The method of reflection-projection for convex feasibility problems with an obtuse cone

Heinz H. Bauschke[*] and **Serge G. Kruk**[†]

February 22, 2002

Abstract

The *convex feasibility problem* asks to find a point in the intersection of finitely many closed convex sets in Euclidean space. This problem is of fundamental importance in mathematics and physical sciences, and it can be solved algorithmically by the classical *method of cyclic projections.*

In this paper, the case where one of the constraints is an *obtuse cone* is considered. Because the nonnegative orthant as well as the set of positive semidefinite symmetric matrices form obtuse cones, we cover a large and substantial class of feasibility problems. Motivated by numerical experiments, the *method of reflection-projection* is proposed: it modifies cyclic projections in that it replaces the projection onto the obtuse cone by the corresponding reflection.

This new method is not covered by the standard frameworks of projection algorithms because of the reflection. The main result states that the method does converge to a solution whenever the underlying convex feasibility problem is consistent. As prototypical applications, we discuss in detail the implementation of two-set feasibility problems aiming to find a nonnegative (resp. positive semidefinite) solution to linear constraints in $\mathbb{R}^n$ (resp. in $\mathbb{S}^n$, the space of symmetric n-by-n matrices), and we report on numerical experiments. The behavior of the method for two *inconsistent* constraints is analyzed as well.

Keywords: convex feasibility problem, obtuse cone, projection methods, self-dual cone.

1 Introduction

Throughout this paper, we assume that

$$\boxed{\mathbb{X} \text{ is a Euclidean space with inner product } \langle\cdot,\cdot\rangle \text{ and induced norm } \|\cdot\|,}$$

[*]Department of Mathematics and Statistics, University of Guelph, Ontario N1G 2W1, Canada. Email: bauschke@cecm.sfu.ca. Research supported by the Natural Sciences and Engineering Research Council of Canada.

[†]Department of Mathematics and Statistics, Oakland University, Rochester, MI 48309-4485, USA. Email: sgkruk@acm.org.

and that

$$\boxed{C_1, \ldots, C_N \text{ are closed convex sets in } \mathbb{X}, \text{ with corresponding projectors } P_1, \ldots, P_N.}$$

(The projector corresponding to a a closed convex set is explained in Definition 2.2.) Moreover, suppose

$$\boxed{K \text{ is a closed convex cone in } \mathbb{X}, \text{ with reflector } R_K = 2P_K - I.}$$

We will require that K be *obtuse*, a notion made precise in Definition 2.5 and broad enough to cover many interesting cones arising in optimization, including the non-negative orthant and the cone of positive semidefinite matrices.

Let

$$\boxed{C := K \cap C_1 \cap \cdots \cap C_N.}$$

Our aim is to solve the *convex feasibility problem*

$$\text{find } x \in C,$$

where, for the most part of this paper, we assume that $C \neq \varnothing$. The convex feasibility problem is of fundamental importance in mathematics and the physical sciences and there exists a multitude of projection algorithms for solving it; see, for instance, [1, 4, 14, 17, 31].

The motivation for this paper stems from a method that works very well in numerical experiments but falls outside the scope of the standard frameworks. Specifically, we propose the *method of reflection-projection*, which, after fixing a starting point x_0, generates a sequence via

$$\boxed{\begin{aligned} x_0 &\mapsto R_K x_0 \mapsto P_1 R_K x_0 \mapsto P_2 P_1 R_K x_0 \mapsto \cdots \mapsto P_N \cdots P_1 R_K x_0 =: x_1 \\ &\mapsto R_K x_1 \mapsto P_1 R_K x_1 \mapsto P_2 P_1 R_K x_1 \mapsto \cdots \mapsto P_N \cdots P_1 R_K x_1 =: x_2 \\ &\mapsto R_K x_2 \mapsto \cdots \quad . \end{aligned}}$$

The terms just displayed form a sequence which has (x_k) as a subsequence. The update operation for (x_k) can be described more concisely by

$$\boxed{x_{k+1} := \big(P_N P_{N-1} \cdots P_1 R\big) x_k.}$$

In Figure 1, we visualize the method of reflection-projection and contrast it with the classical method of cyclic projections (which arises when the reflection is replaced by the corresponding projection) for a two-set convex feasibility problem involving an icecream cone and a plane in $\mathbb{R}^3$. Of course, this particular example favors the method of reflection-projection, but experiments described later bear out this advantage more generally.

The aim of this paper is show that the method of reflection-projection generates a sequence which converges to a solution of the convex feasibility problem. Moreover, experiments demonstrate that the method can yield a solution faster than other standard methods.

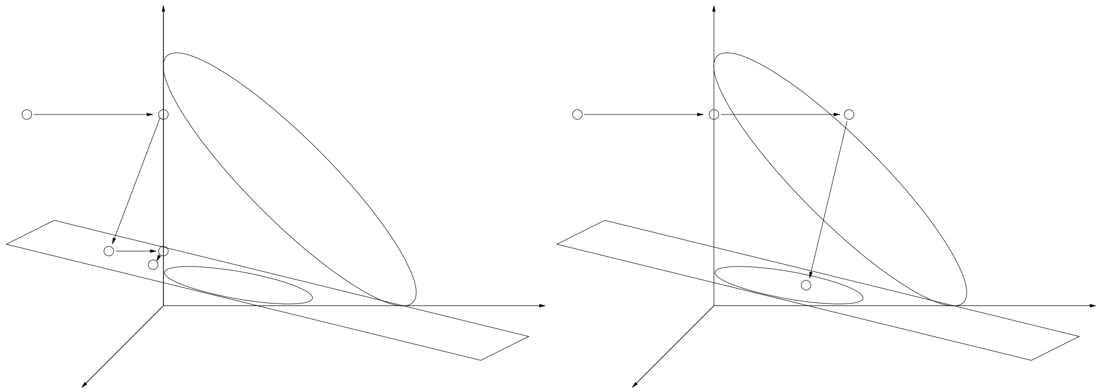

Figure 1: Contrasting behavior of cyclic projections (left) and the method of reflection-projection for the intersection of an icecream cone and a plane (right).

We point out that the standard theory is not applicable, since the reflector R_K is nonexpansive (Lemma 2.12.(ii)), but it does not share any of the common properties (Remark 2.13) typically imposed on the operators in the general frameworks presented in [1, 4, 17, 31].

In fact, the only algorithmic schemes utilizing true reflections are classical and due to Motzkin and Schoenberg [37] and Cimmino [16]. However, none of the convergence results associated with these methods cover the method of reflection-projection presented here.

The paper is organized as follows: Section 2 introduces the cones of interest along with classical convergence results based on Fejér monotone sequences. Section 3 introduces abstractly our feasibility algorithm and the convergence proof in the consistent case. In Section 4, we review affine space projections and the Moore-Penrose inverse. This material is necessary for the practical implementations in $\mathbb{R}^n$ (see Section 5) and in $\mathbb{S}^n$ (see Section 6). Section 7 offers partial results on the inconsistent case and we conclude in Section 8.

2 Preliminaries

Projections

Definition 2.1 (projection and projector) Suppose S is a closed convex nonempty set in $\mathbb{X}$, and $x \in \mathbb{X}$. Then there exists a unique point in S nearest to x, denoted $P_S(x)$ or P_Sx, and called the *projection* of x onto S. Note that P_Sx realizes the *distance* from x to S: $\|x - P_Sx\| = d(x, S) := \min_{s\in S} \|x - s\|$. The induced map $P_S : \mathbb{X} \to S$ is called the *projector*.

Fact 2.2 The projection P_Sx is characterized by $P_Sx \in S$ and $\sup\langle S - P_Sx, x - P_Sx\rangle \leq 0$. In particular, the projector P_S is *firmly nonexpansive*, *i.e.*,

$$(\forall x \in \mathbb{X})(\forall y \in \mathbb{X}) \quad \|P_S x - P_S y\|^2 + \|(I - P_S)x - (I - P_S)y\|^2 \leq \|x - y\|^2.$$

Proof. See, *e.g.*, [22, Chapter 12] or [44]. ☐

Moreau decomposition and obtuse cones

Definition 2.3 (polar cone) Suppose S is a closed convex cone in $\mathbb{X}$. Then

$$S^{\ominus} := \{x \in \mathbb{X} : \sup\langle x, S\rangle \leq 0\}$$

is the *(negative) polar cone* of S. Also, $S^{\oplus} := -S^{\ominus}$ is the *positive polar cone* of S. Given $x \in \mathbb{X}$, we write $x^+ := P_S x$ and $x^- := P_{S^{\ominus}} x$.

Fact 2.4 (Moreau) $P_{K^{\ominus}} = I - P_K$. Let $x \in \mathbb{X}$. Then $x = x^+ + x^-$ and $\langle x^+, x^-\rangle = 0$.

Proof. See [36], or the discussion following [40, Theorem 31.5]. ☐

Definition 2.5 (obtuse and self-dual cones) A closed convex cone K in $\mathbb{X}$ is *obtuse* (resp. *self-dual*), if $K^{\oplus} \subseteq K$ (resp. $K^{\oplus} = K$).

