Dykstra's Alternating Projection for Polyhedral Sets

Claudio Vestini and Idris Kempf

November 25, 2024

1 Introduction

Euclidean projections are ubiquitous in engineering, optimisation, and machine learning, playing a central role in many algorithms for solving problems constrained by complex sets. Mathematically, a projection of a point $x^\circ \in \mathbb{R}^n$ onto a closed convex set $\mathcal{H} \subset \mathbb{R}^p$ is defined as the point $x^\star := \mathcal{P}_\mathcal{H}(x)$ that minimises the Euclidean distance between x° and any point in $\mathcal{H}$:

$$\mathcal{P}_\mathcal{H}(x) := \underset{x\in\mathcal{H}}{\arg\min}\, \|x - x^\circ\|_2. \tag{1}$$

By the convexity of the set $\mathcal{C}$, problem (1) admits a unique solution [?, Ch. 3.2]. Here, we assume that $\mathcal{H}$ is a convex polyhedral set that can be represented as

$$\mathcal{H} := \{x \in \mathbb{R}^p \mid Ax \le c\}, \tag{2}$$

where $A \in \mathbb{R}^{n\times p}$ and $c \in \mathbb{R}^n$. Each row of $\mathcal{H}$ corresponds to a half-space $\mathcal{H}_i$,

$$\mathcal{H}_i := \left\{x \in \mathbb{R}^p \;\middle|\; x^\mathrm{T} f_i \le c_i\right\}, \tag{3}$$

where f_i, $\|f_i\|_2 = 1$, is the normal vector of the plane H_i,

$$H_i := \left\{x \in \mathbb{R}^p \;\middle|\; x^\mathrm{T} f_i = c_i\right\}. \tag{4}$$

The polyhedral set (2) can also be represented as the intersection of the half-spaces $\mathcal{H}_i$, i.e. $\mathcal{H} = \bigcap_{i=0}^{n-1} \mathcal{H}_i$. In many engineering applications, constraint sets can be approximated using (2).

While for some sets the solution of (2) can be obtained explicitly and in closed form, no closed-form solution is known for an arbitrary polyhedral sets. In these cases, the solution can be obtained using a solver for constrained quadratic programs. To solve (1), most solvers require introducing an additional variable $z := Ax$ that is projected onto the set $\{z \in \mathbb{R}^n \mid z \le c\}$, for which an explicit solution exists. However, for large n, this variable augmentation can reduce the computational efficiency, in particular when the projection is part of a larger iterative algorithm.

As an alternative to variable augmentation, problem (1) can be solved using *Dykstra's Alternating Projection Algorithm* [?]. Dykstra's algorithm, first published in 1983, extends von Neumann's *Method of Alternating Projections* (MAP) [?], which is designed to find a point lying in the intersection of n closed convex sets by cyclically projecting onto each individual set. While von Neumann's algorithm identifies *some* point in the intersection, Dykstra's algorithm determines the Euclidean projection (1). Both algorithms circumvent the potentially complex projection $\mathcal{P}_\mathcal{H}$ by iteratively applying the (known) projections $\mathcal{P}_{\mathcal{H}_0}, \dots, \mathcal{P}_{\mathcal{H}_{n-1}}$.

Although Dykstra's algorithm is proven to eventually converge to the projection, it has shown to be prone to *stalling* [?] for certain sets, including polyhedral sets like (2). In such cases, the iterates of Dykstra's method remain unchanged for a number of iterations. The duration of the stalling period cannot be determined *a priori*, and depending on the starting point, can be arbitrary long. This phenomenon hinders the application of Dykstra's method in practice. Additionally, the algorithm *cannot* be run for a fixed number of iterations with a guarantee that the output will be closer to the projection than the initial point – a guarantee typically required when embedding Dykstra's method in iterative schemes, such as gradient methods [?].

