Forward–Backward Splitting with Deviations for Monotone Inclusions

Hamed Sadeghi∗  Sebastian Banert∗  Pontus Giselsson∗

August 15, 2022

Abstract

We propose and study a weakly convergent variant of the forward–backward algorithm for solving structured monotone inclusion problems. Our algorithm features a per-iteration deviation vector which provides additional degrees of freedom. The only requirement on the deviation vector to guarantee convergence is that its norm is bounded by a quantity that can be computed online. This approach provides great flexibility and opens up for the design of new and improved forward–backward-based algorithms, while retaining global convergence guarantees. These guarantees include linear convergence of our method under a metric subregularity assumption without the need to adapt the algorithm parameters.

Choosing suitable monotone operators allows for incorporating deviations into other algorithms, such as Chambolle–Pock and Krasnosel’ski–Mann iterations. We propose a novel inertial primal–dual algorithm by selecting the deviations along a momentum direction and deciding their size using the norm condition. Numerical experiments demonstrate our convergence claims and show that even this simple choice of deviation vector can improve the performance, compared, e.g., to the standard Chambolle–Pock algorithm.

Key words. forward–backward splitting, monotone inclusions, global convergence, linear convergence rate, inertial forward–backward method, inertial primal–dual algorithm.

AMS Subject Classification (2020). 47H05, 47N10, 49M29, 65K05, 90C25.

1 Introduction

Forward–backward (FB) splitting [6, 22, 25] has been extensively used to solve structured monotone inclusion problems of finding $x \in \mathcal{H}$ such that

$$0 \in Ax + Cx,$$

Choosing suitable monotone operators allows for incorporating deviations into other algorithms, such as Chambolle–Pock and Krasnosel’ski–Mann iterations. We propose a novel inertial primal–dual algorithm by selecting the deviations along a momentum direction and deciding their size using the norm condition. Numerical experiments demonstrate our convergence claims and show that even this simple choice of deviation vector can improve the performance, compared, e.g., to the standard Chambolle–Pock algorithm.
where \( A: \mathcal{H} \to 2^\mathcal{H} \) is a maximally monotone operator, \( C: \mathcal{H} \to \mathcal{H} \) is a cocoercive operator, and \( \mathcal{H} \) is a real Hilbert space. The algorithm sequentially performs a forward step with the operator \( C \) followed by a backward step with \( A \) to arrive at the iteration

\[
x_{n+1} = (\text{Id} + \gamma_n A)^{-1} \circ (\text{Id} - \gamma_n C)x_n,
\]

where \( \gamma_n > 0 \) is a step-size parameter.

One of the most important special cases of this setting is first-order algorithms for convex optimization: let \( f: \mathcal{H} \to \mathbb{R} \) be a convex, differentiable function whose gradient is Lipschitz continuous and \( g: \mathcal{H} \to \mathbb{R} \cup \{\pm \infty\} \) be a proper, convex and lower semicontinuous function, and let \( A = \partial g \) (the subdifferential of \( g \)) and \( C = \nabla f \). Then, \( (1) \) is the problem of finding a minimizer of \( f + g \), and \( (2) \) describes the proximal gradient method \([11]\).

In this paper, we present a weakly convergent extension to the standard FB splitting method to solve the monotone inclusion \( (1) \). A simplified instance of our algorithm is given by

\[
p_n = (\text{Id} + \frac{1}{\beta} A)^{-1} \circ (\text{Id} - \frac{1}{\beta} C)(x_n + u_n)
\]

\[
x_{n+1} = p_n - u_n
\]

where \( u_n \) is a deviation (vector) and \( \frac{1}{\beta} > 0 \) is a cocoercivity constant of \( C \). By letting \( u_n = 0 \), a step of \( (3) \) reduces to the standard FB step in \( (2) \). The addition of \( u_n \) therefore gives added flexibility that can be utilized to improve performance. In order to ensure convergence of this algorithm, \( u_n \) has to satisfy the norm condition

\[
\|u_n\|^2 \leq \frac{1-\epsilon}{\epsilon^2} \|p_{n-1} - x_{n-1} + u_{n-1}\|^2,
\]

where \( \epsilon \in [0,1) \) is arbitrary and the quantity to the right-hand side of the inequality is computable online since the variables are known from previous iterations.

Safeguarding is a common technique to ensure global convergence in optimization algorithms, for instance the Wolfe conditions in line-search \([24] \) Chapter 3 ensure a sufficient decrease in the objective function value, and trust-region methods \([24] \) Chapter 4 are based on a quadratic model having sufficient accuracy within a given radius. Recently, a norm condition similar to \( (4) \) has been combined with a deep-learning approach to speed up the convergence \([4]\). Even for monotone operators, line-search strategies with safeguarding have been developed, see \([37] \) Eq. (2.4) for an example. In contrast to line search, \( (4) \) does not require to compute (and possibly reject) several steps per iteration. For other examples of safeguarding, see \([19, 32, 36, 40]\).

Our main algorithm (Algorithm \([1]\)) is more general than \( (3) \). It uses two deviation vectors and a slightly more involved safeguard condition. A similar algorithm with deviation vectors has been proposed in \([4]\) to extend the proximal gradient method for convex minimization. The fact that we consider the more general monotone inclusion setting, allows us to apply our results, e.g., to the
Chambolle–Pock [7] and Condat–Vũ [13, 39] methods that both are preconditioned FB methods [20]. To facilitate the derivation of these special cases, we derive our algorithm with explicit preconditioning, such as in [10, 12, 15, 16, 17, 18, 26, 29].

Our algorithm is also related to inexact FB methods, which are studied in the framework of monotone inclusions [28, 34, 35, 39] and in a convex optimization setting [13, 33, 38]. By including error terms in the FB splitting algorithms, these works allow for inaccuracies in the forward and backward step evaluations. The convergence of the algorithm is usually based on a summability assumption on the error sequences and would therefore allow arbitrarily large errors as long as they only happen for a finite number of iterations. The idea behind our method is in stark contrast to these methods, as our method is designed for actively choosing the deviations with the aim to improve performance.

We instantiate our general scheme in three special settings: the standard FB setting, the primal-dual setting of Condat–Vũ, and the Krasnosel’skiǐ–Mann setting. We also propose a further specialization of the primal-dual setting of Chambolle–Pock in which we select the deviations in a heavy-ball type [27] momentum direction (see [30] for another novel usage of the deviations in a primal–dual setting). The resulting algorithm bears similarities with the inertial FB methods [1, 2, 3, 9, 23] when applied in a primal–dual setting. Numerical experiments show improved performance of our method compared to Chambolle–Pock and a primal–dual version of Lorenz–Pock [23].

Contributions. The most notable differences of this work to existing literature can be summarized as follows:

– Compared to the standard FB, we extend the degrees of freedom by allowing the input argument to the FB operator to deviated from a pre-specified point.

– Unlike various known examples of momentum methods, the increase is not achieved with a fixed number of parameters, but the design parameter has the dimension of the underlying problem.

– In contrast to inexact FB algorithms [13, 28, 39, 38], the bound on the deviations is a scalar condition with known quantities in each step instead of a summability condition that has limited meaning for a finite number of steps.

– In contrast to the deviation-based FB method for convex optimization in [4], our work considers more general monotone inclusion problems. Hence, we immediately obtain the algorithms of Chambolle–Pock [7] and Krasnosel’skiǐ–Mann with deviations as special cases. Moreover, our convergence result is slightly stronger than [4, Theorem 3.2]. To the best of our knowledge, neither algorithm is a special case of the other.
In addition to showing weak convergence of our algorithm, we show that under a metric subregularity assumption the algorithm converges strongly to a point in the solution set of the problem with a linear rate of convergence.

As an example for the expressiveness of the deviation-based approach, we introduce a novel inertial primal–dual algorithm by selecting the deviations along a momentum direction—in the sense of Polyak [27]—and deciding their size using the norm condition.

Outline of the paper. The organization of the paper is as follows. In Section 2, we provide notations and some definitions. In Section 3, the proposed algorithm is introduced. In Section 4, we prove weak convergence of the method and linear and strong convergence under a metric subregularity assumption. In Section 5, some special cases of the proposed algorithm are presented and Section 6 further specializes one of these to arrive at a novel inertial primal–dual algorithm. We conclude the paper by presenting the numerical results in Section 7.

2 Preliminaries

Throughout the paper, the set of real numbers is denoted by \( \mathbb{R} \); \( \mathcal{H} \) and \( \mathcal{K} \) denote real Hilbert spaces that are equipped with inner products and induced norms, which are denoted by \( \langle \cdot, \cdot \rangle \) and \( \| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle} \), respectively. A bounded, self-adjoint operator \( M : \mathcal{H} \to \mathcal{H} \) is said to be strongly positive if there exists some \( c > 0 \) such that \( \langle x, Mx \rangle \geq c \|x\|^2 \) for all \( x \in \mathcal{H} \). We use the notation \( \mathcal{M}(\mathcal{H}) \) to denote the set of linear, self-adjoint, strongly positive operators on \( \mathcal{H} \). For \( M \in \mathcal{M}(\mathcal{H}) \) and for all \( x, y \in \mathcal{H} \), the \( M \)-induced inner product and norm are denoted by \( \langle x, y \rangle_M = \langle x, My \rangle \) and \( \|x\|_M = \sqrt{\langle x, Mx \rangle} \), respectively.

Young’s inequality \( \langle x, y \rangle \leq \frac{\omega}{2} \|x\|^2_M + \frac{1}{2\omega} \|y\|^2_{M^{-1}} \) holds for all \( x, y \in \mathcal{H} \), \( \omega > 0 \), and \( M \in \mathcal{M}(\mathcal{H}) \). Hence, with the same variables, \( \|x + y\|^2_M = \|x\|^2_M + \|y\|^2_M + 2\langle x, My \rangle \leq (1 + \omega)\|x\|^2_M + \frac{1 + \omega}{\omega} \|y\|^2_M \).

Let \( M \in \mathcal{M}(\mathcal{H}) \), \( x \in \mathcal{H} \), and \( S \subset \mathcal{H} \) be a nonempty closed convex set. The \( M \)-induced projection of \( x \) onto the set \( S \) is defined as \( \Pi_S^M x = \arg \min_{y \in S} \|x - y\|^2_M \), and the \( M \)-induced distance from \( x \) to \( S \) is defined by \( \text{dist}_M(x, S) = \|x - \Pi_S^M x\|^2_M \).