Remark 2.6 The notion of an obtuse cone was coined by Goffin; see [23, Section 3.2]. An obtuse cone is "large" in the following sense:

(i) The affine span of a closed convex obtuse cone K is equal to the entire space $\mathbb{X}$; in particular, K has nonempty interior: indeed, let $\mathbb{Y}$ be the linear (equivalently, affine) span of K. Then $\mathbb{Y}^{\perp} \subseteq K^{\oplus}$. On the one hand, this implies (multiply by -1) the inclusion $\mathbb{Y}^{\perp} \subseteq K^{\ominus}$. On the other hand, since K is obtuse, we conclude $\mathbb{Y}^{\perp} \subseteq K^{\oplus} \subseteq K$. Altogether, $\mathbb{Y}^{\perp} \subseteq K \cap K^{\ominus} = \{0\}$, and so $\mathbb{Y} = \mathbb{X}$.

(ii) [23, Theorem 3.2.1] Suppose K is a closed convex cone in $\mathbb{X}$. Then K is obtuse if and only if $K^{\oplus}$ is *acute*, *i.e.*, $\inf\langle K^{\oplus}, K^{\oplus}\rangle = 0$. ("$\Rightarrow$" is easy to see; for "$\Leftarrow$", use a separation argument.)

The notions of an acute and an obtuse cone have proven quite useful in optimization; see, for instance, [10–13, 23, 29, 30]. The self-dual cones form an important subclass of the obtuse cones, as they include the nonnegative orthant as well as the cone of positive semidefinite matrices — these two cones are of central importance in modern interior point methods [27, 38]. We will discuss these cones in detail in Sections 5 and 6 below.

To provide some examples right now, let us consider a class of halfspaces, and ice cream cones.

Example 2.7 (halfspaces with zero in the boundary) Fix $a \in \mathbb{X} \setminus \{0\}$ and let $K := \{x \in \mathbb{X} : \langle a, x\rangle \geq 0\}$. Then $K^{\oplus} = \{\rho a : \rho \geq 0\}$. Hence $K^{\oplus} \subseteq K$, and therefore K is obtuse.

Ice cream cones

Definition 2.8 The *ice cream cone with parameter* $\alpha > 0$, denoted $\mathrm{ice}(\alpha)$, is defined by

$$\mathrm{ice}(\alpha) := \{(x, r) \in \mathbb{X} \times \mathbb{R} : \|x\| \le \alpha r\}.$$

Note that $\mathrm{ice}(\alpha)$ is a closed convex cone in $\mathbb{X} \times \mathbb{R}$. When $\alpha = 1$, one obtains the so-called *second-order cone* which has found important applications because of the recent successes of interior point methods for convex programming (see [34]). If $\mathbb{X} = \mathbb{R}^3$, the second-order cone becomes

$$\big\{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 : x_4 \ge \sqrt{x_1^2 + x_2^2 + x_3^2}\big\},$$

i.e., the *future light-cone* or *Lorentz cone* from theoretical physics.

The dual cone and the projector of an ice cream cone is known explicitly:

Fact 2.9 Suppose $\alpha > 0$ and $(x, r) \in \mathbb{X} \times \mathbb{R}$. Then $\mathrm{ice}^{\oplus}(\alpha) = \mathrm{ice}(1/\alpha)$, and

$$P_{\mathrm{ice}(\alpha)}(x, r) = \begin{cases} (x, r), & \text{if } \|x\| \le \alpha r; \\ (0, 0), & \text{if } \alpha\|x\| \le -r; \\ \frac{\alpha\|x\| + r}{\alpha^2 + 1}\big(\alpha \frac{x}{\|x\|}, 1\big), & \text{otherwise.} \end{cases}$$

Proof. See [1, Theorem 3.3.6]. ☐

Corollary 2.10 Suppose $\alpha > 0$. Then: $\mathrm{ice}(\alpha)$ is obtuse $\Leftrightarrow \alpha \ge 1$; $\mathrm{ice}(\alpha)$ is self-dual $\Leftrightarrow \alpha = 1$.

Proof. If $\beta > 0$, then $\mathrm{ice}(\alpha) \subseteq \mathrm{ice}(\beta) \Leftrightarrow \alpha \le \beta$; this and Fact 2.9 readily yield the result. ☐

Reflector

Definition 2.11 (reflector) Suppose K is a closed convex set in $\mathbb{X}$. Then the *reflector* corresponding to K is defined by $R_K := 2P_K - I$. If K is a cone and $x \in \mathbb{X}$, we also write $x^{++} := R_K x$.

The following lemma collects various useful results on reflectors and obtuse cones.

Lemma 2.12 Suppose K is a closed convex cone in $\mathbb{X}$, and x, y are two points in $\mathbb{X}$. Then:

(i) $x^{++} = x^+ - x^-$.

(ii) $\|x - y\|^2 - \|x^{++} - y^{++}\|^2 = 4\langle x^+, -y^-\rangle + 4\langle y^+, -x^-\rangle \ge 0$.

(iii) The reflector R_K is *nonexpansive*: $\|x^{++} - y^{++}\| \le \|x - y\|$.

(iv) If $y \in K$, then $y^{++} = y$ and $\|x - y\| \geq \|x^{++} - y\|$.

(v) K is obtuse if and only if R_K maps $\mathbb{X}$ onto K.

Proof. In view of Fact 2.4, we have $x = x^+ + x^-$. Now $x^- \in K^\ominus$, hence $-x^- \in -K^\ominus = K^\oplus$; similarly, $-y^- \in K^\oplus$.
(i): $x^{++} = R_K x = (2P_K - I)(x) = 2x^+ - (x^+ + x^-) = x^+ - x^-$.
(ii): Using Fact 2.4, $\{x^+, y^+\} \subseteq K$, and $\{-x^-, -y^-\} \subseteq K^\oplus$, we obtain

$$\begin{aligned}
\|x - y\|^2 - \|x^{++} - y^{++}\| &= \|(x^+ + x^-) - (y^+ + y^-)\|^2 - \|(x^+ - x^-) - (y^+ - y^-)\|^2 \\
&= \|(x^+ - y^+) + (x^- - y^-)\|^2 - \|(x^+ - y^+) - (x^- - y^-)\|^2 \\
&= 4\langle x^+ - y^+, x^- - y^- \rangle \\
&= 4\langle x^+, -y^- \rangle + 4\langle y^+, -x^- \rangle \\
&\geq 0.
\end{aligned}$$

(iii): This is immediate from (ii).
(iv): If $y \in K$, then $y = P_K y$ and hence $y^+ = y$ and $y^- = 0$. By (i), $y^{++} = y^+ - y^- = y$. The result now follows from (iii).
(v): "$\Rightarrow$": By assumption on K, we have $x^{++} = x^+ + (-x^-) \in K + K^\oplus \subseteq K + K = K$. "$\Leftarrow$": Fix $x \in K^\ominus$. Then $x^+ = 0$ and $x^- = x$. By assumption and (i), $x^{++} = x^+ - x^- \in K$, hence $-x \in K$. Since x was chosen arbitrarily in K, we conclude that $-K^\ominus = K^\oplus \subseteq K$. ☐

Remark 2.13 The reflector R_K (Lemma 2.12.(iii)) is nonexpansive even when K is merely assumed to be a closed convex nonempty set. (Reason: P_K is firmly nonexpansive $\Leftrightarrow$ R_K is nonexpansive; see, for instance, [22, Theorem 12.1].) Let R_K be the reflector corresponding to the nonnegative orthant in the Euclidean plane. Lemma 2.12.(ii) shows not only that R_K is nonexpansive, but it can be also used to demonstrate that R_K *does not satisfy any of the following stronger notions*: • strongly nonexpansive [9]; • nonexpansive in the sense of De Pierro and Iusem [20]; • firmly nonexpansive [4]; • averaging [4]; • strongly attracting [4]; • attracting [4].

It is this lack of additional good properties in the sense of nonexpansive mappings that makes the analysis of the method of reflection-projection within standard frameworks impossible.

Fejér monotone sequences

Definition 2.14 Suppose S is a closed convex nonempty set in $\mathbb{X}$, and $(y_k)_{k\geq 0}$ is a sequence in $\mathbb{X}$. Then (y_k) is *Fejér monotone with respect to S*, if

$$(\forall k \geq 0)(\forall s \in S) \quad \|y_{k+1} - s\| \leq \|y_k - s\|.$$

Fejér monotone sequences are very useful in the analysis of optimization algorithms; see, for instance, [2, 4, 18]. We now record a selection of good properties that will be handy later:

Fact 2.15 Suppose S is a closed convex nonempty set in $\mathbb{X}$, and $(y_k)_{k\geq 0}$ is Fejér monotone with respect to S. Then:

(i) (y_k) is a bounded sequence.

(ii) $\big(d(y_k, S)\big)$ is decreasing and nonnegative, hence convergent.

(iii) The sequence $(P_S y_k)$ converges to some point $\bar{s} \in S$.