2 Dykstra's Alternating Projection

Given n convex sets $\mathcal{H}_0, \dots, \mathcal{H}_{n-1}$, Dykstra's alternating projection algorithm [**?**] finds the orthogonal projection $x^\star$ of x° onto $\mathcal{H} \coloneqq \bigcap_{i=0}^{n-1} \mathcal{H}_i$ by generating a series of iterates $\{x_m\}$ using the scheme

$$x_{m+1} = \mathcal{P}_{\mathcal{H}_{[m]}} \left(x_m + e_{m-n}\right), \tag{5a}$$

$$e_m = e_{m-n} + x_m - x_{m+1}, \tag{5b}$$

where $[m] \coloneqq m \bmod n$, $x_0 = x$, $\mathcal{P}_{[m]} \coloneqq \mathcal{P}_{\mathcal{H}_{[m]}}$, and the auxiliary variables e_m are initialised as

$$e_{-n} = e_{-(n-1)} = \dots = e_{-1} = 0. \tag{6}$$

Note that von Neumann's MAP can be obtained from (5) by setting $e_m \equiv 0 \quad \forall m$. The Boyle-Dykstra theorem [**?**] implies that $\lim_{m\to\infty} \|x_m - \mathcal{P}_{\mathcal{H}}(x)\| = 0$. For a finite number of iterations, there is no guarantee that $x_m \in \mathcal{H}$ nor that $x_m \neq x$.

2.1 The Polyhedral Case

For polyhedral sets (3), the projection step (5a) can be simplified to

$$x_{m+1} = \begin{cases} x_m + e_{m-n} & \text{if } x_m + e_{m-n} \in \mathcal{H}_{[m]} \\ x_m + e_{m-n} - \left((x_m + e_{m-n})^{\mathrm{T}} f_{[m]} - c_{[m]}\right) f_{[m]} & \text{if } x_m + e_{m-n} \notin \mathcal{H}_{[m]} \end{cases}, \tag{7}$$

and the update for the auxiliary vector to

$$e_m = \begin{cases} 0 & \text{if } x_m + e_{m-n} \in \mathcal{H}_{[m]}, \\ \left((x_m + e_{m-n})^{\mathrm{T}} f_{[m]} - c_{[m]}\right) f_{[m]} & \text{if } x_m + e_{m-n} \notin \mathcal{H}_{[m]}, \end{cases}. \tag{8}$$

The auxiliary vector e_m is either 0 or parallel to $f_{[m]}$, so that it can be represented as $e_m = k_m f_{[m]}$ with $k_m = \operatorname{dist}_{\mathcal{H}_{[m]}}(x_{m-1} + e_{m-n})$, further simplifying (12) to

$$k_m = k_{m-n} + x_m^{\mathrm{T}} f_{[m]} - c_{[m]}. \tag{9}$$

The convergence of Dykstra's method has been analysed in [**?**, **?**, **?**] for polyhedral sets. The proof is based on partitioning the sets into inactive ($x^\star \notin H_i$) and active sets ($x^\star \in H_i$), i.e.

$$A = \{i \in \{0, \dots, n-1\} \mid x_\infty \in H_i\}, \qquad B = \{0, \dots, n-1\} \backslash A, \tag{10}$$

where $x_\infty = \lim_{m\to\infty} x_m$. It can be shown that there exists a number N_1 such that whenever

$$[m] \in B, \quad m \geq N_1 \quad \Rightarrow \quad x_m = x_{m-1}, \quad e_m = 0, \tag{11}$$

i.e. the half-spaces that become "inactive" remain inactive. Furthermore, there exists $N_2 \geq N_1$ such that whenever $n \geq N_2$, it holds that

$$\|x_{m+n} - x_\infty\|_2 \leq \alpha_{[m]} \|x_m - x_\infty\|_2, \tag{12}$$

where $0 \leq \alpha_{[m]} < 1$ are numbers related to angles between half-spaces. The number N_2 describes the iteration from which on the algorithm has determined the inactive half-spaces. Finally, it is shown that the algorithm satisfies the following inequality:

Theorem 1 (Deutsch and Hundal [**?**]). *There exist constants $0 \leq c < 1$ and $\rho > 0$ such that*

$$\|x_m - x_\infty\| \leq \rho c^m.$$

The factor c can be estimated from the smallest $\alpha_{[m]}$, which is characterized by the angle between certain subspaces (subspaces formed by the “active” halfspaces). The factor $\alpha_{[m]}$ can be upper-bounded by considering the “worst” angles in the polyhedron. The constant ρ, however, depends on a number $N_3 \geq N_2$ and on x. It is unclear how to obtain that constant ρ. In fact, the authors of [?] and [?] emphasize that ρ cannot be computed in advance, and that the inability to compute a bound on the projection error makes the application of Dykstra's method difficult. The authors of [?] proposed a combined Dysktra-conjugate-gradient method that allows for computing an upper bound on $\|x_m - x_\infty\|$. The authors of [?] proposed an alternative algorithm called *successive approximate algorithm*, which promises fast convergence, conditioned on knowing a point $x \in \mathcal{H}$ in advance.

Remark 1 (Random thoughts)**.** *Dykstra's method could be interpreted as an autonomous non-linear discrete-time system with initial condition x°. For $m \geq N_2$, I believe that the discrete-time system becomes* linear *and could be reformulated as $x_{m+n} = Gx_m + b$. Since by the Boyle-Dykstra theorem x_{m+n} necessarily converges to the projection for $n \to \infty$, the matrix G must be Schur stable, so that the projection can be obtained from $x^\star = (I - G)^{-1} b$. The question is whether N_2 can be detected (may have been answered in the literature already).*

3 Stalling

In [?], the behaviour of Dykstra's method is analysed for two sets. The authors give conditions on Dykstra's algorithm for (i) finite convergence, (ii) infinite convergence, and (iii) stalling followed by infinite convergence. A specific example is given for the case that the set is provided by the intersection of a line with a unit box in $\mathbb{R}^2$ ($\mathcal{H}$ is a polyhedron). It can be shown that cases (i)–(iii) depend on the starting point x_0, and one can determine the 3 regions shown in Figure 1a that yield different convergence behaviour. Convergence case (i) is obtained when starting in the green region, case (ii) when starting in the blue region, and case (iii) when starting in the red region.

To understand the stalling effect, consider Figure 1b, which shows the first iterations of Dykstra's algorithm with starting point in the red region. Note that the outcome of Dykstra's algorithm depends on the order of the sets $\mathcal{H}_i, \dots, \mathcal{H}_n$. In Figure 1b, the algorithm starts by projecting onto the box and then onto the line. It can be seen that for the first 6 iterations[1], Dykstra's algorithm returns the top left corner of the box (“stalling”). The authors also determine the exact number of iterations required to break free from the red region, and show that if the starting point is arbitrarily far to the left, the algorithm will need an arbitrarily large iteration number to break free from the red region.

3.1 Early Termination of Stalling

By graphical examination of Figure 1 and formulae (5), we make the following observations:

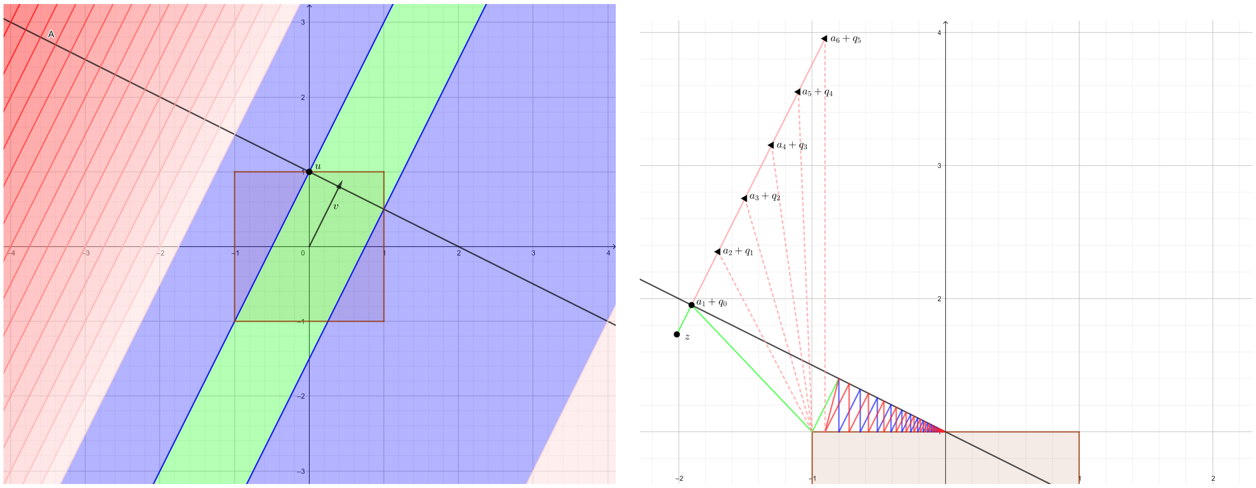
1. The stalling period continues until one of the half-spaces that is active for x° and inactive for $x^\star$ changes from active to inactive. This is a necessary condition.