The notation \( 2^\mathcal{H} \) denotes the power set of \( \mathcal{H} \). A map \( A : \mathcal{H} \to 2^\mathcal{H} \) is characterized by its graph \( \text{gra}(A) = \{(x, u) \in \mathcal{H} \times \mathcal{H} : u \in Ax \} \). An operator \( A : \mathcal{H} \to 2^\mathcal{H} \) is monotone if \( \langle u - v, x - y \rangle \geq 0 \) for all \( (x, u), (y, v) \in \text{gra}(A) \). A monotone operator \( A : \mathcal{H} \to 2^\mathcal{H} \) is maximally monotone if there exists no monotone operator \( B : \mathcal{H} \to 2^\mathcal{H} \) such that \( \text{gra}(B) \) properly contains \( \text{gra}(A) \).

Let \( M \in \mathcal{M}(\mathcal{H}) \). An operator \( T : \mathcal{H} \to \mathcal{H} \) is said to be...
(i) $L$-Lipschitz continuous ($L \geq 0$) w.r.t. $\|\cdot\|_M$ if

$$\|Tx - Ty\|_{M^{-1}} \leq L\|x - y\|_M$$

for all $x, y \in \mathcal{H}$;

(ii) $\frac{1}{\beta}$-cocoercive ($\beta > 0$) w.r.t. $\|\cdot\|_M$ if

$$\langle Tx - Ty, x - y \rangle \geq \frac{1}{\beta} \|Tx - Ty\|_M^{2}$$

for all $x, y \in \mathcal{H}$;

(iii) nonexpansive if it is 1-Lipschitz continuous w.r.t. $\|\cdot\|$;

(iv) firmly nonexpansive if

$$\|Tx - Ty\|^2 + \|(Id - T)x - (Id - T)y\|^2 \leq \|x - y\|^2$$

for all $x, y \in \mathcal{H}$.

By the Cauchy–Schwarz inequality, a $\frac{1}{\beta}$-cocoercive operator is $\beta$-Lipschitz continuous. The resolvent of a maximally monotone operator $A: \mathcal{H} \to 2^{\mathcal{H}}$ is denoted by $J_{\gamma A}: \mathcal{H} \to \mathcal{H}$ and defined as $J_{\gamma A} := (Id + \gamma A)^{-1}$. $J_{\gamma A}$ has full domain, is firmly nonexpansive [5, Corollary 23.8], and is single-valued.

### 3 Forward–backward splitting with deviations

We consider structured monotone inclusion problems of the form

$$0 \in Ax + Cx,$$  \hspace{1cm} (5)

that satisfy the following assumptions.

**Assumption 1.** Assume that $\beta > 0$,  

(i) $A: \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone.

(ii) $C: \mathcal{H} \to \mathcal{H}$ is $\frac{1}{\beta}$-cocoercive with respect to $\|\cdot\|_M$ with $M \in \mathcal{M}(\mathcal{H})$.

(iii) The solution set $\text{zer}(A + C) := \{x \in \mathcal{H} : 0 \in Ax + Cx\}$ is nonempty.

Observe that, as a cocoercive operator is maximally monotone [5, Corollary 20.28], and since $C$ has a full domain, the operator $A + C$ is maximally monotone [5, Corollary 25.5].

We present and prove convergence for the following extended variant of FB splitting for solving (5).
Algorithm 1 Forward–backward splitting with deviations

1: **Input:** initial point \( x_0 \in \mathcal{H} \), the sequences \((\zeta_n)_{n \in \mathbb{N}}, (\lambda_n)_{n \in \mathbb{N}}, \text{ and } (\gamma_n)_{n \in \mathbb{N}}\) as per Assumption 2 and the metric \( \| \cdot \|_M \) with \( M \in \mathcal{M}(\mathcal{H}) \).

2: **set:** \( u_0 = v_0 = 0 \)

3: for \( n = 0, 1, 2, \ldots \) do

4: \( y_n = x_n + u_n \)

5: \( z_n = x_n + \frac{(1-\lambda_n)\gamma_n}{2-\lambda_n\gamma_n} u_n + v_n \)

6: \( p_n = (M + \gamma_n A)^{-1}(Mz_n - \gamma_n C y_n) \)

7: \( x_{n+1} = x_n + \lambda_n(p_n - z_n) \)

8: choose \( u_{n+1} \) and \( v_{n+1} \) such that

\[
\frac{\lambda_{n+1} \gamma_{n+1} \beta}{2-\lambda_{n+1} \gamma_{n+1} \beta} \|u_{n+1}\|_M^2 + \frac{\lambda_{n+1} (2-\lambda_{n+1} \gamma_{n+1} \beta)}{4-2\lambda_{n+1} \gamma_{n+1} \beta} \|v_{n+1}\|_M^2 \leq \zeta_n \ell_n^2
\]  

(6)

is satisfied, where

\[
\ell_n^2 = \frac{\lambda_n (4-2\lambda_n \gamma_n \beta)}{2} \|p_n - x_n + \frac{\lambda_n \gamma_n \beta}{2-\lambda_n \gamma_n \beta} u_n - \frac{2(1-\lambda_n)}{4-2\lambda_n - \gamma_n \beta} v_n\|_M^2
\]  

(7)

9: **end for**

Our convergence analysis requires that the parameter sequences \((\zeta_n)_{n \in \mathbb{N}}, (\lambda_n)_{n \in \mathbb{N}}, \text{ and } (\gamma_n)_{n \in \mathbb{N}}\) satisfy the following assumption.

**Assumption 2.** Choose \( \epsilon \in \left(0, \min \left(1, \frac{4}{3\beta} \right) \right) \), and assume that, for all \( n \in \mathbb{N} \), the following hold:

(i) \( 0 \leq \zeta_n \leq 1 - \epsilon \);

(ii) \( \epsilon \leq \gamma_n \leq \frac{4-3\epsilon}{\beta} \); and

(iii) \( \epsilon \leq \lambda_n \leq 2 - \frac{\gamma_n \beta}{2} - \frac{\epsilon}{2} \).

The sequence \((\zeta_n)_{n \in \mathbb{N}}\) relates the norm of the deviation vector \((u_{n+1}, v_{n+1})\) in (6) to its maximum permissible value; \((\gamma_n)_{n \in \mathbb{N}}\) is a sequence of step-size parameters for the FB step (6); and \((\lambda_n)_{n \in \mathbb{N}}\) can be seen as a sequence of relaxation parameters for \((x_n)_{n \in \mathbb{N}}\) in step 7.

For our convergence analysis in Section 4, we have to choose these sequences in such a way that all the coefficients multiplying the norms in (6) and (7) have a positive lower bound. Indeed, if \((\gamma_n)_{n \in \mathbb{N}}\) and \((\lambda_n)_{n \in \mathbb{N}}\) satisfy Assumption 2 then

\[
4 - 2\lambda_n - \gamma_n \beta \geq \epsilon
\]  

(8)

and

\[
2 - \lambda_n \gamma_n \beta \geq 2 - \left(2 - \frac{\gamma_n \beta}{2} - \frac{\epsilon}{2}\right) \gamma_n \beta = \frac{\epsilon \gamma_n \beta}{2} + 2 \left(1 - \frac{\gamma_n \beta}{2}\right)^2 \geq \frac{\epsilon^2 \beta}{2}.
\]  

(9)

Algorithm 1 handles the evaluation of \( C \) and \( A \) in step 6 differently than the standard FB method 2 in two ways. First, the operator \( M \) acts as a
preconditioning for the resolvent of $A$, and secondly, the points $y_n$ and $z_n$ can be different. Algorithm 1 also allows for deviations $u_n$ and $v_n$, which can be seen as design parameters of the algorithm. They can in general be chosen in a subset of $\mathcal{H}$ with non-empty interior (if $\ell^2_n > 0$ in step 8). Hence, the degrees of freedom in the parameter choice are determined by the dimension of $\mathcal{H}$. It is important to note that the upper bound $\ell^2_n$, as it is seen from (7), is computable at the time of selecting $u_{n+1}$ and $v_{n+1}$. See [31] for a generalization of Algorithm 1.

Below, we present some special cases of our method. We defer a more detailed discussion on special cases to Section 5.

**Example 1.** With the trivial choice of $u_{n+1} = v_{n+1} = 0$, the condition (6) is already satisfied, and Algorithm 1 reduces to the relaxed preconditioned FB iteration

$$p_n = (M + \gamma_n A)^{-1}(Mx_n - \gamma_n Cx_n),$$

$$x_{n+1} = x_n + \lambda_n(p_n - x_n).$$

With $M = \text{Id}$ and $\lambda_n = 1 \ (n \in \mathbb{N})$, we recover (2).

**Example 2.** With $M = \text{Id}$, $\gamma_n = \frac{2}{\beta}$, $\lambda_n = 1$, $v_n = u_n$, and $\zeta_n = 1 - \epsilon \ (n \in \mathbb{N})$, we recover the simplified version from (3) in Section 1. It is easy to see that this choice satisfies Assumption 2.

**Example 3 (no relaxation).** With $\lambda_n = 1$ for all $n \in \mathbb{N}$, Algorithm 1 simplifies to the iteration

$$p_n = (M + \gamma_n A)^{-1}(Mx_n + v_n - \gamma_n C(x_n + u_n)),$$

$$x_{n+1} = p_n - v_n$$

with the norm condition

$$\frac{\gamma_n+1}{2-\gamma_n+1}\|u_{n+1}\|^2_M + \|v_{n+1}\|^2_M \leq \frac{\zeta_n}{2} \left\| p_n - x_n + \frac{\gamma_n \beta}{2} u_n \right\|^2_M.$$

**Example 4 (forward iteration with deviations).** With $Ax = \{0\}$ for all $x \in \mathcal{H}$, $v_n = 0$, and $\gamma_n = 2/\beta$ for all $n \in \mathbb{N}$, Algorithm 1 simplifies to the iteration

$$y_n = x_n + u_n,$$

$$x_{n+1} = x_n - \frac{2\lambda_n}{\beta} M^{-1}Cy_n$$

with the norm condition

$$\frac{\lambda_n+1}{1-\lambda_n} \|u_{n+1}\|^2_M \leq \zeta_n \lambda_n (1 - \lambda_n) \left\| p_n - x_n - \frac{2\lambda_n}{\beta} M^{-1}Cy_n \right\|^2_M.$$

**Example 5 (backward iteration with derivations).** With $Cx = 0$ for all $x \in \mathcal{H}$ and $u_n = 0$ for all $n \in \mathbb{N}$, Algorithm 1 simplifies to the iteration

$$p_n = (M + \gamma_n A)^{-1}M(x_n + v_n),$$

$$x_{n+1} = x_n + \lambda_n(p_n - x_n - v_n).$$
Since $C$ is $1/\beta$-cocoercive for all $\beta > 0$, it is possible to set $\beta = 0$ in the norm condition, which then takes the form
\[
\frac{\lambda_{n+1}}{2 - \lambda_{n+1}} \|v_{n+1}\|_M^2 \leq \zeta_n \lambda_n (2 - \lambda_n) \left\| p_n - x_n - \frac{1 - \lambda_n}{2 - \lambda_n} v_n \right\|_M^2.
\]

**Remark 1.** Many works exist that allow for error terms in FB algorithms [13, 28, 39, 38]. Convergence is often based on a summability argument so that any summable sequence of errors is allowed. The strength of our condition (6) is that it is iteration-wise; hence, arbitrary large errors would not be accepted. A major difference is that our algorithm does not treat the deviations as errors or inaccuracies in the computation. Instead, they are introduced to allow for actively selecting the deviations with the aim to improve performance.

