x \in \operatorname{Fix} T_{A,B,C}$.
- Unfortunately, it is possible that $P_Ax, P_Bx, P_Cx \notin A \cap B \cap C$.

A Many-set Douglas–Rachford Scheme?

A Many-set Douglas–Rachford Scheme?

Is there a Douglas–Rachford variant which can be used to solve the problem in the original space? *i.e.,* Without recourse to a product space formulation?

An obvious candidate is the following: Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

A similar argument shows:

- (x_n) converges weakly to a point $x \in \operatorname{Fix} T_{A,B,C}$.
- Unfortunately, it is possible that $P_A x, P_B x, P_C x \notin A \cap B \cap C$.

A Many-set Douglas–Rachford Scheme?

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

Let $x_0 = (-\sqrt{3}, -1)$ & $2 \leq \alpha \leq \infty$.
Define constraints:

$$A := \{\lambda(0,1) : |\lambda| \leq \alpha\},$$
$$B := \{\lambda(\sqrt{3},1) : |\lambda| \leq \alpha\},$$
$$C := \{\lambda(-\sqrt{3},1) : |\lambda| \leq \alpha\}.$$

Then $A \cap B \cap C = \{0\}$.

We have $x_0 \in \operatorname{Fix} T_{A,B,C}$. However,

$$P_A x_0, P_B x_0, P_C x_0 \neq 0.$$

A Many-set Douglas–Rachford Scheme?

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

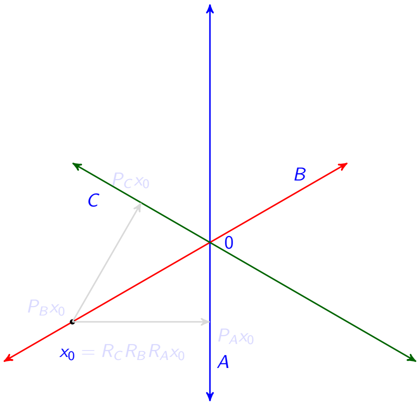
Let $x_0 = (-\sqrt{3}, -1)$ & $2 \leq \alpha \leq \infty$.
Define constraints:

$$A := \{\lambda(0,1) : |\lambda| \leq \alpha\},$$
$$B := \{\lambda(\sqrt{3},1) : |\lambda| \leq \alpha\},$$
$$C := \{\lambda(-\sqrt{3},1) : |\lambda| \leq \alpha\}.$$

Then $A \cap B \cap C = \{0\}$.

We have $x_0 \in \text{Fix}\, T_{A,B,C}$. However,

$$P_A x_0, P_B x_0, P_C x_0 \neq 0.$$



A Many-set Douglas–Rachford Scheme?

$$x_{n+1} = T_{A,B,C} x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

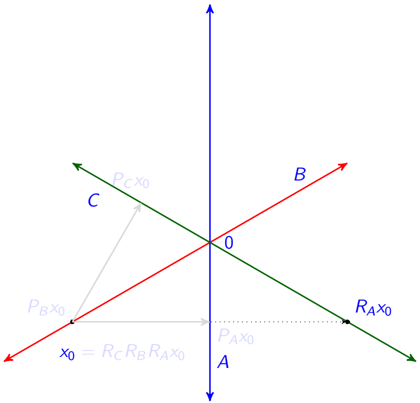
Let $x_0 = (-\sqrt{3}, -1)$ & $2 \leq \alpha \leq \infty$.
Define constraints:

$$A := \{\lambda(0,1) : |\lambda| \leq \alpha\},$$
$$B := \{\lambda(\sqrt{3},1) : |\lambda| \leq \alpha\},$$
$$C := \{\lambda(-\sqrt{3},1) : |\lambda| \leq \alpha\}.$$

Then $A \cap B \cap C = \{0\}$.

We have $x_0 \in \operatorname{Fix} T_{A,B,C}$. However,

$$P_A x_0, P_B x_0, P_C x_0 \neq 0.$$



A Many-set Douglas–Rachford Scheme?

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

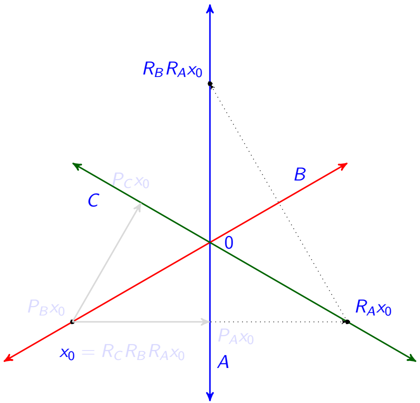
Let $x_0 = (-\sqrt{3}, -1)$ & $2 \leq \alpha \leq \infty$.
Define constraints:

$$A := \{\lambda(0,1) : |\lambda| \leq \alpha\},$$
$$B := \{\lambda(\sqrt{3},1) : |\lambda| \leq \alpha\},$$
$$C := \{\lambda(-\sqrt{3},1) : |\lambda| \leq \alpha\}.$$

Then $A \cap B \cap C = \{0\}$.

We have $x_0 \in \operatorname{Fix} T_{A,B,C}$. However,

$$P_A x_0, P_B x_0, P_C x_0 \neq 0.$$



A Many-set Douglas–Rachford Scheme?

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

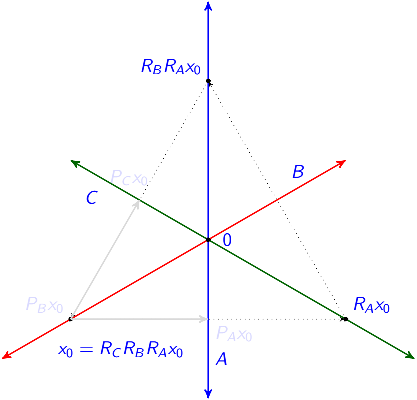
Let $x_0 = (-\sqrt{3}, -1)$ & $2 \leq \alpha \leq \infty$.
Define constraints:

$$A := \{\lambda(0,1) : |\lambda| \leq \alpha\},$$
$$B := \{\lambda(\sqrt{3},1) : |\lambda| \leq \alpha\},$$
$$C := \{\lambda(-\sqrt{3},1) : |\lambda| \leq \alpha\}.$$

Then $A \cap B \cap C = \{0\}$.

We have $x_0 \in \text{Fix}\, T_{A,B,C}$. However,

$$P_A x_0, P_B x_0, P_C x_0 \neq 0.$$



A Many-set Douglas–Rachford Scheme?

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

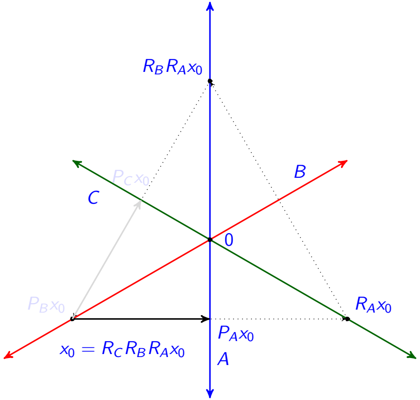
Let $x_0 = (-\sqrt{3}, -1)$ & $2 \leq \alpha \leq \infty$.
Define constraints:

$$A := \{\lambda(0,1) : |\lambda| \leq \alpha\},$$
$$B := \{\lambda(\sqrt{3},1) : |\lambda| \leq \alpha\},$$
$$C := \{\lambda(-\sqrt{3},1) : |\lambda| \leq \alpha\}.$$

Then $A \cap B \cap C = \{0\}$.

We have $x_0 \in \operatorname{Fix} T_{A,B,C}$. However,

$$P_A x_0, P_B x_0, P_C x_0 \neq 0.$$



A Many-set Douglas–Rachford Scheme?

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

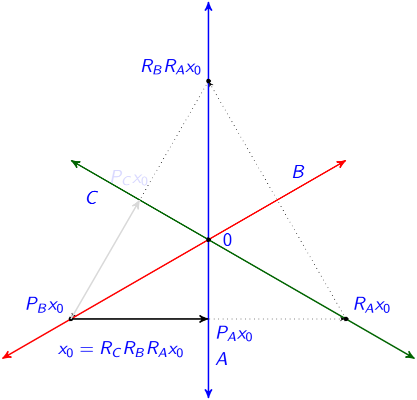
Let $x_0 = (-\sqrt{3}, -1)$ & $2 \leq \alpha \leq \infty$.
Define constraints:

$$A := \{\lambda(0,1) : |\lambda| \leq \alpha\},$$
$$B := \{\lambda(\sqrt{3},1) : |\lambda| \leq \alpha\},$$
$$C := \{\lambda(-\sqrt{3},1) : |\lambda| \leq \alpha\}.$$

Then $A \cap B \cap C = \{0\}$.

We have $x_0 \in \operatorname{Fix} T_{A,B,C}$. However,

$$P_A x_0, P_B x_0, P_C x_0 \neq 0.$$



A Many-set Douglas–Rachford Scheme?

$$x_{n+1} = T_{A,B,C}x_n \quad \text{where} \quad T_{A,B,C} = \frac{I + R_C R_B R_A}{2}.$$

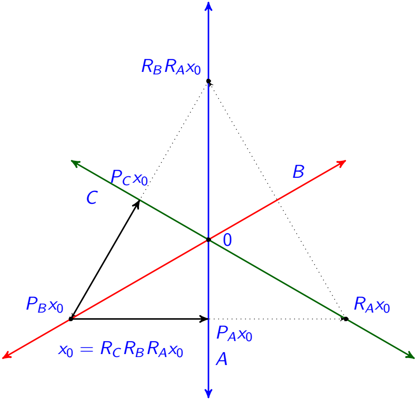
Let $x_0 = (-\sqrt{3}, -1)$ & $2 \leq \alpha \leq \infty$.
Define constraints:

$$A := \{\lambda(0,1) : |\lambda| \leq \alpha\},$$
$$B := \{\lambda(\sqrt{3},1) : |\lambda| \leq \alpha\},$$
$$C := \{\lambda(-\sqrt{3},1) : |\lambda| \leq \alpha\}.$$

Then $A \cap B \cap C = \{0\}$.

We have $x_0 \in \text{Fix}\, T_{A,B,C}$. However,

$$P_A x_0, P_B x_0, P_C x_0 \neq 0.$$



A Common Framework

Theorem (Borwein–Tam 2013)

Let $C_1, \dots, C_N \subseteq \mathcal{H}$ be closed convex sets with nonempty intersection, let $T_j : \mathcal{H} \to \mathcal{H}$ and denote $T := T_M \dots T_2 T_1$. Suppose the following three properties hold.

(i) T is nonexpansive and asymptotically regular,

(ii) $\operatorname{Fix} T = \cap_{j=1}^{M} \operatorname{Fix} T_j \neq \emptyset$,

(iii) $P_{C_j} \operatorname{Fix} T_j \subseteq C_{j+1}$ for each $j = 1, \dots, N$.

Then, for any $x_0 \in \mathcal{H}$, the sequence $x_n := T^n x_0$ converges weakly to some x such that $P_{C_1} x = P_{C_2} x = \dots = P_{C_N} x$. In particular, $P_{C_1} x \in \bigcap_{i=1}^{N} C_i$.

Proof sketch. Denote $C_{N+1} := C_1$.

1. (i) + (ii) $\implies$ (x_n) converges weakly to some $x \in \cap \operatorname{Fix} T$.
2. (iii) + convex projection inequality yields

$$\langle x - P_{C_{j+1}} x, P_{C_j} x - P_{C_{j+1}} x \rangle \leq 0 \text{ for all } j$$

A Common Framework

Theorem (Borwein–Tam 2013)

Let $C_1, \ldots, C_N \subseteq \mathcal{H}$ be closed convex sets with nonempty intersection, let $T_j : \mathcal{H} \to \mathcal{H}$ and denote $T := T_M \ldots T_2 T_1$. Suppose the following three properties hold.

(i) T is nonexpansive and asymptotically regular,

(ii) $\operatorname{Fix} T = \cap_{j=1}^{M} \operatorname{Fix} T_j \neq \emptyset$,

(iii) $P_{C_j} \operatorname{Fix} T_j \subseteq C_{j+1}$ for each $j = 1, \ldots, N$.

Then, for any $x_0 \in \mathcal{H}$, the sequence $x_n := T^n x_0$ converges weakly to some x such that $P_{C_1}x = P_{C_2}x = \cdots = P_{C_N}x$. In particular, $P_{C_1}x \in \bigcap_{i=1}^{N} C_i$.

Proof sketch. Denote $C_{N+1} := C_1$.

1. (i) + (ii) $\implies$ (x_n) converges weakly to some $x \in \cap \operatorname{Fix} T$.
2. (iii) + convex projection inequality yields

$$\langle x - P_{C_{j+1}}x, P_{C_j}x - P_{C_{j+1}}x \rangle \leq 0 \text{ for all } j$$

A Common Framework

To complete the proof observe

$$
\begin{aligned}
&\frac{1}{2}\sum_{j=1}^{N}\|P_{C_{j+1}}x - P_{C_j}x\|^2 \\
&\quad = \langle x, 0\rangle + \frac{1}{2}\sum_{j=1}^{N}\left(\|P_{C_{j+1}}x\|^2 - 2\langle P_{C_{j+1}}x, P_{C_j}x\rangle + \|P_{C_j}x\|^2\right) \\
&\quad = \left\langle x, \sum_{j=1}^{N}(P_{C_j}x - P_{C_{j+1}}x)\right\rangle - \sum_{j=1}^{N}\langle P_{C_{j+1}}x, P_{C_j}x\rangle + \sum_{j=1}^{N}\|P_{C_{j+1}}x\|^2 \\
&\quad = \sum_{j=1}^{N}\langle x, (P_{C_j}x - P_{C_{j+1}}x)\rangle - \sum_{j=1}^{N}\langle P_{C_{j+1}}x, P_{C_j}x - P_{C_{j+1}}x\rangle \\
&\quad = \sum_{j=1}^{N}\langle x - P_{C_{j+1}}x, P_{C_j}x - P_{C_{j+1}}x\rangle \leq 0.
\end{aligned}
$$

Composition of DR-Operators

We require one final theorem.

Theorem (Bauschke *et al.* 2012)

Suppose that each $T_i : \mathcal{H} \to \mathcal{H}$ is firmly nonexpansive and asymptotically regular. Then $T_m T_{m-1} \dots T_1$ is also asymptotically regular.

The proof can be found in:

H.H. Bauschke, V. Martin-Marquez, S.M. Moffat, and X. Wang. **Compositions and convex combinations of asymptotically regular firmly nonexpansive mappings are also asymptotically regular**, *Fixed Point Theory and Applications* 2012, 2012:53.

Composition of DR-Operators

We require one final theorem.

Theorem (Bauschke *et al.* 2012)

Suppose that each $T_i : \mathcal{H} \to \mathcal{H}$ is firmly nonexpansive and asymptotically regular. Then $T_m T_{m-1} \ldots T_1$ is also asymptotically regular.

The proof can be found in:

H.H. Bauschke, V. Martin-Marquez, S.M. Moffat, and X. Wang. **Compositions and convex combinations of asymptotically regular firmly nonexpansive mappings are also asymptotically regular**, *Fixed Point Theory and Applications* 2012, 2012:53.

Cyclic Douglas–Rachford Method

Corollary (Borwein–Tam 2013)

Let $C_1, C_2, \dots, C_N \subseteq \mathcal{H}$ be closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := \underbrace{\left(T_{C_N,C_1} T_{C_{N-1},C_N} \dots T_{C_2,C_3} T_{C_1,C_2}\right)}_{=: T_{[1\,2\,\dots\,N]}} x_n \text{ where } T_{C_j,C_{j+1}} = \frac{I + R_{C_{j+1}} R_{C_j}}{2}.$$

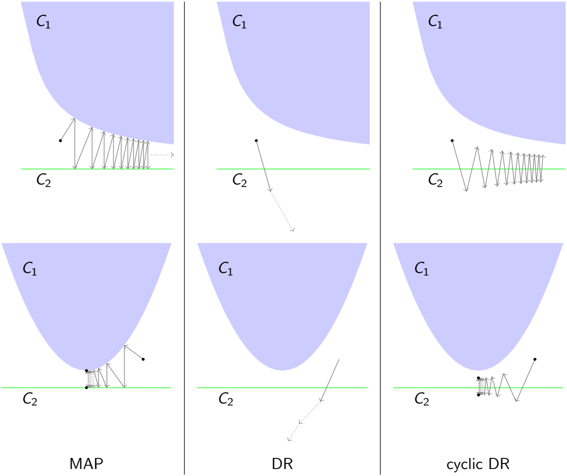
Then (x_n) converges weakly to a point x such that $P_{C_1}x = \cdots = P_{C_N}x$.

- Borwein–Tam (arXiv:1310.2195): Analysed behaviour for empty intersections.
- Using Hundal (2004): There exists a hyperplane and convex cone with nonempty intersection such that convergence is not strong.
- Bauschke–Noll–Phan (2014): If $\dim \mathcal{H} < \infty$ and $\cap_{j=1}^N \operatorname{ri} C_j \neq \emptyset$ then convergence is linear.
- Bauschke–Noll–Phan (2014): If $\operatorname{Fix} T_{[1\,2\,\dots\,N]}$ is bounded linearly regular and $C_j + C_{j+1}$ is closed, for each j, then convergence is linear.

Three Methods: An Example

Consider the following examples with $C_2 := 0 \times \mathbb{R}$, and

$$C_1 := \operatorname{epi}(\exp(\cdot) + 1) \quad \text{or} \quad \operatorname{epi}((\cdot)^2 + 1).$$



Averaged Douglas–Rachford Method

The following variant lends itself to parallel implementation.

Corollary (Borwein-Tam 2013)

Let $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ be closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := \frac{1}{N}\left(\sum_{j=1}^{N} T_{C_j,C_{j+1}}\right) x_n \quad \text{where} \quad T_{C_j,C_{j+1}} = \frac{I + R_{C_{j+1}} R_{C_j}}{2}.$$

Then (x_n) converges weakly to a point x such that $P_{C_1}x = \cdots = P_{C_N}x$.

Proof sketch. For $x_0 \in \mathcal{H}$, set $\mathbf{x}_0 = (x_0, \ldots, x_0) \in \mathcal{H}^N$. Apply the theorem to the product-space iteration

$$\mathbf{x}_{n+1} = P_D\left(\prod_{i=1}^{N} T_{C_i,C_{i+1}}\right)\mathbf{x}_n \in D \subseteq \mathcal{H}^N.$$

Averaged Douglas–Rachford Method

The following variant lends itself to parallel implementation.

Corollary (Borwein-Tam 2013)

Let $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ be closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := \frac{1}{N}\left(\sum_{j=1}^{N} T_{C_j, C_{j+1}}\right) x_n \quad \text{where} \quad T_{C_j,C_{j+1}} = \frac{I + R_{C_{j+1}} R_{C_j}}{2}.$$

Then (x_n) converges weakly to a point x such that $P_{C_1}x = \cdots = P_{C_N}x$.

Proof sketch. For $x_0 \in \mathcal{H}$, set $\mathbf{x}_0 = (x_0, \ldots, x_0) \in \mathcal{H}^N$. Apply the theorem to the product-space iteration

$$\mathbf{x}_{n+1} = P_D\left(\prod_{i=1}^{N} T_{C_i, C_{i+1}}\right)\mathbf{x}_n \in D \subseteq \mathcal{H}^N.$$

•

Cyclically Anchored Douglas–Rachford Method

Choose the first set C_1 to be the anchor set, and think of

$$\bigcap_{j=1}^{N} C_j = C_1 \cap \left(\bigcap_{j=2}^{N} C_j \right).$$

Theorem (Bauschke–Noll–Phan 2014)

Let $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ be closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := \prod_{j=2}^{N} T_{C_1,C_j} x_n \quad \text{where} \quad T_{C_1,C_j} = \frac{I + R_{C_j} R_{C_1}}{2}.$$

Then (x_n) converges weakly to a point x such that $P_{C_1} x \in \bigcap_{j=1}^{N} C_j$.