(iv) (y_k) converges to $\bar{s}$ if and only if all cluster points of (y_k) belong to S.

Proof. See [2], [4], or [18]. □

3 The method of reflection-projection

The method of reflection-projection is formally espressed by Algorithm 1.

Algorithm 1 The method of reflection-projection

$k := 0$	{Iteration index}
Given $x_k \in \mathbb{X}$	{Starting point}
while $\max\big\{d(x_k, K), d(x_k, C_1), \ldots, d(x_k, C_N)\big\} > 0$ **do**	
$\quad x_k^{++} := R_K x_k$	{Reflect into the cone}
$\quad x_{k+1} := P_N P_{N-1} \ldots P_1 x_k^{++}$	{Project cyclically onto the constraints}
$\quad k := k+1$	{Next iterate}
end while	

Theorem 3.1 Suppose $C \neq \emptyset$ and K is obtuse. Let $x_0 \in \mathbb{X}$. Then the sequence (x_k) generated by Algorithm 1 converges to a point in C.

Proof. We proceed in several steps. Let $(y_k) := (x_0, x_0^{++}, P_1 x_0^{++}, \ldots, x_1, x_1^{++}, \ldots)$, *i.e.*, the sequence implicit in the generation of the sequence (x_k) with all the intermediate terms.

`Step 1:` (y_k) is Fejér monotone with respect to C.
The reflector R_K is nonexpansive (Lemma 2.12.(ii)), and so are the projections $P_1, \ldots, P_N$ (Fact 2.2); moreover, the intersection of the fixed point sets of these $N+1$ maps is precisely C. It follows that (y_k) is Fejér monotone with respect to C.

`Step 2:` (x_k^{++}) is contained in K, and each $d(x_k^{++}, C_i) \to 0$.
Since K is obtuse, Lemma 2.12.(v) implies that (x_k^{++}) lies entirely in K. Next, apply firm nonexpansiveness of P_1 to the two points $x_k^{++}, P_C x_k^{++}$ to obtain $\|x_k^{++} - P_C x_k^{++}\|^2 \geq \|P_1 x_k^{++} - P_C x_k^{++}\|^2 + \|x_k^{++} - P_1 x_k^{++}\|^2$. This, `Step 1`, and Fact 2.15.(ii) yield

$$d^2(x_k^{++}, C_1) \leq d^2(x_k^{++}, C) - d^2(P_1 x_k^{++}, C) \to 0.$$

Firm nonexpansiveness of P_2 applied to the two points $P_1 x_k^{++}, P_C P_1 x_k^{++}$ results analogously in

$$d^2(P_1 x_k^{++}, C_2) \leq d^2(P_1 x_k^{++}, C) - d^2(P_2 P_1 x_k^{++}, C) \to 0.$$

Continuing in this fashion yields $N-2$ further results, the last of which states

$$d^2(P_{N-1} \cdots P_1 x_n^{++}, C_N) \leq d^2(P_{N-1} \cdots P_1 x_k^{++}, C) - d^2(x_{k+1}, C) \to 0.$$

In particular,

1. $x_k^{++} - P_1 x_k^{++} \to 0$;
2. $P_1 x_k^{++} - P_2 P_1 x_k^{++} \to 0$;
3. $P_2 P_1 x_k^{++} - P_3 P_2 P_1 x_k^{++} \to 0$;

$\vdots$;

N. $P_{N-1} \cdots P_1 x_k^{++} - x_{k+1} \to 0$;

Now fix $\nu \in \{1, \ldots, N\}$. Summing the null sequences of items 1 to ν, followed by telescoping and taking the norm, yields

$$0 \leq d(x_k^{++}, C_\nu) \leq \|x_k^{++} - P_\nu \cdots P_1 x_k^{++}\| \to 0.$$

Since ν was chosen arbitrarily, we have completed the proof of `Step 2`.

`Step 3:` Each cluster point of (x_k^{++}) lies in C.
Clear from `Step 2` and the continuity of each distance function $d(\cdot, C_i)$.

`Step 4:` (x_k^{++}) converges to some point $\bar{c} \in C$.
On the one hand, by `Step 1`, the sequence (x_k^{++}) is Fejér monotone with respect to C. On the other hand, by `Step 3`, all cluster points of (x_k^{++}) belong to C. Using Fact 2.15.(iv), we conclude altogether that (x_k^{++}) converges to some point in C.

`Step 5:` The entire sequence (y_k) converges to $\bar{c}$.
Using `Step 4` and continuity of P_1 yields the convergence of $(P_1 x_k^{++})$ to $\bar{c}$. Applying continuity of $P_2, \ldots, P_N$ successively in this fashion, we conclude altogether that (y_k) converges to $\bar{c}$.

`Final Step:` (x_k) converges to $\bar{c}$.
Immediate from `Step 5`, since (x_k) is a subsequence of (y_k). ☐

Remark 3.2 Various comments on Theorem 3.1 are in order.

(i) Theorem 3.1 may be extended routinely in various directions by incorporating weights, relaxation and extrapolation parameters as in [1, 4, 14, 17, 31]. However, rather than obtaining a somewhat more general version, we opted to present a setting that not only clearly shows the usefulness of obtuseness but that also works quite well in practice on the sample problems we investigated numerically: in fact, in Subsection 5.4, we compare the method of reflection-projection to relaxed projections — the numerical results presented there strongly support the practical usefulness of the proposed algorithm.

(ii) Similarly, Theorem 3.1 and its proof extend to general Hilbert spaces as follows: the sequence (x_k) converges *weakly* to some point in C, provided that each projector P_i is *weakly continuous*. Thus the method of reflection-projection can be used to solve convex feasibility problems with an obtuse cone constraint along with *affine* constraints (for which the corresponding projections are indeed weakly continuous).

(iii) In Theorem 3.1, it is impossible to strengthen the conclusion to handle *two or more* obtuse cones via reflectors: indeed, consider two neighboring quadrants in the Euclidean plane. The sequence of alternating reflections will not converge if we fix a starting point in the interior of one quadrant.

(iv) The condition $C \neq \emptyset$ is essential, as the algorithm may fail to converge in its absence: consider the nonnegative orthant in $\mathbb{R}^2$ and the half-space $\{(\rho_1, \rho_2) \in \mathbb{R}^2 : \rho_1 + \rho_2 \leq -1\}$. For $x_0 := (0, 1)$, the method of reflection-projection cycles indefinitely: $x_n \equiv \big(0, (-1)^n\big)$. See, however, Section 7 for some positive results on the inconsistent case.

4 Affine subspace projector and the Moore-Penrose inverse

Throughout this short section, we assume that

$$\boxed{\mathbb{X} \text{ and } \mathbb{Y} \text{ are Euclidean spaces, and } A \text{ is a linear operator from } \mathbb{X} \text{ to } \mathbb{Y}.}$$

Because we work with finite-dimensional spaces, the operator A is continuous and its range $\operatorname{ran} A := \{Ax \in \mathbb{Y} : x \in \mathbb{X}\}$ is closed.

We first summarize fundamental properties of the Moore-Penrose inverse, taken from Chapter II of Groetsch's monograph [26].

Fact 4.1 (Moore-Penrose inverse) There exists a unique (continuous) linear operator $A^\dagger$ from $\mathbb{Y}$ to $\mathbb{X}$ with

$$AA^\dagger = P_{\operatorname{ran} A} \quad \text{and} \quad A^\dagger A = P_{\operatorname{ran} A^\dagger}.$$

The operator $A^\dagger$ is called the *Moore-Penrose inverse* of A. Moreover, $\operatorname{ran} A^\dagger = \operatorname{ran} A^*$, and the Moore-Penrose inverse can be computed via

$$A^\dagger = A^*(AA^*)^\dagger = A^*(AA^*|_{\operatorname{ran} A})^{-1} = (A^*A|_{\operatorname{ran} A^*})^{-1}A^* = (A^*A)^\dagger A^*.$$

The next lemma exhibits the main use we intend to make of the Moore-Penrose inverse.

$$\boxed{\text{Fix } b \in \mathbb{Y}, \text{ not necessarily in the range of } A, \text{ and let } b_0 := P_{\operatorname{ran} A}(b).}$$

Then $b_0 \in \operatorname{ran} A$, and

$$\boxed{S := \{x \in \mathbb{X} \mid Ax = b_0\}}$$

is an affine subspace of $\mathbb{X}$.

Lemma 4.2 (affine subspace projector) $P_S(x) = x - A^\dagger(Ax - b)$, for every $x \in \mathbb{X}$.

