Dykstra's Algorithm for Projection onto Convex Sets: Stalling Analysis

Claudio Vestini and Idris Kempf

September 2024

1 Context & Background

The forthcoming upgrade of the Diamond Light Source (DLS) synchrotron presents a complex control challenge in the stabilisation of the electron beam. This is due to the increased number of sensors (from 172 to 252) and actuators (from 173 to 396), as well as the introduction of two different types of corrector magnets [6]. Traditional control methods (*modal decomposition* [7]) are no longer sufficient, and model predictive control (MPC), while promising, faces high computational demands for high-frequency (100 kHz) control. The optimiser has to solve a constrained quadratic programming (CQP) problem at each timestep in less than $10\mu s$ for feasibility. This involves minimising a cost function to obtain a global minimum, and projecting it onto the constraint set. The projection step can be solved using Dykstra's method [3], which has been shown to converge linearly [8]. This algorithm can stall under certain conditions [1], and this paper will investigate stalling and how it may be prevented.

2 Problem Statement

The relationship between the $n_y = 252$ measured beam displacements $y_k \in \mathbb{R}^{n_y}$ and the $n_u = 396$ actuator inputs $u_k \in \mathbb{R}^{n_u}$ can be described by the state-space representation of a linear system in discrete time,

$$x_{k+1} = Ax_k + Bu_k, \quad u_k \in \mathcal{U}(u_{k-1}), \tag{1a}$$

$$y_k = Cx_k + d_k, \tag{1b}$$

where $x_k \in \mathbb{R}^{n_x}$ are the states at time $t = kT_s$. The inputs u_k are subjected to amplitude and slew-rate constraints that can be modelled as $\mathcal{U}(u_{k-1}) := \mathcal{A} \cap \mathcal{R}(u_{k-1})$, where u_{k-1} is the input applied at time $t = (k-1)T_s$ and $\mathcal{A}$ and $\mathcal{R}(u_{k-1})$ are the amplitude and slew-rate constraint sets:

$$\mathcal{A} := \{u_k \in \mathbb{R}^{n_u} \mid -\alpha \le u_k \le \alpha\}, \tag{2a}$$

$$\mathcal{R}(u_{k-1}) := \{u_k \in \mathbb{R}^{n_u} \mid -\rho \le u_k - u_{k-1} \le \rho\}. \tag{2b}$$

The condensed [2] model predictive control problem for (1a) is

$$\min_{u} \frac{1}{2} u^{\mathrm{T}} J u + q(\hat{x}_k, \hat{d}_k)^{\mathrm{T}} u \quad \text{s.t.} \quad u \in \mathcal{S}(u_{-1}), \tag{3}$$

where $u \coloneqq (u_0^{\mathrm{T}}, \dots, u_{N-1}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{N n_u}$, N the horizon, $f(u) \coloneqq \frac{1}{2} u^{\mathrm{T}} J u + q(\hat{x}_k, \hat{d}_k)^{\mathrm{T}} u$ the objective function and $J = J^{\mathrm{T}} \in \mathbb{R}^{N n_u \times N n_u}$ the Hessian. The vector $q(\hat{x}_k, \hat{d}_k)$ is an affine function of the observer output $\hat{x}_k$ and $\hat{d}_k$. The closed convex set $\mathcal{S}(u_{-1})$ is defined as

$$\mathcal{S}(u_{-1}) \coloneqq \mathcal{U}(u_{-1}) \times \cdots \times \mathcal{U}(u_{N-2}), \tag{4}$$

and depends on the input u_{-1} applied at time $t - 1$. The following assumptions on problem (3) are made throughout the paper:

Assumption I.a (Strong convexity)**.** *It holds that $0 \prec \lambda_{min} I \preceq J \preceq \lambda_{max} I$, where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of J.*

Assumption I.b (Bounded inputs)**.** *The set $\mathcal{S}$ is bounded, so that $|\mathcal{S}| \coloneqq \max_{x,y \in \mathcal{S}} \|x - y\|_2 < \infty$.*

To solve problem (3), we are using the fast gradient method [9]. The present variant of the fast gradient method is applicable to strongly convex functions for ill-conditioned Hessians [10, Ch. 2.2.4]. The fast gradient method requires the Hilbert space projection $\mathcal{P}_{\mathcal{S}}$ onto $\mathcal{S}$, and no simple formula exists for $N > 1$. Here, we are replacing the exact projection $\mathcal{P}_{\mathcal{S}}$ with Dykstra's alternating projection algorithm, $\mathcal{D}_{\mathcal{S}}$. Dykstra's method is an iterative algorithm that yields the exact projection if the algorithm is run for an infinite number of iterations. For M iterations, $\mathcal{D}_{\mathcal{S}}(z) \notin \mathcal{S}$ in general[1] and the method yields a projection error that can be quantified as shown in the following assumption.

Assumption II (Approximate projection)**.** *Dykstra's method $\mathcal{D}_{\mathcal{S}}$ returns a point $\mathcal{D}_{\mathcal{S}}(z)$ that satisfies*

$$\|\mathcal{D}_{\mathcal{S}}(z) - \mathcal{P}_{\mathcal{S}}(z)\|_2 \le \delta(z, M, \mathcal{S}), \tag{5}$$

where the upper bound $\delta(z, M, \mathcal{S}) > 0$ depends on the maximum number of iterations M of Dykstra's method.