- Bauschke–Noll–Phan (2014): If $\dim \mathcal{H} < \infty$ and $\cap_{j=1}^{N} \operatorname{ri} C_j \neq \emptyset$ then convergence is linear.
- Bauschke–Noll–Phan (2014): For subspaces, if $\operatorname{Fix} T_{C_1,C_j}$ is bounded linearly regular and $C_1 + C_j$ is closed then convergence is linear.

Averaged Anchored Douglas–Rachford Method

The scheme also has a parallel counterpart:

Theorem

Let $C_1, C_2, \dots, C_N \subseteq \mathcal{H}$ be closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := \frac{1}{N-1}\left(\sum_{j=1}^{N} T_{C_1,C_j}\right) x_n \quad \text{where} \quad T_{C_1,C_j} = \frac{I + R_{C_j} R_{C_i}}{2}.$$

Then (x_n) converges weakly to a point x such that $P_{C_1}x \in \bigcap_{j=1}^{N} C_j$.

Proof sketch. Use the product space (as we did for the averaged DR iteration) up the iteration:

$$\mathbf{x}_{n+1} = P_D\left(\prod_{i=1}^{N} T_{C_1,C_j}\right)\mathbf{x}_n \in D \subseteq \mathcal{H}^{N-1}.$$

Averaged Anchored Douglas–Rachford Method

The scheme also has a parallel counterpart:

Theorem

Let $C_1, C_2, \ldots, C_N \subseteq \mathcal{H}$ be closed and convex with non-empty intersection. Given $x_0 \in \mathcal{H}$ define

$$x_{n+1} := \frac{1}{N-1}\left(\sum_{j=1}^{N} T_{C_1,C_j}\right)x_n \quad \text{where} \quad T_{C_1,C_j} = \frac{I + R_{C_j}R_{C_i}}{2}.$$

Then (x_n) converges weakly to a point x such that $P_{C_1}x \in \bigcap_{j=1}^{N} C_j$.

Proof sketch. Use the product space (as we did for the averaged DR iteration) up the iteration:

$$\mathbf{x}_{n+1} = P_D\left(\prod_{i=1}^{N} T_{C_1,C_j}\right)\mathbf{x}_n \in D \subseteq \mathcal{H}^{N-1}.$$

•

Commentary and Open Questions

- The (classical) Douglas–Rachford method better than theory suggests performance on non-convex problems. Consequently many variants and extensions have recently been proposed.
- Even in the convex setting there are many subtleties and open questions.
  - Norm convergence for realistic moment problems with codimension greater than 1?
- Experimental comparison of the variants needed.

Exercises

1. Let $T_j : \mathcal{H} \to \mathcal{H}$ be firmly nonexpansive, for $j = 1, \ldots, r$, and define $T := T_r \ldots T_2 T_1$. If $\mathrm{Fix}\, T \neq \emptyset$ show that T is asymptotically regular.
2. Show that the cyclic DR method becomes MAP in certain cases. Hence find an example where convergence in cyclic DR is only weak.
3. (**Hard**) Prove or disprove: The Douglas–Rachford algorithm converges in norm for the moment problem when the affine set has codimension 2.

References

- H.H. Bauschke, P.L. Combettes & D.R. Luke. **Finding best approximation pairs relative to two closed convex sets in Hilbert spaces**. *J. Approx. Theory* 127(2):178–192 (2004).
- H.H. Bauschke, J.Y. Cruz, T.T.A. Nghia, H.M. Phan & W. Wang. **The rate of linear convergence of the Douglas-Rachford algorithm for subspaces is the cosine of the Friedrichs angle**. *J. Approx. Theory* 185:63–79 (2014).
- H.H. Bauschke, D. Noll, H.M. Phan. **Linear and strong convergence of algorithms involving averaged nonexpansive operators**. *J. Math. Anal. and Appl.*, 421(1):1–20 (2014).
- J.M. Borwein, B. Sims & M.K. Tam. **Norm convergence of realistic projection and reflection methods**. *Optim.*, 64(1):161–178 (2015).
- J.M. Borwein & M.K. Tam. **A cyclic Douglas–Rachford iteration scheme**. *J. Optim. Theory Appl.*, 160(1):1–29 (2014). http://arxiv.org/abs/1303.1859.
- J.M. Borwein & M.K. Tam. **The cyclic Douglas–Rachford method for inconsistent feasibility problems**. *J. Nonlinear and Convex Anal.*, to appear. http://arxiv.org/abs/1310.2195.
- R. Hesse, D.R. Luke & P. Neumann. **Alternating Projections and Douglas-Rachford for Sparse Affine Feasibility**. *IEEE Trans. Sign. Proc.*, 62(1):4868–4881 (2014).
- H.M. Phan. **Linear convergence of the Douglas–Rachford method for two closed sets**. *arXiv:1401.6509*.

Many resources available at: `http://carma.newcastle.edu.au/DRmethods`

4. Non-Convex Douglas–Rachford

Newcastle in Lonely Planet!

Lonely Planet's top 10 cities

10:30 AEST Mon Nov 1 2010
Adam Bub

10 images in this story

Travel experts Lonely Planet have named the top 10 cities for 2011 in their annual travel bible, *Best in Travel 2011*. The top-listed cities win points for their local cultures, value for money, and overall va-va-voom. So which cities make the cut? Find out here, from 10 to 1...

What do you think of the list? **Tell us here!**

Related links: Lonely Planet destination videos

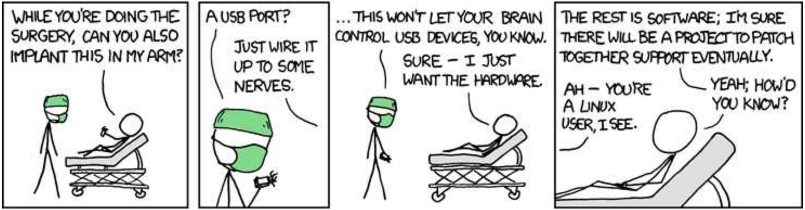
A weekend in Newcastle

Images: ThinkStock/Getaway

9. Newcastle, Australia 2 of 10

The Rest is Software

"It was my luck (perhaps my bad luck) to be the world chess champion during the critical years in which computers challenged, then surpassed, human chess players. Before 1994 and after 2004 these duels held little interest." — Garry Kasparov, 2010



- Likewise much of current Optimization Theory.

Abstract

- The Douglas–Rachford iteration scheme, introduced half a century ago in connection with nonlinear heat flow problems, aims to find a point common to two or more closed constraint sets.
  - Convergence is ensured when the sets are convex subsets of a Hilbert space, however, despite the absence of satisfactory theoretical justification, the scheme has been routinely used to successfully solve a diversity of practical optimization or feasibility problems in which one or more of the constraints involved is non-convex.
- As a first step toward addressing this deficiency, **we provide convergence results for a proto-typical non-convex (phase-recovery) scenario**: Finding a point in the intersection of the Euclidean sphere and an affine subspace.

Abstract

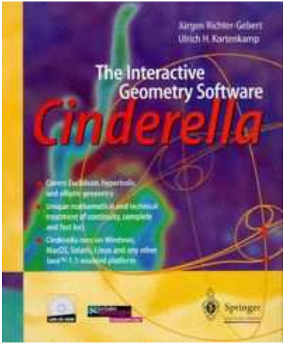
- The Douglas–Rachford iteration scheme, introduced half a century ago in connection with nonlinear heat flow problems, aims to find a point common to two or more closed constraint sets.
  - Convergence is ensured when the sets are convex subsets of a Hilbert space, however, despite the absence of satisfactory theoretical justification, the scheme has been routinely used to successfully solve a diversity of practical optimization or feasibility problems in which one or more of the constraints involved is non-convex.
- As a first step toward addressing this deficiency, **we provide convergence results for a proto-typical non-convex (phase-recovery) scenario**: Finding a point in the intersection of the Euclidean sphere and an affine subspace.

Abstract

- The Douglas–Rachford iteration scheme, introduced half a century ago in connection with nonlinear heat flow problems, aims to find a point common to two or more closed constraint sets.
  - Convergence is ensured when the sets are convex subsets of a Hilbert space, however, despite the absence of satisfactory theoretical justification, the scheme has been routinely used to successfully solve a diversity of practical optimization or feasibility problems in which one or more of the constraints involved is non-convex.
- As a first step toward addressing this deficiency, **we provide convergence results for a proto-typical non-convex (phase-recovery) scenario**: Finding a point in the intersection of the Euclidean sphere and an affine subspace.

An Interactive Presentation

- Much of my lecture will be interactive using the interactive geometry package Cinderella and the HTML applets
  - www.carma.newcastle.edu.au/~jb616/reflection.html
  - www.carma.newcastle.edu.au/~jb616/expansion.html
  - www.carma.newcastle.edu.au/~jb616/lm-june.html



Those Involved

Brailey Sims

Fran Aragon

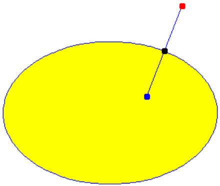
0 Thanks also to Ulli Kortenkamp, Matt Skerritt and Chris Maitland

Phase Reconstruction

Projectors and Reflectors: $P_A(x)$ is the metric projection or nearest point and $R_A(x)$ reflects in the tangent: x is red.

Veit Elser, Ph.D.

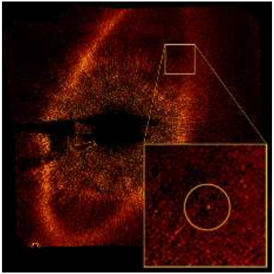
2007 Elser solving Sudoku with **reflectors**.



projection (black) and reflection (blue) of point (red) on boundary (blue) of ellipse (yellow)

"All physicists and a good many quite respectable mathematics are contemptuous about proof."
– **G.H. Hardy** (1877–1947)

2008 Finding exoplanet Fomalhaut in Piscis with **projectors**.

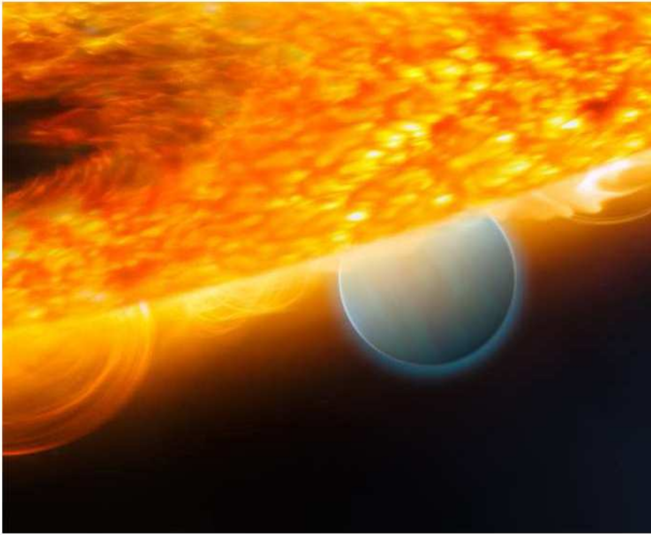


The story of Hubble's 1.3mm error in the "upside down" lens (1990).

And Kepler's hunt for exo-planets (launched March 2009).

We wrote:
"We should add, however, that many Kepler sightings in particular remain to be 'confirmed'. Thus one might legitimately wonder how mathematical robust are the underlying determinations of velocity, imaging, transiting, timing, micro-lensing, etc.?

http://experimentalmath.info/blog/2011/09/where-is-everybody/



Feeling the heat: Kepler scientists justify why some exoplanet data needs to be held back, for now. Image: A "Hot Jupiter" exoplanet close to its host star (ESO).

One of the biggest astronomical stories to unfold over the last decade or so is the story of exoplanets (or "extra-solar planets"). The theory of the formation of our solar system predicts that there should be many more such systems out there. And there certainly are, in fact, 461 at time of writing.

The story of Hubble's 1.3mm error in the "upside down" lens (1990).

And Kepler's hunt for exo-planets (launched March 2009).

We wrote:
"We should add, however, that many Kepler sightings in particular remain to be 'confirmed'. Thus one might legitimately wonder how mathematical robust are the underlying determinations of velocity, imaging, transiting, timing, micro-lensing, etc.?

http://experimentalmath.info/blog/2011/09/where-is-everybody/

THE CONVERSATION BETA
Academic rigour, journalistic flair

26 September 2011, 8.59am AEST

The exoplanet that wasn't. Or was it?

AUTHOR

Sunanda Creagh
Editor

DISCLOSURE STATEMENT

Our goal is to ensure the content is not compromised in any way. We therefore ask all authors to disclose any potential conflicts of interest before publication.

LICENCE TO REPUBLISH

We license our articles under Creative Commons — attribution, no derivatives.

Click here to get a copy of this article to republish.

An exoplanet called Fomalhaut b has been photographed in an unexpected spot — so is it even an exoplanet at all? NASA/http://www.nasa.gov

A distant planet that made its name as the world's first directly photographed exoplanet is at the centre of an astronomical stoush, after it veered off course and new doubts were raised about its existence.

It was in 2008 that Hubble astronomer Paul Kalas from the University of California at Berkeley and NASA announced that Fomalhaut b had been photographed orbiting a star called Fomalhaut around 25 light years from Earth.

Why Does it Work?

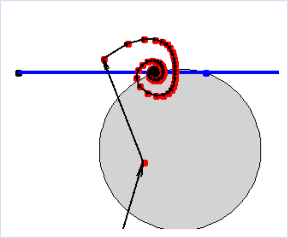
In a wide variety of large hard problems (protein folding, 3SAT, Sudoku) A is non-convex but DR and "divide and concur" (below) works better than theory can explain. It is:

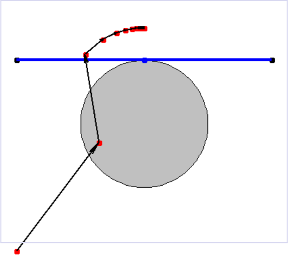
$$R_A(x) := 2P_A(x) - x \text{ and } x \mapsto \frac{x + R_B\left(R_A(x)\right)}{2}.$$

Consider the simplest case of a line B of height h and the unit circle A. With $z_n := (x_n, y_n)$ the iteration becomes

$$x_{n+1} := \cos\theta_n,\ y_{n+1} := y_n + h - \sin\theta_n, \quad (\theta_n := \arg z_n).$$

For $h = 0$: We prove convergence to one of the two points in $A \cap B$ iff we do not start on the vertical axis (where we have chaos). For $h > 1$: (infeasible) it is easy to see the iterates go to infinity (vertically). For $h = 1$: We converge to an infeasible point. For $h \in (0,1)$: The pictures are lovely but proofs escaped us for 9 months. Two representative Maple pictures follow:





sphere and segment at height h

An ideal problem for introducing early undergraduates to research, with many many accessible extensions in 2 or 3 dimensions.

Why Does it Work?

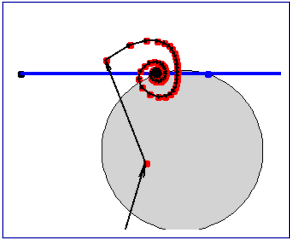
In a wide variety of large hard problems (protein folding, 3SAT, Sudoku) A is non-convex but DR and "divide and concur" (below) works better than theory can explain. It is:

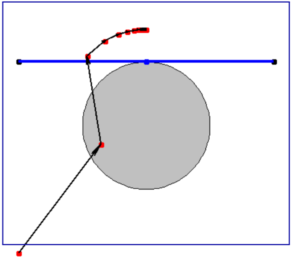
$$R_A(x) := 2P_A(x) - x \text{ and } x \mapsto \frac{x + R_B\left(R_A(x)\right)}{2}.$$

Consider the simplest case of a line B of height h and the unit circle A. With $z_n := (x_n, y_n)$ the iteration becomes

$$x_{n+1} := \cos\theta_n,\ y_{n+1} := y_n + h - \sin\theta_n, \quad (\theta_n := \arg z_n).$$

For $h = 0$: We prove convergence to one of the two points in $A \cap B$ iff we do not start on the vertical axis (where we have chaos). For $h > 1$: (infeasible) it is easy to see the iterates go to infinity (vertically). For $h = 1$: We converge to an infeasible point. For $h \in (0, 1)$: The pictures are lovely but proofs escaped us for 9 months. Two representative Maple pictures follow:





sphere and segment at height h

An ideal problem for introducing early undergraduates to research, with many many accessible extensions in 2 or 3 dimensions.

Why Does it Work?

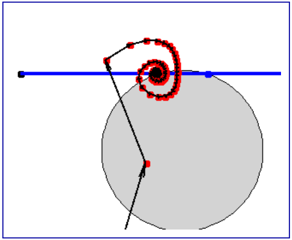
In a wide variety of large hard problems (protein folding, 3SAT, Sudoku) A is non-convex but DR and "divide and concur" (below) works better than theory can explain. It is:

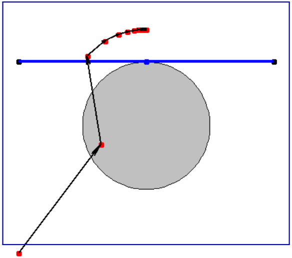
$$R_A(x) := 2P_A(x) - x \text{ and } x \mapsto \frac{x + R_B\left(R_A(x)\right)}{2}.$$

Consider the simplest case of a line B of height h and the unit circle A. With $z_n := (x_n, y_n)$ the iteration becomes

$$x_{n+1} := \cos\theta_n,\ y_{n+1} := y_n + h - \sin\theta_n, \quad (\theta_n := \arg z_n).$$

For $h = 0$: We prove convergence to one of the two points in $A \cap B$ iff we do not start on the vertical axis (where we have chaos). For $h > 1$: (infeasible) it is easy to see the iterates go to infinity (vertically). For $h = 1$: We converge to an infeasible point. For $h \in (0, 1)$: The pictures are lovely but proofs escaped us for 9 months. Two representative Maple pictures follow:





sphere and segment at height h

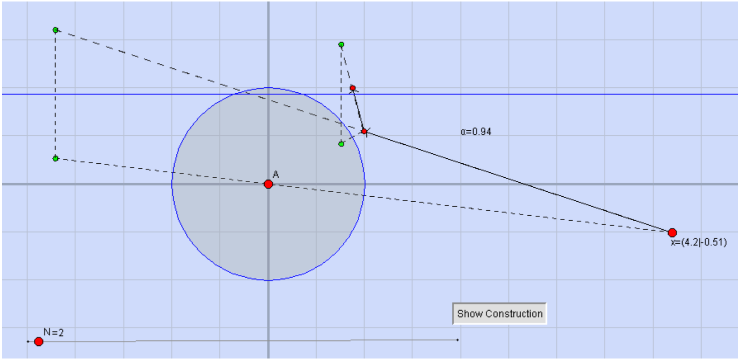
An ideal problem for introducing early undergraduates to research, with many many accessible extensions in 2 or 3 dimensions.