Proof. Pick $x \in \mathbb{X}$ and let $s := x - A^\dagger(Ax - b)$. In view of Fact 2.2, we need to show that (i) $s \in S$, and that (ii) $\sup\langle S - s, x - s\rangle \leq 0$. Now, using Fact 4.1,

$$\begin{aligned} As &= A(x - A^\dagger(Ax - b)) = Ax - AA^\dagger(Ax) + AA^\dagger(b) \\ &= Ax - P_{\operatorname{ran} A}(Ax) + P_{\operatorname{ran} A}(b) = Ax - Ax + b_0 \\ &= b_0. \end{aligned}$$

Hence $s \in S$ and (i) holds. Since $s \in S$, we have $S = s + \ker A$, where $\ker A := \{x \in \mathbb{X} : Ax = 0\}$ is the *kernel* of A. By Fact 4.1, $A^\dagger(Ax - b) \in \operatorname{ran} A^\dagger = \operatorname{ran} A^*$. Hence $A^\dagger(Ax - b) \in (\ker A)^\perp$. This implies $0 = \langle \ker A, A^\dagger(Ax - b)\rangle = \langle S - s, x - s\rangle$. Therefore, (ii) is verified and we are done. ☐

As an illustration, let us re-derive the well-known formula for the projection onto a hyperplane.

Example 4.3 (hyperplane projection) Suppose $a \in \mathbb{X} \setminus \{0\}$ and $b \in \mathbb{R}$. Let $S = \{x \in \mathbb{X} : \langle a, x\rangle = b\}$. Then $P_S(x) = x - \frac{\langle a,x\rangle - b}{\|a\|^2} a$, for every $x \in \mathbb{X}$.

Proof. Let $\mathbb{Y} := \mathbb{R}$, and define $A : \mathbb{X} \to \mathbb{Y}$ by $Ax = \langle a, x\rangle$. It is easy to see that $A^\dagger(y) = \frac{y}{\|a\|^2} a$. The result now follows from Lemma 4.2. ☐

The following remark discusses the complications arising from considering two or more hyperplanes.

Remark 4.4 (Gram matrix) Let $\mathbb{Y} = \mathbb{R}^m$ and $a_1, a_2, \ldots, a_m$ be m vectors in $\mathbb{X}$. This induces a linear operator $A : \mathbb{X} \to \mathbb{Y} : x \mapsto (\langle a_i, x\rangle)_{i=1}^m$. (Unless $m = 1$, there is no closed form available for $A^\dagger$, unfortunately.) Note that AA^* maps $\mathbb{R}^m$ to itself. Hence, after fixing a basis and switching to coordinates, AA^* is represented by a matrix $G \in \mathbb{R}^{m \times m}$. It is not hard to see that G is the *Gram matrix* of the vectors $a_1, \ldots, a_m$, i.e, $G_{i,j}$, the (i,j)-entry of G, is equal $\langle a_i, a_j\rangle$, the inner product of the vectors a_i and a_j. Fact 4.1 results in $A^\dagger = A^*(AA^*)^\dagger = A^*G^\dagger$. Thus: finding the Moore-Penrose inverse of A essentially boils down to computing the Moore-Penrose inverse of the Gram matrix G. (See also the end of Chapter 8 in Deutsch's recent monograph [21].)

Remark 4.5 Everything we recorded in this section holds true provided that $\mathbb{X}$ and $\mathbb{Y}$ are Hilbert space, and that A has closed range. (This is so because all results cited from [26] hold in this setting.) For various algorithms on computing the Moore-Penrose inverse, see [26, Sections 3–5 in Chapter II].

5 Euclidean space $\mathbb{R}^n$ and the nonnegative orthant $\mathbb{R}^n_+$

Throughout this section, we assume that

$$\boxed{\mathbb{X} := \mathbb{R}^n,}$$

and the obtuse cone is simply the positive orthant:

$$\boxed{K := \mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x_i \geq 0, \forall i\}.}$$

It is easy to see that K is a closed convex self-dual cone. We consider an additional affine constraint, derived as follows. Let $A \in \mathbb{R}^{m \times n}$ represent a linear operator from $\mathbb{X}$ to $\mathbb{Y} := \mathbb{R}^m$. Suppose $b \in \operatorname{ran} A$, and define

$$\boxed{L := \{x \in \mathbb{X} : Ax = b\}.}$$

Our interest concerns the basic two-set convex feasibility problem

$$\text{find } x \in C := K \cap L.$$

This feasibility problem is of fundamental importance in various areas of mathematics, including Medical Imaging [14].

5.1 Implementation of the cone projector P_K and cone reflector R_K

The projection onto the cone K is simply $(P_K x)_i = x_i^+ = \max\{x_i, 0\}$, for every $i \in \{1, \ldots, n\}$; thus the reflector $R_K = 2P_K - I$ is given by $(R_K x)_i = |x_i| = \max\{x_i, -x_i\} = \operatorname{abs}(x_i)$.

5.2 Implementation of the affine space projector P_L

Lemma 4.2 gives us a handle on computing P_L; the key step is to find an efficient and robust representation of $A^\dagger$, the Moore-Penrose inverse of A. We briefly review three possible paths to an actual implementation: the first two — which can be found many text books — are included for completeness; the third one has the best numerical properties when used in the context of projections.

Let

$$\boxed{r = \operatorname{rank}(A).}$$

We assume without loss of generality that $r \geq 1$. (If $r = 0$, then $A = 0$ and hence $A^\dagger = 0 \in \mathbb{R}^{n \times m}$.)

$A^\dagger$ via singular value decomposition

This approach is outlined in almost every text covering the Moore-Penrose inverse, including [28,42]. Decompose $A = USV^*$, where $U \in \mathbb{R}^{m\times m}$ is orthogonal, $S \in \mathbb{R}^{n\times m}$ has only $S_{1,1}, \ldots, S_{r,r}$ as nonzero entries (which are, in fact, the strictly positive singular values of A), and $V \in \mathbb{R}^{n\times n}$ is orthogonal. Then

$$A^\dagger = VS^\dagger U^*,$$

where $S^\dagger \in \mathbb{R}^{m\times n}$ with the only nonzero entries being $(S^\dagger)_{1,1} = 1/S_{1,1}, \ldots, (S^\dagger)_{r,r} = 1/S_{r,r}$. (The notation is justified as $S^\dagger$ is indeed the Moore-Penrose inverse of S.)

$A^\dagger$ via full-rank factorizations

Factor (see [6,42]) $A = FG$, where $F \in \mathbb{R}^{m\times r}$, $G = \mathbb{R}^{r\times n}$, and $\operatorname{rank}(F) = \operatorname{rank}(G) = r$. Then the *MacDuffee formula* for $A^\dagger$ states

$$A^\dagger = G^*(F^*AG^*)^{-1}F^* = G^*(GG^*)^{-1}(F^*F)^{-1}F^*.$$

The full-rank factors F and G may be constructed as follows: recall the rectangular LU decomposition $PA = LU$, where P is a permutation matrix, and the last $m - r$ rows of U are all zero. Then let G be the submatrix of U consisting only of the first r rows, and F be the submatrix of $P^{-1}L$ consisting of only the first r columns.

$A^\dagger$ via QR factorization

[25, Algorithm 5.4.1] describes an efficient implementation of the following factorization of A^*:

$$A^* = \begin{bmatrix} Q & Q_0 \end{bmatrix} \begin{bmatrix} R & D \\ 0 & 0 \end{bmatrix} P^*,$$

where both $Q \in \mathbb{R}^{n\times r}$ and $Q_0 \in \mathbb{R}^{n\times(n-r)}$ have orthonormal columns, $R \in \mathbb{R}^{r\times r}$ is upper triangular and full-rank, $D \in \mathbb{R}^{r\times(m-r)}$, and finally $P \in \mathbb{R}^{m\times m}$ is a permutation matrix (and hence orthogonal). In practice, Q_0 is not computed since $A^*P = Q\begin{bmatrix} R & D \end{bmatrix}$. By Fact 4.1,

$$\begin{aligned} A^\dagger &= A^*(AA^*)^\dagger \\ &= Q\begin{bmatrix} R & D \end{bmatrix} P^* \left[P \begin{bmatrix} R^* \\ D^* \end{bmatrix} Q^*Q \begin{bmatrix} R & D \end{bmatrix} P^* \right]^\dagger \\ &= Q\begin{bmatrix} R & D \end{bmatrix} P^* \left[P \begin{bmatrix} R^*R & R^*D \\ D^*R & D^*D \end{bmatrix} P^* \right]^\dagger. \end{aligned}$$

Implementation of P_L via QR factors

After permuting the rows of $\begin{bmatrix} A & b \end{bmatrix}$ according to P^* and removing redundant constraints if necessary, we assume without loss of generality that $P^* = I$ and $r = m \leq n$. Then D disappears altogether and the previous expression for $A^\dagger$ simplifies to

$$A^\dagger = QR^{-*}.$$

We can detect whether L is nonempty by comparing the the column rank of $\begin{bmatrix} A & b \end{bmatrix}$ to r. Assuming $L \neq \emptyset$ and utilizing Lemma 4.2, the projection of $x \in \mathbb{X}$ onto L now becomes

$$P_L(x) = x - A^\dagger(Ax - b) = x - QR^{-*}(Ax - b).$$

(In large-scale applications, one needs to store Q and R in compact form using Householder reflections; see [25, Section 5.2.1] for further information.) Since $A = R^*Q^*$, this projection can also be expressed as

$$P_L(x) = x - QR^{-*}(Ax - b) = x - QQ^*x + QR^{-*}b = (I - QQ^*)x + QR^{-*}b;$$

however, especially when A is sparse, this is not preferable in terms of cost or robustness because of the term involving QQ^*.