We wish to characterize $\delta(z, M, \mathcal{S})$ for $\mathcal{S}$ and $z \in \mathcal{Z}$, where $\mathcal{Z} = \mathcal{Z}(J, q, u_{-1})$ is to be defined.

[1] For example, consider the projection onto a corner (or edge) of two intersecting hyperplanes.

3 Dykstra's Alternating Projection

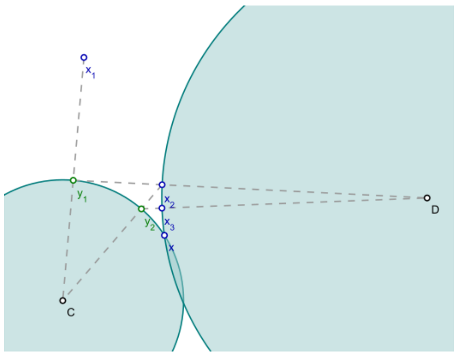
The projection step of the fast gradient method can be solved using Dykstra's algorithm. Given n convex sets $\mathcal{H}_1, \ldots, \mathcal{H}_n$, Dykstra's alternating projection algorithm [11] finds the orthogonal projection $x^\star$ of x onto $\mathcal{H} \coloneqq \bigcap_{i=1}^n \mathcal{H}_i$ by initially setting

$$e_{-n} = e_{-(n-1)} = \ldots = e_{-1} = \vec{0} \tag{6}$$

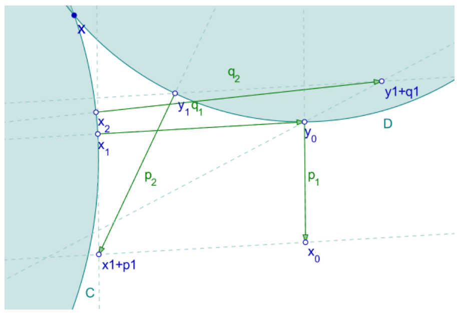
and generating a series of iterates $\{x_m\}$ using the scheme

$$x_{m+1} = \mathcal{P}_{\mathcal{H}_{[m]}}(x_m + e_{m-n}), \qquad e_m = x_m + e_{m-n} - x_{m+1}, \tag{7}$$

for $[m] = 0, 1, \ldots$ and $x_0 = x$.
This is a variation of the simpler Von Neumann's *Method of Alternating Projections* (MAP), which can be obtained from (7) with the errors $e_m = \vec{0}$. A visual explanation of the difference between the two methods is displayed in Figure 1



(a) MAP algorithm with successive projections y and x onto sets C and D respectively

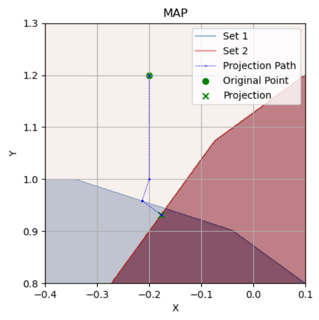


(b) Dykstra's algorithm with associated errors p and q

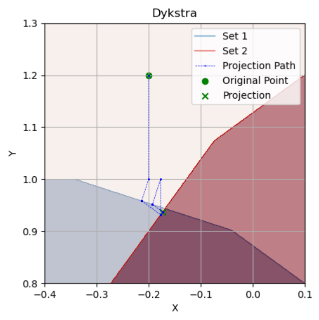
Figure 1: A demonstration of MAP vs Dykstra. Dykstra has intermediate steps

Despite MAP being much simpler, it exhibits weak convergence [4] and does not always produce an optimal output, as illustrated in Figure 2.

The Boyle-Dykstra theorem [3] implies that $\lim_{m\to\infty} \|x_m - \mathcal{P}_{\mathcal{H}}(x)\| = \vec{0}$. For a finite number of iterations, there is no guarantee that $x_m \in \mathcal{H}$.



(a) Sub-optimal solution for MAP after 1 iteration. Nothing happens in second iteration



(b) Dykstra gets closer to optimal solution in second iteration

Figure 2: A demonstration of solution optimality. Both algorithms were ran for two iterations. Dykstra's will produce an optimal output after infinitely many iterations

Here, we assume that $\mathcal{H}$ is a polyhedron and the $\mathcal{H}_i$ are halfspaces given by

$$\mathcal{H}_i \coloneqq \{x \mid \langle x, f_i\rangle \leq c_i\}, \tag{8}$$

where $\|f_i\| = 1$. In addition, define boundaries

$$H_i \coloneqq \{x \mid \langle x, f_i\rangle = c_i\}, \tag{9}$$

so that $\operatorname{int}\mathcal{H}_i \coloneqq \mathcal{H}_i \backslash H_i$. The projections onto H_i and $\mathcal{H}_i$ are given by

$$\mathcal{P}_{H_i}(x) = x - (\langle x, f_i\rangle - c_i) f_i, \qquad \mathcal{P}_{\mathcal{H}_i}(x) = \begin{cases} x & \text{if } x \in \mathcal{H}_i \\ \mathcal{P}_{H_i}(x) & \text{if } x \notin \mathcal{H}_i\,. \end{cases} \tag{10}$$