### 4 Convergence analysis

In this section, we provide a convergence analysis for Algorithm 1. We start by describing the points in the graph of $A + C$ constructed by Algorithm 1 (Lemma 1) and introducing a Lyapunov inequality in Lemma 2. Both results are later used to show weak convergence in Theorem 1 and strong and linear convergence under a metric subregularity assumption in Theorem 2.

**Lemma 1.** Suppose that Assumption 1 holds. Let $(x_n)_{n \in \mathbb{N}}$, $(y_n)_{n \in \mathbb{N}}$, $(z_n)_{n \in \mathbb{N}}$, and $(p_n)_{n \in \mathbb{N}}$ be sequences generated by Algorithm 1. Then, for all $n \in \mathbb{N}$, $(p_n, \Delta_n) \in \text{gra}(A + C)$, where
\[
\Delta_n := \frac{Mz_n - Mp_n}{\gamma_n} - (Cy_n - Cp_n).
\]

Moreover,
\[
\|\Delta_n\|_{M^{-1}} \leq \frac{1}{2\gamma_n} \left\| (2 - \beta \gamma_n)(x_n - p_n) - \frac{\lambda_n \gamma_n (2 - \gamma_n \beta)}{2 - \lambda_n \gamma_n \beta} u_n + 2v_n \right\|_M + \beta \|y_n - p_n\|_M
\]
\[
= \frac{1}{2\gamma_n} \left\| (2 - \beta \gamma_n)(x_n - p_n) - \frac{\lambda_n \gamma_n (2 - \gamma_n \beta)}{2 - \lambda_n \gamma_n \beta} u_n + 2v_n \right\|_M
\]
\[
+ \beta \|x_n - p_n + u_n\|_M.
\]

Before we prove Lemma 1, note that the right-hand side of (10) only contains data that is computed in Algorithm 1, whereas evaluating $\Delta_n$ requires the knowledge of $Cp_n$. Therefore, (10) can be used to check the accuracy of the current iteration or to define a stopping criterion without any extra evaluations of $C$.

In Example 2, (10) reduces to
\[
\|\Delta_n\| \leq \beta \|y_n - p_n\| = \beta \left\| \left( \text{Id} - \left( \text{Id} + \frac{1}{\beta} A \right)^{-1} \circ \left( \text{Id} - \frac{1}{\beta} C \right) \right) y_n \right\|.
\]
Hence, the right-hand side of (10) plays the role of a residual for the iteration in Algorithm 1.

**Proof of Lemma 2** Let $n \in \mathbb{N}$. Step 6 in Algorithm 1 is equivalent to the inclusion

$$
Mz_n - Mp_n - C y_n \in A p_n,
$$

(11)

to which adding $C p_n$ on both sides yields the desired inclusion $\Delta_n \in (A + C) p_n$. Furthermore, we have

\[
\begin{align*}
\|\Delta_n\|_{M^{-1}}^2 & - \left(1 + \frac{1}{2\gamma_n^2}\|2(z_n - p_n) - \beta\gamma_n(y_n - p_n)\|_M^2 + \frac{\beta}{2}\|y_n - p_n\|_M^2\right)^2 \\
= & \frac{1}{\gamma_n^2}\|z_n - p_n\|_M^2 + \|C y_n - C p_n\|_{M^{-1}}^2 - \frac{2}{\gamma_n}(z_n - p_n, C y_n - C p_n) \\
& - \frac{\beta}{2\gamma_n^2}\|y_n - p_n\|_M^2 + \frac{\beta\gamma_n}{2}(z_n - p_n, y_n - p_n)_M \\
& - \frac{1}{4\gamma_n}\|2(z_n - p_n) - \beta\gamma_n(y_n - p_n)\|_M^2 - \frac{\beta^2}{4}\|y_n - p_n\|_M^2 \\
& = \|C y_n - C p_n\|_{M^{-1}}^2 - \frac{2}{\gamma_n}(z_n - p_n, C y_n - C p_n) \\
& - \frac{\beta^2}{2}\|y_n - p_n\|_M^2 + \frac{\beta}{2\gamma_n}(z_n - p_n, y_n - p_n)_M \\
& - \frac{\beta}{2\gamma_n}\|2(z_n - p_n) - \beta\gamma_n(y_n - p_n)\|_M\|y_n - p_n\|_M \\
& + \frac{1}{\gamma_n}\left(2(z_n - p_n) - \beta\gamma_n(y_n - p_n)_M, \frac{\beta}{2}M(y_n - p_n) - (C y_n - C p_n)\right)_M \\
& - \frac{\beta}{2\gamma_n}\|2(z_n - p_n) - \beta\gamma_n(y_n - p_n)\|_M\|y_n - p_n\|_M.
\end{align*}
\]

(12)

Notice that, by the $1/\beta$-cocoercivity of $C$ w.r.t. $\|\cdot\|_M$,

$$
\|y_n - p_n\|_M \geq \frac{2}{\beta}\|C y_n - C p_n - \frac{\beta}{2}M(y_n - p_n)\|_{M^{-1}}.
$$

(13)

The inequality part in (10) then follows from (12), using the $1/\beta$-cocoercivity again, inserting (13), and applying the Cauchy–Schwarz inequality. The equality in (10) is easily obtained by inserting the definitions of $y_n$ and $z_n$. $\square$

**Lemma 2** (Lyapunov inequality). Suppose that Assumptions 1 and 2 hold. Let $(x_n)_{n\in\mathbb{N}}, (u_n)_{n\in\mathbb{N}}, (v_n)_{n\in\mathbb{N}}, (\ell_n^2)_{n\in\mathbb{N}}$ be sequences generated by Algorithm 1 and $x^*$ be an arbitrary point in zero $(A + C)$. Then,

\[
\|x_{n+1} - x^*\|_M + \ell_n^2 \leq \|x_n - x^*\|_M + \frac{\lambda_n\gamma_n^2}{2 - \lambda_n\gamma_n^2}\|u_n\|_M^2 + \frac{\lambda_n(2 - \lambda_n\gamma_n^2)}{4 - 2\lambda_n - \gamma_n^2}\|v_n\|_M^2
\]

(14)

9
and
\[ \|x_{n+1} - x^*\|^2_M + \ell_n^2 \leq \|x_n - x^*\|^2_M + \zeta_n \ell_{n-1}^2 \] (15)
hold for all \( n \in \mathbb{N} \).

**Proof.** Let \( n \in \mathbb{N} \) be arbitrary. Step 6 in Algorithm 1 is equivalent to the inclusion
\[ \frac{Mz_n - Mp_n}{\gamma_n} - Cy_n \in Ap_n. \] (16)
Since \( x^* \in \text{zer}(A + C) \), we also have
\[ -Cx^* \in Ax^*. \] (17)
Using (16), (17), and the monotonicity of \( A \) gives
\[ 0 \leq \left\langle \frac{Mz_n - Mp_n}{\gamma_n} - Cy_n + Cx^*, p_n - x^* \right\rangle. \] (18)
By the \( 1/\beta \)-cocoercivity of \( C \) w.r.t. \( \| \cdot \|_M \) we have
\[ \frac{1}{\beta} \|Cy_n - Cx^*\|_M^2 \leq \langle Cy_n - Cx^*, y_n - x^* \rangle. \] (19)
Adding (18) and (19) yields
\[ 0 \leq \left\langle \frac{Mz_n - Mp_n}{\gamma_n}, p_n - x^* \right\rangle + \langle Cy_n - Cx^*, y_n - p_n \rangle - \frac{1}{\beta} \|Cy_n - Cx^*\|_M^2. \]
Then, from step 7 in Algorithm 1 we substitute \( z_n - p_n = \frac{1}{x_n}(x_n - x_{n+1}) \) to obtain
\[
0 \leq \frac{1}{2 \gamma_n \lambda_n} \langle x_n - x_{n+1}, p_n - x^* \rangle_M + \langle Cy_n - Cx^*, y_n - p_n \rangle - \frac{1}{\beta} \|Cy_n - Cx^*\|_M^2.
\]
where we use the identity \( 2(a - b, c - d)_M = \|a - d\|_M^2 + \|b - c\|_M^2 - \|a - c\|_M^2 - \|b - d\|_M^2 \) for all \( a, b, c, d \in \mathcal{H} \) and Young’s inequality. Multiplying both sides of the last inequality by \( 2 \gamma_n \lambda_n \) and reordering the terms yield
\[ \|x_{n+1} - x^*\|^2_M - \|x_n - x^*\|^2_M \]
\[ \leq \|x_{n+1} - p_n\|^2_M - \|x_n - p_n\|^2_M + \frac{\lambda_n \gamma_n}{2} \|y_n - p_n\|^2_M \]
\[ = \|x_n - p_n + \lambda_n(p_n - z_n)\|^2_M - \|x_n - p_n\|^2_M + \frac{\lambda_n \gamma_n}{2} \|y_n - p_n\|^2_M \]
\[ = \lambda_n^2 \|p_n - z_n\|^2_M + 2 \lambda_n \langle x_n - p_n, p_n - z_n \rangle_M + \frac{\lambda_n \gamma_n}{2} \|y_n - p_n\|^2_M \]
\[ = -\lambda_n (2 - \lambda_n) \|p_n - z_n\|^2_M + 2 \lambda_n \langle p_n - z_n, x_n - z_n \rangle_M + \frac{\lambda_n \gamma_n}{2} \|y_n - p_n\|^2_M \] (20)