5.3 Complete implementation of the algorithm

We now present the method of reflection-projection (Algorithm 2) for an affine constraint and the nonnegative orthant, based on the material developed earlier in this section. An important feature is that we allow arbitrary input data A, b, with no restrictions on the relative size of the matrix $A \in \mathbb{R}^{m \times n}$ or on its rank. We first compare the (numerical) ranks of A and $\begin{bmatrix} A & b \end{bmatrix}$, to determine whether L is nonempty. If it is, then the constraints are permuted and redundant constraints are removed.

The termination criteria in the actual implementation go somewhat further than the abstract formulation of Algorithm 1: Based on the analysis in Section 7, a heuristic attempts to detect a possible inconsistency of the feasibility problem, *i.e.*, $C = K \cap L = \emptyset$ even though $L \neq \emptyset$. To recognize these cases we use Lemma 7.5, more specifically, the convergence of the difference of consecutive points inside the cone and on the *flat* (the affine subspace) to a vector realizing the minimum distance between the two convex sets.

After the initial cost of the QR factorization, which is $O(n^3)$ in the dense matrix case [25, 5.2.1], each iteration is fast since the cost of a triangular solve and a matrix-vector multiplication is only $O(n^2)$, see [25, 3.1].

5.4 Numerical experiments

To highlight some advantages of using the method of reflection-projection over (relaxed) projections, we devised an experiment whose results we illustrate now. The Euclidean space chosen was $\mathbb{X} = \mathbb{R}^{64}$

Algorithm 2 The Method of Reflection-Projection for $\{x \in \mathbb{R}^n \mid Ax = b\} \cap \mathbb{R}^n_+$.

```
Given A ∈ R^{m×n}, b ∈ R^m, T ∈ R; x := e ∈ R^n          {Data, tolerance and initial iterate}
Factor [A  b]^* P = QR                                   {Rank-revealing QR}
I_Ab := (|diag(R)| > 10^{-13}||A||);                     {Index of non-zeros}
Factor A^* P = QR                                        {QR update}
I := (|diag(R)| > 10^{-13}||A||);                        {Index of non-zeros}
if |I_Ab| ≠ |I| then
  Quit                                                   {Problem is infeasible}
else
  A := P(I)^*A; b := P(I)^*b; Q := Q(I); R := R(I);      {Eliminate redundancy and permute}
end if
r := Ax − b; Solve R^*y = r; x^+ := x − Qy; d_l := ||x − x^+||;   {Flat Projection}
v := 2x + e; a := b := 0;                                {Initialize variables}
while (||d_l|| > T and ||v − (a − b)||/(1 + ||v||) > T) do
  v := a − b; b := x; x := x^+; a := x;                  {Gap vector and new iterate}
  x := abs(x^+);                                         {Cone reflection}
  r := Ax − b; Solve R^*y = r; x^+ := x − Qy; d_l := ||x − x^+||;   {Flat Projection}
end while
```

and we generated a set of 1000 random feasible problems for m linear constraints, for $2 \leq m \leq 62$. We then ran a sequence of alternating relaxed projection algorithms on each problem, and we averaged the number of iterations needed. More precisely, representing the relaxed projection onto the cone by

$$x \mapsto \big((1-\alpha_K)I + \alpha_K P_K\big)x, \quad \alpha_K \in (0, 2],$$

and the relaxed projection onto the flat by

$$x \mapsto \big((1-\alpha_L)I + \alpha_L P_L\big)x, \quad \alpha_L \in (0, 2),$$

we measured the performance of iterating the map $\big((1-\alpha_L)I + \alpha_L P_L\big)\big((1-\alpha_K)I + \alpha_K P_K\big)$, for a fixed pair of relaxation parameters $(\alpha_K, \alpha_L) \in (0, 2] \times (0, 2)$.

If the relaxation parameter is equal to 1, then the relaxed projection is actually an exact projection; similarly, if it is equal to 2, then we obtain a reflection. Thus, the method of alternating projections corresponds to the choice $(\alpha_K, \alpha_L) = (1, 1)$, whereas the new method of reflection-projection is obtained by setting $(\alpha_K, \alpha_L) = (2, 1)$. (If $\alpha_K < 2$, then the iterates are known to converge; see [1, 4, 17, 31]. And if $\alpha_K = 2$ but $\alpha_L \neq 1$, then a convergence results can be easily obtained by a straight-forward modification of the proof of Theorem 3.1, see also Remark 3.2.(i).)

We searched experimentally for the optimal relaxation parameters by varying them independently in multiples of 0.1. Figure 2 displays the average number of iterations for the problems, for every combination of the relaxation parameters. This experiment suggests that the optimal strategy is to project exactly on the flat ($\alpha_L = 1$), but to reflect into the cone ($\alpha_K = 2$) — this corresponds precisely to the method of reflection-projection (Algorithm 2)! Of course, a different set of problems may suggest a different combination of the relaxation parameters.

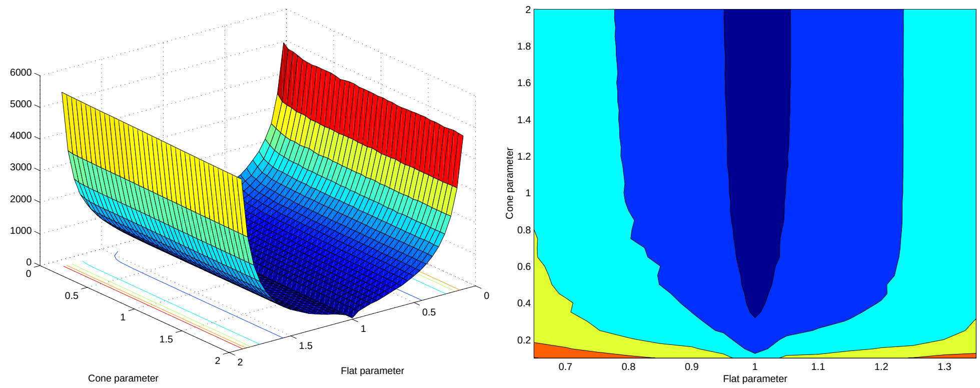

Figure 2: Iteration count for alternating relaxed projections.

6 Euclidean space $\mathbb{S}^n$ and the positive semidefinite cone $\mathbb{S}^n_+$

In this section, we consider the Euclidean space of all real symmetric n-by-n matrices,

$$\boxed{\mathbb{X} := \mathbb{S}^n := \{X \in \mathbb{R}^{n\times n} : X = X^*\}, \quad \text{with } \langle X, Y\rangle := \operatorname{trace}(XY), \text{ for } X, Y \in \mathbb{X}.}$$

For $X \in \mathbb{X}$, we write $X \succeq 0$ to indicate that x is positive semidefinite, and we collect all such matrices in the set

$$\boxed{K := \mathbb{S}^n_+ := \{X \in \mathbb{X} : X \succeq 0\}.}$$

Fejér's Theorem states that K is a closed convex self-dual cone; see [28, Corollary 7.5.4]. This setting lies at the heart of modern optimization; see, for instance, [7] and [43].

Building upon Section 5, we consider an affine constraint given by finitely many linearly independent vectors $A_1, \ldots, A_m$ in $\mathbb{X}$, and a vector $b \in \mathbb{R}^m$. (Linear independence may be enforced as described in our discussion of Algorithm 2 in Subsection 5.3.) Our assumption is equivalent to the surjectivity of the operator

$$A : \mathbb{X} \to \mathbb{R}^m : X \mapsto \begin{bmatrix} \langle A_1, X\rangle \\ \langle A_2, X\rangle \\ \vdots \\ \langle A_m, X\rangle \end{bmatrix}.$$

Hence

$$\boxed{L := \{X \in \mathbb{X} : A(X) = b\} \neq \varnothing,}$$

and we aim to solve the two-set feasibility problem

$$\text{find } X \in C := K \cap L.$$

For simplicity, we assume consistency, *i.e.*, $C \neq \emptyset$. The inconsistent case is quite subtle: the two constraints may have no points in common yet their gap may be zero (see Remark 7.8) — such behavior is impossible for the (polyhedral) setting of the previous Section 5!

The motivation for considering this problem stems from *Interior Point Methods* for solving *Semidefinite Programming problems*. These algorithms fall into two disjoint classes: the so-called *infeasible methods* (which do not require a feasible starting point) and the *feasible methods*. The feasible starting point required for algorithms of the latter class is precisely a solution of the above feasibility problem. See [8, 43] and references therein for further details and background on semidefinite programming. As we illustrate in this section, the method of reflection-projection is well-suited to find such a feasible starting point.