Interactive Phase Recovery in Cinderella

Recall the simplest case of a line B of height h and the unit circle A. With $z_n := (x_n, y_n)$ the iteration becomes

$$R_A(x) := 2P_A(x) - x \text{ and } x \mapsto \frac{x + R_B\left(R_A(x)\right)}{2}.$$

A Cinderella picture of two steps from $(4.2, -0.51)$ follows:

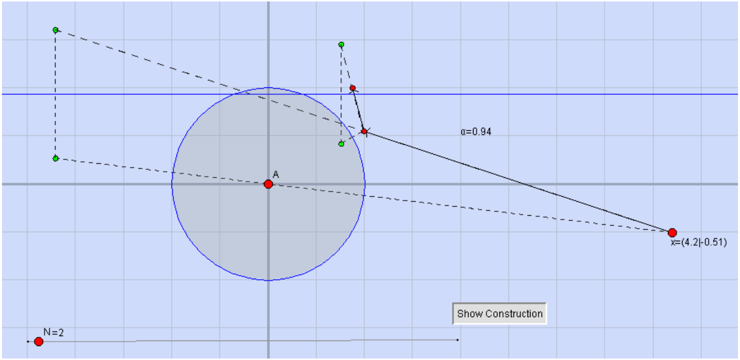


Interactive Phase Recovery in Cinderella

Recall the simplest case of a line B of height h and the unit circle A. With $z_n := (x_n, y_n)$ the iteration becomes

$$R_A(x) := 2P_A(x) - x \text{ and } x \mapsto \frac{x + R_B\left(R_A(x)\right)}{2}.$$

A Cinderella picture of two steps from $(4.2, -0.51)$ follows:

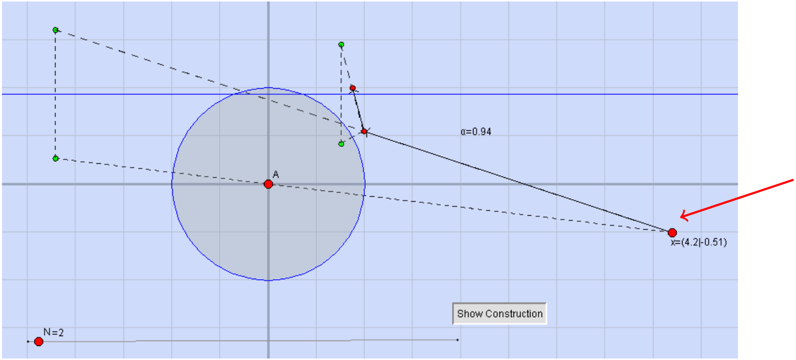


Interactive Phase Recovery in Cinderella

Recall the simplest case of a line B of height h and the unit circle A. With $z_n := (x_n, y_n)$ the iteration becomes

$$R_A(x) := 2P_A(x) - x \text{ and } x \mapsto \frac{x + R_B\left(R_A(x)\right)}{2}.$$

A Cinderella picture of two steps from $(4.2, -0.51)$ follows:

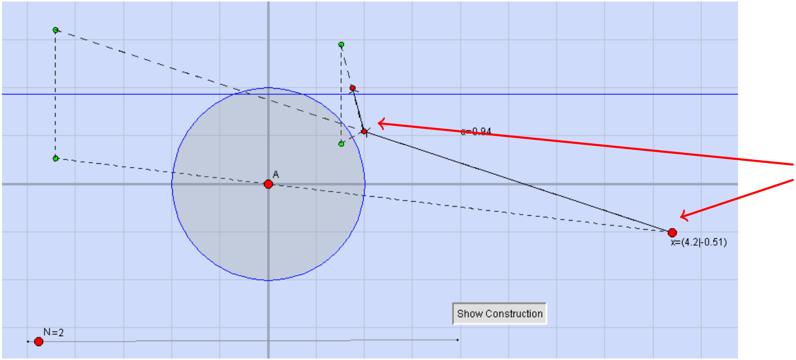


Interactive Phase Recovery in Cinderella

Recall the simplest case of a line B of height h and the unit circle A. With $z_n := (x_n, y_n)$ the iteration becomes

$$R_A(x) := 2P_A(x) - x \text{ and } x \mapsto \frac{x + R_B\left(R_A(x)\right)}{2}.$$

A Cinderella picture of two steps from $(4.2, -0.51)$ follows:

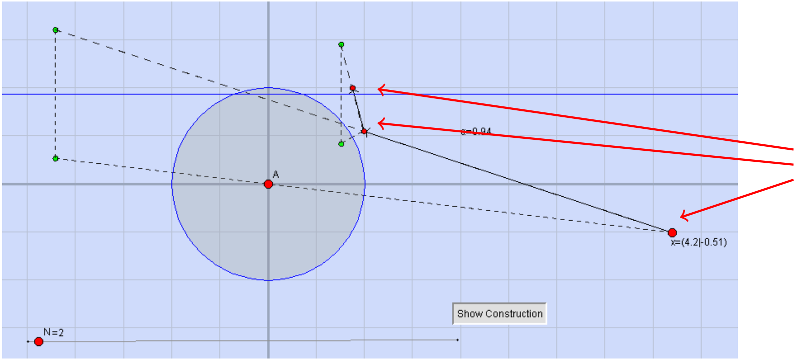


Interactive Phase Recovery in Cinderella

Recall the simplest case of a line B of height h and the unit circle A. With $z_n := (x_n, y_n)$ the iteration becomes

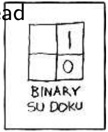
$$R_A(x) := 2P_A(x) - x \text{ and } x \mapsto \frac{x + R_B\left(R_A(x)\right)}{2}.$$

A Cinderella picture of two steps from $(4.2, -0.51)$ follows:



Divide and Concur

To find a point in the intersection of M-sets A_k and in X we can instead consider the subset $A := \prod_{k=1}^M A_k$ and the linear subset



$$B := \{x = (x_1, x_2, \ldots, x_M) : x_1 = x_2 = \cdots = x_M\},$$

of the product Hilbert space $\widetilde{X} := \left(\prod_{k=1}^M X\right)$. We observe

$$R_A(x) = \prod_{k=1}^M R_{A_k}(x_k),$$

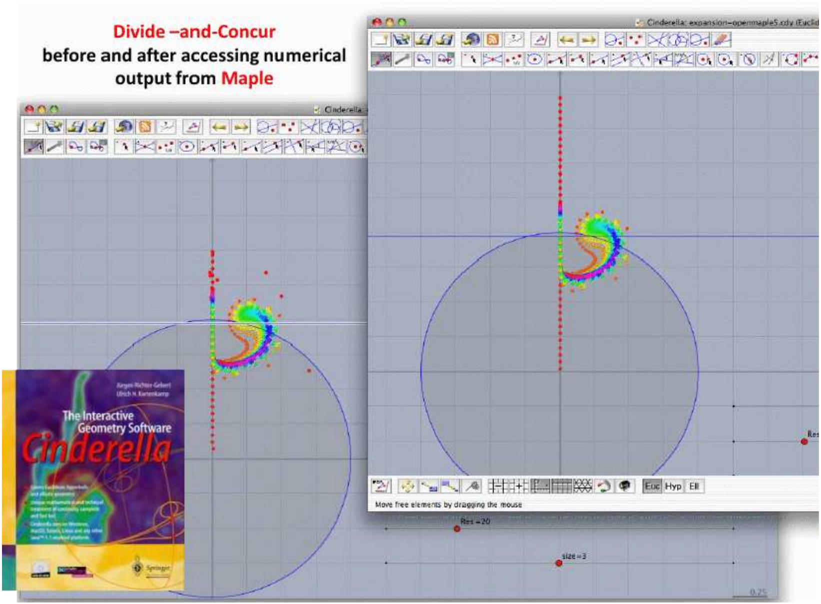
Serial (L) & Parallel (R).

hence the reflection may be 'divided' up and

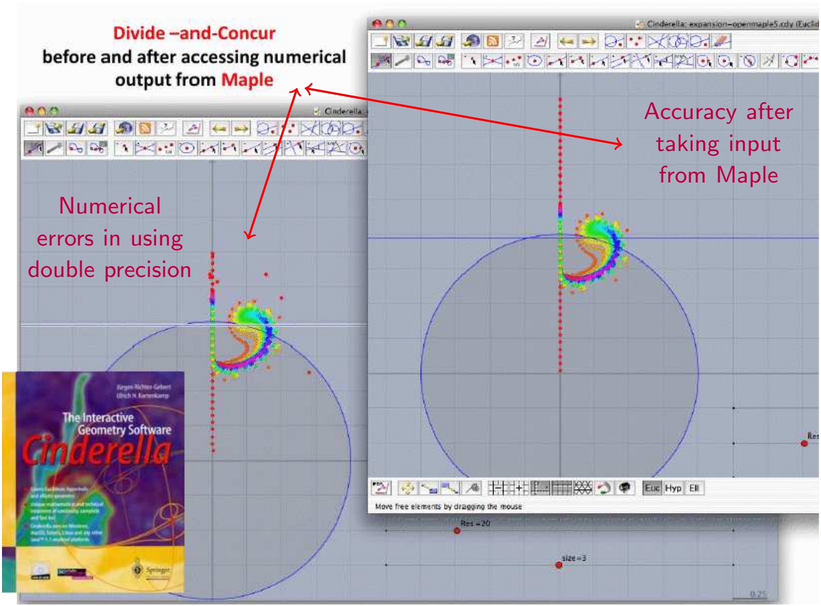
$$P_B(x) = \left(\frac{x_1 + x_2 + \cdots + x_M}{M}, \ldots, \frac{x_1 + x_2 + \cdots + x_M}{M}\right),$$

so that the projection and reflection on B are averaging ('concurrences'), hence the name. In this form the algorithm is suited to parallelization. We can also compose more reflections in serial—we still observe iterates spiralling to a feasible point.

CAS+IGP: The Grief is in the GUI

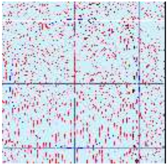


CAS+IGP: The Grief is in the GUI



The Route to Discovery

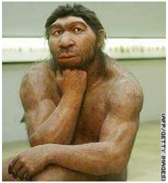
- Exploration first in Maple and then in Cinderella (SAGE)
  - ability to look at orbits/iterations dynamically is great for insight
  - allows for rapid reinforcement and elaboration of intuition



- Decided to look at ODE analogues
  - and their linearizations
  - hoped for Lyapunov like results

$$x'(t) = \frac{x(t)}{r(t)} - x(t), \quad y'(t) = h - \frac{y(t)}{r(t)},$$

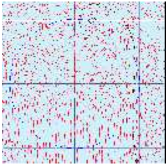
where $r(t) := \sqrt{x(t)^2 + y(t)^2}$, is a reasonable counterpart to the Cartesian formulation —replacing $x_{n+1} - x_n$ by $x'(t)$, etc.—as in Figure.



- Searched literature for a discrete version
  - found Perron's work

The Route to Discovery

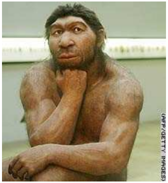
- Exploration first in Maple and then in Cinderella (SAGE)
  - ability to look at orbits/iterations dynamically is great for insight
  - allows for rapid reinforcement and elaboration of intuition



- Decided to look at ODE analogues
  - and their linearizations
  - hoped for Lyapunov like results

$$x'(t) = \frac{x(t)}{r(t)} - x(t), \quad y'(t) = h - \frac{y(t)}{r(t)},$$

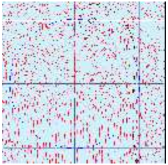
where $r(t) := \sqrt{x(t)^2 + y(t)^2}$, is a reasonable counterpart to the Cartesian formulation —replacing $x_{n+1} - x_n$ by $x'(t)$, etc.—as in Figure.



- Searched literature for a discrete version
  - found Perron's work

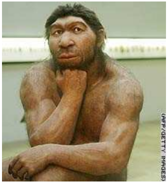
The Route to Discovery

- Exploration first in Maple and then in Cinderella (SAGE)
  - ability to look at orbits/iterations dynamically is great for insight
  - allows for rapid reinforcement and elaboration of intuition
- Decided to look at ODE analogues
  - and their linearizations
  - hoped for Lyapunov like results



$$x'(t) = \frac{x(t)}{r(t)} - x(t), \quad y'(t) = h - \frac{y(t)}{r(t)},$$

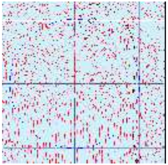
where $r(t) := \sqrt{x(t)^2 + y(t)^2}$, is a reasonable counterpart to the Cartesian formulation —replacing $x_{n+1} - x_n$ by $x'(t)$, etc.—as in Figure.



- Searched literature for a discrete version
  - found Perron's work

The Route to Discovery

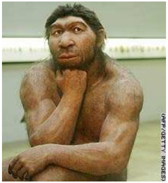
- Exploration first in Maple and then in Cinderella (SAGE)
  - ability to look at orbits/iterations dynamically is great for insight
  - allows for rapid reinforcement and elaboration of intuition



- Decided to look at ODE analogues
  - and their linearizations
  - hoped for Lyapunov like results

$$x'(t) = \frac{x(t)}{r(t)} - x(t), \quad y'(t) = h - \frac{y(t)}{r(t)},$$

where $r(t) := \sqrt{x(t)^2 + y(t)^2}$, is a reasonable counterpart to the Cartesian formulation —replacing $x_{n+1} - x_n$ by $x'(t)$, etc.—as in Figure.



- Searched literature for a discrete version
  - found Perron's work

The Basis of the Proof

Theorem (Perron)

If $f : \mathbb{N} \times \mathbb{R}^m \to \mathbb{R}^m$ satisfies

$$\lim_{x \to 0} \frac{\|f(n,x)\|}{\|x\|} = 0,$$

uniformly in n and M is a constant $n \times n$ matrix all of whose eigenvalues lie inside the unit disk, then the zero solution (provided it is an isolated solution) of the difference equation,

$$x_{n+1} = Mx_n + f(n, x_n),$$

is *exponentially asymptotically stable*; that is, there exists $\delta > 0$, $K > 0$ and $\zeta \in (0,1)$ such that $\|x_0\| < \delta$ then $\|x_n\| \le K\|x_0\|\zeta^n$.

In our case:

$$M = \begin{pmatrix} \alpha^2 & -\alpha\sqrt{1-\alpha^2} & 0 & \dots & 0 \\ \alpha\sqrt{1-\alpha^2} & \alpha^2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$

and the spectrum of the gradient comprises 0, and $\alpha^2 \pm i\alpha\sqrt{1-\alpha^2}$.

The Basis of the Proof

Theorem (Perron)

If $f : \mathbb{N} \times \mathbb{R}^m \to \mathbb{R}^m$ satisfies

$$\lim_{x \to 0} \frac{\|f(n,x)\|}{\|x\|} = 0,$$

uniformly in n and M is a constant $n \times n$ matrix all of whose eigenvalues lie inside the unit disk, then the zero solution (provided it is an isolated solution) of the difference equation,

$$x_{n+1} = Mx_n + f(n, x_n),$$

is *exponentially asymptotically stable*; that is, there exists $\delta > 0, K > 0$ and $\zeta \in (0,1)$ such that $\|x_0\| < \delta$ then $\|x_n\| \leq K\|x_0\|\zeta^n$.

In our case:

$$M = \begin{pmatrix} \alpha^2 & -\alpha\sqrt{1-\alpha^2} & 0 & \dots & 0 \\ \alpha\sqrt{1-\alpha^2} & \alpha^2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$

and the spectrum of the gradient comprises 0, and $\alpha^2 \pm i\alpha\sqrt{1-\alpha^2}$.

The Basis of the Proof

Theorem (Perron)

If $f : \mathbb{N} \times \mathbb{R}^m \to \mathbb{R}^m$ satisfies

$$\lim_{x \to 0} \frac{\|f(n,x)\|}{\|x\|} = 0,$$

uniformly in n and M is a constant $n \times n$ matrix all of whose eigenvalues lie inside the unit disk, then the zero solution (provided it is an isolated solution) of the difference equation,

$$\mathbf{x_{n+1} = Mx_n + f(n, x_n)},$$

is *exponentially asymptotically stable*; that is, there exists $\delta > 0, K > 0$ and $\zeta \in (0,1)$ such that $\|x_0\| < \delta$ then $\|x_n\| \leq K\|x_0\|\zeta^n$.

In our case:

$$M = \begin{pmatrix} \alpha^2 & -\alpha\sqrt{1-\alpha^2} & 0 & \dots & 0 \\ \alpha\sqrt{1-\alpha^2} & \alpha^2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$

Explains spin for height in $(0,1)$

and the spectrum of the gradient comprises 0, and $\alpha^2 \pm i\alpha\sqrt{1-\alpha^2}$.

What We Can Now Show

Theorem (Borwein–Sims 2009)

For the case of a sphere in n-space and a line of height α (normalized so that we have $x(2) = \alpha, a = e_1, b = e_2$):

(a) If $0 \leq \alpha < 1$ then the Douglas–Rachford scheme is locally convergent at each of the critical points $\pm\sqrt{1-\alpha^2}a + \alpha b$.

(b) If $\alpha = 0$ and the initial point has $x_0(1) > 0$ then the scheme converges to the feasible point $(1, 0, 0, \ldots, 0)$.

(c) When L is tangential to S at b (*i.e.*, when $\alpha = 1$), starting from any initial point with $x_0(1) \neq 0$, the scheme converges to a point yb with $y > 1$.

(d) If there are no feasible solutions (*i.e.*, when $\alpha > 1$) then for any non-zero initial point $x_n(2)$ and hence $\|x_n\|$ diverge at at least linear rate to $+\infty$.

- The same result applies to the sphere S and any *affine* subset B.
- For non-affine B things are substantially more complex — even in $\mathbb{R}^2$.

What We Can Now Show

Theorem (Borwein–Sims 2009)

For the case of a sphere in n-space and a line of height α (normalized so that we have $x(2) = \alpha, a = e_1, b = e_2$):

(a) If $0 \leq \alpha < 1$ then the Douglas–Rachford scheme is locally convergent at each of the critical points $\pm\sqrt{1-\alpha^2}a + \alpha b$.

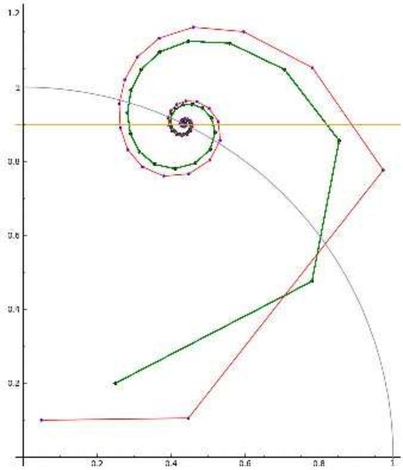
(b) If $\alpha = 0$ and the initial point has $x_0(1) > 0$ then the scheme converges to the feasible point $(1, 0, 0, \ldots, 0)$.

(c) When L is tangential to S at b (*i.e.*, when $\alpha = 1$), starting from any initial point with $x_0(1) \neq 0$, the scheme converges to a point yb with $y > 1$.

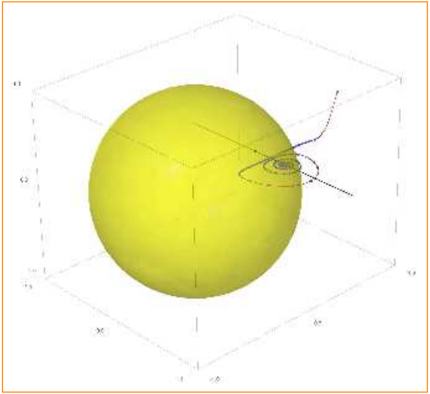
(d) If there are no feasible solutions (*i.e.*, when $\alpha > 1$) then for any non-zero initial point $x_n(2)$ and hence $\|x_n\|$ diverge at at least linear rate to $+\infty$.

- The same result applies to the sphere S and any *affine* subset B.
- For non-affine B things are substantially more complex — even in $\mathbb{R}^2$.

Algorithms *Appears* to be Stable

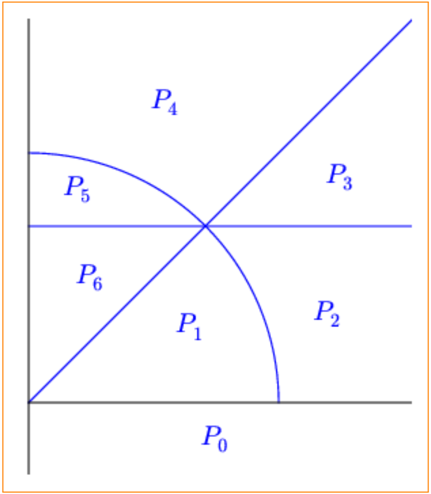


Three and Higher Dimensions



$$x_{n+1}(1) = x_n(1)/\rho_n,$$
$$x_{n+1}(2) = \alpha + (1 - 1/\rho_n)x_n(2), \quad \text{and}$$
$$x_{n+1}(k) = (1 - 1/\rho_n)x_n(k), \quad \text{for } k = 3, \ldots, N$$
where $\rho_n := \|x_n\| = \sqrt{x_n(1)^2 + \cdots + x_n(N)^2}$.