10
where we, once again, used step 7 in Algorithm 1 to substitute back $x_{n+1} = x_n + \lambda_n (p_n - z_n)$ into the expression to the right-hand side of the inequality. Now, using the definitions of $y_n$ and $z_n$ in steps 4 and 5 of Algorithm 1, we observe that

$$
\ell_n^2 = \left( \lambda_n (2 - \lambda_n) - \frac{\lambda_n \gamma_n \beta}{2} \right) \left| p_n - x_n + \frac{\lambda_n \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n + \frac{2(1 - \lambda_n)}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right|^2_M \tag{21}
$$

$$= \lambda_n (2 - \lambda_n) \left| p_n - z_n \right|^2_M + \lambda_n (2 - \lambda_n) \left| \frac{\gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - \frac{2 - 2 \gamma_n \beta}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right|^2_M$$

$$+ 2 \lambda_n (2 - \lambda_n) \left( p_n - z_n, \frac{\gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - \frac{2 - 2 \gamma_n \beta}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right)_M$$

$$- \frac{\lambda_n \gamma_n \beta}{2} \left| p_n - y_n \right|^2_M - \frac{\lambda_n \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n + \frac{2(1 - \lambda_n)}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right|^2_M$$

$$- \lambda_n \gamma_n \beta \left( p_n - y_n, \frac{2}{2 - \lambda_n \gamma_n \beta} u_n + \frac{2(1 - \lambda_n)}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right)_M.$$ 

We can estimate the left-hand side of (14) by adding (20) and (21). Let us do this step by step. First, let us look at the two inner products with $p_n - z_n$.

$$2 \lambda_n \left( p_n - z_n, x_n - z_n + (2 - \lambda_n) \left( \frac{\gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - \frac{2 - 2 \gamma_n \beta}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right) \right)_M$$

$$= 2 \lambda_n \left( p_n - z_n, \frac{\gamma_n \beta(2 - \lambda_n)}{2 - \lambda_n \gamma_n \beta} u_n - \frac{(1 - \lambda_n) \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} v_n - \left( 1 + \frac{(2 - \lambda_n)(2 - 2 \gamma_n \beta)}{\gamma_n \beta - 2(2 - \lambda_n)} \right) v_n \right)_M$$

$$= 2 \lambda_n \left( p_n - z_n, \frac{\gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n + \frac{(1 - \lambda_n) \gamma_n \beta}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right)_M.$$

This can be combined with the last term in (21), so that we get

$$\left\| x_{n+1} - x^* \right\|^2_M - \left\| x_n - x^* \right\|^2_M + \ell_n^2 \leq 2 \lambda_n \gamma_n \beta \left( y_n - z_n, \frac{1}{2 - \lambda_n \gamma_n \beta} u_n + \frac{(1 - \lambda_n)}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right)_M$$

$$+ \lambda_n (2 - \lambda_n) \left| \frac{\gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - \frac{2 - 2 \gamma_n \beta}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right|^2_M$$

$$- \lambda_n \gamma_n \beta \left| \frac{1}{2 - \lambda_n \gamma_n \beta} u_n + \frac{(1 - \lambda_n)}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right|^2_M \tag{22}.$$ 

With $y_n - z_n = \frac{2 - 2 \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - v_n$, the right-hand side of (22) is a quadratic expression in $u_n$ and $v_n$ alone:

$$\left\| x_{n+1} - x^* \right\|^2_M - \left\| x_n - x^* \right\|^2_M + \ell_n^2 \leq 2 \lambda_n \gamma_n \beta \left( \left\| \frac{1 - \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - \frac{(1 - \lambda_n)}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right\|_M + \frac{1}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right\|_M$$

$$+ \lambda_n (2 - \lambda_n) \left| \frac{\gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - \frac{2 - 2 \gamma_n \beta}{\gamma_n \beta - 2(2 - \lambda_n)} v_n \right|^2_M.$$ 

In order to verify (14), it suffices to check the coefficients of $\left\| u_n \right\|^2_M$, $\left\| v_n \right\|^2_M$, and
\[ \langle u_n, v_n \rangle_M \] on the right-hand side. This results in
\[
\|x_{n+1} - x^*\|_M^2 - \|x_n - x^*\|_M^2 + \ell_n^2
\]
\[
\leq 2 \lambda_n \gamma_n \beta (1 - \gamma_n \beta) + \lambda_n \gamma_n \beta (2 - \lambda_n) \|u_n\|_M^2 + \frac{-2 \lambda_n \gamma_n \beta (1 - \gamma_n \beta) (1 - \lambda_n) + \lambda_n (2 - \lambda_n) (2 - \gamma_n \beta)^2}{(2 - \lambda_n \gamma_n \beta)^2} \|v_n\|_M^2 + \frac{2 \lambda_n \gamma_n \beta (1 - \gamma_n \beta) (1 - \lambda_n) - 2 \lambda_n \gamma_n \beta (2 - \lambda_n) (2 - \gamma_n \beta)}{(2 - \lambda_n \gamma_n \beta)^2} \|u_n - x^*\|_M^2 + \frac{2 \lambda_n \gamma_n \beta (2 - \lambda_n) (2 - \gamma_n \beta)}{(2 - \lambda_n \gamma_n \beta)^2} \langle u_n, v_n \rangle_M
\]
showing (14). Finally, (15) follows from inserting (6).

The following theorem is the main convergence result of the paper that guarantees weak convergence for the sequence of iterates obtained from Algorithm 1.

**Theorem 1.** Suppose that Assumptions 1 and 2 hold. Let the sequences \((x_n)_{n \in \mathbb{N}}, (u_n)_{n \in \mathbb{N}}, (v_n)_{n \in \mathbb{N}}, \) and \((\ell_n^2)_{n \in \mathbb{N}}\) be generated by Algorithm 1. Then, the following hold:

(i) The sequence \((\ell_n^2)_{n \in \mathbb{N}}\) is summable and the sequences \((u_n)_{n \in \mathbb{N}}\) and \((v_n)_{n \in \mathbb{N}}\) are convergent to zero.

(ii) For all \(x^* \in \text{zer}(A + C)\), the sequence \((\|x_n - x^*\|_M)_{n \in \mathbb{N}}\) converges.

(iii) The sequence \((x_n)_{n \in \mathbb{N}}\) converges weakly to a point in \(\text{zer}(A + C)\).

**Proof.** We start by proving (14) via a telescoping argument for (15). To this end, let \(N \in \mathbb{N}\). We sum (15) for \(n = 1, 2, \ldots, N\) to obtain
\[
\|x_{N+1} - x^*\|_M^2 + \ell_N^2 + \sum_{n=1}^{N-1} (1 - \zeta_n) \ell_n^2 \leq \|x_1 - x^*\|_M^2 + \zeta_0 \ell_0^2.
\]
Then, rearranging the terms gives
\[
\sum_{n=1}^{N} (1 - \zeta_n) \ell_n^2 \leq \|x_1 - x^*\|_M^2 - \|x_{N+1} - x^*\|_M^2 - \zeta_N \ell_N^2
\]
\[
\leq \|x_1 - x^*\|_M^2 + \zeta_0 \ell_0^2.
\]
Since the right hand side of the last inequality is independent of \(N\), we conclude that
\[
\sum_{n=0}^{\infty} (1 - \zeta_n) \ell_n^2 < \infty,
\]
which, along with \(\zeta_n \leq 1 - \epsilon\) from Assumption 2 implies that
\[
\ell_n^2 \to 0
\]
as \(n \to \infty\). Then, (6) implies that \(u_n \to 0\) and \(v_n \to 0\) as \(n \to \infty\). This proves (14).
The proof of Theorem 1(ii) follows from the property that \(\|x_n - x^*\|_M^2 + \ell_n^2 \leq \|x_n - x^*\|_M^2 + \ell_{n-1}^2\), i.e., the sequence \(\left(\|x_n - x^*\|_M^2 + \ell_n^2\right)_{n \in \mathbb{N}}\) is nonincreasing. As it is also nonnegative, it is convergent, say \(\|x_n - x^*\|_M^2 + \ell_n^2 \to \ell_{x^*} \geq 0\) as \(n \to \infty\). Moreover, \(\ell_n^2 \to 0\) by Theorem 1(i) as \(n \to \infty\), so \(\|x_n - x^*\|_M^2 \to \ell_{x^*}\), proving Theorem 1(ii).

For the proof of Theorem 1(iii) recall that \((p_n, \Delta_n) \in \text{gra}(A+C)\) for all \(n \in \mathbb{N}\) by Lemma 1. Now, by (8), we have \(\frac{\lambda_n(4 - 2\gamma_n \gamma_n)}{2 - 2\lambda_n - \gamma_n \beta} \geq \epsilon^2 / 2\) for all \(n \in \mathbb{N}\). By this and \(\ell_n \to 0\) as \(n \to \infty\), we have that
\[
p_n - x_n + \frac{\lambda_n \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n + \frac{2(\lambda_n - 1) \gamma_n \beta}{4 - 2\lambda_n - \gamma_n \beta} v_n \to 0.
\]
Next, from \(u_n \to 0\) and \(v_n \to 0\), together with (8) and (9), we conclude that \(p_n - x_n \to 0\) as \(n \to \infty\). Then, by Lemma 4
\[
\|\Delta_n\|_M^{-1} \leq \frac{1}{2\gamma_n} \left\| (2 - \beta \gamma_n) (x_n - p_n) - \frac{\lambda_n \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n + 2v_n \right\|_M
\]
\[
\quad \quad \quad \quad \quad + \frac{\beta}{2} \|x_n - p_n + u_n\|_M,
\]
hence, \(\Delta_n \to 0\) as \(n \to \infty\).

Now, from Theorem 1(ii) we know that \(\left(\|x_n - x^*\|_M^2\right)_{n \in \mathbb{N}}\) is convergent, which implies that the sequence \((x_n)_{n \in \mathbb{N}}\) is bounded. Therefore, the latter has at least one weakly convergent subsequence \((x_{k_n})_{n \in \mathbb{N}}\), say \(x_{k_n} \rightharpoonup x^\ast_{wc} \in \mathcal{H}\) as \(n \to \infty\). By the arguments above, we have \(p_{k_n} \to x^\ast_{wc}\) and \(\Delta_{k_n} \to 0\). Therefore, \((x^\ast_{wc}, 0) \in \text{gra}(A+C)\) by the weak–strong closedness of \(\text{gra}(A+C)\) [5, Proposition 20.38]. Then, Theorem 1(iii) follows from [5 Lemma 2.47], and the proof is complete.

### 4.1 Linear convergence

In this section, we show the linear convergence of Algorithm 1 under the following metric subregularity assumption.