6.1 Implementation of the cone projector P_K and the cone reflector R_K

Fix an arbitrary $X \in \mathbb{X}$. Since X is symmetric, we can factor $X = U^*DU$, where U is an orthogonal matrix whose columns are the eigenvectors $u_1, \ldots, u_n$ of X, and D is a diagonal matrix whose diagonal entries λ_i are the corresponding eigenvalues: $\lambda_i u_i = X u_i$, for all i. Denote by D^+ the diagonal matrix in $\mathbb{X}$ with $(D^+)_{i,i} = (\lambda_i)^+ = \max\{\lambda_i, 0\}$. Using the complete eigenvalue decomposition (see, *e.g.*, [25, 5.5.4]), we have $P_K(X) = U^*D^+U$; equivalently,

$$P_K(X) = \sum_{i:\lambda_i>0} \lambda_i u_i u_i^*.$$

The bulk of the work in computing the projection $P_K(X)$ or the reflection $R_K(X) = 2P_K(X) - X$ lies thus in the determination of the eigenvalues and eigenvectors of X.

The eigen decomposition of a symmetric matrix is an intricate but well-studied problem, and algorithms have been developed for which code is (sometimes freely) available. We refer the reader to the classical work [39] and to the more recent treatment [41]. Note that in order to compute $P_K(X)$, we do not need the complete decomposition; rather, the eigen pairs corresponding to either the positive or the negative eigenvalues are sufficient. For actual numerical implementations, a Lanczos (or Arnoldi) process appears to be most appropriate, especially for large sparse matrices [41]. From now on, we consider the decomposition as a given black-box routine.

6.2 Implementation of the affine subspace projector P_L

The space of real symmetric n-by-n matrices is s a proper subspace of the space of real n-by-n matrices: $\mathbb{S}^n \subsetneqq \mathbb{R}^{n \times n}$; in fact, the dimension of $\mathbb{X} = \mathbb{S}^n$ is

$$t(n) := 1 + 2 + \cdots + n = \frac{n(n+1)}{2},$$

the n^{th} *triangular number*. From a numerical point of view, it is much faster and more memory efficient to work with corresponding vectors in $\mathbb{R}^{t(n)}$ rather than with (symmetric and hence redundant) matrices in $\mathbb{R}^{n\times n}$. Consequently, we start by describing the isometry

$$\text{svec} : \mathbb{S}^n \to \mathbb{R}^{t(n)},$$

which takes the first i entries in column i, stacks them (proceeding from left to right) into a long column vector

$$\left[X_{1,1}, X_{1,2}, X_{2,2}, X_{1,3}, X_{2,3}, X_{3,3}, \ldots, X_{1,n}, X_{2,n}, \ldots, X_{n,n}\right]^*,$$

and finally multiplies each off-diagonal element by $\sqrt{2}$ (to guarantee that the norm $\|X\|$, taken in $\mathbb{S}^n$, agrees with the norm $\|\text{svec}\,(X)\,\|$, taken in $\mathbb{R}^n$). More formally, define the following two index functions

$$\begin{aligned} \text{svecind}(i,j) &:= t(j-1)+i, \quad \text{for } 1 \le i \le j \le n; \\ \text{smatind}(k) &:= \left(k - \tfrac{j(j-1)}{2}, j\right), \quad \text{where } j := \left\lceil \tfrac{1}{2}\left(\sqrt{1+8k}-1\right)\right\rceil \text{ and } 1 \le k \le t(n), \end{aligned}$$

which are inverses of each other [32]. For $X \in \mathbb{S}^n$ and $1 \le k \le t(n)$, the isometry svec is described by

$$\text{svec}(X)_k = \begin{cases} X_{\text{smatind}(k)}, & \text{if } k \text{ is triangular;} \\ \sqrt{2}X_{\text{smatind}(k)}, & \text{otherwise.} \end{cases}$$

And, for $x \in \mathbb{R}^{t(n)}$ and $1 \le i, j \le n$, the inverse smat of svec is given explicitly by

$$\text{smat}(x)_{i,j} = \begin{cases} \frac{1}{\sqrt{2}} x_{\text{svecind}(i,j)}, & \text{if } i < j; \\ x_{\text{svecind}(i,j)}, & \text{if } i = j; \\ \frac{1}{\sqrt{2}} x_{\text{svecind}(j,i)}, & \text{if } i > j. \end{cases}$$

Now define the m-by-$t(n)$ matrix

$$\text{sop}(A) := \begin{bmatrix} \left(\text{svec}\,(A_1)\,\right)^* \\ \left(\text{svec}\,(A_2)\,\right)^* \\ \vdots \\ \left(\text{svec}\,(A_m)\,\right)^* \end{bmatrix}.$$

For $X \in \mathbb{X}$, the affine constraint in $\mathbb{S}^n$ is thus reformulated equivalently in $\mathbb{R}^{t(n)}$ by

$$A(X) = b \quad \Leftrightarrow \quad \big(\text{sop}(A)\big)\text{svec}(X) = b.$$

Therefore, the computation of $P_L(X)$ is reduced to the case considered previously in Subsection 5.2.

6.3 The formulation of the algorithm

For the sake of brevity, we shall omit the steps dealing with the infeasibility and redundancy since they are similar to those taken at the beginning of Algorithm 2 (see also [33]).

Algorithm 3 The method of reflection-projection for $\{X \in \mathbb{S}^n \mid A(X) = b\} \cap \mathbb{S}^n_+$.

Given $A_1, A_2, \ldots, A_m \in \mathbb{S}^n, b \in \mathbb{R}^m, T \in \mathbb{R}$; {Data and tolerance}

$\mathrm{sop}(A) := \begin{bmatrix} (\mathrm{svec}\,(A_1))^* \\ (\mathrm{svec}\,(A_2))^* \\ \vdots \\ (\mathrm{svec}\,(A_m))^* \end{bmatrix}$ {Handle infeasibility and redundancy. Ensure sop(A) is full-rank.}

Factor $\big(\mathrm{sop}(A)\big)^* = QR$

$X := I \in \mathbb{S}^n;\ x := \mathrm{svec}\,(X)\,;$ {Start X_0^{++} at the identity, matrix and vector forms}

$r := \mathrm{sop}(A)x - b;$ Solve $R^*y = r;\ x^+ := x - Qy;$ {Flat Projection}

$d_l := \|x - x^+\|;$ {Keep distance to flat}

while $\|d_l\| < T$ **do**

  $X := \mathrm{smat}\,(x^+)\,;$ {Flat projection X^+}

  Factor $X = V\Lambda V^*$ {Eigen decomposition}

  $\lambda := \mathrm{diag}\,(\Lambda);\ \Lambda^{++} := \mathrm{Diag}\,(\mathrm{abs}\,(\lambda));\ X := V\Lambda^{++}V^*;\ x := \mathrm{svec}\,(X)\,;$ {Cone reflection X^{++}}

  $r := \mathrm{sop}(A)x - b;$ Solve $R^*y = r;\ x^+ := x - Qy;$ {Flat Projection x^+}

  $d_l := \|x - x^+\|;$ {Keep distance to flat}

end while

Algorithm 3 produces sequences denoted $\big(X_k^+\big)$ and $\big(X_k^{++}\big)$, and representing the successive iterates onto the flat and into the cone:

$$X_0^{++} \overset{P_L}{\longmapsto} X_1^+ \overset{R_K}{\longmapsto} X_1^{++} \overset{P_L}{\longmapsto} X_2^+ \overset{R_K}{\longmapsto} X_2^{++} \overset{P_L}{\longmapsto} X_3^+ \overset{R_K}{\longmapsto} \cdots$$

where the loop invariant maintains the iterates within the positive definite cone. The termination criterion therefore only involves the distance to the flat. The reason for exiting the inner loop after a reflection is that the solution returned by the algorithm is numerically positive definite in most cases. This may be useful when strictly interior solutions are sought as is the case for feasible interior-point algorithms of semidefinite programming.

6.4 Numerical experiments

The reader will find in [33] Octave[1] and Matlab[2] implementations of the algorithms described here. These implementations were used to generate all numerical results in this paper.

For the first experiment, feasibility tolerance was set to 10^{-5}. Table 1 presents the results of Algorithm 3 on problems generated by creating random matrices A of increasing sizes. The actual distance to the flat is indicated by the column $\|x - P_L(x)\|$. We do not indicate the distance to the cone since this is always 0.

Figure 3 helps to visualize the increase in iteration count as the feasible set shrinks. For these problems $n = 15$ and the figure averages the number of iterations for each size after 40 runs. The

[1] Octave is freely re-distributable software available at `http://www.octave.org`.