For this particular choice of sets, $e_m = k_m f_{[m]}$ with $k_m = \operatorname{dist}_{\mathcal{H}_{[m]}}(x_{m-1} + e_{m-n})$, i.e. the auxiliary vector is either 0 or parallel to $f_{[m]}$. The convergence of Dykstra's method has been analyzed in [8, 5], where it has been shown that the convergence is linear. In [5], the proof is based on partitioning the set $\{1, \dots, n\}$ into

$$A = \{i \in \{1, \dots, n\} \mid x_\infty \in H_i\}, \quad B = \{1, \dots, n\} \backslash A = \{i \in \{1, \dots, n\} \mid x_\infty \in \operatorname{int}\mathcal{H}_i\}, \tag{11}$$

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

$$[m] \in B, \quad m \geq N_1 \quad \Rightarrow \quad x_m = x_{m-1}, \quad e_m = \vec{0}, \tag{12}$$

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

$$\|x_{m+n} - x_\infty\|_2 \leq \alpha_{[m]} \|x_m - x_\infty\|_2, \tag{13}$$

where $0 \leq \alpha_{[m]} < 1$. With these ingredients, the following result is obtained:

Theorem 1. *There exist constants $0 \leq c < 1$ and $\rho > 0$ such that*

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

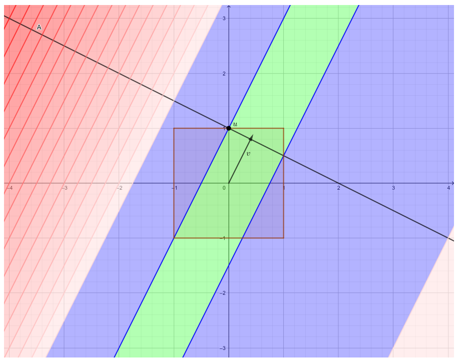
The factor c can be estimated from the smallest $\alpha_{[m]}$, which is characterized by the angle between certain subspaces (subspaces formed by the "active" halfspaces). The factor $\alpha_{[m]}$ can be upper-bounded by considering the "worst" angles in the polyhedron. The constant ρ, however, depends 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 [11] and [13] 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 [11] proposed a combined Dysktra-conjugate-gradient method that allows for computing an upper bound on $\|x_m - x_\infty\|$. The authors of [13] proposed an alternative algorithm called *successive approximate algorithm*, which promises fast convergence, conditioned on knowing a point $x \in \mathcal{H}$ in advance.

4 Stalling

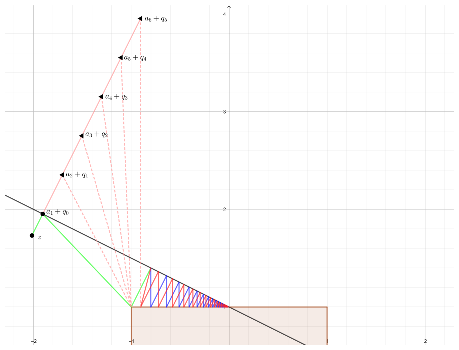
In [1], the behaviour of Dykstra's method is analysed for two sets. The authors give conditions on Dykstra's algorithm for (1) finite convergence, (2) infinite convergence and (3) stalling followed by infinite convergence. A specific example is given for the case that the set is provided by the intersection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

To understand the stalling effect, consider Figure 3b, 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 3b, the algorithm starts by projecting onto the box and then onto the line. It can be seen that for the first 6 iterations[2], 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.

[2]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.

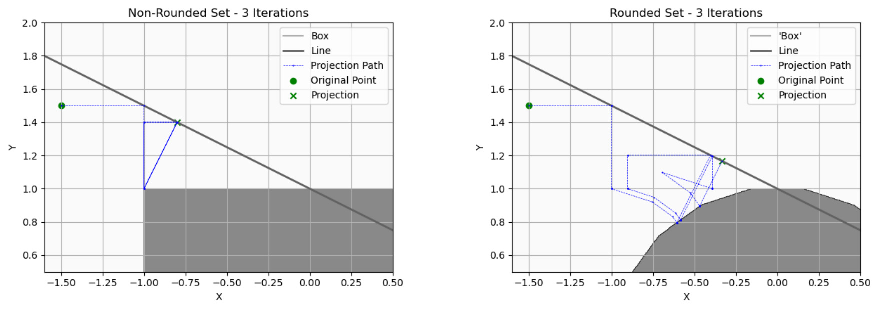
Figure 3: A demonstration of the stalling problem for a box and a line. Note how MAP applied to the same constraint sets would not result in any stalling: MAP follows the green line, and subsequently converges via the blue line path.