An "Even Simpler" Case



Intersection at $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$.

If $(x_n, y_n) \in P_1 \cup P_2 \cup P_3$ then

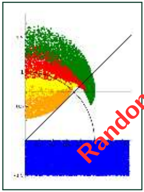
$$|(x_{n+1}, y_{n+1} - (x^*, y^*)|^2 \leq \frac{1}{2}|(x_n, y_n - (x^*, y^*)|^2.$$

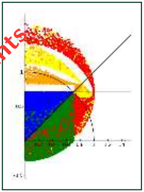
If $(x_n, y_n) \in P_4$ then

$$|(x_{n+1}, y_{n+1} - (x^*, y^*)|^2 \leq |(x_n, y_n - (x^*, y^*)|^2.$$

If $(x_n, y_n) \in P_5 \cup P_6$ then

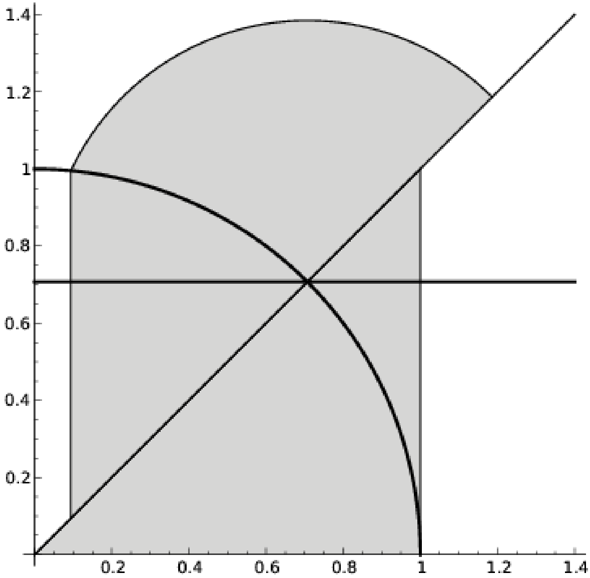
$$|(x_{n+1}, y_{n+1} - (x^*, y^*)|^2 \leq \underbrace{\left(\frac{5}{2} - \sqrt{2} + \frac{1}{2}\sqrt{29 - 20\sqrt{2}}\right)}_{\approx 1.51}|(x_n, y_n - (x^*, y^*)|^2.$$





Random points

Aragón–Borwein Region of Convergence



The Search for a Lyapunov Function

Recent progress has been made by Joël Benoist. His idea is to search for a Lyapunov function V such that ∇V is perpendicular to the DR trajectories. That is,

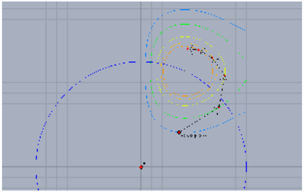
$$\langle \nabla V(x_n,\ y_n), (x_{n-1}, y_{n-1}) - (x_n, y_n)\rangle = 0.$$

Expressing (x_{n-1}, y_{n-1}) is terms of (x_n, y_n) gives the PDE:

$$(y-\lambda)\frac{\partial V}{\partial x}(x,y) + \frac{-\lambda\sqrt{1-x^2}+1-x^2}{x}\frac{\partial V}{\partial y}(x,y) = 0.$$

One solution to this PDE is the following:

$$V(x,y) = \frac{1}{2}(y-\lambda)^2 - \lambda\ln(1+\sqrt{1-x^2}) + \lambda\sqrt{1-x^2} + (\lambda-1)\ln x + \frac{1}{2}x^2.$$



The Search for a Lyapunov Function

Recent progress has been made by Joël Benoist. His idea is to search for a Lyapunov function V such that ∇V is perpendicular to the DR trajectories. That is,

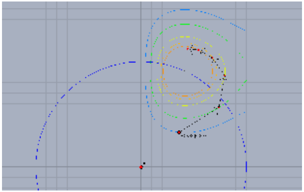
$$\langle \nabla V(x_n,\ y_n), (x_{n-1}, y_{n-1}) - (x_n, y_n) \rangle = 0.$$

Expressing (x_{n-1}, y_{n-1}) is terms of (x_n, y_n) gives the PDE:

$$(y-\lambda)\frac{\partial V}{\partial x}(x,y) + \frac{-\lambda\sqrt{1-x^2}+1-x^2}{x}\frac{\partial V}{\partial y}(x,y) = 0.$$

One solution to this PDE is the following:

$$V(x,y) = \frac{1}{2}(y-\lambda)^2 - \lambda \ln(1+\sqrt{1-x^2}) + \lambda\sqrt{1-x^2} + (\lambda-1)\ln x + \frac{1}{2}x^2.$$



The Search for a Lyapunov Function

Recent progress has been made by Joël Benoist. His idea is to search for a Lyapunov function V such that ∇V is perpendicular to the DR trajectories. That is,

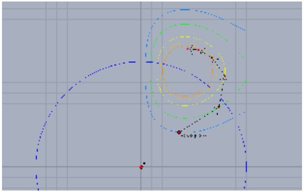
$$\langle \nabla V(x_n,\ y_n), (x_{n-1}, y_{n-1}) - (x_n, y_n)\rangle = 0.$$

Expressing (x_{n-1}, y_{n-1}) is terms of (x_n, y_n) gives the PDE:

$$(y-\lambda)\frac{\partial V}{\partial x}(x,y) + \frac{-\lambda\sqrt{1-x^2}+1-x^2}{x}\frac{\partial V}{\partial y}(x,y) = 0.$$

One solution to this PDE is the following:

$$V(x,y) = \frac{1}{2}(y-\lambda)^2 - \lambda\ln(1+\sqrt{1-x^2}) + \lambda\sqrt{1-x^2} + (\lambda-1)\ln x + \frac{1}{2}x^2.$$



The Search for a Lyapunov Function

Denote the solution $(x^*, y^*) := (\sqrt{1-h^2}, h)$. Recall the Benoist's Lyapunov candidate function

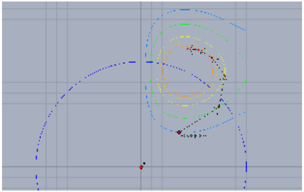
$$V(x,y) = \frac{1}{2}(y-\lambda)^2 - \lambda \ln(1+\sqrt{1-x^2}) + \lambda\sqrt{1-x^2} + (\lambda-1)\ln x + \frac{1}{2}x^2.$$

In the right half-space it is shown that:

1. (V decreases along DR trajectories): For all $\epsilon > 0$,

$$\sup_{\|(x,y)-(x^*,y^*)\|\geq\epsilon} (V(T(x,y)) - V(x,y)) < 0.$$

2. $V(T(x,y)) = V(x,y)$ if and only if $(x,y) = (x^*,y^*)$.



The Search for a Lyapunov Function

Denote the solution $(x^*, y^*) := (\sqrt{1-h^2}, h)$. Recall the Benoist's Lyapunov candidate function

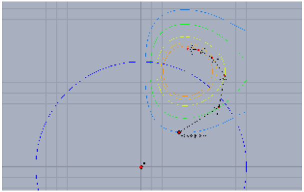
$$V(x,y) = \frac{1}{2}(y-\lambda)^2 - \lambda \ln(1+\sqrt{1-x^2}) + \lambda\sqrt{1-x^2} + (\lambda-1)\ln x + \frac{1}{2}x^2.$$

In the right half-space it is shown that:

1. (V decreases along DR trajectories): For all $\epsilon > 0$,

$$\sup_{\|(x,y)-(x^*,y^*)\|\geq\epsilon} \left(V(T(x,y)) - V(x,y)\right) < 0.$$

2. $V(T(x,y)) = V(x,y)$ if and only if $(x,y) = (x^*,y^*)$.



Global Convergence with a Half-Space Constraint

Consider the two-set feasibility problem given by a closed set $Q \subseteq \mathbb{R}^m$, and the half-space

$$H := \{x \in \mathbb{R}^m : \langle a, x \rangle \leq b\}.$$

where $b \in \mathbb{R}$, and $a \in \mathbb{R}^m$ with $\|a\| = 1$.

In this case, the Douglas–Rachford iteration simplifies to

$$x_{k+1} = \begin{cases} q_k & \text{if } \langle a, 2q_k - x_k \rangle \leq b, \\ q_k + (\langle a, x_k \rangle + b - 2\langle a, q_k \rangle)a & \text{otherwise,} \end{cases}$$

where, at each iteration, a point $q_k \in P_Q(x_k)$ is selected.

Motivated by experimental evidence, we first consider the case in which the set Q is finite.

Global Convergence with a Half-Space Constraint

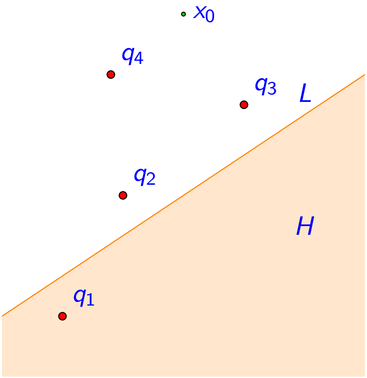


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

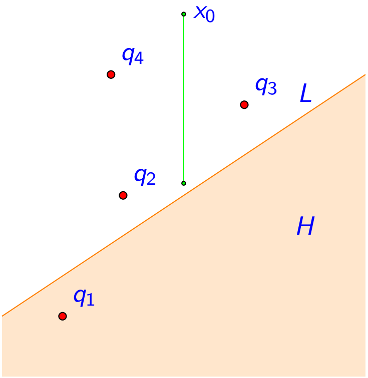


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

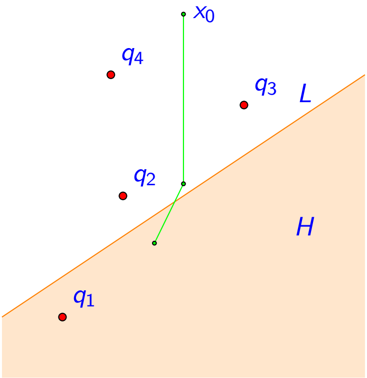


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

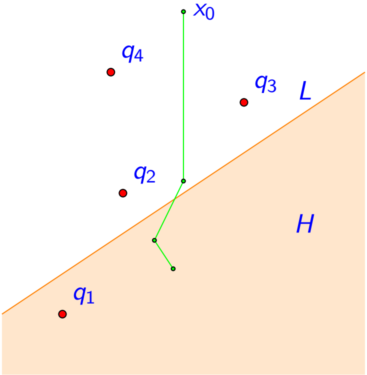


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

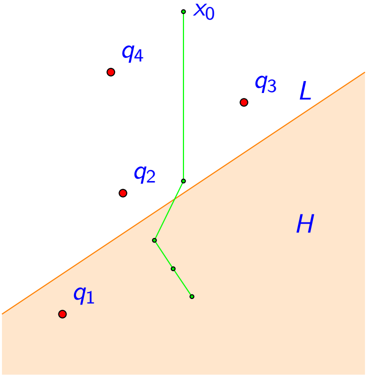


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

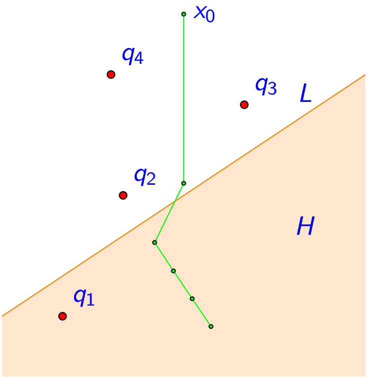


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

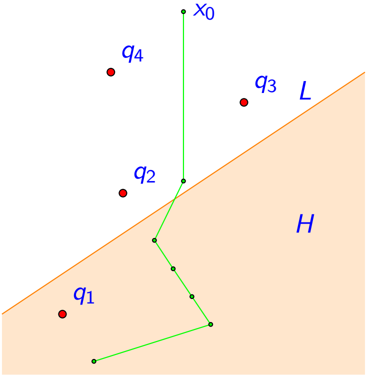


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

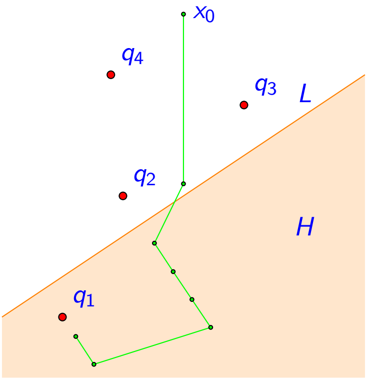


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

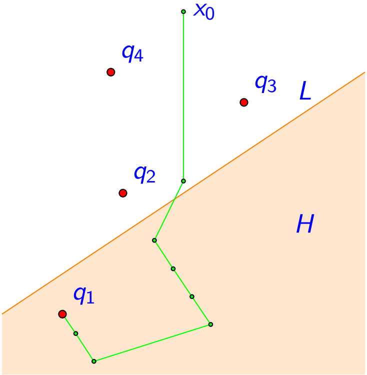


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

Global Convergence with a Half-Space Constraint

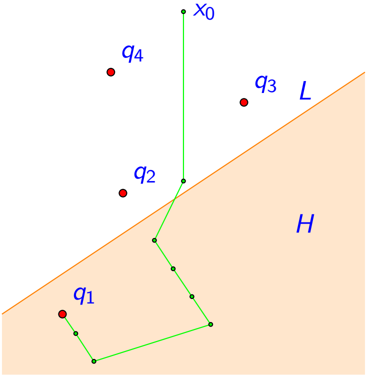


Fig. 1 A Douglas–Rachford iteration in $\mathbb{R}^2$ with the set $Q = \{q_1, q_2, q_3, q_4\}$ finds a solution in eight iterations.

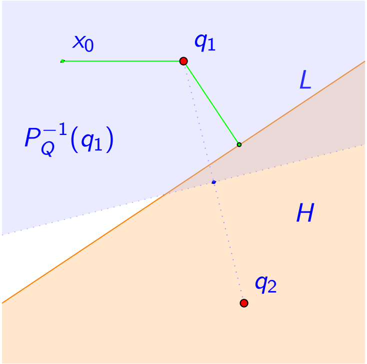


Fig. 2 The alternating projection algorithm fails to find a solution for any initial point in the set $P_Q^{-1}(q_1)$ where $Q = \{q_1, q_2\}$.

Global Convergence with a Half-Space Constraint

Theorem (Aragón Artacho–Borwein–Tam, 2015)

Suppose Q is a compact set. Let $\{x_k\}$ be a Douglas–Rachford sequence and $q_k \in P_Q(x_k)$ for all $k \in \mathbb{N}$. Then either:

(i) $d(q_k, H) \to 0$ and the set of cluster points $\{q_k\}$ is non-empty and contained in $Q \cap H$, or

(ii) $d(q_k, H) \to \beta$ for some $\beta > 0$ and $H \cap Q = \emptyset$.

Moreover, in the latter case, $\|x_k\| \to +\infty$.

It is worth noting that:

1. The set Q is not assumed to satisfy any (local) regularity properties (*e.g.*, strongly regular intersection, prox-regularity, ...).
2. The behaviour of the method does not depend on how q_k is chosen. The result holds for *any* choice.
3. The theorem remains true if one assume that the function

$$x \mapsto \iota_Q(x) + d(x, H),$$

has compact lower-level sets.

Global Convergence with a Half-Space Constraint

Theorem (Aragón Artacho–Borwein–Tam, 2015)

Suppose Q is a compact set. Let $\{x_k\}$ be a Douglas–Rachford sequence and $q_k \in P_Q(x_k)$ for all $k \in \mathbb{N}$. Then either:

(i) $d(q_k, H) \to 0$ and the set of cluster points $\{q_k\}$ is non-empty and contained in $Q \cap H$, or

(ii) $d(q_k, H) \to \beta$ for some $\beta > 0$ and $H \cap Q = \emptyset$.

Moreover, in the latter case, $\|x_k\| \to +\infty$.

It is worth noting that:

1. The set Q is not assumed to satisfy any (local) regularity properties (*e.g.*, strongly regular intersection, prox-regularity, . . .).
2. The behaviour of the method does not depend on how q_k is chosen. The result holds for *any* choice.
3. The theorem remains true if one assume that the function

$$x \mapsto \iota_Q(x) + d(x, H),$$

has compact lower-level sets.

Global Convergence with a Half-Space Constraint

This theorem allows us to deduce global convergence of the Douglas–Rachford method applied to a sphere and a half-space (instead of an affine line).

Example (Global convergence for the sphere and half-space)

Let Q be the unit sphere and H a half-space in $\mathbb{R}^2$. By symmetry, we may assume $a = (0,1)$. Let $x_0 \neq 0$ with $x_0(1) > 0$. Then $x_k(1) > 0$ and $q_k = \frac{x_k}{\|x_k\|}$ for all $k \in \mathbb{N}$, and the iteration becomes

$$x_{k+1}(1) = \frac{x_k(1)}{\|x_k\|}, \quad x_{k+1}(2) = \begin{cases} \frac{x_k(2)}{\|x_k\|} & \text{if } \left(\frac{2}{\|x_k\|} - 1\right) x_k(2) \leq b, \\ \left(1 - \frac{1}{\|x_k\|}\right) x_k(2) + b & \text{otherwise.} \end{cases}$$

If $Q \cap H \neq \emptyset$ (or equivalently $b \geq -1$) then the previous theorem ensures $d(q_k, H) \to 0$. It then follows that either:

1. $q_{k_0} \in H \cap Q$ for some $k_0 \in \mathbb{N}$ (*i.e.*, a solution is found in finitely many iterations), or
2. $q_k(2) \to b$ and hence $q_k \to (\sqrt{1-b^2}, b) \in Q \cap H$.

Global Convergence with a Half-Space Constraint

Specialising to the finite case, we have the following.

Corollary (Aragón Artacho–Borwein–Tam, 2015)

Suppose Q is finite. Let $\{x_k\}$ be a Douglas–Rachford sequence and $q_k \in P_Q(x_k)$ for all $k \in \mathbb{N}$. Then either:

(i) $\{x_k\}$ and $\{q_k\}$ are eventually constant and the limit of $\{q_k\}$ is contained in $H \cap Q \neq \emptyset$, or

(ii) $H \cap Q = \emptyset$ and $\|x_k\| \to +\infty$.

- This corollary explains our previous example.
- First global convergence result for the Douglas–Rachford applicable to discrete/combinatorial constraint sets.
- Bauschke & Noll (2014) proved if the constraints are finite unions of convex sets, then method is locally convergent (in neighbourhoods of strong fixed points).

Global Convergence with a Half-Space Constraint

We give one further example from binary linear programming.

Example (Knapsack lower bound feasibility)

The classical 0-1 knapsack problem is the binary program

$$\min \{\langle c, x\rangle \mid x \in \{0,1\}^n,\ \langle a, x\rangle \leq b\},$$

for vectors $a, c \in \mathbb{R}^m_+$ and $b \geq 0$.

The 0-1 knapsack lower-bound feasibility problem is the problem with constraints

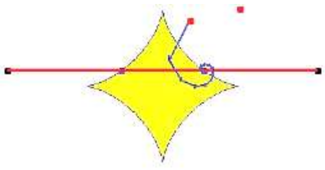
$$H := \{x \in \mathbb{R}^n \mid \langle a, x\rangle \leq b\}, \quad Q := \{x \in \{0,1\}^n \mid \langle c, x\rangle \geq \lambda\},$$

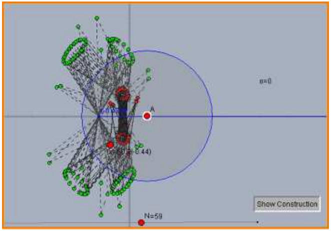
where $\lambda \geq 0$. As a decision problem it is NP-complete.