**Definition 1 (M-metric subregularity).** Let \(M \in \mathcal{M}(\mathcal{H})\). A mapping \(T : \mathcal{H} \to 2^\mathcal{H}\) is called \(M\)-metrically subregular at \(x^\ast\) for \(y^\ast\) if \((x^\ast, y^\ast) \in \text{gra}(T)\) and there exists a \(\kappa \geq 0\) along with neighborhoods \(U\) of \(x^\ast\) and \(V\) of \(y^\ast\) such that
\[
\text{dist}_M(x, T^{-1}(y^\ast)) \leq \kappa \text{dist}^{-1}_M(y^\ast, T(x) \cap V)
\]
for all \(x \in U\).
This definition is equivalent to that in [14], but uses the $M$- and $M^{-1}$-induced norm distances instead of the standard canonical norm distance. Using this definition simplifies the notation in the linear convergence analysis. Metric subregularity is an important notion in numerical analysis. For a set-valued operator $T$ and an input vector $\bar{y}$, it simply provides an upper bound of how far a point $x$ is from being a solution to inclusion problem $\bar{y} \in T(x)$. This upper bound is given by [24] in terms of the distance of $T(x)$ from the input vector $\bar{y}$. For a detailed discussion on this subject, see [14].

**Theorem 2** (linear convergence). Consider the monotone inclusion problem [5] and suppose that Assumptions 2 and 3 hold, that $A + C$ is $M$-metrically subregular at all $x^* \in \text{zer}(A + C)$ for 0, and that either $\mathcal{H}$ is finite-dimensional or that in Definition 7 the neighborhood $\mathcal{U}$ at all $x^* \in \text{zer}(A + C)$ is the whole space $\mathcal{H}$. Then, there exists $0 \leq q < 1$ such that the following statements hold.

(i) There exists $0 < \delta < 1$ such that

$$
dist_M^2(x_{n+1}, \text{zer}(A + C)) + (1 - \delta)\ell_n^2 
\leq q \left( dist_M^2(x_n, \text{zer}(A + C)) + (1 - \delta)\ell_{n-1}^2 \right)
$$

for all $n \geq 1$;

(ii) there exist $x^* \in \text{zer}(A + C)$ and $c > 0$ such that $\|x_n - x^*\|^2 \leq cq^n$ for all $n \geq 1$. Hence, $x_n \to x^*$ even if $\mathcal{H}$ is infinite-dimensional.

**Proof.** We start by proving (i). Let $x^* \in \text{zer}(A + C)$ be the weak cluster point of the sequences $(x_n)_{n \in \mathbb{N}}$ and $(p_n)_{n \in \mathbb{N}}$ according to Theorem 4. From the metric subregularity of $A + C$ at $x^*$ for 0, we get $\kappa \geq 0$ and neighborhoods $\mathcal{U}$ of $x^*$ and $\mathcal{V}$ of 0 such that

$$
dist_M(x, \text{zer}(A + C)) \leq \kappa \text{dist}_{M^{-1}}(0, (A + C)(x) \cap \mathcal{V})
$$

(25)

for all $x \in \mathcal{U}$.

If $\mathcal{H}$ is finite-dimensional, then $p_n \to x^*$, and there exists $n_0 \in \mathbb{N}$ such that $p_n \in \mathcal{U}$ for all $n \geq n_0$. If $\mathcal{H}$ is infinite-dimensional, then $\mathcal{U} = \mathcal{H}$, and $p_n \in \mathcal{U}$ for all $n \in \mathbb{N}$.

Now, Lemma 1 gives $\Delta_n \in (A + C)p_n$ for all $n \in \mathbb{N}$, and $\Delta_n \to 0$ by the proof of Theorem 1. Let $n_0 \in \mathbb{N}$ be chosen such that $\Delta_n \in \mathcal{V}$ in addition to $p_n \in \mathcal{U}$ for all $n \geq n_0$. Setting $x = p_n$ in (25) hence gives

$$
dist_M(p_n, \text{zer}(A + C)) 
\leq \kappa \text{dist}_{M^{-1}}(0, (A + C)(p_n) \cap \mathcal{V}) 
\leq \kappa \|\Delta_n\|_{M^{-1}} 
\leq \frac{\kappa}{2\gamma_n} \left( (2 - \beta \gamma_n)(x_n - p_n) - \frac{\lambda_n \gamma_n \beta(2 - \gamma_n \beta)}{2 - \lambda_n \gamma_n \beta} u_n + 2v_n \right)_{M} 
\quad + \frac{\beta \kappa}{2} \|x_n - p_n + u\|_{M}
$$

(26)
for all $n \geq n_0$, where we used (10) in the last step. From Lemma 2 we have that

$$\|x_{n+1} - x^*\|_M^2 + \ell_n^2 \leq \|x_n - x^*\|_M^2 + \zeta_{n-1} \ell_{n-1}^2.$$  \hspace{1cm} (27)

Now, set $x^*_n := \Pi_{\zer(A+C)}^n(x_n)$. Then, from (27), we get

$$\text{dist}^2_M(x_{n+1}, \zer(A+C)) + \ell_n^2 \leq \|x_{n+1} - x^*_n\|_M^2 + \ell_n^2 \leq \|x_n - x^*_n\|_M^2 + \zeta_{n-1} \ell_{n-1}^2 = \text{dist}^2_M(x_n, \zer(A+C)) + \zeta_{n-1} \ell_{n-1}^2.$$  \hspace{1cm} (28)

Next, we will estimate both sides of (26) in terms of $\text{dist}^2_M(x_n, \zer(A+C))$, $\ell_n^2$, and $\ell_{n-1}^2$. Let $p^*_n := \Pi_{\zer(A+C)}^n(p_n)$. Then, since $\Pi_{\zer(A+C)}$ is the projection onto a convex set w.r.t. the $M$-induced metric, [5] Theorem 3.16 yields

$$\text{dist}^2_M(p_n, \zer(A+C)) \geq \|p_n - p^*_n\|_M^2 - 2(x_n - x_n, p_n - p^*_n)_M = \|p_n - p^*_n\|_M^2 - 2(x_n - x_n, p_n - p^*_n)_M - 2(x_n - x_n)_M \leq \|x_n - x^*_n\|_M^2 - 2(x_n - x_n)_M \geq \frac{1}{2} \|x_n - x^*_n\|_M^2 - 2\|p_n - x_n\|_M^2,$$

where we used Young’s inequality in the last step. Combining this with (26) gives

$$\frac{1}{2} \text{dist}^2_M(x_n, \zer(A+C)) \leq \frac{\kappa}{2\gamma_n} \left| (2 - \beta \gamma_n)(x_n - p_n) - \frac{\lambda_n \gamma_n \beta (2 - \gamma_n \beta)}{2 - \lambda_n \gamma_n \beta} u_n + 2v_n \right|_M^2 \leq \frac{\beta \kappa}{2} \|x_n - p_n + u_n\|_M^2 + 2\|p_n - x_n\|_M^2 \leq \frac{\kappa^2}{2\gamma_n^2} \left| (2 - \beta \gamma_n)(x_n - p_n) - \frac{\lambda_n \gamma_n \beta (2 - \gamma_n \beta)}{2 - \lambda_n \gamma_n \beta} u_n + 2v_n \right|_M^2 \leq \frac{\beta^2 \kappa^2}{2} \|x_n - p_n + u_n\|_M^2 + 2\|p_n - x_n\|_M^2,$$

where we used Young’s inequality in the last step. It remains to estimate the right-hand side of (29) in terms of $\ell_n^2$ and $\ell_{n-1}^2$. To this end, we use the following lemma.

**Lemma 3.** Let $(x_n)_{n \in \mathbb{N}}$, $(p_n)_{n \in \mathbb{N}}$, $(u_n)_{n \in \mathbb{N}}$, $(v_n)_{n \in \mathbb{N}}$, and $(\ell_n^2)_{n \in \mathbb{N}}$ be generated by Algorithm 1 under Assumption 3 and let $(a_n)_{n \in \mathbb{N}}$, $(b_n)_{n \in \mathbb{N}}$, and $(\ell_n)_{n \in \mathbb{N}}$ be bounded sequences of real numbers. Then there exist $c_1, c_2 > 0$ (which do not depend on $n$) such that

$$\|a_n(p_n - x_n) + b_n u_n + c_n v_n\|_M^2 \leq c_1 \ell_n^2 + c_2 \ell_{n-1}^2.$$
Proof. The assertion is proven by repeatedly applying Young’s inequality and subsequently using the norm condition (6):

\[ \|a_n(p_n - x_n) + b_n u_n + c_n v_n\|_M^2 \]

\[ = \|a_n \left( p_n - x_n + \frac{\lambda_n \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - \frac{2(1 - \lambda_n)}{4 - 2 \lambda_n - \gamma_n \beta} v_n \right) \]

\[ + \left( b_n - \frac{\lambda_n \gamma_n \beta a_n}{2 - \lambda_n \gamma_n \beta} \right) u_n + \left( c_n + \frac{2a_n(1 - \lambda_n)}{4 - 2 \lambda_n - \gamma_n \beta} \right) v_n \|_M^2 \]

\[ \leq 2a_n^2 \|p_n - x_n + \frac{\lambda_n \gamma_n \beta}{2 - \lambda_n \gamma_n \beta} u_n - \frac{2(1 - \lambda_n)}{4 - 2 \lambda_n - \gamma_n \beta} v_n\|_M^2 \]

\[ + 2 \left\| \left( b_n - \frac{\lambda_n \gamma_n \beta a_n}{2 - \lambda_n \gamma_n \beta} \right) u_n + \left( c_n + \frac{2a_n(1 - \lambda_n)}{4 - 2 \lambda_n - \gamma_n \beta} \right) v_n \right\|_M^2 \]

\[ \leq \frac{4a_n^2}{\lambda_n(4 - 2 \lambda_n - \gamma_n \beta)} \ell_n^2 \]

\[ + 4 \left( b_n - \frac{\lambda_n \gamma_n \beta a_n}{2 - \lambda_n \gamma_n \beta} \right)^2 \|u_n\|_M^2 + 4 \left( c_n + \frac{2a_n(1 - \lambda_n)}{4 - 2 \lambda_n - \gamma_n \beta} \right)^2 \|v_n\|_M^2 \]

\[ \leq \frac{4a_n^2}{\lambda_n(4 - 2 \lambda_n - \gamma_n \beta)} \ell_n^2 + 4d_n \zeta_n - 1 \ell_n^2 \]

with

\[ d_n := \max \left\{ \frac{2 - \lambda_n \gamma_n \beta}{\lambda_n \gamma_n \beta} \left( b_n - \frac{\lambda_n \gamma_n \beta a_n}{2 - \lambda_n \gamma_n \beta} \right)^2 , \right. \]

\[ \left. \frac{4 - 2 \lambda_n - \gamma_n \beta}{\lambda_n(2 - \lambda_n \gamma_n \beta)} \left( c_n + \frac{2a_n(1 - \lambda_n)}{4 - 2 \lambda_n - \gamma_n \beta} \right)^2 \right\} \].

It is straightforward to show, by using Assumption 2, that \( \frac{4a_n^2}{\lambda_n(4 - 2 \lambda_n - \gamma_n \beta)} \) and \( 4d_n \zeta_n - 1 \ell_n^2 \) are bounded, completing the proof.