5 Boundary Smoothness

From an analysis of the behaviour of Dykstra's algorithm in the example of Figure 3b, a possible reason behind stalling can be identified in the non-smooth ([12]) nature of the boundary of the box's convex set. The algorithm keeps projecting onto the box's corner at the start of each iteration, since this is the closest point to any x_m with x coordinate less than -1. This is repeated until $x >= -1$, at which point the algorithm can project onto the top side of the box and exit stalling. If the boundary of the box was strictly "smooth", there would be no corner to get stuck at. The algorithm would get closer to the optimal solution at every step. A non-strict example is illustrated in Figure 4, where the box is rounded by introducing 5 "rounding corners" (essentially transforming the box into a solid icosagon).

This idea originally seemed to be very promising. However, there are several problems associated with the smoothing out of convex sets' boundaries to prevent stalling:

- The smoothing adds space and time complexity to the algorithm, both directly and by adding new halfspaces to project upon.
- The smoothing may cast the optimal solution onto a non-optimal approximation. This is not the case in our box and line example, but it may represent a serious issue in the MPC implementation.
- Most important of all, the added presence of new halfspaces hinders the algorithm's convergence rate. This is further discussed below.



(a) Dykstra's algorithm on the box and line example

(b) Rounded edges prevent algorithm from stalling

Figure 4: A demonstration of how roundedness reduces stalling

Let us consider the same line-box constraint set as 3a, but now let us place the initial guess at $(-10,\ 10)$ to increase the count of iterations required to exit stalling. Dykstra's algorithm applied with these conditions, as well as its convergence behaviour, are displayed in Figure 5

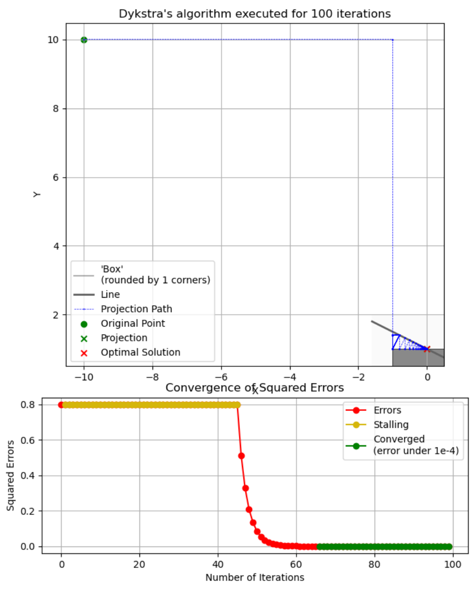


Figure 5: Dykstra's algorithm applied to initial point $z = (-10,\ 10)$

Now compare this to the same box, rounded by 10 corners. The algorithm requires fewer iterations to exit stalling, but convergence is much slower. In this case, the algorithm outputs a much better approximation for the non-rounded set, as shown by comparing Figure 5 and Figure 6.

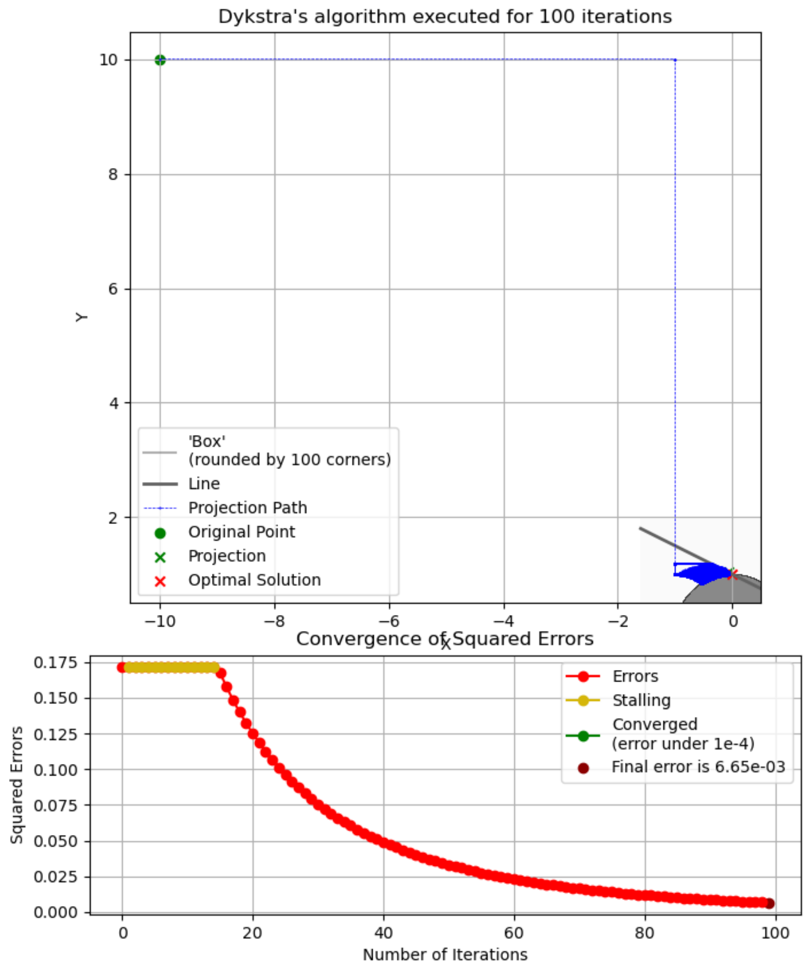


Figure 6: Dykstra's algorithm applied to initial point $z = (-10, 10)$, box is now rounded by 100 corners

6 Acceleration for Dykstra's Method

Consider introducing the step size parameter $\beta_m \geq 0$:

$$x_m = \mathcal{P}_{\mathcal{H}_{[m]}}\left(x_{m-1} + e_{m-n}\right), \qquad e_m = e_{m-n} + \beta_m(x_{m-1} - x_m). \tag{14}$$

For $\beta_m = 0$, we obtain MAP and for $\beta_m = 1$, we obtain Dykstra's method (7). We proceed by characterising the term e_m.