Applied to this problem, the corollary shows that the Douglas–Rachford method either finds a solution in finitely many iterations, or none exists and the norm of the Douglas–Rachford sequence diverges to infinity. Note that, in general, P_Q usually cannot be computed efficiently.

Commentary and Open Questions

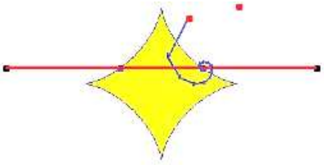
- As noted, the method parallelizes very well.
- Can one work out rates in the general convex case?
- Why does alternating projection (no reflection) work well for optical aberration but not phase reconstruction?
- Other cases of Lyapunov arguments for global convergence?
  - in the appropriate basins?
- Study general sets (in so-called CAT(0)metrics)
  - even the half-line case is much more complex
  - as I may now demo
- Why does the method work for a half-space but not a hyperplane?

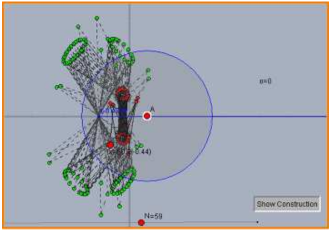




Commentary and Open Questions

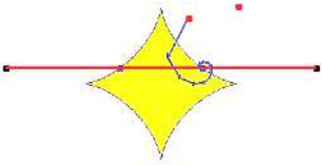
- As noted, the method parallelizes very well.
- Can one work out rates in the general convex case?
- Why does alternating projection (no reflection) work well for optical aberration but not phase reconstruction?
- Other cases of Lyapunov arguments for global convergence?
  - in the appropriate basins?
- Study general sets (in so-called CAT(0)metrics)
  - even the half-line case is much more complex
  - as I may now demo
- Why does the method work for a half-space but not a hyperplane?

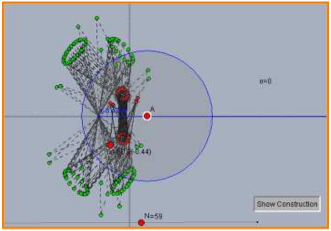




Commentary and Open Questions

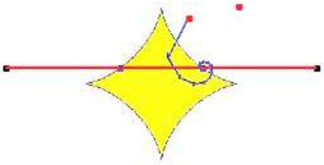
- As noted, the method parallelizes very well.
- Can one work out rates in the general convex case?
- Why does alternating projection (no reflection) work well for optical aberration but not phase reconstruction?
- Other cases of Lyapunov arguments for global convergence?
  - in the appropriate basins?
- Study general sets (in so-called CAT(0)metrics)
  - even the half-line case is much more complex
  - as I may now demo
- Why does the method work for a half-space but not a hyperplane?

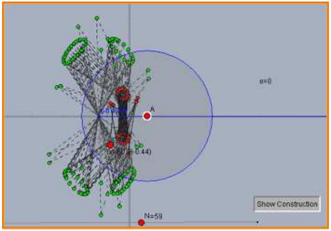




Commentary and Open Questions

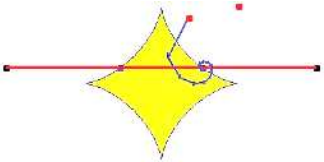
- As noted, the method parallelizes very well.
- Can one work out rates in the general convex case?
- Why does alternating projection (no reflection) work well for optical aberration but not phase reconstruction?
- Other cases of Lyapunov arguments for global convergence?
  - in the appropriate basins?
- Study general sets (in so-called CAT(0)metrics)
  - even the half-line case is much more complex
  - as I may now demo
- Why does the method work for a half-space but not a hyperplane?

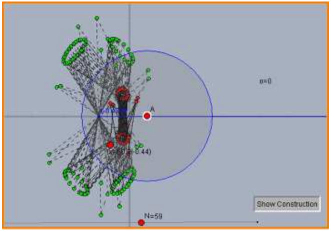




Commentary and Open Questions

- As noted, the method parallelizes very well.
- Can one work out rates in the general convex case?
- Why does alternating projection (no reflection) work well for optical aberration but not phase reconstruction?
- Other cases of Lyapunov arguments for global convergence?
  - in the appropriate basins?
- Study general sets (in so-called CAT(0)metrics)
  - even the half-line case is much more complex
  - as I may now demo
- Why does the method work for a half-space but not a hyperplane?





Exercises

1. (A lemma toward global convergence) The Douglas–Rachford iteration for the line and circle with $\alpha = 1/\sqrt{2}$. Is given by

$$x_{n+1} = \frac{x_n}{\rho_n}, \quad y_{n+1} = \alpha + \left(1 - \frac{1}{\rho_n}\right) y_n = \alpha + (\rho_n - 1)\sin\theta_n,$$

where $\rho_n = \sqrt{x_n^2 + y_n^2}$ and $\theta_n = \arg(x_n, y_n)$. Show if

$$(x_0, y_0) \in \{(x, y) : y \leq 0 < x\},$$

then $y_n > 0$ for some $n \in \mathbb{N}$.

2. (Existence of 2-cycles) Consider the sets

$$C_1 := \{(x, y) : x^2 + y^2 = 1\} \text{ and } C_2 := (x_1, 0) : x_1 \leq a\}.$$

Show that for each $a \in (0, 1)$ there is a point x such that $T_{C_1,C_2}x \neq x$ and $T^2_{C_1,C_2}x = x$. What happens instead if C_2 is merely the singleton $\{(a, 0)\}$?

3. Investigate the behavior of the Douglas–Rachford algorithm applied to two set feasibility problems with one of the sets finite (assume whatever structure you see fit on the other set).

4. (**Very Hard**) Complete the guided exercise (next slide) of Benoist's global convergence proof

Guided Exercise: Benoist's Global Convergence Proof

Consider the Lyapunov candidate function

$$V(x,y) = \frac{1}{2}(y-\lambda)^2 - \lambda \ln(1+\sqrt{1-x^2}) + \lambda\sqrt{1-x^2} + (\lambda - 1)\ln x + \frac{1}{2}x^2.$$

Let $\Delta :=]0,1] \times \mathbb{R}$ and define $G : \Delta \to \Delta$ by

$$G(x,y) := V \circ T - V,$$

where T is the DR operator.

Consider $W : [0,1[\times[0,1[\to \mathbb{R}$ defined using a change of variables on G:

$$W(u,v) := G(a,b) \text{ where } u^2 = 1 - a^2 \text{ and } v^2 = \frac{b^2}{a^2+b^2}.$$

Guided Exercise: Benoist's Global Convergence Proof

Prove the following two lemmas.

Lemma 0

Show that W may be expressed as

$$W(u,v) := A(u) - A(v) + \sqrt{1-u^2}B(v) + \frac{u^2-h^2}{2},$$

where $A(t) := \frac{1+h}{2}\ln(1+t) + \frac{1-h}{2}\ln(1-t) - h$, $B(t) := \frac{t(h-t)}{\sqrt{1-t^2}}$.

Lemma 1

There exists a unique real number μ such that $0 < \mu < h$: (i) B is increasing on $[0,\mu]$ from 0 to $B(\mu)$, and (ii) B is decreasing in $[\mu, 1[$ from $B(\mu)$ to $-\infty$ with $B(h) = 0$.

Hint: Consider $B'(t)$.

Guided Exercise: Benoist's Global Convergence Proof

Prove the following lemma.

Lemma 2

For all $v \in [0, 1[$, we have $W(0, v) < 0$.

Hint: Show that

$$W(0, v) = -\frac{1}{2}h^2 + S(v)h + R(v),$$

where $S(t) := \frac{1}{2}\ln\left(\frac{1-t}{1+t}\right) + \frac{t}{\sqrt{1-t^2}} + t$, $R(t) : -\frac{1}{2}\ln(1-t^2) - \frac{t^2}{\sqrt{1-t^2}}$.
Argue that there exists a unique $v^* < 0.8$ such that $S(v^*) = 1$, and distinguish three cases: (i) $v^* \leq v < 1$, (ii) $0 < v \leq v^*$, and (iii) $v = 0$.

Guided Exercise: Benoist's Global Convergence Proof

Using Lemmas 1 and 2 to prove the following.

Proposition 1.

For all $(u, v) \in [0, 1[\times[0, 1[$ we have

$$W(u, v) \leq 0 \text{ with equality if and only if } u = v = h.$$

Hint: Show that

$$\frac{\partial W(u, v)}{\partial u} > 0 \iff B(u) > B(v).$$

Distinguish four cases: (i) $h \leq v < 1$, (ii) $\mu < v < h$, (iii) $v = \mu$, and (iv) $0 \leq v < \mu$.

Guided Exercise: Benoist's Global Convergence Proof

Using Proposition 1 prove the following.

Proposition 2.

For all $\epsilon > 0$ we have

$$\sup_{(x,y)\in\Delta(\epsilon)} G(x,y) < 0,$$

where $\Delta(\epsilon) := \{(x,y) \in \Delta : d((x,y),(\sqrt{1-h^2},h)) > \epsilon\}$.

Hint: If $\sup_{(x,y)\in\Delta(\epsilon)} G(x,y) \geq 0$, use Proposition 1 to argue the existence of a subsequence such that $W(u_{n_k}, v_{n_k}) = G(x_{n_k}, y_{n_k}) \to 0$ such that $u_{n_k}, v_{n_k} \to (u, v)$ for some u and v.
Distinguish two cases: (i) $u \neq 1$ and $v \neq 1$, (ii) $u = 1$ or $v = 1$.

Guided Exercise: Benoist's Global Convergence Proof

Using Proposition 2 prove the main result.

Theorem (Benoist, 2015)

If $(x_0, y_0) \in \Delta$ then the Douglas–Rachford sequence converges to $(\sqrt{1-h^2}, h)$.

Hint: By telescoping, show that

$$\sum_{n\in\mathbb{N}} G(x_n, y_n)$$

converges and deduce $G(x_n, y_n) \to 0$ which contradicts Proposition 2.

References

- J. Benoist. **The Douglas–Rachford algorithm for the case of the sphere and the line**, *J. Global Optimization*, in press, 2015.
- J.M. Borwein & R.L. Luke. **Duality and Convex Programming**, pp. 229–270 in *Handbook of Mathematical Methods in Imaging*, O. Scherzer (Editor-in-Chief), Springer-Verlag. E-pub. 2010.
- Jonathan M. Borwein & Brailey Sims, **The Douglas-Rachford algorithm in the absence of convexity**. Chapter 6, pp. 93–109 in Fixed-Point Algorithms for Inverse Problems in Science and Engineering in Springer Optimization and Its Applications, vol. 49, 2011. http://www.carma.newcastle.edu.au/~jb616/dr.pdf
- J. M. Borwein and J. Vanderwerff, **Convex Functions: Constructions, Characterizations and Counterexamples**. Encyclopedia of Mathematics and Applications, 109 Cambridge Univ Press, 2010. http://projects.cs.dal.ca/ddrive/ConvexFunctions/
- V. Elser, I. Rankenburg, & P. Thibault, **Searching with iterated maps**," Proc. National Academy of Sciences, 104 (2007), 418–423.

Many resources available at:

http://carma.newcastle.edu.au/DRmethods

5. Applications to Matrix Completion

Those Involved

Note the fluid flow being studied

Fran Aragón Jon Borwein

Matt Tam

Matrix Completion Preliminaries

Many successful non-convex applications of the Douglas–Rachford method can be considered as matrix completion problems (a well studied topic).

In the remainder of this series, we shall focus on recent successful applications of the method to a variety of (real) matrix reconstruction problems.

In particular, consider matrix completion in the context of:

1. Positive semi-definite matrices.
2. Stochastic matrices.
3. Euclidean distance matrices, esp. those in protein reconstruction.
4. Hadamard matrices together with their specialisations.
5. Nonograms – a Japanese number “painting” game.
6. Sudoku – a Japanese number game.

The framework is flexible and there are many other actual and potential applications. Our exposition will highlight the importance of the model.

Matrix Completion Preliminaries

Many successful non-convex applications of the Douglas–Rachford method can be considered as matrix completion problems (a well studied topic).

In the remainder of this series, we shall focus on recent successful applications of the method to a variety of (real) matrix reconstruction problems.

In particular, consider matrix completion in the context of:

1. Positive semi-definite matrices.
2. Stochastic matrices.
3. Euclidean distance matrices, esp. those in protein reconstruction.
4. Hadamard matrices together with their specialisations.
5. Nonograms – a Japanese number "painting" game.
6. Sudoku – a Japanese number game.

The framework is flexible and there are many other actual and potential applications. Our exposition will highlight the importance of the model.

Matrix Completion

From herein, we consider $\mathcal{H} = \mathbb{R}^{m \times n}$ equipped with the trace inner product and induced (Frobenius) norm:

$$\langle A, B \rangle := \mathrm{tr}(A^T B), \quad \|A\|_F := \sqrt{\mathrm{tr}(A^T A)} = \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij}^2}.$$

- A partial 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 $b_{ij} = a_{ij}$.

Abstractly matrix completion is the following:

> Given a partial matrix, find a completion which belongs to some prescribed family of matrices.

Matrix Completion

From herein, we consider $\mathcal{H} = \mathbb{R}^{m\times n}$ equipped with the trace inner product and induced (Frobenius) norm:

$$\langle A, B\rangle := \operatorname{tr}(A^T B), \quad \|A\|_F := \sqrt{\operatorname{tr}(A^T A)} = \sqrt{\sum_{j=1}^{n}\sum_{i=1}^{m} a_{ij}^2}.$$

- A partial 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 $b_{ij} = a_{ij}$.

Abstractly matrix completion is the following:

> Given a partial matrix, find a completion which belongs to some prescribed family of matrices.

Matrix Completion: Example

Suppose the partial matrix $D = (D_{ij}) \in \mathbb{R}^{4\times 4}$ is known to contains the pair-wise distances between four points $x_1, \dots, x_m \in \mathbb{R}^2$. That is,

$$D_{ij} = \|x_i - x_j\|^2.$$

$$D = \begin{pmatrix} 0 & 3.1 & ? & ? \\ 3.1 & 0 & ? & ? \\ ? & ? & 0 & 4.3 \\ ? & ? & 4.3 & 0 \end{pmatrix}$$

four points in $\mathbb{R}^2$

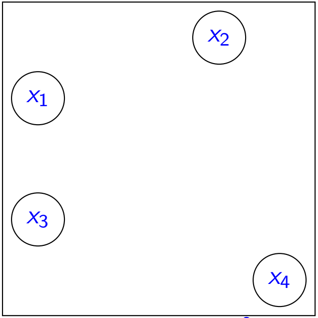
$\longrightarrow$ Reconstruct D from known entries and *a priori* information.

Matrix Completion: Example

Suppose the partial matrix $D = (D_{ij}) \in \mathbb{R}^{4\times 4}$ is known to contains the pair-wise distances between four points $x_1, \dots, x_m \in \mathbb{R}^2$. That is,

$$D_{ij} = \|x_i - x_j\|^2.$$

$$D = \begin{pmatrix} 0 & 3.1 & ? & ? \\ 3.1 & 0 & ? & ? \\ ? & ? & 0 & 4.3 \\ ? & ? & 4.3 & 0 \end{pmatrix}$$



four points in $\mathbb{R}^2$

$\longrightarrow$ Reconstruct D from known entries and *a priori* information.

Matrix Completion: Example

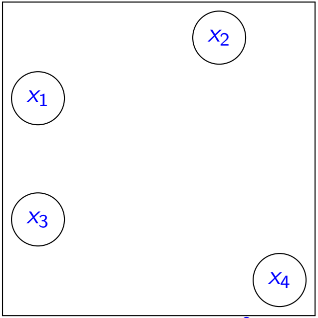
Suppose the partial matrix $D = (D_{ij}) \in \mathbb{R}^{4\times 4}$ is known to contains the pair-wise distances between four points $x_1, \dots, x_m \in \mathbb{R}^2$. That is,

$$D_{ij} = \|x_i - x_j\|^2.$$

$$D = \begin{pmatrix} 0 & 3.1 & \mathbf{?} & \mathbf{?} \\ 3.1 & 0 & \mathbf{?} & \mathbf{?} \\ ? & ? & 0 & 4.3 \\ ? & ? & 4.3 & 0 \end{pmatrix}$$

$$\Downarrow ???$$

$$D = \begin{pmatrix} 0 & 3.1 & \mathbf{2.0} & \mathbf{5} \\ 3.1 & 0 & \mathbf{4.2} & \mathbf{4.1} \\ \mathbf{2.0} & \mathbf{4.2} & 0 & 4.3 \\ \mathbf{5} & \mathbf{4.1} & 4.3 & 0 \end{pmatrix}$$



four points in $\mathbb{R}^2$

$\longrightarrow$ Reconstruct D from known entries and *a priori* information.

Matrix Completion Preliminaries

It is natural to formulate matrix completions as the feasibility problem:

$$\text{find } X \in \bigcap_{i=1}^{N} C_i \subseteq \mathbb{R}^{m\times n}.$$

Let A be the partial matrix to be completed. We (mostly) choose

- C_1 to be the set of all matrix completions of A.
- $C_2, \ldots, C_N$ s.t. their intersection equals the prescribed matrix family.

Let Ω denote the set of indices for the entry in A is known. Then

$$C_1 := \{X \in \mathbb{R}^{m\times n} : X_{ij} = A_{ij} \text{ for all } (i,j) \in \Omega\}.$$

The projection of $X \in \mathbb{R}^{m\times n}$ onto C_1 is given pointwise by

$$P_{C_1}(X)_{ij} = \begin{cases} A_{ij}, & \text{if } (i,j) \in \Omega, \\ X_{ij}, & \text{otherwise.} \end{cases}$$

The remainder of the talk will focus on choosing $C_2, \ldots, C_N$.

Positive Semi-Definite Matrices

Denote the symmetric matrices by $\mathbb{S}^n$, and the positive semi-definite matrices by $\mathbb{S}^n_+$. Our second constraint set is

$$C_2 := \mathbb{S}^n_+ = \{X \in \mathbb{R}^{n\times n} : X = X^T, y^T X y \geq 0 \text{ for all } y \in \mathbb{R}^n\}.$$

The matrix X is a PSD completion of A if and only if $X \in C_1 \cap C_2$.

Theorem (Higham 1986)

For any $X \in \mathbb{R}^{n\times n}$, define $Y = (X + X^T)/2$ and let $Y = UP$ be a polar decomposition of Y (*i.e.*, U unitary, $P \in \mathbb{S}^n_+$.). Then

$$P_{C_2}(X) = \frac{Y+P}{2}.$$

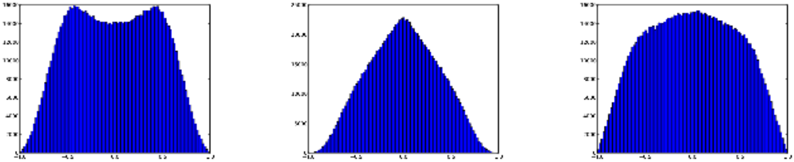
An important class of PSD matrices are the correlation matrices.

Positive Semi-Definite Matrices: Correlation Matrices

For random variables $X_1, X_2, \ldots, X_n$, the ij-th entry of the corresponding correlation matrix contains the correlation between X_i and X_j. This is incorporated into C_1 by enforcing that

$$(i,i) \in \Omega \text{ with } A_{ii} = 1 \text{ for } i = 1, 2, \ldots, n. \tag{4}$$

Moreover, whenever (4) holds for a matrix its entries are necessarily contained in $[-1, 1]$.
Apply this formulation for different starting points yields:



$X_0 := Y.$ $\quad X_0 := \frac{1}{2}(Y + Y^T) \in S_5.$ $\quad X_0 := YY^T \in S_5.$

Figure. Distribution of entries for correlation matrices generated by choosing different initial points. Y is a random matrix in $[-1, 1]^{5\times 5}$.