[2] Matlab is a registered trademark of The MathWorks, Inc.

m	n	$\|x-P_L(x)\|$	Iter	m	n	$\|x-P_L(x)\|$	Iter
1	15	1.000799e-16	5	2	15	9.279782e-16	30
3	15	8.530127e-07	54	4	15	8.917017e-07	75
5	15	9.767166e-07	96	6	15	9.867892e-07	118
7	15	9.977883e-07	134	8	15	9.543684e-07	158
9	15	9.400719e-07	175	10	15	9.402718e-07	197
11	15	9.411396e-07	217	12	15	9.500699e-07	246
13	15	9.707506e-07	263	14	15	9.703939e-07	296
15	15	9.802466e-07	319	16	15	9.923299e-07	351
17	15	9.856825e-07	386	18	15	9.878404e-07	428
19	15	9.874863e-07	454	20	15	9.621321e-07	495
21	15	9.959647e-07	531	22	15	9.912171e-07	576
23	15	9.950764e-07	627	24	15	9.826544e-07	790
25	15	9.918322e-07	748	26	15	9.981378e-07	883
27	15	9.989317e-07	931	28	15	9.938861e-07	961
29	15	9.928170e-07	1079	30	15	9.959961e-07	1327
31	15	9.897021e-07	1406	32	15	9.973459e-07	1864

Table 1: Random problems

reader will notice that the algorithm performs very well if the number of constraints is low, but the performance then degrades as the feasible set gets smaller.

7 Results for the inconsistent case

In this final section, we discuss the behavior of the algorithm for two possibly nonintersecting constraints. The reason for this restriction is this: even for the mathematically easier case of cyclic projections, the geometry and behavior is only fully understood for two sets; see [5] for a survey. On the other hand, the results of this section do hold for two general closed convex sets, *i.e.*, neither is assumed to be an obtuse cone.

We start by reviewing the geometry of the problem, which is independent of the algorithm under consideration. For the rest of this section, we assume that

$$\boxed{A \text{ and } B \text{ are two closed convex nonempty sets in } \mathbb{X}.}$$

We let

$$\boxed{v := P_{\mathrm{cl}(B-A)}(0) \text{ and } \delta := \|v\| = \inf\|A-B\| \text{ be the \textit{gap} between } A \text{ and } B.}$$

Here $\mathrm{cl}(B-A)$ denotes the closure of the Minkowski difference $B - A := \{b - a : a \in A, b \in B\}$. We collect the points in A and B where the gap is attained in the following sets

$$\boxed{E := \{a \in A : d(a, B) = \delta\} \text{ and } F := \{b \in B : d(b, A) = \delta\}.}$$

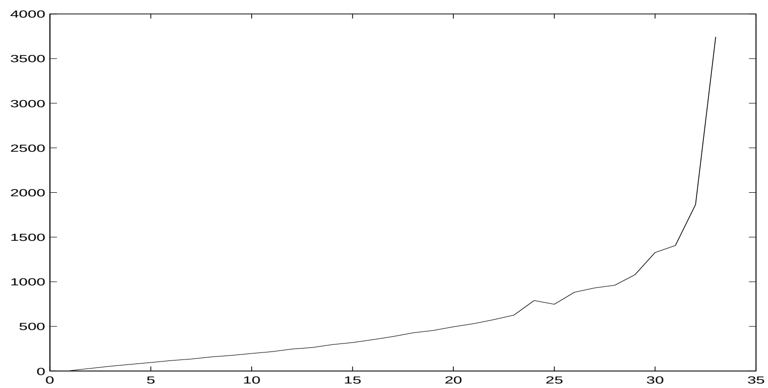

Figure 3: Iteration count as problem size increases.

The sets E and F thus generalize the idea of the intersection of the two sets A and B. Note, however, that E and F may be empty: consider in the Euclidean plane the horizontal axis and the epigraph of $\rho \mapsto 1/\rho$, *i.e.*, $\{(\rho_1, \rho_2) \in \mathbb{R}^2 : 0 < 1/\rho_1 \leq \rho_2\}$.

Definition 7.1 (set of fixed points) If $T : \mathbb{X} \to \mathbb{X}$ is a map, then $\mathrm{Fix}(T) = \{x \in \mathbb{X} : T(x) = x\}$ denotes the *set of fixed points* of T.

The next result provides basic properties of E and F.

Fact 7.2

(i) $E = \mathrm{Fix}(P_A P_B)$ and $F = \mathrm{Fix}(P_B P_A)$.

(ii) $E + v = F$, $E = A \cap (B - v)$, and $F = (A + v) \cap B$.

(iii) Suppose $e \in E$ and $f \in F$. Then $P_B e = e + v$ and $P_A f = f - v$.

Proof. (i) is in Cheney and Goldstein's [15]; for (ii) and (iii), see [3]. □

The method of reflection-projection consists of computing the iterates of the maps $P_A R_B$ and $R_B P_A$; consequently, we are interested in the fixed point sets of these compositions:

Lemma 7.3 $\mathrm{Fix}(P_A R_B) = E$ and $\mathrm{Fix}(R_B P_A) = F + v$.

Proof. Let $(\bar{a}, \bar{b})$ be a fixed point pair: $\bar{a} = P_A \bar{b}$ and $\bar{b} = R_B \bar{a}$. Fix $a \in A$ and $b \in B$ arbitrarily. Then, using Fact 2.2, $\bar{a} = P_A \bar{b}$ is characterized by $\bar{a} \in A$ and $\langle a - \bar{a}, \bar{b} - \bar{a} \rangle \leq 0$. Similarly, $\bar{b} = R_B \bar{a}$ is equivalent to $(\bar{a} + \bar{b})/2 \in B$, and $\langle b - (\bar{a} + \bar{b})/2, \bar{a} - \bar{b} \rangle \leq 0$. Adding the inequalities yields

$$\Big\langle b - a - \big(\tfrac{\bar{a}+\bar{b}}{2} - \bar{a}\big), \bar{a} - \bar{b} \Big\rangle \leq 0,$$

which shows that

$$\tfrac{\bar a+\bar b}{2} - \bar a = P_{\mathrm{cl}(B-A)}(0) = v.$$

Fact 7.2 now shows that $\bar a \in E$ and $\bar b = R_B\bar a = \bar a + 2v \in F + v$. Hence $\mathrm{Fix}(P_A R_B) \subseteq E$, and $\mathrm{Fix}(R_B P_A) \subseteq F + v$. The reverse inclusions are shown similarly, using once again Fact 7.2. ☐

Remark 7.4 (compositions are not asymptotically regular) Viewed from fixed point theory, the compositions R_BP_A and P_AR_B are nonexpansive maps with little extra structure. They lack, for instance, asymptotic regularity: indeed, in $\mathbb{X} = \mathbb{R}^2$, let B be the nonnegative orthant, and A be the line $\{(\rho, -\rho - 1) : \rho \in \mathbb{R}\}$. Let $a_0 = (0, -1)$ be the starting point for the sequence $a_k := P_AR_Ba_{k-1}$ generated by the method of reflection-projection. Then the orbit (a_k) consists of two distinct subsequences $a_{2k} = (0, -1)$ and $a_{2k+1} = (-1, 0)$. Hence (a_k) does not converge even though the distance between the two sets is uniquely attained at $(-\frac{1}{2}, -\frac{1}{2}) \in A$ and $(0, 0) \in B$. In particular, $a_k - a_{k+1} \not\to 0$, which means that the sequence (a_k) is not asymptotically regular.

On the other hand, we have the following positive result.

Lemma 7.5 Let $b_k := R_Ba_k$ and $a_{k+1} := P_Ab_k$ be the sequence generated by the method of reflection-projection, with starting point a_0. Then:

$$(\forall a \in A) \quad \|a_k - a\|^2 \geq \|a_{k+1} - P_AR_Ba\|^2 + \|(b_k - a_{k+1}) - (R_Ba - P_AR_Ba)\|^2.$$

Furthermore: if the gap between A and B is realized, *i.e.*, $E \neq \emptyset$, then:

(i) $(\forall e \in E) \quad \|a_k - e\|^2 \geq \|a_{k+1} - e\|^2 + \|(b_k - a_{k+1}) - 2v\|^2$. In particular, (a_k) is Fejér monotone with respect to E.

(ii) $b_k - a_{k+1} \to 2v$.

(iii) Every cluster point of $\big(\frac{a_k+a_{k+1}}{2}\big)$ belongs to E.

(iv) Every cluster point of $\big(P_Ba_k\big)$ belongs to F.

Proof. The inequality follows, since R_B is nonexpansive and P_A is firmly nonexpansive. (i): is a special case of the inequality, since $R_Be = e + 2v$ and $P_AR_Be = e$ (Lemma 7.3 and Fact 7.2). (ii): is clear from (i). (iii) and (iv): (ii) is equivalent to $P_Ba_k - (a_k + a_{k+1})/2 \to v$. Now the first term in this difference belongs to B and the second one to A. The result now follows directly from [3, Lemma 2.3]. ☐

Remark 7.6 We don't know whether the sequence of averages $\big(\frac{a_k+a_{k+1}}{2}\big)$ must converge to a point in E. This will happen if E is a singleton, as is the case in example presented in Remark 7.4.