Lemma 1. *It holds that $e_m = y_m f_{[m]}$, where*

$$y_n = (1 - \beta_m)y_{m-n} + k_m,$$

and $k_m = dist_{\mathcal{H}_{[m]}}(x_{m-1} + e_{m-r})$.

Proof. Suppose that $x_{m-1} + e_{m-n} \in \operatorname{int}\mathcal{H}_{[m]}$. Then, $x_n = x_{m-1} + e_{m-n}$ and

$$e_m = (1 - \beta_m)e_{m-n}.$$

Suppose that $x_{m-1}+e_{m-n} \notin \operatorname{int}\mathcal{H}_{[m]}$. Then, $x_m = x_{m-1}+e_{m-n}-\left(\left\langle x_{m-1} + e_{m-n}, f_{[m]}\right\rangle - c_i\right) f_{[m]}$ and

$$e_m = (1 - \beta_m)e_{m-n} + k_m f_{[m]}.$$

Note that $e_m = 0$ for $m \leq 0$. By induction, e_m is always parallel to $f_{[m]}$ or zero. Substituting $e_m = y_n f_{[m]}$ yields

$$y_m = (1 - \beta_m)y_{m-n} + k_m.$$

□

The introduction of β_m adds another degree of freedom to Dykstra's algorithm. This parameter could be chosen before the algorithm is applied, or even before each iteration. How to compute an optimal choice of β_m is currently under investigation.

For our beloved line-box example with starting choice of $z = (-10,\ 10)$, it is easy to spot how a choice of $\beta_m = 0$ would be optimal. The algorithm would converge as shown in Figure 7

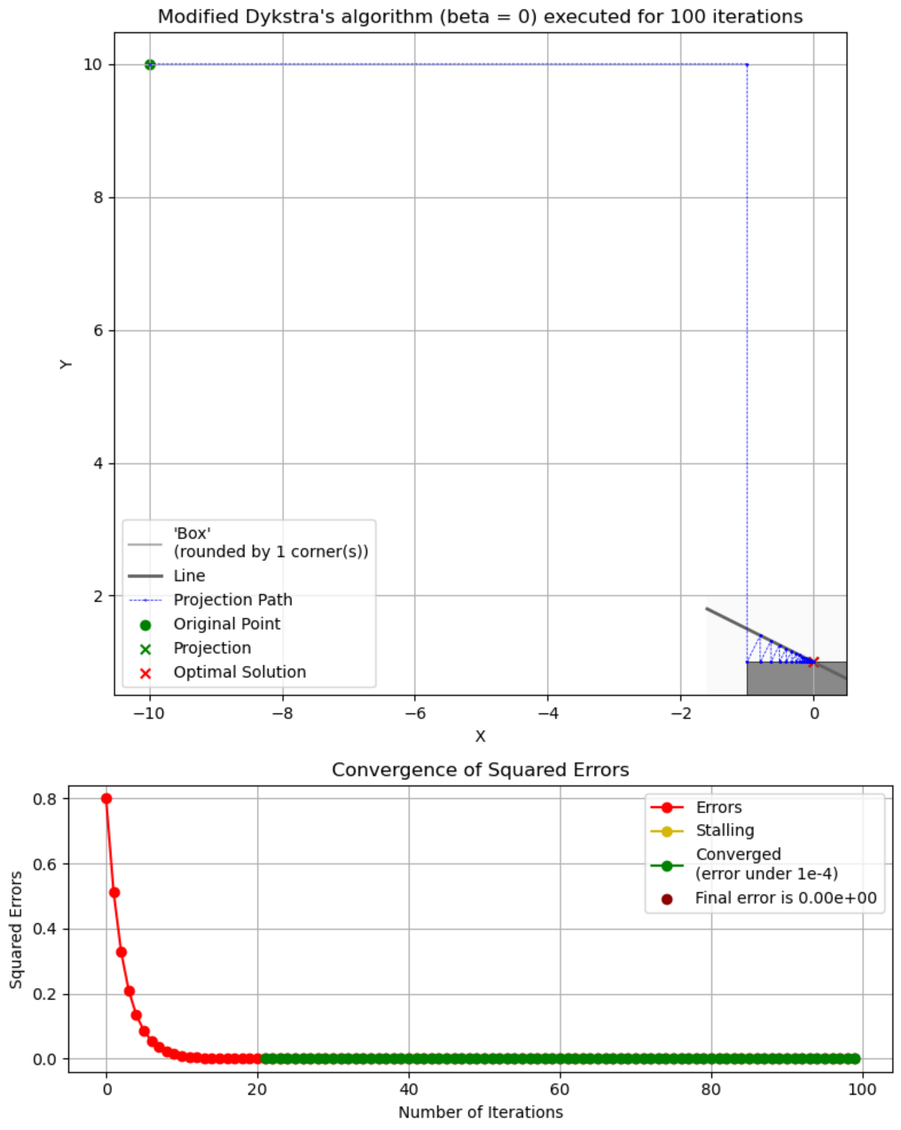


Figure 7: Dykstra's algorithm applied to initial point $z = (-10,\ 10)$, with $\beta_m = 0$. This is essentially applying MAP