Now, we are in the position to complete the argument of this section’s main result.

Proof of Theorem 2 continued. Since all the relevant coefficients on the right-hand side of (29) are bounded due to Assumption 2 using Lemma 3 on all the norms and combining the results yields \( c_1, c_2 > 0 \) such that

\[ \frac{1}{2} \text{dist}_M^2(x_n, \text{zer}(A + C)) \leq c_1 \ell_n^2 + c_2 \ell_n^2 \text{.} \]
Multiplying this with any $\delta' > 0$ and adding $28$ gives
\[
\text{dist}_M^2(x_{n+1}, \text{zer}(A + C)) + (1 - \delta' c_1)\ell_n^2 \\
\leq \left(1 - \frac{\delta'}{2}\right)\text{dist}_M^2(x_n, \text{zer}(A + C)) + (\zeta_{n-1} + \delta' c_2)\ell_{n-1}^2 \\
\leq \left(1 - \frac{\delta'}{2}\right)\text{dist}_M^2(x_n, \text{zer}(A + C)) + (1 - \epsilon + \delta' c_2)\ell_{n-1}^2.
\]
Choosing (for example) $\delta'$ as the smaller of the two solutions to
\[
\left(1 - \frac{\delta'}{2}\right)(1 - \delta' c_1) = (1 - \epsilon + \delta' c_2),
\]
namely
\[
\delta' = \frac{1 + 2c_1 + 2c_2}{2c_1} - \sqrt{\frac{(1 + 2c_1 + 2c_2)^2 - 2\epsilon}{4c_1^2}},
\]
proves Item (i) with $\delta = \delta' c_1$ and $q = 1 - \delta'/2$. For the proof of Item (ii) choose $c_1', c_2' > 0$ according to Lemma 3 such that
\[
\|x_{n+1} - x_n\|^2_M = \lambda_n^2\|p_n - x_n - \frac{1 - \lambda_n}{2 - \lambda_n \gamma_n \beta} u_n - v_n\|^2_M \\
\leq c_1' \ell_n + c_2' \ell_{n-1}^2
\]
for all $n \geq 1$. From Item (i) we get $\delta > 0$ and $0 \leq q < 1$ such that
\[
\text{dist}_M^2(x_{n+1}, \text{zer}(A + C)) + (1 - \delta)\ell_n^2 \leq q\left(\text{dist}_M^2(x_n, \text{zer}(A + C)) + (1 - \delta)\ell_{n-1}^2\right)
\]
for all $n \geq 1$. Repeatedly applying this relation gives
\[
\ell_n^2 \leq \frac{1}{1 - \delta}\left(\text{dist}_M^2(x_{n+1}, \text{zer}(A + C)) + (1 - \delta)\ell_n^2\right) \\
\leq \frac{q^n}{1 - \delta}\left(\text{dist}_M^2(x_1, \text{zer}(A + C)) + (1 - \delta)\ell_0^2\right).
\]
Inserting into $31$ and taking square roots on both sides yields
\[
\|x_{n+1} - x_n\|_M \leq q^{n/2}\sqrt{\frac{c_1' + c_2' / q}{1 - \delta}}\left(\text{dist}_M^2(x_1, \text{zer}(A + C)) + (1 - \delta)\ell_0^2\right).
\]
Let us choose $m > n \geq 1$ and apply the triangle inequality,
\[
\|x_m - x_n\|_M \leq \sum_{k=n}^{m-1} \|x_{k+1} - x_k\|_M \\
\leq \sum_{k=n}^{m-1} q^{k/2}\sqrt{\frac{c_1' + c_2' / q}{1 - \delta}}\left(\text{dist}_M^2(x_1, \text{zer}(A + C)) + (1 - \delta)\ell_0^2\right) \\
\leq \sum_{k=n}^{\infty} q^{k/2}\sqrt{\frac{c_1' + c_2' / q}{1 - \delta}}\left(\text{dist}_M^2(x_1, \text{zer}(A + C)) + (1 - \delta)\ell_0^2\right) \\
= q^{n/2}\frac{1}{1 - \sqrt{q}}\sqrt{\frac{c_1' + c_2' / q}{1 - \delta}}\left(\text{dist}_M^2(x_1, \text{zer}(A + C)) + (1 - \delta)\ell_0^2\right)
\]
(32)
showing that \((x_n)_{n \in \mathbb{N}}\) is a Cauchy sequence, hence \(x_n \to x^*\) as \(n \to \infty\) with \(x^*\) from Theorem 1. The other claim of Item (ii) follows by letting \(m \to \infty\) in (32).

**Remark 2.** The analysis in Section 4 requires \(\beta > 0\), but it can in an analogous way be done with the choice \(C = 0\) and \(\beta = 0\) without division by zero, leading to the iteration and safeguarding condition mentioned in Example 5.

## 5 Special cases

In this section, we present some special cases of our algorithm.

### 5.1 Primal–dual splitting with deviations

We are concerned with the primal inclusion problem of finding \(x \in \mathcal{H}\) such that

\[
0 \in Ax + L^*B(Lx) + Cx
\]

under the following assumption.

**Assumption 3.** We assume that

(i) \(A : \mathcal{H} \to 2^{\mathcal{H}}\) is a maximally monotone operator;

(ii) \(B : \mathcal{K} \to 2^{\mathcal{K}}\) is a maximally monotone operator;

(iii) \(L : \mathcal{H} \to \mathcal{K}\) is a bounded linear operator;

(iv) \(C : \mathcal{H} \to \mathcal{H}\) is a \(1/\beta\)-cocoercive operator with respect to \(\|\cdot\|\);

(v) the solution set \(\text{zer}(A + L^*B + C) := \{x \in \mathcal{H} : 0 \in Ax + L^*B(Lx) + Cx\}\) is nonempty.

Problem (33) can be translated to a primal–dual problem [20]: \(x \in \mathcal{H}\) is a solution to (33) if and only if there exists \(\mu \in B(Lx)\) (the dual variable) such that

\[
0 \in Ax + L^*\mu + Cx,
0 \in -Lx + B^{-1}\mu.
\]

Define the primal–dual pair \(w := (x, \mu) \in \mathcal{H} \times \mathcal{K}\). Then, (34) can be restated as

\[
0 \in Aw + Cw,
\]

where (with slight abuse of notation in the infinite-dimensional setting)

\[
A = \begin{bmatrix} A & L^* \\ -L & B^{-1} \end{bmatrix}, \quad C = \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix}.
\]
The operator \( A \) is maximally monotone by [5, Proposition 26.32] and \( C \) is \( 1/\beta \)-cocoercive with respect to the metric \( \| \cdot \|_M \), with
\[
M = \begin{bmatrix}
I & -\tau L^* \\
-\tau L & \tau \sigma^{-1} I
\end{bmatrix}
\] (37)
where \( \sigma, \tau > 0 \) such that \( \sigma \tau \| L \|_2^2 < 1 \).

The translation of (33) to (35) via the two operators \( A \) and \( C \) shows that Algorithm 1 using the metric \( M \) can be used to solve problem (33). We present this special case in Algorithm 2 along with the subsequent result on its convergence.

**Algorithm 2**

1: **Input:** \((x_0, \mu_0) \in \mathcal{H} \times \mathcal{K}\), the sequences \((\lambda_n)_{n \in \mathbb{N}}\) and \((\zeta_n)_{n \in \mathbb{N}}\) as defined in Assumption 2, and \( \sigma, \tau > 0 \) such that \( \sigma \tau \| L \|_2^2 < 1 \).

2: **set:** \( u_{x,0} = v_{x,0} = 0 \), \( v_{\mu,0} = 0 \).

3: **for** \( n = 0, 1, 2, \ldots \) **do**

4: \( \tilde{x}_n = x_n + u_{x,n} \)

5: \[
\begin{bmatrix}
\tilde{x}_n \\
\tilde{\mu}_n
\end{bmatrix} = \begin{bmatrix}
x_n \\
\mu_n
\end{bmatrix} + \begin{bmatrix}
(1-\lambda_n) \frac{\tau^2}{2} u_{x,n} + v_{x,n} \\
\frac{\tau^2}{2} v_{x,n}
\end{bmatrix}
\]

6: \[
\begin{bmatrix}
p_{x,n} \\
p_{\mu,n}
\end{bmatrix} = \begin{bmatrix}
J_{\tau A} (\tilde{x}_n - \tau L^* \tilde{\mu}_n - \tau C \tilde{x}_n) \\
J_{\tau B^{-1}} (\tilde{\mu}_n + \sigma L(2p_{x,n} - \tilde{x}_n))
\end{bmatrix}
\]

7: \[
\begin{bmatrix}
x_{n+1} \\
\mu_{n+1}
\end{bmatrix} = \begin{bmatrix}
x_n \\
\mu_n
\end{bmatrix} + \lambda_n \begin{bmatrix}
p_{x,n} - \frac{\tau}{\sigma} \tilde{x}_n \\
p_{\mu,n} - \frac{\tau}{\sigma} \tilde{\mu}_n
\end{bmatrix}
\]

8: choose \( u_{n+1} = (u_{x,n+1}, u_{\mu,n+1}) \) and \( v_{n+1} = (v_{x,n+1}, v_{\mu,n+1}) \) such that
\[
\begin{aligned}
\frac{\lambda_n}{2-\lambda_n} \| u_{x,n+1} \|^2 + \frac{\lambda_{n+1}}{2-\lambda_{n+1}} \| p_{x,n} \|_M + \frac{\lambda_n}{2-\lambda_n} \| v_{x,n} \|^2 & \\
\leq \zeta_n \frac{\lambda_n}{2-\lambda_n} \| u_{x,n} \|^2 + \frac{\lambda_{n+1}}{2-\lambda_{n+1}} \| p_{x,n} \|_M + \frac{\lambda_{n+1}}{2-\lambda_{n+1}} \| v_{x,n} \|^2 & \\
- \frac{\lambda_n}{2-\lambda_n} \| u_{x,n+1} \|^2 & \\
\end{aligned}
\] (38)

9: **end for**

**Corollary 1.** Consider monotone inclusions (35) and suppose that Assumption 3 holds. Let \((x_n)_{n \in \mathbb{N}}\) and \((\mu_n)_{n \in \mathbb{N}}\) denote the primal and the dual sequences, respectively, that are obtained from Algorithm 2. Then \((x_n)_{n \in \mathbb{N}}\) converges weakly to a point in \( \text{zer}(A + L^* BL + C) \).