Remark 7.7 (lack of monotonicity) In the Euclidean plane, let $A := \{(\rho, -3\rho - 3) : \rho \in \mathbb{R}\}$ and B be the nonnegative orthant. Further, set $a_0 := (0, -3) \in A$, and define recursively $b_n := R_B(a_n)$ and $a_{n+1} := P_A(b_n)$, for $n \geq 0$. It is easy to see that $\|a_1 - b_0\| < \|a_2 - b_1\|$. Hence the sequence

$\big(\|a_{k+1} - b_k\|\big)$ is not decreasing. However, monotonicity properties of this kind lie at the heart of the analysis of the method of cyclic projections (and also Dykstra's algorithm); see [3, Lemma 4.4.(ii) and Lemma 3.1.(iv)]. The lack of this type of monotonicity appears to make the analysis of the inconsistent case much more difficult.

Remark 7.8 (attainment versus nonattainment) Whether or not the gap between the constraints A and B is realized depends essentially on the relative geometry of the sets. Some sufficient conditions for attainment are discussed in [3, Section 5]. The perhaps most important case in applications occurs when one constraint is affine, and the other either the nonnegative orthant or the cone of positive-definite matrices. We explicitly record the following.

(i) If A is affine and B is the nonnegative orthant, then the gap between A and B is always realized. (*Reason:* If A and B are both polyhedral, then so is their difference; in particular, $B - A$ is closed. See [3, Facts 5.1(ii)] and also [5] for additional information.)

(ii) If A is affine and B is the cone of positive semidefinite matrices, then the gap between A and B need not be realized at a pair of points; see [19] and [35] for concrete examples.

8 Conclusion

We presented a new algorithm, the method of reflection-projection, for solving the convex feasibility problem. It aims to find a point in the intersection of finitely many closed convex sets, where one of these sets is an obtuse cone. The method is similar to cyclic projections but it is falls outside the standard frameworks and hence requires a separate proof. Experimental results indicate better performance on some problem sets.

We have given detailed instructions for the feasibility problems involving affine constraints and either the nonnegative orthant or the positive semidefinite cone, both of which are of practical importance. In the former case, in addition to a convergence proof of the algorithm on consistent problems, we have a theoretical detection mechanism of inconsistency. This criterion leads to the implementation of an effective heuristic.

One particularity of the method of reflection-projection is that it easily yields a strictly positive solution in most instances if the last step of the inner loop is the reflection (as we have described and implemented). This is of particular interest in the case of the semidefinite feasibility problem since the method may then be used as a preliminary phase for feasible interior-point methods of semidefinite programming. We are currently considering such a multi-phase approach.

Acknowledgment

We wish to thank Andrzej Cegielski for sending us [10–13]. Part of this work was carried out while HHB was a visiting member of the Fields Institute in Winter 2002. HHB wishes to thank the Fields Institute for their hospitality and support.

References

[1] H. H. Bauschke. *Projection Algorithms and Monotone Operators*, PhD Thesis, Simon Fraser University, 1996. Available at `www.cecm.sfu.ca/preprints/1996pp.html` as 96:080.

[2] H. H. Bauschke. Projection algorithms: results and open problems. In *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications (Haifa 2000)*, D. Butnariu, Y. Censor, and S. Reich (editors), Elsevier 2001, pages 11–22.

[3] H. H. Bauschke and J. M. Borwein. Dykstra's Alternating Projection Algorithm for Two Sets. *J. Approx. Theory* 79(3):418–443, 1994.

[4] H. H. Bauschke and J. M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Rev.* 38(3):367–426, 1996.

[5] H. H. Bauschke, J. M. Borwein, and A. S. Lewis. The method of cyclic projections for closed convex sets in Hilbert space. In *Recent developments in optimization theory and nonlinear analysis (Jerusalem, 1995)*, 1–38, Contemp. Math., 204, Amer. Math. Soc., Providence, RI, 1997.

[6] A. Ben-Israel and N. E. Greville. *Generalized inverses: theory and applications.* Wiley, 1974.

[7] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization.* Springer-Verlag, 2000.

[8] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[9] R. E. Bruck and S. Reich. Nonexpansive projections and resolvents of accretive operators in Banach spaces. *Houston J. Math.* 3:459–470, 1977.

[10] A. Cegielski. Projection onto an acute cone and convex feasibility problem. In *System modelling and optimization (Compiègne, 1993)*, 187–194, Lecture Notes in Control and Inform. Sci., 197, Springer, London, 1994.

[11] A. Cegielski. A method of projection onto an acute cone with level control in convex minimization. *Math. Program.* 85:469–490, 1999.

[12] A. Cegielski. Obtuse cones and Gram matrices with non-negative inverse. *Linear Algebra Appl.* 335:167–181, 2001.

[13] A. Cegielski and R. Dylewski. Residual selection in a projection method for convex minimization. Preprint, 2001.

[14] Y. Censor and S. A. Zenios. *Parallel Optimization.* Oxford University Press, 1997.

[15] W. Cheney and A. A. Goldstein. Proximity maps for convex sets. *Proc. Amer. Math. Soc.* 10:448–450, 1959.

[16] G. Cimmino. Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari. *La Ricerca scientifica ed il Progresso tecnico nell' Economia nazionale (Roma)* 1:326–333, 1938. In Italian.

[17] P. L. Combettes. Hilbertian convex feasibility problem: convergence of projection methods. *Appl. Math. Optim.* 35(3):311–330, 1997.

[18] P. L. Combettes. Quasi-fejérian analysis of some optimization algorithms. In *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications (Haifa 2000)*, D. Butnariu, Y. Censor, and S. Reich (editors), Elsevier 2001, pages 115–152.

[19] E. de Klerk, C. Roos, and T. Terlaky. Infeasible-start semidefinite programming algorithms via self-dual embeddings. *Topics in semidefinite and interior-point methods (Toronto, ON, 1996)*, 215–236, Amer. Math. Soc., Providence, RI, 1998.

[20] A. R. De Pierro and A. N. Iusem. On the asymptotic behaviour of some alternate smoothing series expansion iterative methods. *Linear Algebra Appl.* 130:3–24, 1990.

[21] F. Deutsch. *Best approximation in inner product spaces.* Springer, New York, 2001.

[22] K. Goebel and W. A. Kirk. *Topics in metric fixed point theory.* Cambridge University Press, 1990.

[23] J. L. Goffin. The relaxation method for solving systems of linear inequalities. *Math. Oper. Res.* 5(3):388–414, 1980.

[24] G. H. Golub. Numerical methods for solving linear least squares problems. *Apl. Mat.*, 10:213–216, 1965.

[25] G. H. Golub and C. F. Van Loan. *Matrix computations.* Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[26] C. W. Groetsch. *Generalized inverses of linear operators: representation and approximation.* Dekker, New York, 1977.

[27] O. Güler. Barrier functions in interior point methods. *Math. Oper. Res.*, 21(4):860–885, 1996.

[28] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[29] K. C. Kiwiel. The efficiency of subgradient projection methods for convex optimization, part II: implementations and extensions. *SIAM J. Control Optim.* 34(2):677–697, 1996.

[30] K. C. Kiwiel. Monotone Gram Matrices and Deepest Surrogate Inequalities in Accelerated Relaxation Methods for Convex Feasibility Problems. *Linear Algebra Appl.* 252:27–33, 1997.

[31] K. C. Kiwiel and B. Łopuch. Surrogate projection methods for finding fixed points of firmly nonexpansive mappings. *SIAM J. Optim.* 7(4):1084–1102, 1997.

[32] S. Kruk. *High Accuracy Algorithms for the Solutions of Linear Programs.* PhD thesis, University of Waterloo, 2001.

[33] S. Kruk. `www.oakland.edu/~kruk/research/feasible`. (Implementation of the algorithms discussed in this manuscript.)

[34] M. Sousa Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra Appl.*, 284(1-3):193–228, 1998. ILAS Symposium on Fast Algorithms for Control, Signals and Image Processing (Winnipeg, MB, 1997).

[35] Z.-Q. Luo, J. F. Sturm, and S. Zhang. Conic convex programming and self-dual embedding. *Optim. Methods Softw.* 14 (2000), no. 3, 169–218.

[36] J. J. Moreau. Décomposition orthogonale d'un espace hilbertien selon deux cônes mutuellement polaires. *C. R. Acad. Sci. Paris Sér. A-B* 255:238–240, 1962. In French.

[37] T. S. Motzkin and I. J. Schoenberg. The relaxation method for linear inequalities. *Canad. J. Math.* 6:393–404, 1954.

[38] Y. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Res.*, 22(1):1–42, 1997.

[39] B. N. Parlett. *The symmetric eigenvalue problem.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Corrected reprint of the 1980 original.

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

[41] G. W. Stewart. *Matrix algorithms. Vol. II. Eigensystems.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[42] G. Strang. *Linear Algebra and Its Applications.* Academic Press, 1976.

[43] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming: theory, algorithms and applications.* Number 27 in International series in operations research & management science. Kluwer Academic Publishers, 101 Phillip Drive, Assinippe Park, Norwell MA 02061, 2000.

[44] E. H. Zarantonello. Projections on convex sets in Hilbert space and spectral theory. In E. H. Zarantonello, editor, *Contributions to Nonlinear Functional Analysis*, pages 237–424, New York, 1971. Academic Press. University of Wisconsin. Mathematics Research Center; Publication No. 27.