7 Choice of β_m

As discussed in Section 3, MAP does not always produce the optimal solution as its output. In general, MAP will converge weakly to *some* point on the feasible region (i.e. locus of points which constitute the intersection of all halfspaces) [4]. More specifically, if $x_m \in \bigcap_{i=1}^{n} \mathcal{H}_i$, then $x_{m+1},\ x_{m+2},\ \ldots = x_m$. This constitutes MAP's biggest flaw, as seen in Figure 8.

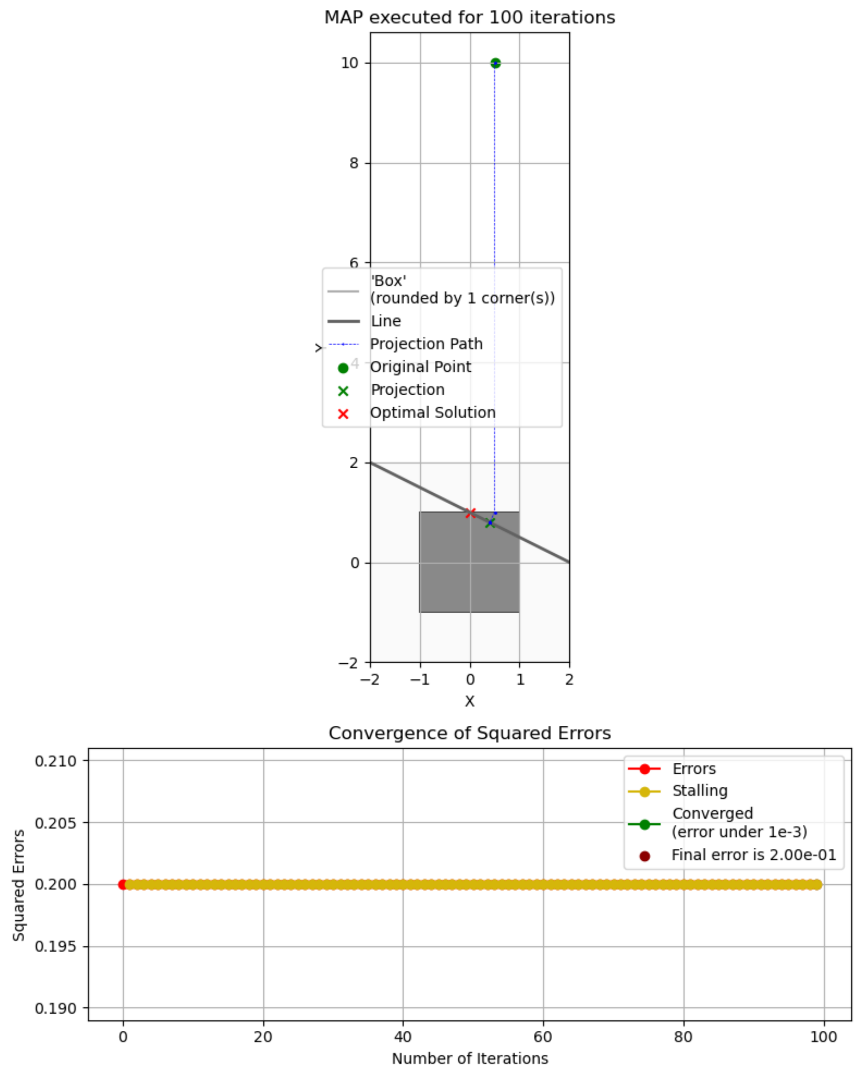


Figure 8: MAP applied to initial point $z = (0.5,\ 10)$. We land on a point on the intersection of all halfspaces on first iteration, hence algorithm does not proceed any further

However, when the initial point to be projected lies outside of $\bigcap_{i=1}^{n} \mathcal{H}_i$, and if $x_m \notin \bigcap_{i=1}^{n} \mathcal{H}_i \forall m$, then MAP will not stall and the series of iterates produced by MAP and Dykstra's method coincide, with infinite convergence (at least in this line and box example, but I think this can be generalised).