**Proof.** In Algorithm 1, replace \( A \) by \( A \) and \( C \) by \( C \) as devised by (36), and substitute \((x_n, \mu_n)\) in place of \( x_n \), and also set \( p_n = (p_{x,n}, p_{\mu,n}) \), \( y_n = (\tilde{x}_n, \mu_n) \),
where $T$ is a nonexpansive operator. Then, by [5] Remark 4.34, Corollary 23.9, there is a maximally monotone operator $A : \mathcal{H} \to 2^\mathcal{H}$ for which $J_{\gamma A} = \frac{1}{2} \text{Id} + \frac{1}{2} T$, with $\gamma > 0$. This correspondence suggests that Algorithm 1 can be used to solve (39). Letting $C = 0$, $\beta = 0$, $M = \text{Id}$, and $u_{\mu} = 0$ for all $n \in \mathbb{N}$ in Algorithm 1 results in Algorithm 3 that can be used to solve problem (39). Weak convergence of Algorithm 3 is shown in Corollary 2.
Corollary 2. Consider the fixed-point problem (39); suppose that its solution set is nonempty and let \( J_{\gamma A} = \frac{1}{2} I + \frac{1}{2} T \). Then, the sequence \((x_n)_{n \in \mathbb{N}}\), that is generated by Algorithm 3, converges weakly to a point in the solution set of the problem.

Algorithm 3

1: Input: \( x_0 \in H \) and the sequences \((\lambda_n)_{n \in \mathbb{N}}\), \((\gamma_n)_{n \in \mathbb{N}}\), and \((\zeta_n)_{n \in \mathbb{N}}\) according to Assumption 2.
2: set: \( v_0 = 0 \)
3: for \( n = 0, 1, \ldots \) do
4: \( z_n = x_n + v_n \)
5: \( p_n = \frac{1}{2}(I + T)(x_n + v_n) \)
6: \( x_{n+1} = (1 - \lambda_n)x_n + \lambda_n(p_n - v_n) \)
7: choose \( v_{n+1} \) such that
\[
\|v_{n+1}\|^2 \leq \zeta_n \frac{\lambda_n (2 - \lambda_n)(2 - \lambda_{n+1})}{\lambda_{n+1}} \left\| p_n - x_n + \frac{\lambda_n - 1}{2 - \lambda_n} v_n \right\|^2
\]
(40)
8: end for

Setting \( v_n = 0 \) for all \( n \in \mathbb{N} \) in Algorithm 3 results in
\[
x_{n+1} = (1 - \frac{\lambda_n}{2})x_n + \frac{\lambda_n}{2} T(x_n),
\]
which is the standard Krasnosel’kii–Mann iteration [5, Corollary 5.17].

6 A novel inertial primal–dual splitting algorithm

In this section, we present a novel inertial primal–dual method to solve problem (33) with \( C = 0 \). We construct this algorithm from Algorithm 2 by considering a special structure for the deviation vector. We preset the deviation vector direction at the \( n \)-th iteration to be aligned with the momentum direction, i.e., \( v_n = a_n(x_n - x_{n-1}, \mu_n - \mu_{n-1}) \), and use the bound on the norm of deviations to compute \( a_n \). Since this algorithm is an instance of Algorithm 2, its convergence is guaranteed by Corollary 1.

Remark 6. Even though Algorithm 4 has similarities with translations of the algorithms of [1, 2, 3, 9, 23] to a primal–dual framework, to the best of our knowledge, the former and the latter cannot be derived from each other, and thus, are essentially different.

6.1 Efficient evaluation of the norm condition

In order to compute the bound on the coefficients \( a_n \) using (41), one needs to compute some \( M \)-induced norms, which involves evaluating \( L \) and \( L^* \). Depending on the complexity of evaluating \( L \) and \( L^* \), these evaluations may be
Algorithm 4

1: **Input:** \((x_0, \mu_0) \in \mathcal{H} \times \mathcal{K}\), and the sequences \((\lambda_n)_{n \in \mathbb{N}}\) and \((\zeta_n)_{n \in \mathbb{N}}\) as stated in Assumption 2.

2: set: \(a_0 = 0\)

3: for \(n = 0, 1, 2, \ldots\) do

4: \[
\begin{bmatrix}
\hat{x}_n \\
\hat{\mu}_n
\end{bmatrix} = \begin{bmatrix} x_n \\
\mu_n
\end{bmatrix} + a_n \begin{bmatrix} x_n - x_{n-1} \\
\mu_n - \mu_{n-1}
\end{bmatrix}
\]

5: \[
\begin{bmatrix}
\hat{p}_{x,n} \\
\hat{p}_{\mu,n}
\end{bmatrix} = \begin{bmatrix}
J_{\tau A} \left( \hat{x}_n - \tau L^* \hat{\mu}_n \right) \\
J_{\sigma B}^{-1} \left( \hat{\mu}_n + \sigma L \left( 2 \hat{p}_{x,n} - \hat{x}_n \right) \right)
\end{bmatrix}
\]

6: \[
\begin{bmatrix}
x_{n+1} \\
\mu_{n+1}
\end{bmatrix} = \begin{bmatrix} x_n \\
\mu_n
\end{bmatrix} + \lambda_n \left( \begin{bmatrix} p_{x,n} \\
p_{\mu,n}
\end{bmatrix} - \hat{p}_{x,n} \right)
\]

7: choose \(a_{n+1}\) such that

\[
a_{n+1}^2 \| \begin{bmatrix} x_{n+1} - x_n \\
\mu_{n+1} - \mu_n
\end{bmatrix} \|_M^2 \leq \zeta_n \frac{\lambda_n (2 - \lambda_n) (2 - \lambda_{n+1})}{\lambda_{n+1}} \| \begin{bmatrix} p_{x,n} - x_n \\
p_{\mu,n} - \mu_n
\end{bmatrix} \|_M^2 + \lambda_n^{-1} a_n \| \begin{bmatrix} x_n - x_{n-1} \\
\mu_n - \mu_{n-1}
\end{bmatrix} \|_M^2 \quad (41)
\]

8: end for

computationally expensive. However, by scrutinizing Algorithm 4 it is observed that some of the previous evaluations can be reused to keep the additional computational cost low compared to the standard Chambolle–Pock algorithm. In what follows, we provide more details on how to compute the required scaled norm of the vector quantities in a computationally efficient manner.

As seen in line 7 of Algorithm 4, at each iteration one of each \(L\) and \(L^*\) evaluations are performed. Similar operations take place at each iteration of, e.g., the Chambolle–Pock algorithm. However, in our algorithm, we have other operations involving evaluations of \(L\) and \(L^*\). Those are due to verification of the norm condition in line 8 of Algorithm 4. More specifically, since the kernel \(M\) is given by (37) for each evaluation of \(\| \cdot \|_M\), we have one more evaluation each of \(L\) and \(L^*\). This can lead to a substantially higher computational cost. However, except for the first iteration, the extra \(L\) and \(L^*\) evaluations can be computed from the computations which are already available from previous iterations. That is possible due to the relations

\[
\begin{align*}
L \hat{x}_n &= L x_n + b_n (L x_n - L \hat{x}_{n-1}), \\
L^* \hat{\mu}_n &= L^* \mu_n + b_n (L^* \mu_n - L^* \mu_{n-1}), \\
L x_{n+1} &= L x_n + \lambda_n (L p_{x,n} - L \hat{x}_n), \\
L^* \mu_{n+1} &= L^* \mu_n + \lambda_n (L^* p_{\mu,n} - L^* \hat{\mu}_n),
\end{align*}
\]

which are derived from lines 5 and 7 of Algorithm 4. In the relations above, for \(n > 0\), all quantities to the right hand side are already computed and can be reused, except for \(L p_{x,n}\) and \(L^* p_{\mu,n}\) that need to be computed via direct
evaluation.

Table 1 provides the list of evaluations involving $L$ and $L^*$ that we need to perform at the first three iterations. It reveals that at the first iteration, we need to perform six different evaluations involving $L$ or $L^*$, of which four might be computationally heavy and two can be done cheaply. After that, i.e. for $n > 0$, we only need to perform two such heavy evaluations per iteration; namely, $L_{p,x,n}$ and $L^*_{p,\mu,n}$. The rest of the $L$ and $L^*$ evaluations can be done efficiently by exploiting previously computed quantities and (42). This keeps the computational per-iteration cost of our algorithm basically the same as that of the Chambolle–Pock algorithm.

| $n$ | Expensive evaluations | Cheap evaluations |
|-----|------------------------|-------------------|
| 0   | $L_{x,0}$, $L^*_{\mu,0}$, $L_{p,x,0}$, $L^*_{p,\mu,0}$ | $L_{x,1}$, $L^*_{\mu,1}$ |
| 1   | $L_{p,x,1}$, $L^*_{p,\mu,1}$ | $L_{x,2}$, $L_{x,2}$, $L^*_{\mu,2}$, $L^*_{\mu,2}$ |
| 2   | $L_{p,x,2}$, $L^*_{p,\mu,2}$ | $L_{x,3}$, $L_{x,3}$, $L^*_{\mu,3}$, $L^*_{\mu,3}$ |

Table 1: List of evaluations that involve $L$ and $L^*$ for the first three iterations. The second column shows direct and potentially expensive evaluations and the third column shows evaluations that can be done cheaply via the relations in (42).

7 Numerical Experiments

We solve an $l_1$-norm regularized SVM problem for classification of the form

$$\min_x f(Lx) + g(x),$$

(43)
given a labeled training data set $\{\theta_i, \phi_i\}_{i=1}^N$, where $\theta_i \in \mathbb{R}^d$ and $\phi_i \in \{-1, 1\}$ are training data and labels, respectively, and with

$$f(Lx) = 1^T \max(0, 1 - Lx), \quad g(x) = \xi \|\omega\|_1, \quad L = \begin{bmatrix} \phi_1 \theta_1^T & \phi_1 \\ \vdots & \vdots \\ \phi_N \theta_N^T & \phi_N \end{bmatrix},$$

where $0 = (0, \ldots, 0)^T$, $1 = (1, \ldots, 1)^T$, $x = (\omega, b)$ is the decision variable with $b \in \mathbb{R}$ and $\omega \in \mathbb{R}^d$, $\max(\cdot, \cdot)$ acts element-wise, and $\xi \geq 0$ is the regularization parameter.

A point $x^*$ is a solution to (43) if and only if it satisfies

$$0 \in L^* \partial f(Lx^*) + \partial g(x^*).$$

This holds, since $f$ and $g$ are proper, closed, and convex functions with full domains, and thus, $\partial f$ and $\partial g$ are maximally monotone and $L$ is a linear operator [5, Proposition 16.42]. This monotone inclusion problem is an instance of (33).
with \( A = \partial g, B = \partial f, \) and \( C = 0. \) As in Section 5.1 we transform the problem into a primal–dual problem and solve it with primal–dual algorithms.