Stochastic matrices

Recall that a matrix $A = (A_{ij}) \in \mathbb{R}^{m \times n}$ is said to be doubly stochastic if

$$\sum_{i=1}^{m} A_{ij} = \sum_{j=1}^{n} A_{ij} = 1, A_{ij} \geq 0. \tag{5}$$

These matrices describe the transitions of a Markov chain (in this case $m = n$), amongst other things. We use the following constraint sets

$$C_2 := \left\{ X \in \mathbb{R}^{m \times n} | \sum_{i=1}^{m} X_{ij} = 1 \text{ for } j = 1, \dots, n \right\},$$

$$C_3 := \left\{ X \in \mathbb{R}^{m \times n} | \sum_{j=1}^{n} X_{ij} = 1 \text{ for } i = 1, \dots, m \right\},$$

$$C_4 := \{ X \in \mathbb{R}^{m \times n} | X_{ij} \geq 0 \text{ for } i = 1, \dots, m \text{ and } j = 1, \dots, n \}.$$

The matrix X is a double stochastic matrix completing A if and only if

$$X \in C_1 \cap C_2 \cap C_3 \cap C_4.$$

Stochastic matrices

Recall that a matrix $A = (A_{ij}) \in \mathbb{R}^{m \times n}$ is said to be doubly stochastic if

$$\sum_{i=1}^{m} A_{ij} = \sum_{j=1}^{n} A_{ij} = 1, A_{ij} \geq 0. \tag{5}$$

These matrices describe the transitions of a Markov chain (in this case $m = n$), amongst other things. We use the following constraint sets

$$C_2 := \left\{ X \in \mathbb{R}^{m \times n} | \sum_{i=1}^{m} X_{ij} = 1 \text{ for } j = 1, \ldots, n \right\},$$

$$C_3 := \left\{ X \in \mathbb{R}^{m \times n} | \sum_{j=1}^{n} X_{ij} = 1 \text{ for } i = 1, \ldots, m \right\},$$

$$C_4 := \{ X \in \mathbb{R}^{m \times n} | X_{ij} \geq 0 \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n \}.$$

The matrix X is a double stochastic matrix completing A if and only if

$$X \in C_1 \cap C_2 \cap C_3 \cap C_4.$$

Stochastic matrices

$$C_2 := \left\{ X \in \mathbb{R}^{m \times n} | \sum_{i=1}^{m} X_{ij} = 1 \text{ for } j = 1, \dots, n \right\},$$
$$C_4 := \{ X \in \mathbb{R}^{m \times n} | X_{ij} \geq 0 \text{ for } i = 1, \dots, m \text{ and } j = 1, \dots, n \}.$$

Denote $\mathbf{e} = (1, 1, \dots, 1) \in \mathbb{R}^m$. Since C_2 applies to each column independently, a column-wise formula for P_{C_2} is given by

$$P_E(x) = x + \frac{1}{m}\left(1 - \sum_{i=1}^{m} x_j\right)\mathbf{e} \quad \text{where} \quad E := \{x \in \mathbb{R}^m : \mathbf{e}^T x = 1\}.$$

The projection of X onto C_4 is given pointwise by

$$P_{C_4}(X)_{ij} = \max\{0, X_{ij}\}.$$

- Singly stochastic matrix completion can be consider by dropping C_3.
- Related work of Thakouda applies Dykstra's algorithm to a two set model. The corresponding projections are less straight-forward.

Hadamard Matrices

A matrix $H = (H_{ij}) \in \{-1, 1\}^{n \times n}$ is said to be a Hadamard matrix of order n if [1]

$$H^T H = nI.$$

A classical result of Hadamard asserts that Hadamard matrices exist only if $n = 1, 2$ or a multiple of 4. For orders 1 and 2, such matrices are easy to find. For example,

$$\begin{bmatrix} 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}.$$

The (open) Hadamard conjecture is concerned with the converse:

There exists a Hadamard matrices of order $4n$ for all $n \in \mathbb{N}$.

[1] There are many equivalent characterizations and many local experts.

Hadamard Matrices

A matrix $H = (H_{ij}) \in \{-1, 1\}^{n \times n}$ is said to be a Hadamard matrix of order n if [1]

$$H^T H = nI.$$

A classical result of Hadamard asserts that Hadamard matrices exist only if $n = 1, 2$ or a multiple of 4. For orders 1 and 2, such matrices are easy to find. For example,

$$\begin{bmatrix} 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}.$$

The (open) Hadamard conjecture is concerned with the converse:

> There exists a Hadamard matrices of order $4n$ for all $n \in \mathbb{N}$.

[1] There are many equivalent characterizations and many local experts.

Hadamard Matrices

Consider now the problem of finding a Hadamard matrix of a given order – an important completion problem with structure restriction but no fixed entries. We use the following constraint sets:

$$C_1 := \{X \in \mathbb{R}^{n\times n} | X_{ij} = \pm 1 \text{ for } i,j = 1,\ldots,n\},$$
$$C_2 := \{X \in \mathbb{R}^{n\times n} | X^T X = nI\}.$$

Then X is a Hadamard matrix if and only if $X \in C_1 \cap C_2$.

The projection of X on C_1 is given by pointwise rounding to ± 1.

Proposition (A projection onto C_2)

Let $X = USV^T$ be a singular value decomposition. Then

$$\sqrt{n}UV^T \in P_{C_2}(X).$$

Hadamard Matrices

Consider now the problem of finding a Hadamard matrix of a given order – an important completion problem with structure restriction but no fixed entries. We use the following constraint sets:

$$C_1 := \{X \in \mathbb{R}^{n\times n} | X_{ij} = \pm 1 \text{ for } i,j = 1,\dots,n\},$$
$$C_2 := \{X \in \mathbb{R}^{n\times n} | X^T X = nI\}.$$

Then X is a Hadamard matrix if and only if $X \in C_1 \cap C_2$.

The projection of X on C_1 is given by pointwise rounding to ± 1.

Proposition (A projection onto C_2)

Let $X = USV^T$ be a singular value decomposition. Then

$$\sqrt{n}UV^T \in P_{C_2}(X).$$

Hadamard Matrices

Let H_1 and H_2 be Hadamard matrices. We say H_1 are H_2 are:

- Distinct if $H_1 \neq H_2$,
- Equivalent if H_2 can be obtained from H_1 by performing row/column permutations, and/or multiplying rows/columns by -1.

For order $4n$:

- Number of Distinct Hadamard matrices is OEIS A206712:

$$768, 4954521600, 20251509535014912000, \ldots$$

- Number of Inequivalent Hadamard matrices is OEIS A00729:

$$1, 1, 1, 1, 5, 3, 60, 487, 13710027, \ldots$$

With increasing order, the number of Hadamard matrices is a faster than exponentially decreasing proportion of total number of ± 1–matrices (there are 2^{n^2} ± 1–matrices or order n).

Hadamard Matrices

Let H_1 and H_2 be Hadamard matrices. We say H_1 are H_2 are:

- Distinct if $H_1 \neq H_2$,
- Equivalent if H_2 can be obtained from H_1 by performing row/column permutations, and/or multiplying rows/columns by -1.

For order $4n$:

- Number of Distinct Hadamard matrices is OEIS A206712:

$$768, 4954521600, 20251509535014912000, \dots$$

- Number of Inequivalent Hadamard matrices is OEIS A00729:

$$1, 1, 1, 1, 5, 3, 60, 487, 13710027, \dots$$

With increasing order, the number of Hadamard matrices is a faster than exponentially decreasing proportion of total number of ± 1–matrices (there are 2^{n^2} ± 1–matrices or order n).

Hadamard Matrices

Let H_1 and H_2 be Hadamard matrices. We say H_1 are H_2 are:

- Distinct if $H_1 \neq H_2$,
- Equivalent if H_2 can be obtained from H_1 by performing row/column permutations, and/or multiplying rows/columns by -1.

For order $4n$:

- Number of Distinct Hadamard matrices is OEIS A206712:

$$768, 4954521600, 20251509535014912000, \dots$$

- Number of Inequivalent Hadamard matrices is OEIS A00729:

$$1, 1, 1, 1, 5, 3, 60, 487, 13710027, \dots$$

With increasing order, the number of Hadamard matrices is a faster than exponentially decreasing proportion of total number of ± 1–matrices (there are 2^{n^2} ± 1–matrices or order n).

Hadamard Matrices

Table: Number of Hadamard matrices found from 1000 instances

Order	$C_1 \cap C_2$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	1.1371	534	8	1
4	1.0791	627	422	1
8	0.7368	996	996	1
12	7.1298	0	0	0
16	9.4228	0	0	0
20	20.6674	0	0	0

Checking if two Hadamard matrices are equivalent can be cast as a problem of graph isomorphism (McKay '79).

- In Sage use `is_isomorphic(graph1,graph2)`.

Hadamard Matrices

We give an alternative formulation. Define:

$$C_1 := \{X \in \mathbb{R}^{n\times n} | X_{ij} = \pm 1 \text{ for } i,j = 1,\dots,n\},$$
$$C_3 := \{X \in \mathbb{R}^{n\times n} | X^T X = \|X\|_F I\}.$$

Then X is a Hadamard matrix if and only if $X \in C_1 \cap C_2 = C_1 \cap C_3$.

Proposition (A projection onto C_3)

Let $X = USV^T$ be a singular value decomposition. Then

$$\sqrt{\|X\|_F}\, UV^T \in P_{C_3}(X).$$

Hadamard Matrices

Table: Number of Hadamard matrices found from 1000 instances

Order	$C_1 \cap C_2$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	1.1371	534	8	1
4	1.0791	627	422	1
8	0.7368	996	996	1
12	7.1298	**0**	**0**	0
16	9.4228	0	0	0
20	20.6674	0	0	0

Order	$C_1 \cap C_3$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	1.1970	505	8	1
4	0.2647	921	541	1
8	0.0117	1000	1000	1
12	0.8337	**1000**	**1000**	1
16	11.7096	16	16	4
20	22.6034	0	0	0

- A more obvious formulation is can be less effective.

Hadamard Matrices

Table: Number of Hadamard matrices found from 1000 instances

Order	$C_1 \cap C_2$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	1.1371	534	8	1
4	1.0791	627	422	1
8	0.7368	996	996	1
12	7.1298	**0**	**0**	0
16	9.4228	0	0	0
20	20.6674	0	0	0

Order	$C_1 \cap C_3$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	1.1970	505	8	1
4	0.2647	921	541	1
8	0.0117	1000	1000	1
12	0.8337	**1000**	**1000**	1
16	11.7096	16	16	4
20	22.6034	0	0	0

- A more obvious formulation is can be less effective.

Skew-Hadamard Matrices

Recall that a matrix $X \in \mathbb{R}^{n\times n}$ is skew-symmetric if $X^T = -X$. A skew-Hadamard matrix is a Hadamard matrix H such that $(I - H)$ is skew-symmetric. That is,

$$H + H^T = 2I.$$

Skew-Hadamard matrices are of interest, for example, in the construction of various combinatorial designs. The number of inequivalent skew-Hadamard matrices of order $4n$ is OEIS A001119 (for $n = 2, 3, \dots$):

$$1, 1, 2, 2, 16, 54, \dots$$

It is convenient to redefine the constraint C_1 to be

$$C_1 = \{X \in \mathbb{R}^{n\times n} | X + X^T = 2I,\ X_{ij} = \pm 1 \text{ for } i, j = 1, \dots, n\}.$$

A projection of X onto C_1 is given pointwise by

$$P_{C_1}(X) = \begin{cases} -1 & \text{if } i \neq j \text{ and } X_{ij} < X_{ji}, \\ 1 & \text{otherwise.} \end{cases}$$

Skew-Hadamard Matrices

Table: Number of skew-Hadamard matrices found from 1000 instances

Order	$C_1 \cap C_2$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	0.0003	1000	2	1
4	1.1095	719	16	1
8	0.7039	902	889	1
12	14.1835	43	43	1
16	19.3462	0	0	0
20	29.0383	0	0	0

Order	$C_1 \cap C_3$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	0.0004	1000	2	1
4	1.6381	634	16	1
8	0.0991	986	968	1
12	0.0497	999	999	1
16	0.2298	1000	1000	2
20	20.0296	495	495	2

- Adding constraints can help.

Skew-Hadamard Matrices

Table: Number of skew-Hadamard matrices found from 1000 instances

Order	$C_1 \cap C_2$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	0.0003	1000	2	1
4	1.1095	719	16	1
8	0.7039	902	889	1
12	14.1835	43	43	1
16	19.3462	0	0	0
20	29.0383	0	0	0

Order	$C_1 \cap C_3$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	0.0004	1000	2	1
4	1.6381	634	16	1
8	0.0991	986	968	1
12	0.0497	999	999	1
16	0.2298	1000	1000	2
20	20.0296	495	495	2

- Adding constraints can help.

Sudoku Puzzles

In Sudoku the player fills entries of an incomplete Latin square subject to the constraints:

- Each row contains the numbers 1 through 9 exactly once.
- Each column contains the numbers 1 through 9 exactly once.
- Each 3×3 sub-block contains the numbers 1 through 9 exactly once.

		5	3					
8							2	
	7			1		5		
4					5	3		
	1			7				6
		3	2				8	
	6		5					9
		4					3	
					9	7		

1	4	5	3	2	7	6	9	8
8	3	9	6	5	4	1	2	7
6	7	2	9	1	8	5	4	3
4	9	6	1	8	5	3	7	2
2	1	8	4	7	3	9	5	6
7	5	3	2	9	6	4	8	1
3	6	7	5	4	2	1	8	9
9	8	4	7	6	1	2	3	5
5	2	1	8	3	9	7	6	4

Figure. An incomplete Sudoku (left) and its unique solution (right).

- The Douglas–Rachford algorithm applied to the natural integer feasibility problem fails (exception: $n^2 \times n^2$ Sudokus where $n = 1, 2$).

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

> The idea: Reformulate integer entries as binary vectors.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

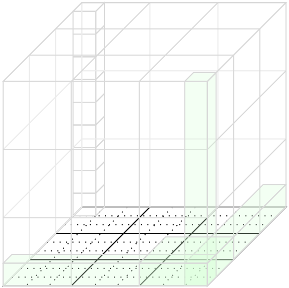
[5] Veit Elser was the first to realise the usefulness of this binary formulation for

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

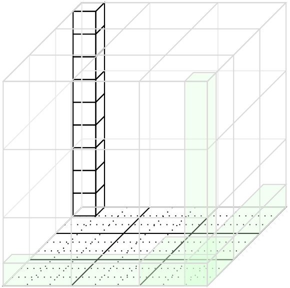
A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^9 \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times9\times9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

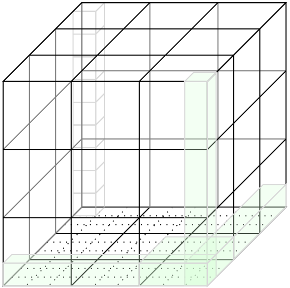
A solution is any $X \in \bigcap_{i=1}^5 C_i$.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$$C_1 = \{X : X_{ij} \in E\}$$
$$C_2 = \{X : X_{ik} \in E\}$$
$$C_3 = \{X : X_{jk} \in E\}$$
$$C_4 = \{X : \text{vec}(3\times 3 \text{ submatrix}) \in E\}$$
$$C_5 = \{X : X \text{ matches original puzzle}\}$$

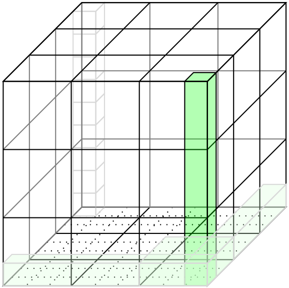
A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times9\times9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

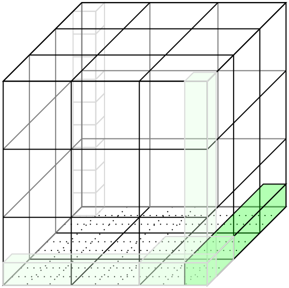
A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times9\times9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

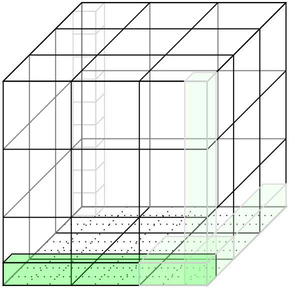
A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times9\times9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

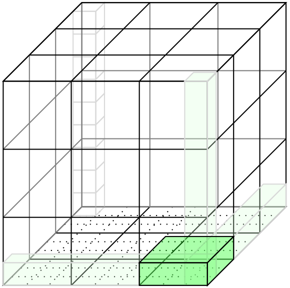
A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

Sudoku Puzzles: Computing projections

Proposition (projections onto permutation sets)

Denote by $\mathcal{C} \subset \mathbb{R}^m$ the set of all vector whose entries are permutations of $c_1, c_2, \dots, c_m \in \mathbb{R}$. Then for any $x \in \mathbb{R}^m$,

$$P_{\mathcal{C}} x = [\mathcal{C}]_x,$$

where $[\mathcal{C}]_x$ is the set of vectors $y \in \mathcal{C}$ such that ith largest index of y has the same index in y as the ith largest entry of x, for all indices i.

- $[\mathcal{C}]_x$ be computed efficiently using sorting algorithms.
- Choosing $c_1 = 1$ and $c_2 = \cdots = c_m = 0$ gives[2]

$$P_E x = \{e_i : x_i = \max\{x_1, \dots, x_m\}\}.$$

  Formulae for $P_{C_1}, P_{C_2}, P_{C_3}$ and P_{C_4} easily follow.
- P_{C_5} is given by setting the entries corresponding to those in the incomplete puzzle to 1, and leaving the remaining untouched.

[2] A direct proof of this special case appears in Jason Schaad's Masters thesis.

Sudoku Puzzles:Algorithm Details

1. Initialize: $x_0 := (y, y, y, y, y) \in D$ for some random $y \in [0, 1]^{9\times9\times9}$.
2. Iteration: By setting

$$x_{n+1} := T_{D,C}x_n = \frac{x_n + R_C R_D x_n}{2}.$$

3. Termination: Either if a solution is found, or 10000 iteration have been performed. More precisely, $\text{round}(P_D x_n)$ ($P_D x_n$ pointwise rounded to the nearest integer) is a solution if

$$\text{round}(P_D x_n) \in C \cap D.$$

Taking $\text{round}(\cdot)$ is valid since the solution is binary.

Sudoku Puzzles: An Experiment

We consider the following test libraries frequently used by programmers to test their solvers.

1. Dukuso's top95 and top1465.
2. First 1000 puzzles from Gordan Royle's minimum Sudoku – puzzles with 17 entries (the best known lower bound on the entries required for a unique solution).
3. reglib-1.3 – 1000 test puzzle suited to particular human style techniques.
4. ksudoku16 and ksudoku25 – a collection around 30 instances (various difficulties) generated with *KSudoku*. Contains larger 16×16 and 25×25 puzzles.[3]

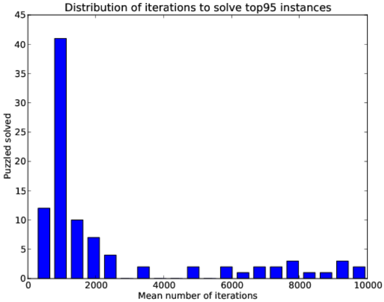
[3] Generating "hard" instances is a difficult problem.