It follows that, if we combine the non-stalling nature of MAP with the certainty of producing an optimal solution of Dykstra's method, we could altogether avoid stalling. This is how we have decided to choose parameter β_m:

1. β_0 is initialised to 1
2. At the start of each iteration, if $x_m \notin \bigcap_{i=1}^{n} \mathcal{H}_i$, then β_{m+1} is set to 0, employing MAP for the next iteration.
3. Otherwise, β_{m+1} is set to 1, employing Dykstra's method instead

This simple modification will ensure we do not stall even if we are arbitrarily deep into the red region, and also that we will not terminate if at any point we land in the feasible region. It is important to note that, in order for this to work, we still have to *keep track* of the sequence of errors e_m we would obtain using Dykstra's algorithm, in case we need to use them in the next iteration. The improvements brought forward by this modification are highlighted in Figure 9 (compare this to Figure 5).

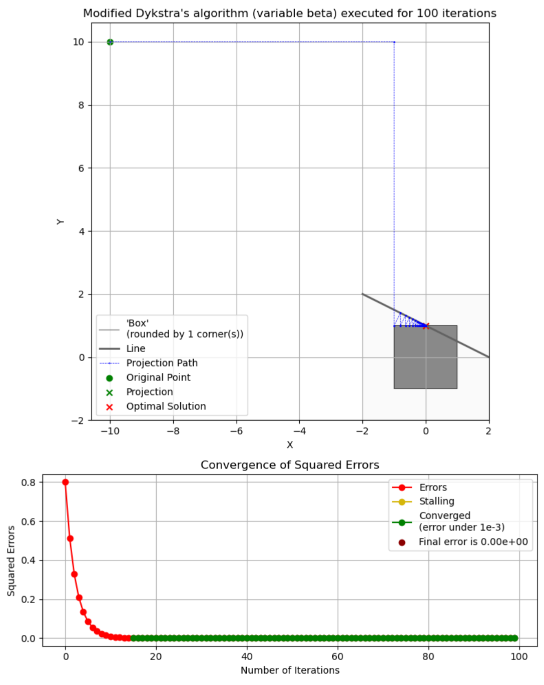


Figure 9: Modified method applied to initial point $z = (-10, 10)$.

It is also worth remarking that this new modified algorithm will also work in the circumstance where we land in the intersection at some point x_m. Compare Figure 8 to Figure 10 below:

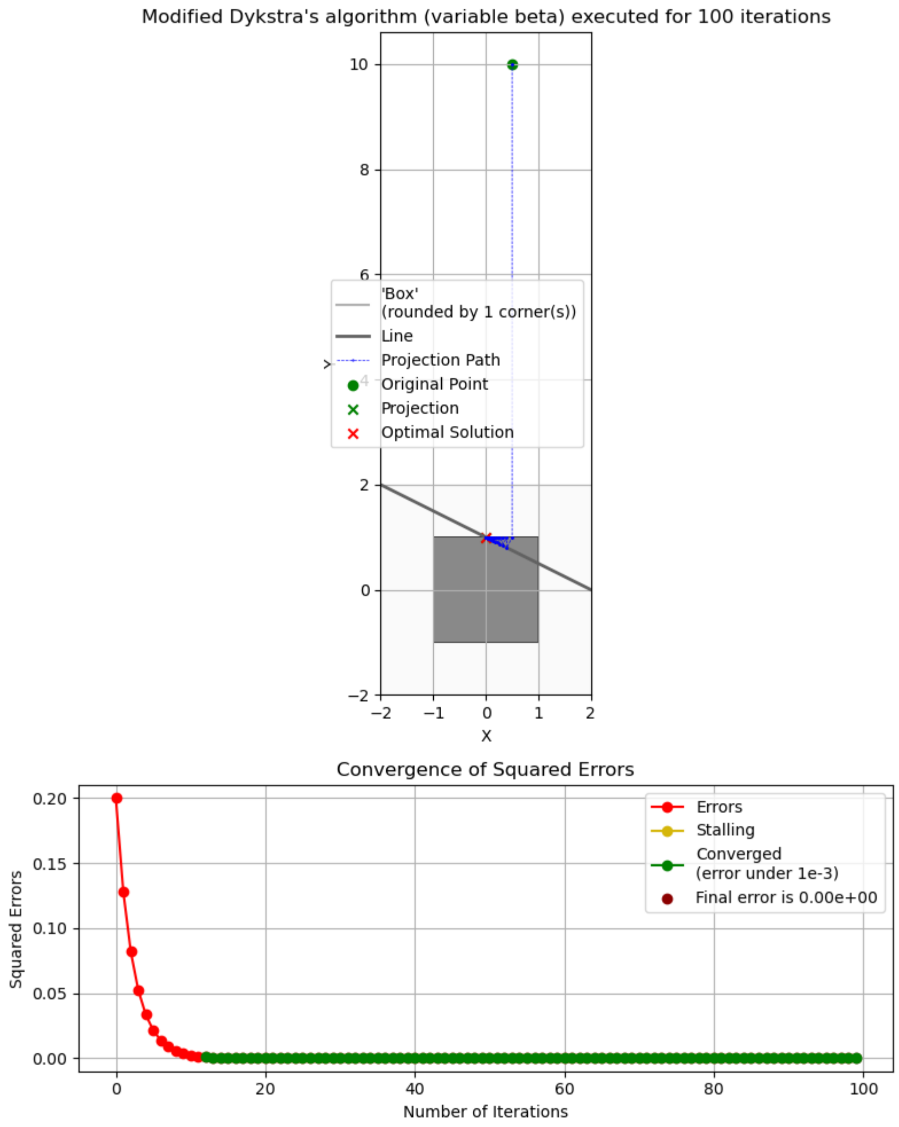


Figure 10: Modified method applied to initial point $z = (0.5,\ 10)$. In this case, β_m is set to 1 for all $m > 1$.