2. During stalling, it holds that $x_m \equiv x_{m-n} \quad \forall m$, so that every iteration, a constant vector $x_{m-1} - x_m$ is added to e_m.

Since during stalling all x_m are constant for each half-space $[m]$, the only way a half-space can become inactive is through change of the auxiliary variables e_m or their length k_m. Moreover, increasing k_m will only move $x_m + e_{m-n}$ away from the half-space, from which we follow that if a half-space is to become inactive, then the magnitude of the auxiliary variables must decrease. From (9), k_m decreases iff

$$x_m^{\mathrm{T}} f_{[m]} - c_{[m]} < 0. \tag{13}$$

[1] By one iteration we mean one cycle of n projections here.



(a) Line-box example with different regions that yield different convergence properties.

(b) Stalling for the line-box example when x_0 is in the red region.

Figure 1: A demonstration of the stalling problem for a box and a line. Note how MAP applied to the same constraint sets would not result in any stalling: MAP follows the green line, and subsequently converges via the blue line path. Figure taken from [**?**]

Moreover, since during stalling x_m is constant, there must exist at least one m for which (13) holds. If (13) for multiple m, then the stalling period can be exited by making several half-spaces inactive. Suppose that stalling starts at iteration m with $x_{m+i} = x_{m+n+i}$ for $i = 0, \dots, n-1$, then the length of the stalling period can be computed from choosing the smallest positive integer $p_{[m+i]}$ for which

$$k_{m+i} + p_{[m+i]} \left(x_{m+i}^{\mathrm{T}} f_{[m+i]} - c_{[m+i]} \right). \tag{14}$$

Alternatively, suppose that $\mathcal{S} := \left\{ i \in \{0, \dots, n-1\} \;\middle|\; x_{m+i}^{\mathrm{T}} f_{[m+i]} - c_{[m+i]} < 0 \right\}$, then the length of the stalling period is

$$N_{stall} := \arg\min_{i \in \mathcal{S}} \left\lfloor -k_{m+i} / \left(x_{m+i}^{\mathrm{T}} f_{[m+i]} - c_{[m+i]} \right) \right\rfloor. \tag{15}$$

Suppose that stalling is registered at iteration m, i.e. $x_{m+i} = x_{m+n+i}$ for $i = 0, \dots, n-1$, then the stalling period can be skipped by applying N_{stall} steps.

4 Additional Ideas

For the polyhedral case, Dykstra's method can be split into two phases: An first phase of length N_2 during which some half-spaces become inactive, and a second phase during which the algorithm iterates between active half-spaces. Note that stalling can only occur during the first phase.

For the first phase, the term $x_m^{\mathrm{T}} f_{[m]} - c_{[m]}$ in (9) seems to play an important role: the only way a half-space can become inactive is by obtaining a negative $x_m^{\mathrm{T}} f_{[m]} - c_{[m]}$. In [**?**], it is shown that

$$x_{m+n} = \operatorname{span} \{ x_m, f_0, \dots, f_{n-1} \}. \tag{16}$$

Moreover, by examination of (7), if $x_{m+n} + e_m \notin \mathcal{H}_{[m]}$, we see that

$$x_{m+n} = x_m + a_{[m]} f_{[m]} - \sum_{i \in \mathcal{I} \setminus [m]} a_i f_i, \tag{17}$$

where $a_i \geq 0$, $a_{[m]} \equiv k_m$, and $\mathcal{I} := \{0, \dots, n-1\}$.