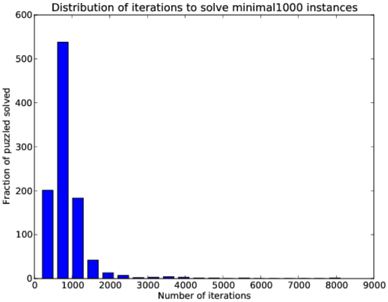
Computational Results: Success Rate

From 10 random replications of each puzzle:

Table. % Solved by the Douglas–Rachford method

top95	top1465	reglib-1.3	minimal1000	ksudoku16	ksudoku25
86.53	93.69	99.35	99.59	92	100





- If a instance was solved, the solution was usually found within the first 2000 iterations.

Computational Example: A 'Nasty' Sudoku

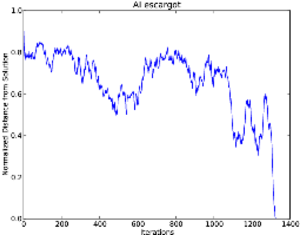
This 'nasty' Sudoku[4] cannot be solved reliably (20.2% success rate) by the Douglas–Rachford method.

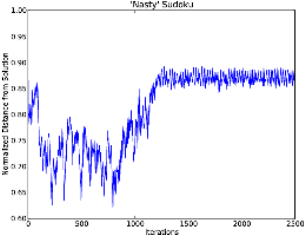
7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Other "difficult" Sudoku puzzles do not cause the Douglas–Rachford method any trouble.

- AI escargot = 98.5% success rate.

Figure. Distance to the solution by iterations





[4] This is a modified version of an example due to Veit Elser.

Computational Example: A 'Nasty' Sudoku

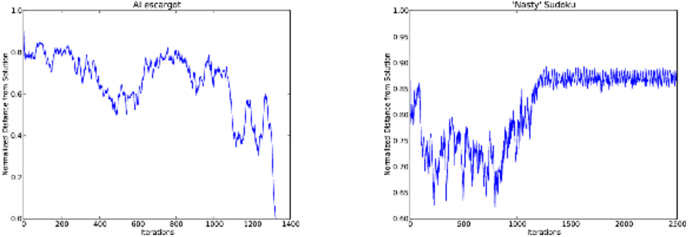
This 'nasty' Sudoku[4] cannot be solved reliably (20.2% success rate) by the Douglas–Rachford method.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Other "difficult" Sudoku puzzles do not cause the Douglas–Rachford method any trouble.

- AI escargot = 98.5% success rate.

Figure. Distance to the solution by iterations



[4] This is a modified version of an example due to Veit Elser.

Computational Example: A ‘Nasty’ Sudoku

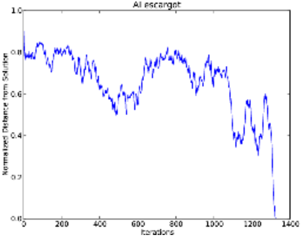
This ‘nasty’ Sudoku[4] cannot be solved reliably (20.2% success rate) by the Douglas–Rachford method.

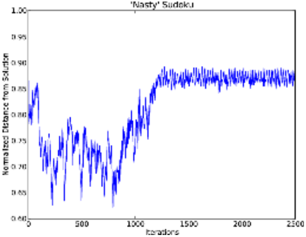
7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Other “difficult” Sudoku puzzles do not cause the Douglas–Rachford method any trouble.

- AI escargot = 98.5% success rate.

Figure. Distance to the solution by iterations





[4]This is a modified version of an example due to Veit Elser.

Computational Example: A 'Nasty' Sudoku

We considered solving the puzzles obtained by removing any single entry from the 'Nasty' Sudoku.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Success rate when any single entry is removed:

- Top left 7 = 24%
- Any other entry = 99%

Number of solutions when any single entry is removed:

- Top left 7 = 5
- Any other entry = 200–3800

Is the Douglas–Rachford method hindered by an abundance of 'near' solutions?

Computational Example: A 'Nasty' Sudoku

We considered solving the puzzles obtained by removing any single entry from the 'Nasty' Sudoku.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Success rate when any single entry is removed:

- Top left 7 = 24%
- Any other entry = 99%

Number of solutions when any single entry is removed:

- Top left 7 = 5
- Any other entry = 200–3800

Is the Douglas–Rachford method hindered by an abundance of 'near' solutions?

Computational Example: A 'Nasty' Sudoku

We considered solving the puzzles obtained by removing any single entry from the 'Nasty' Sudoku.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Success rate when any single entry is removed:

- Top left 7 = 24%
- Any other entry = 99%

Number of solutions when any single entry is removed:

- Top left 7 = 5
- Any other entry = 200–3800

Is the Douglas–Rachford method hindered by an abundance of 'near' solutions?

Computational Results: Performance Comparison

We compared the Douglas–Rachford method to the following solvers:

1. Gurobi binary program – Solves the same binary model using integer programming techniques.
2. YASS (Yet another Sudoku solver) – First applies a reasoning algorithm to determine possible candidates for each empty square. If this does not completely solve the puzzle, a deterministic recursive algorithm is used.
3. DLX – Solves an exact cover formulation using the *Dancing Links* implementation of Knuth's *Algorithm X* (non-deterministic, depth-first, back-tracking).

Table. Average Runtime (seconds).[5]

	top95	reglib-1.3	minimal1000	ksudoku16	ksudoku25
DR	1.432	0.279	0.509	5.064	4.011
Gurobi	0.063	0.059	0.063	0.168	0.401
YASS	2.256	0.039	0.654	-	-
DLX	1.386	0.105	3.871	-	-

[5] Some solvers are only designed to handle 9×9 puzzles.

Nonograms

A nonogram puzzle consists of a blank $m \times n$ grid of "pixels" together with $(m+n)$ cluster-size sequences (*i.e.*, for each row and each column). The goal is to "paint" the canvas with a picture such that:

1. Each pixel must be either black or white.
2. If a row (resp. column) has a cluster-size sequences $s_1, \ldots, s_k$ then it must contain k cluster of black pixels, each separated by at least one white pixel. The ith leftmost (resp. uppermost) cluster contains s_i black pixels.

						1			
			2			4	1	2	2
2	3	1	1	5	4	1	5	2	1

1	2										
	2										
	1										
	1										
	2										
2	4										
2	6										
	8										
1	1										
2	2										

Nonograms

A nonogram puzzle consists of a blank $m \times n$ grid of "pixels" together with $(m+n)$ cluster-size sequences (*i.e.*, for each row and each column). The goal is to "paint" the canvas with a picture such that:

1. Each pixel must be either black or white.
2. If a row (resp. column) has a cluster-size sequences $s_1, \dots, s_k$ then it must contain k cluster of black pixels, each separated by at least one white pixel. The ith leftmost (resp. uppermost) cluster contains s_i black pixels.

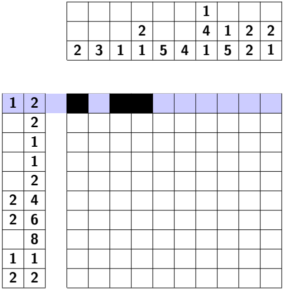
						1			
			2			4	1	2	2
2	3	1	1	5	4	1	5	2	1

1	2
	2
	1
	1
	2
2	4
2	6
	8
1	1
2	2

Nonograms

A nonogram puzzle consists of a blank $m \times n$ grid of "pixels" together with $(m+n)$ cluster-size sequences (*i.e.*, for each row and each column). The goal is to "paint" the canvas with a picture such that:

1. Each pixel must be either black or white.
2. If a row (resp. column) has a cluster-size sequences $s_1, \ldots, s_k$ then it must contain k cluster of black pixels, each separated by at least one white pixel. The ith leftmost (resp. uppermost) cluster contains s_i black pixels.

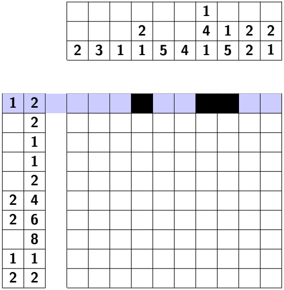


Legal row.

Nonograms

A nonogram puzzle consists of a blank $m \times n$ grid of "pixels" together with $(m+n)$ cluster-size sequences (*i.e.*, for each row and each column). The goal is to "paint" the canvas with a picture such that:

1. Each pixel must be either black or white.
2. If a row (resp. column) has a cluster-size sequences $s_1, \ldots, s_k$ then it must contain k cluster of black pixels, each separated by at least one white pixel. The ith leftmost (resp. uppermost) cluster contains s_i black pixels.



Legal row.

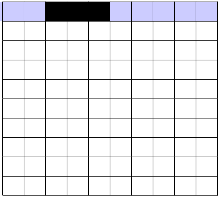
Nonograms

A nonogram puzzle consists of a blank $m \times n$ grid of "pixels" together with $(m+n)$ cluster-size sequences (*i.e.*, for each row and each column). The goal is to "paint" the canvas with a picture such that:

1. Each pixel must be either black or white.
2. If a row (resp. column) has a cluster-size sequences $s_1, \dots, s_k$ then it must contain k cluster of black pixels, each separated by at least one white pixel. The ith leftmost (resp. uppermost) cluster contains s_i black pixels.

						1			
			2			4	1	2	2
2	3	1	1	5	4	1	5	2	1

1	2
	2
	1
	1
	2
2	4
2	6
	8
1	1
2	2



Illegal row.

Nonograms

We model nonograms as a binary feasibility problem. The $m \times n$ grid is represented as a matrix $A \in \mathbb{R}^{m\times n}$ with

$$A[i,j] = \begin{cases} 0 & \text{if the } (i,j)\text{-th entry of the grid is white,} \\ 1 & \text{if the } (i,j)\text{-th entry of the grid is black.} \end{cases}$$

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). The constraints are:

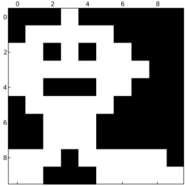
$$C_1 = \{A : A[i,:] \in \mathcal{R}_i \text{ for } i = 1,\dots,m\},$$
$$C_2 = \{A : A[:,j] \in \mathcal{C}_j \text{ for } j = 1,\dots,n\}.$$

Given an incomplete nonogram puzzle, A is a solution if and only if

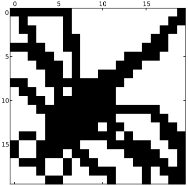
$$A \in C_1 \cap C_2.$$

Nonograms: Computational Results

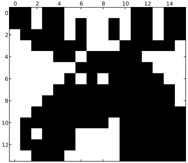
From 1000 random replications, the following nonograms were solved in every instance.



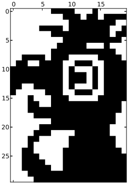
A spaceman.



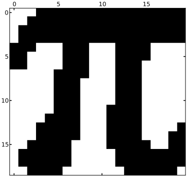
A dragonfly.



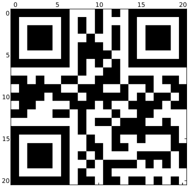
A moose.



A parrot.



The number π.



“Hello from CARMA”.

Nonograms: Computational Details

- Computing the projections onto C_1 and C_2 is difficult.
- We do not know an efficient way to do so.
  - Our approach: Pre-compute all legal cluster size sequences (slow).
- Only a few Douglas–Rachford iterations are required to solve (fast).

In contrast other problems, frequently, have relatively simple projections but require many more iterations.

This suggests the following:

> Trade-off between simplicity of projection operators and the number of iterations required.

Nonograms: Computational Details

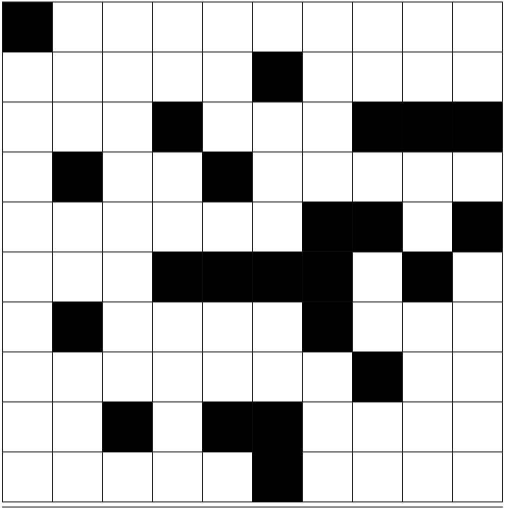
- Computing the projections onto C_1 and C_2 is difficult.
- We do not know an efficient way to do so.
  - Our approach: Pre-compute all legal cluster size sequences (slow).
- Only a few Douglas–Rachford iterations are required to solve (fast).

In contrast other problems, frequently, have relatively simple projections but require many more iterations.

This suggests the following:

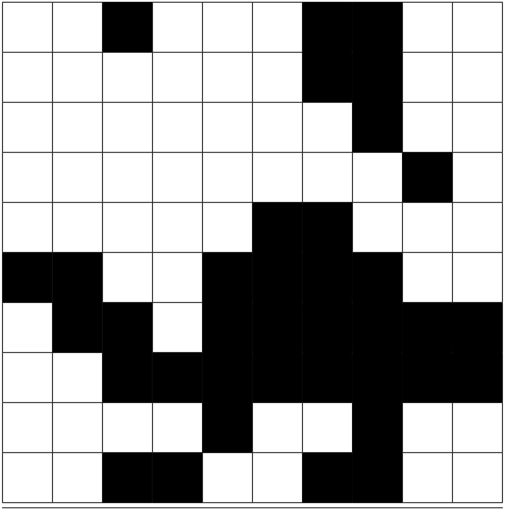
> Trade-off between simplicity of projection operators and the number of iterations required.

Nonograms: An example



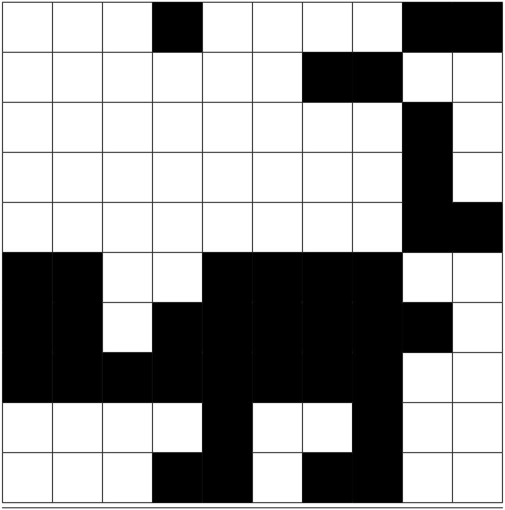
Iteration: 0 (random initialisation)

Nonograms: An example



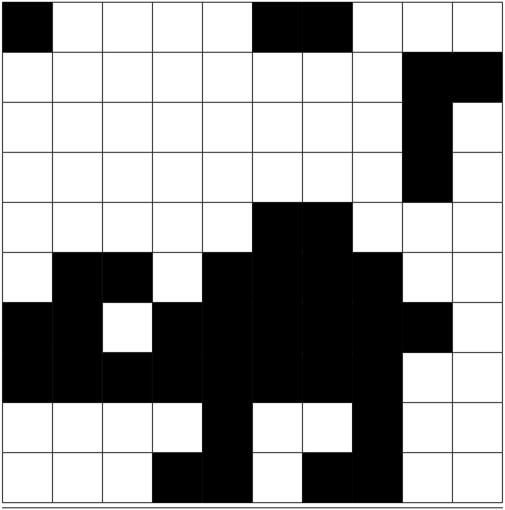
Iteration: 1

Nonograms: An example



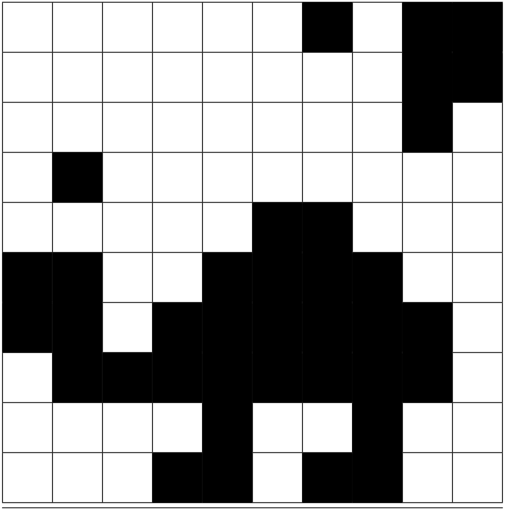
Iteration: 2

Nonograms: An example



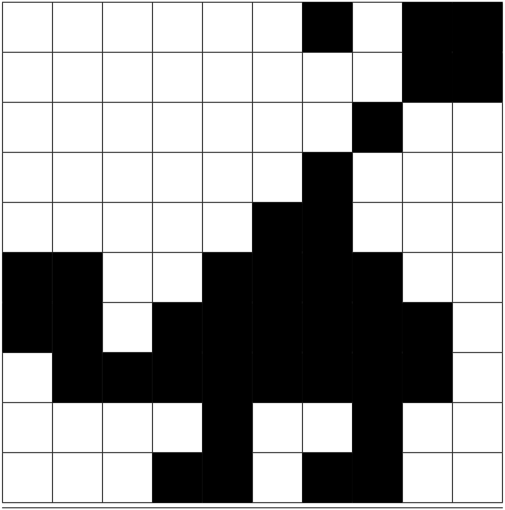
Iteration: 3

Nonograms: An example



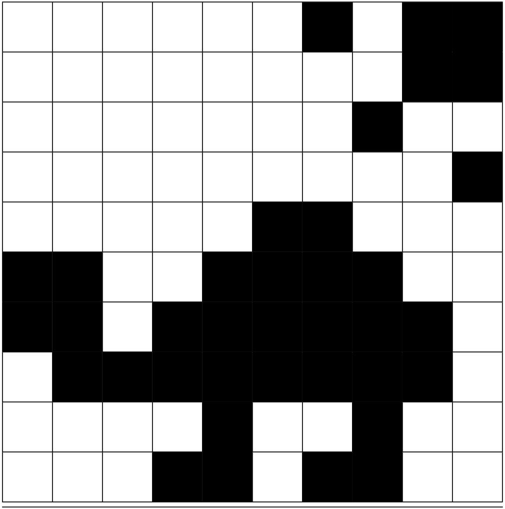
Iteration: 4

Nonograms: An example



Iteration: 5

Nonograms: An example



Iteration: 6 (solved)

GCHQ's 2015 Christmas Puzzle

GCHQ

WHO WE ARE   WHAT WE DO   HOW WE WORK   CESG   CAREERS
PRESS & MEDIA

You are here > Home > Press & media > News & features > A Christmas card with a cryptographic twist for charity

A Christmas card with a cryptographic twist for charity

News article - 7 Dec 2015

This year, along with his traditional Christmas cards, Director GCHQ Robert Hannigan is including a brain-teasing puzzle that seems certain to exercise the grey matter of participants over the holiday season.

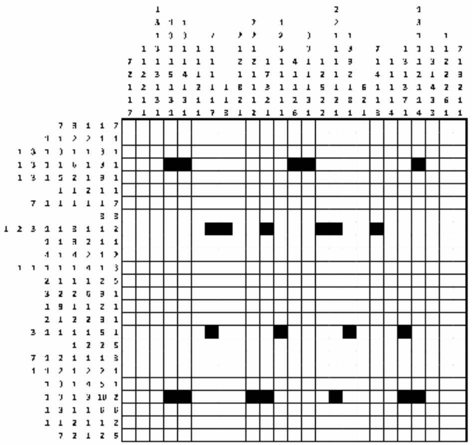
The card, which features the 'Adoration of the Shepherds' by a pupil of Rembrandt, includes traditional Christmas greetings from Director on behalf of the department. However, unlike previous years, the 2015 card will contain a grid-shading puzzle and instructions on how it should be completed. By solving this first puzzle players will create an image that leads to a series of increasingly complex challenges.