8 Thoughts

It seems that the modified algorithm proposed in Section 7 is able to completely avoid stalling in the line-box example proposed by Bauschke. We have implemented a new parameter β_m, which allows us to choose between MAP and Dykstra's method based on the current projection x_m. This new algorithm demonstrates at worst infinite convergence with no stalling, for any $z \in \mathbb{R}^2$, which constitutes a drastic improvement compared to the results of Figure 3a.

If we assume that MAP cannot stall (which follows intuition for convex sets), then to demonstrate our algorithm never stalls it would be sufficient to prove that *any intermediate step x_m which lies in the feasible region cannot lead to stalling* (note how an initial point $z \in \bigcap_{i=1}^{n} \mathcal{H}_i$ already constitutes the optimal solution). See Figure 11 for a practical example. This should cast our initial problem into a much more confined one.

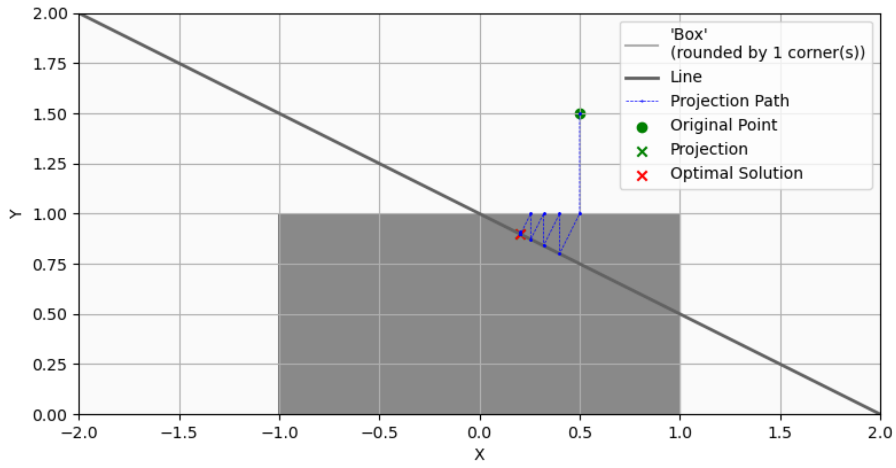


Figure 11: In an example such as this one, to show no stalling, one would have to prove that, if we land on one of the blue dots within the feasible region (line segment between (0, 1) and (1, 0.5)), then it is not possible to stall.

However, some questions still remain unanswered:

- Can we generalise the algorithm outlined in Section 7 to work with any choice of halfspaces and in any number of dimensions?
- The new algorithm requires we check every halfspace at the start of each iteration, which is expensive: can we keep track of which halfspaces remain active, and only check those to obtain a value of β_{m+1}?
- Is there any value in rounding the edges of our constraint sets?
- Can Dysktra's method be accelerated after escaping the stalling region (e.g. by choosing a value of $\beta_m \neq 1$)?

References

[1] H. H. Bauschke, R. S. Burachik, D. B. Herman, and C. Y. Kaya. On Dykstra's algorithm: finite convergence, stalling, and the method of alternating projections. *ArXiv e-prints*, pages 1–14, January 2020.

[2] F. Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[3] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in Order Restricted Statistical Inference*, pages 28–47. Springer New York, 1986.

[4] L.M. Bregman. The method of projections for finding the common point of convex sets. *Sov. Math. Dokl.*, 6:688–692, 1965.

[5] F. Deutsch and H. Hundal. The rate of convergence of Dykstra's cyclic projections algorithm: The polyhedral case. *Numerical Functional Analysis and Optimization*, 15(6):537–565, May 1994.

[6] Idris Kempf et al. Model predictive control for electron beam stabilization in a synchrotron. *IFAC*, Jul. 2020.

[7] W.P. Heath. Orthogonal functions for cross-directional control of web forming processes. *Automatica*, 32(2):183–198, February 1996.

[8] A. N. Iusem and A. R. de Pierro. On the convergence properties of Hildreth's quadratic programming algorithm. *Math. Prog. Ser. A and B*, 47(1):37–51, May 1990.

[9] Idris Kempf, Paul Goulart, and Stephen Duncan. Fast gradient method for model predictive control with input rate and amplitude constraints. *ArXiv e-prints*, March 2020.

[10] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer Science & Business Media, 2003.

[11] C. Perkins. A convergence analysis of Dykstra's algorithm for polyhedral sets. *SIAM J. Numer. Anal.*, 40(2):732–804, July 2002.

[12] Nguyen Mau Nam Shyan S. Akmal and J. J. P. Veerman. On a convex set with nondifferentiable metric projection. *El Pais, Spain*, 1989.

[13] S. Xu. Successive approximate algorithm for best approximation from a polyhedron. *J. Approx. Theory*, 93(3):415–433, June 1998.