Once all stages have been unlocked and completed successfully, players are invited to submit their answer via a given GCHQ email address by 31 January 2016. The winner will then be drawn from all the successful entries and notified soon after. Players are invited to make a donation to the National Society for the Prevention of Cruelty to Children, if they have enjoyed the puzzle.

People who enjoy puzzles, but who are not yet on Director's Christmas card list, need not worry. The first puzzle can be seen below.

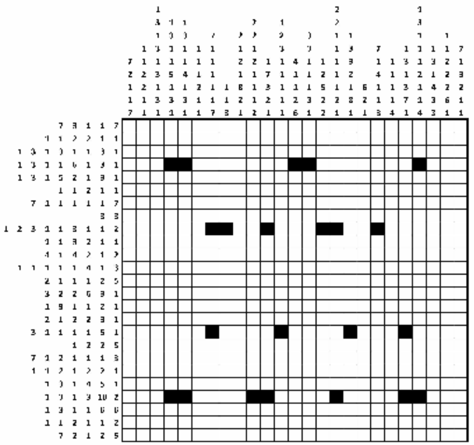
[5] Kudos to Veit Elser who made us aware of the puzzle.

GCHQ's 2015 Christmas Puzzle



```
========= DR Nonogram Solver =========
Precomputing row/column clusters...
Precomputing done!
Time spent precomputing: 33.9s

Running DR...
Solution found!
Iterations: 10
Time spent running DR: 9.9s

Total time: 43.8s
======================================
```

GCHQ's 2015 Christmas Puzzle



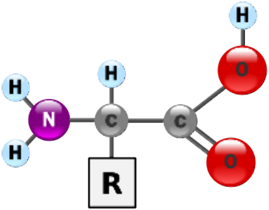
```
========= DR Nonogram Solver =========
Precomputing row/column clusters...
Precomputing done!
Time spent precomputing: 33.9s

Running DR...
Solution found!
Iterations: 10
Time spent running DR: 9.9s

Total time: 43.8s
======================================
```

GCHQ's 2015 Christmas Puzzle

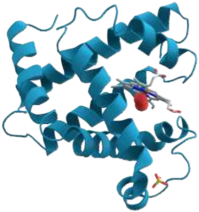
The solution is a QR code which directs to the following website.

Protein Conformation Determination and EDMs

Proteins are large biomolecules comprising of multiple amino acid chains.



Generic amino acid



Myoglobin

They participate in virtually every cellular process, and knowledge of structural conformation gives insights into the mechanisms by which they perform.

Protein Conformation Determination and EDMs

One technique that can be used to determine conformation is nuclear magnetic resonance (NMR) spectroscopy. However, NMR is only able to resolve short inter-atomic distances (*i.e.*, < 6Å). For 1PTQ (404 atoms) this corresponds to $< 8\%$ of the total inter-atomic distances.

We say $D = (D_{ij}) \in \mathbb{R}^{m \times m}$ is a Euclidean distance matrix (EDM) if there exists points $p_1, \dots, p_m \in \mathbb{R}^q$ such that

$$D_{ij} = \|p_i - p_j\|^2.$$

When this holds for points in $\mathbb{R}^q$, we say that D is embeddable in $\mathbb{R}^q$.

We formulate protein reconstruction as a matrix completion problem:

Find a EDM, embeddable in $\mathbb{R}^s$ where $s := 3$,
knowing only short inter-atomic distances.

Protein Conformation Determination and EDMs

One technique that can be used to determine conformation is nuclear magnetic resonance (NMR) spectroscopy. However, NMR is only able to resolve short inter-atomic distances (*i.e.*, < 6Å). For 1PTQ (404 atoms) this corresponds to $< 8\%$ of the total inter-atomic distances.

We say $D = (D_{ij}) \in \mathbb{R}^{m \times m}$ is a Euclidean distance matrix (EDM) if there exists points $p_1, \ldots, p_m \in \mathbb{R}^q$ such that

$$D_{ij} = \|p_i - p_j\|^2.$$

When this holds for points in $\mathbb{R}^q$, we say that D is embeddable in $\mathbb{R}^q$.

We formulate protein reconstruction as a matrix completion problem:

Find a EDM, embeddable in $\mathbb{R}^s$ where $s := 3$,
knowing only short inter-atomic distances.

Protein Conformation Determination and EDMs

One technique that can be used to determine conformation is nuclear magnetic resonance (NMR) spectroscopy. However, NMR is only able to resolve short inter-atomic distances (*i.e.*, < 6Å). For 1PTQ (404 atoms) this corresponds to $< 8\%$ of the total inter-atomic distances.

We say $D = (D_{ij}) \in \mathbb{R}^{m\times m}$ is a Euclidean distance matrix (EDM) if there exists points $p_1, \dots, p_m \in \mathbb{R}^q$ such that

$$D_{ij} = \|p_i - p_j\|^2.$$

When this holds for points in $\mathbb{R}^q$, we say that D is embeddable in $\mathbb{R}^q$.

We formulate protein reconstruction as a matrix completion problem:

Find a EDM, embeddable in $\mathbb{R}^s$ where $s := 3$,
knowing only short inter-atomic distances.

A Feasibility Problem Formulation

Denote by Q the Householder matrix defined by

$$Q := I - \frac{2vv^T}{v^Tv}, \text{where } v = \left[1, 1, \dots, 1, 1 + \sqrt{m}\right]^T \in \mathbb{R}^m.$$

Theorem (Hayden–Wells 1988)

A nonnegative, symmetric, hollow matrix X, is a EDM iff $\widehat{X} \in \mathbb{R}^{(m-1)\times(m-1)}$ in

$$Q(-X)Q = \begin{bmatrix} \widehat{X} & d \\ d^T & \delta \end{bmatrix} \qquad (*)$$

is positive semi-definite (PSD). In this case, X is embeddable in $\mathbb{R}^q$ where $q = \operatorname{rank}(\widehat{X}) \leq m-1$ but not in $\mathbb{R}^{q-1}$.

Let D denote the partial EDM (obtained from NMR), and $\Omega \subset \mathbb{N} \times \mathbb{N}$ the set of indices for known entries. The problem of low-dimensional EDM reconstruction can thus be case as a feasibility problem with constraints:

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

$$C_2 = \{X \in \mathbb{R}^{m\times m} : \widehat{X} \text{ in } (*) \text{ is PSD with } \operatorname{rank} \widehat{X} \leq s := 3\}.$$

A Feasibility Problem Formulation

Recall the constraint sets:

$$C_1 = \{X \in \mathbb{R}^{m\times m} : X \geq 0, X_{ij} = D_{ij} \text{ for } (i,j) \in \Omega\},$$
$$C_2 = \{X \in \mathbb{R}^{m\times m} : \widehat{X} \text{ in } (*) \text{ is PSD with } \operatorname{rank} \widehat{X} \leq s := 3\}.$$

Now,

- C_1 is a convex set (intersection of cone and affine subspace).
- C_2 is convex iff $m \leq 2$ (in which case $C_2 = \mathbb{R}^{m\times m}$).

For interesting problems, C_2 is **never convex**.

Computing Projections and Reflections

Recall the constraint sets:

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

$$C_2 = \{X \in \mathbb{R}^{m\times m} : \widehat{X} \text{ in } (*) \text{ is PSD with } \operatorname{rank} \widehat{X} \leq s := 3\}.$$

The projection onto C_1 is given (point-wise) by

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

The projection onto C_2 is the set

$$P_{C_2}(X) = \left\{ -Q \begin{bmatrix} \widehat{Y} & d \\ d^T & \delta \end{bmatrix} Q : Q(-X)Q = \begin{bmatrix} \widehat{X} & d \\ d^T & \delta \end{bmatrix}, \begin{array}{l} \widehat{X} \in \mathbb{R}^{(m-1)\times(m-1)}, \\ d \in \mathbb{R}^{m-1}, \delta \in \mathbb{R}, \end{array} \widehat{Y} \in P_{\mathcal{S}_3}\widehat{X} \right\},$$

where $\mathcal{S}_s$ is the set of PSD matrices of rank s or less.

- Computing $P_{\mathcal{S}_s}(\widehat{X})$ = spectral decomposition → threshold eigenvalues.

Computing Projections and Reflections

Recall the constraint sets:

$$C_1 = \{X \in \mathbb{R}^{m\times m} : X \geq 0, X_{ij} = D_{ij} \text{ for } (i,j) \in \Omega\},$$
$$C_2 = \{X \in \mathbb{R}^{m\times m} : \widehat{X} \text{ in } (*) \text{ is PSD with } \operatorname{rank} \widehat{X} \leq s := 3\}.$$

The projection onto C_1 is given (point-wise) by

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

The projection onto C_2 is the set

$$P_{C_2}(X) = \left\{ -Q \begin{bmatrix} \widehat{Y} & d \\ d^T & \delta \end{bmatrix} Q : Q(-X)Q = \begin{bmatrix} \widehat{X} & d \\ d^T & \delta \end{bmatrix}, \begin{array}{l} \widehat{X} \in \mathbb{R}^{(m-1)\times(m-1)}, \\ d \in \mathbb{R}^{m-1}, \delta \in \mathbb{R}, \end{array} \widehat{Y} \in P_{\mathcal{S}_3}\widehat{X} \right\},$$

where $\mathcal{S}_s$ is the set of PSD matrices of rank s or less.

- Computing $P_{\mathcal{S}_s}(\widehat{X})$ = spectral decomposition → threshold eigenvalues.

Computing Projections and Reflections

Recall the constraint sets:

$$C_1 = \{X \in \mathbb{R}^{m\times m} : X \geq 0, X_{ij} = D_{ij} \text{ for } (i,j) \in \Omega\},$$
$$C_2 = \{X \in \mathbb{R}^{m\times m} : \widehat{X} \text{ in } (*) \text{ is PSD with } \operatorname{rank} \widehat{X} \leq s := 3\}.$$

The projection onto C_1 is given (point-wise) by

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

The projection onto C_2 is the set

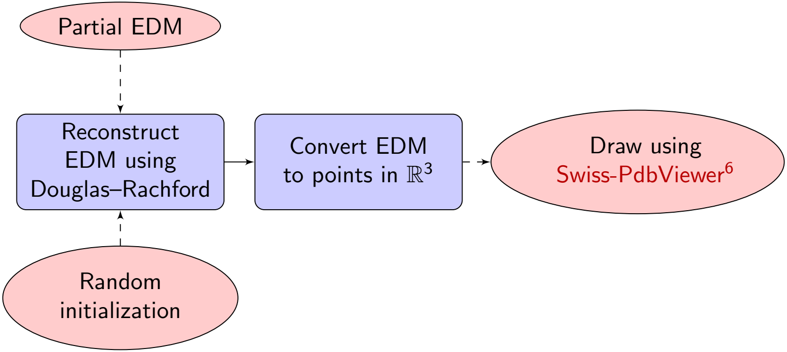
$$P_{C_2}(X) = \left\{ -Q \begin{bmatrix} \widehat{Y} & d \\ d^T & \delta \end{bmatrix} Q : Q(-X)Q = \begin{bmatrix} \widehat{X} & d \\ d^T & \delta \end{bmatrix}, \begin{array}{l} \widehat{X} \in \mathbb{R}^{(m-1)\times(m-1)}, \\ d \in \mathbb{R}^{m-1}, \delta \in \mathbb{R}, \end{array} \widehat{Y} \in P_{\mathcal{S}_3}\widehat{X} \right\},$$

where $\mathcal{S}_s$ is the set of PSD matrices of rank s or less.

- Computing $P_{\mathcal{S}_s}(\widehat{X})$ = spectral decomposition $\rightarrow$ threshold eigenvalues.

The Algorithmic Approach

The reconstruction approach can be summarised as follows:



[1] http://spdbv.vital-it.ch/

Experiment: Six Test Proteins

Experiment: We consider the simplest realistic protein conformation determination problem.

NMR experiments were simulated for proteins with known conformation by computing the partial EDM containing all inter-atomic distances < 6Å.

Table: Six proteins from the RCSB Protein Data Bank.[7]

Protein	# Atoms	# Residues	Known Distances
1PTQ	404	50	8.83%
1HOE	581	74	6.35%
1LFB	641	99	5.57%
1PHT	988	85	4.57%
1POA	1067	118	3.61%
1AX8	1074	146	3.54%

[2] http://www.rcsb.org/

Experiment: Six Test Proteins

Table: Average (worst) results: **5,000** iterations, five random initializations.

Protein	Problem Size	Rel. Error (dB)	RMS Error	Max Error
1PTQ	81,406	-83.6 (-83.7)	0.02 (0.02)	0.08 (0.09)
1HOE	168,490	-72.7 (-69.3)	0.19 (0.26)	2.88 (5.49)
1LFB	205,120	-47.6 (-45.3)	3.24 (3.53)	21.68 (24.00)
1PHT	236,328	-60.5 (-58.1)	1.03 (1.18)	12.71 (13.89)
1POA	568,711	-49.3 (-48.1)	34.09 (34.32)	81.88 (87.60)
1AX8	576,201	-46.7 (-43.5)	9.69 (10.36)	58.55 (62.65)

- The reconstructed EDM is compared to the actual EDM using:

$$\text{Relative error (decibels)} = 10\log_{10}\left(\frac{\|P_A x_n - P_B R_A x_n\|^2}{\|P_A x_n\|^2}\right).$$

- The reconstructed points in $\mathbb{R}^3$ are then compared using:

$$\text{RMS Error} = \left(\sum_{k=1}^{m}\|z_k - z_k^{\text{actual}}\|^2\right)^{1/2}, \quad \text{Max Error} = \max_{k=1,\dots,m}\|z_k - z_k^{\text{actual}}\|,$$

which are computed up to translation, reflection and rotation.

Experiment: Six Test Proteins

Table: Average (worst) results: 5,000 iterations, five random initializations.

Protein	Problem Size	Rel. Error (dB)	RMS Error	Max Error
1PTQ	81,406	-83.6 (-83.7)	0.02 (0.02)	0.08 (0.09)
1HOE	168,490	-72.7 (-69.3)	0.19 (0.26)	2.88 (5.49)
1LFB	205,120	-47.6 (-45.3)	3.24 (3.53)	21.68 (24.00)
1PHT	236,328	-60.5 (-58.1)	1.03 (1.18)	12.71 (13.89)
1POA	568,711	-49.3 (-48.1)	34.09 (34.32)	81.88 (87.60)
1AX8	576,201	-46.7 (-43.5)	9.69 (10.36)	58.55 (62.65)

- The reconstructed EDM is compared to the actual EDM using:

$$\text{Relative error (decibels)} = 10\log_{10}\left(\frac{\|P_A x_n - P_B R_A x_n\|^2}{\|P_A x_n\|^2}\right).$$

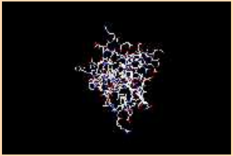
- The reconstructed points in $\mathbb{R}^3$ are then compared using:

$$\text{RMS Error} = \left(\sum_{k=1}^{m}\|z_k - z_k^{\text{actual}}\|^2\right)^{1/2}, \quad \text{Max Error} = \max_{k=1,\dots,m}\|z_k - z_k^{\text{actual}}\|,$$

which are computed up to translation, reflection and rotation.

Experiment: Six Test Proteins

1HOE (actual)

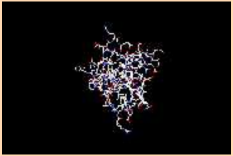


1LFB (actual)

1POA (actual)

Experiment: Six Test Proteins

1HOE (actual)

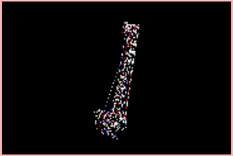


1LFB (actual)

1POA (actual)

1HOE (-72.7dB)

1LFB (-60.5dB)



1POA (-49.3dB)

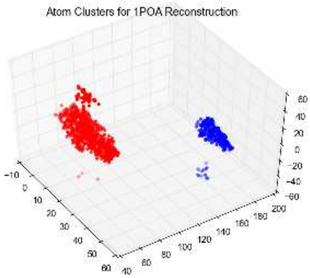
1HOE is good, 1LFB is mostly good, and 1POA has two good pieces.

Experiment: Six Test Proteins

Let's take a closer look at the bad 1POA reconstructions.

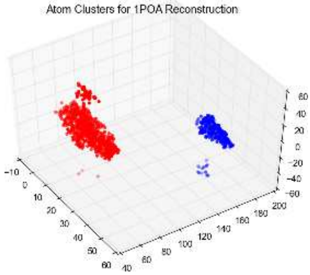
Experiment: Six Test Proteins

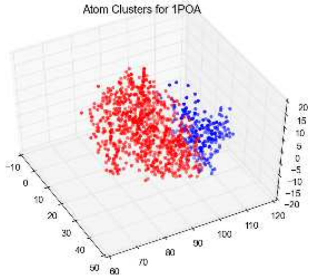
Let's take a closer look at the bad 1POA reconstructions. We *partition* the bad protein's atoms into two clusters: blue and red. We colour the same atoms in the actual structure.



Experiment: Six Test Proteins

Let's take a closer look at the bad 1POA reconstructions. We *partition* the bad protein's atoms into two clusters: blue and red. We colour the same atoms in the actual structure.





- The reconstructed protein's clusters splits actual conformation nicely in two 'halves'.

Experiment: A Better Stopping Criterion?

Optimising our implementation gave a ten-fold speed-up. We performed the following experiment:

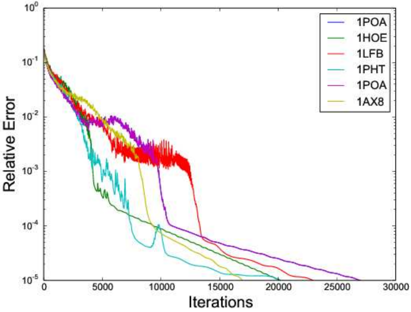
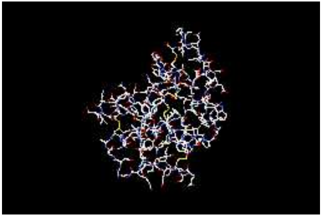


Figure: Relative error by iterations (vertical axis logarithmic).

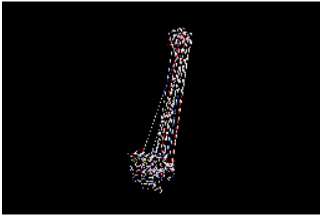
- For $< 5,000$ iterations, the error exhibits non-monotone oscillatory behaviour. It then decreases sharply. Beyond this progress is slower.
- Early termination to blame? $\longrightarrow$ Terminate when error < -100dB.

A More Robust Stopping Criterion

The “un-tuned” implementation (worst reconstruction from previous slide):



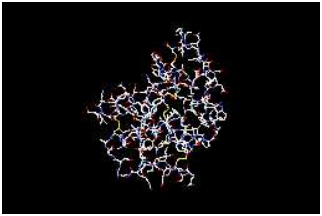
1POA (actual)



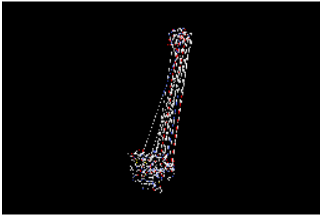
5,000 steps, -49.3dB

A More Robust Stopping Criterion

The "un-tuned" implementation (worst reconstruction from previous slide):

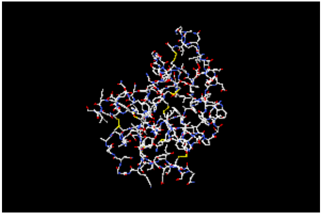


1POA (actual)

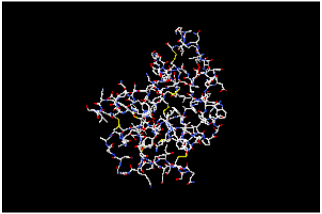


5,000 steps, -49.3dB

The optimised implementation:



1POA (actual)



28,500 steps, -100dB (perfect!)

- Similar results observed for the other test proteins.