We compare our inertial primal–dual method, Algorithm 4, to the standard Chambolle–Pock (CP) [7], and to the inertial primal–dual algorithm of Lorenz–Pock (LP) [23]. In all experiments, we set the primal and the dual step-sizes to \( \tau = \sigma = \frac{0.99}{\|L\|} \), the regularization parameter of problem (43) to \( \xi = 0.1 \), and \( \zeta_n \) is, for each \( n \in \mathbb{N} \), sampled from a uniform distribution on \([0, 1 - 10^{-6}]\). The experiments are done using the liver disorders data-set [8] which has 145 samples and 5 features. The solution \((x^*, \mu^*)\) is found by running the standard Chambolle–Pock algorithm until the residual gets smaller than \(10^{-15}\).

![Figure 1](image1.png)

Figure 1: Distance to the solution vs. iteration number for the \(l_1\)-norm regularized SVM (43) with \( \xi = 0.1 \), on the liver disorders data-set [8] with 145 samples and 5 features. Solved using Chambolle–Pock primal–dual algorithm (CP), Lorenz–Pock inertial primal–dual method (LP), and Algorithm 4 with \( \lambda = 1.0 \). The primal and dual step-sizes are set to \( \tau = \sigma = \frac{0.99}{\|L\|} \) for all algorithms.

![Figure 2](image2.png)

Figure 2: Scaling factor \( a_n \) of Algorithm 4 in the experiment shown in Fig. 1 vs. iteration number for the first 1000 iterations.

For the \(l_1\)-norm regularized SVM problem, since \( f \) and \( g \) are piece-wise linear,
The resulting (primal–dual) monotone operator

\[ \mathcal{A} = \begin{bmatrix} \partial g^* - L \quad L^* \\ -L & \partial f^* \end{bmatrix} \]

is metrically subregular at any point in the solution set of the problem for 0, see [21] Lemma IV.4. It therefore follows from Theorem 2 that the algorithm exhibits local linear convergence, see Fig. 1 and Fig. 3. The figures reveal that our method needs about half the number of iterations to reach the same accuracy as the other two methods. This improvement comes at essentially no extra computational cost.

Figure 2 shows the first one thousand scaling factors \( a_n \) of Algorithm 4 for the same implementation as in Fig. 1. It is seen that the scaling factor attains mostly values close to one.

In Fig. 3 the impact of the relaxation parameter \( \lambda \) is investigated. In the sense of convergence rate, it interestingly seems that \( \lambda = 1.0 \) yields the best performance in this example.

Acknowledgement. The authors would like to thank Bo Bernhardsson for his valuable feedback on this work. This research was partially supported by Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Sebastian Banert was partially supported by ELLIIT.

References

[1] Alvarez, F. “On the minimizing property of a second order dissipative system in Hilbert spaces”. In: SIAM Journal on Control and Optimization 38.4 (2000), pp. 1102–1119. doi:10.1137/s0363012998335802
[2] Alvarez, F. and Attouch, H. “An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping”. In: *Set-Valued Analysis* 9.1/2 (2001), pp. 3–11. doi: 10.1023/a:1011253113155

[3] Attouch, H. and Cabot, A. “Convergence of a relaxed inertial forward–backward algorithm for structured monotone inclusions”. In: *Applied Mathematics & Optimization* 80.3 (2019), pp. 547–598. doi: 10.1007/s00245-019-09584-2.

[4] Banert, S. et al. *Accelerated forward–backward optimization using deep learning*. 2021. arXiv: 2105.05210v1 [math.OC].

[5] Bauschke, H. H. and Combettes, P. L. *Convex analysis and monotone operator theory in Hilbert spaces*. 2nd ed. CMS Books in Mathematics. Springer, 2017. doi: 10.1007/978-3-319-48311-5.

[6] Bruck, R. E. “An iterative solution of a variational inequality for certain monotone operators in Hilbert space”. In: *Bulletin of the American Mathematical Society* 81 (1975), pp. 890–892. DOI: 10.1090/S0002-9904-1975-13874-2.

[7] Chambolle, A. and Pock, T. “A first-order primal–dual algorithm for convex problems with applications to imaging”. In: *Journal of Mathematical Imaging and Vision* 40.1 (2011), pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

[8] Chang, C.-C. and Lin, C.-J. “LIBSVM: A library for support vector machines”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), pp. 1–27. DOI: 10.1145/1961189.1961199.

[9] Cholamjiak, W., Cholamjiak, P., and Suantai, S. “An inertial forward–backward splitting method for solving inclusion problems in Hilbert spaces”. In: *Journal of Fixed Point Theory and Applications* 20.1 (2018). DOI: 10.1007/s11784-018-0526-5.

[10] Chouzenoux, E., Pesquet, J.-C., and Repetti, A. “Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function”. In: *Journal of Optimization Theory and Applications* 162.1 (2013), pp. 107–132. DOI: 10.1007/s10957-013-0465-7.

[11] Combettes, P. L. and Pesquet, J.-C. “Proximal splitting methods in signal processing”. In: *Fixed-point algorithms for inverse problems in science and engineering*, Ed. by Bauschke, H. H. et al. Springer New York, 2011, pp. 185–212. DOI: 10.1007/978-1-4419-9569-8_10.

[12] Combettes, P. L. and Vu, B. C. “Variable metric forward–backward splitting with applications to monotone inclusions in duality”. In: *Optimization* 63.9 (2012), pp. 1289–1318. DOI: 10.1080/02331934.2012.733883.

[13] Condat, L. “A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms”. In: *Journal of Optimization Theory and Applications* 158.2 (2013), pp. 460–479. DOI: 10.1007/s10957-012-0245-9.
[14] Dontchev, A. L. and Rockafellar, R. T. Implicit functions and solution mappings. A view from variational analysis. Springer Monographs in Mathematics. Springer, 2009. doi: 10.1007/978-0-387-87821-8.

[15] Giselsson, P. “Nonlinear forward–backward splitting with projection correction”. In: SIAM Journal on Optimization 31.3 (2021), pp. 2199–2226. doi: 10.1137/20M1345062.

[16] Giselsson, P. and Boyd, S. “Metric selection in fast dual forward–backward splitting”. In: Automatica 62 (2015), pp. 1–10. doi: 10.1016/j.automatica.2015.09.010.

[17] Giselsson, P. and Boyd, S. P. “Diagonal scaling in Douglas–Rachford splitting and ADMM”. In: 53rd IEEE Conference on Decision and Control. IEEE. 2014, pp. 5033–5039. doi: 10.1109/CDC.2014.7040176.

[18] Giselsson, P. and Boyd, S. P. “Preconditioning in fast dual gradient methods”. In: 53rd IEEE Conference on Decision and Control. IEEE. 2014, pp. 5040–5045. doi: 10.1109/CDC.2014.7040176.

[19] Giselsson, P., Fält, M., and Boyd, S. “Line search for averaged operator iteration”. In: 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE. 2016, pp. 1015–1022. doi: 10.1109/CDC.2016.7798401.

[20] He, B. and Yuan, X. “Convergence analysis of primal–dual algorithms for a saddle-point problem: from contraction perspective”. In: SIAM Journal on Imaging Sciences 5.1 (2012), pp. 119–149. doi: 10.1137/100814494.

[21] Latafat, P., Freris, N., and Patrinos, P. “A new randomized block-coordinate primal–dual proximal algorithm for distributed optimization”. In: IEEE Transactions on Automatic Control 64.10 (2019), pp. 4050–4065. doi: 10.1109/TAC.2019.2906924.

[22] Lions, P. L. and Mercier, B. “Splitting algorithms for the sum of two nonlinear operators”. In: SIAM Journal on Numerical Analysis 16.6 (1979), pp. 964–979. doi: 10.1137/0716071.

[23] Lorenz, D. A. and Pock, T. “An inertial forward–backward algorithm for monotone inclusions”. In: Journal of Mathematical Imaging and Vision 51.2 (2015), pp. 311–325. doi: 10.1007/s10851-014-0523-2.

[24] Nocedal, J. and Wright, S. J. Numerical optimization. 2nd ed. Springer Series in Operations Research and Financial Engineering. Springer, 2006. doi: 10.1007/978-0-387-40065-5.

[25] Passty, G. B. “Ergodic convergence to a zero of the sum of monotone operators in Hilbert space”. In: Journal of Mathematical Analysis and Applications 72.2 (1979), pp. 383–390. doi: 10.1016/0022-247x(79)90234-8.

[26] Pock, T. and Chambolle, A. “Diagonal preconditioning for first order primal–dual algorithms in convex optimization”. In: 2011 International Conference on Computer Vision. IEEE. 2011, pp. 1762–1769. doi: 10.1109/ICCV.2011.6126441.
[27] Polyak, B. T. “Some methods of speeding up the convergence of iteration methods”. In: USSR Computational Mathematics and Mathematical Physics 4.5 (1964), pp. 1–17. DOI: 10.1016/0041-5553(64)90137-5.

[28] Raguet, H., Fadili, J., and Peyré, G. “A generalized forward–backward splitting”. In: SIAM Journal on Imaging Sciences 6.3 (2013), pp. 1199–1226. DOI: 10.1137/120872802.

[29] Raguet, H. and Landrieu, L. “Preconditioning of a generalized forward–backward splitting and application to optimization on graphs”. In: SIAM Journal on Imaging Sciences 8.4 (2015), pp. 2706–2739. DOI: 10.1137/15m1018253.

[30] Sadeghi, H., Banert, S., and Giselsson, P. DWIFOB: A Dynamically Weighted Inertial Forward–Backward Algorithm for Monotone Inclusions. 2022. arXiv: 2203.00028 [math.OC].

[31] Sadeghi, H., Banert, S., and Giselsson, P. Incorporating History and Deviations in Forward–Backward Splitting. 2022. arXiv: 2208.05498 [math.OC].

[32] Sadeghi, H. and Giselsson, P. Hybrid acceleration scheme for variance reduced stochastic optimization algorithms. 2021. arXiv: 2111.06791 [math.OC].

[33] Schmidt, M., Roux, N., and Bach, F. “Convergence rates of inexact proximal–gradient methods for convex optimization”. In: Advances in Neural Information Processing Systems (NIPS 2011). Ed. by Shawe-Taylor, J. et al. Vol. 24. Curran Associates, Inc., 2011, pp. 1458–1466. URL: https://proceedings.neurips.cc/paper/2011/hash/8f7d807e1f53eff5f9efbe5cb81090fb-Abstract.html.

[34] Solodov, M. V. and Svaiter, B. F. “An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions”. In: Mathematics of Operations Research 25.2 (2000), pp. 214–230. DOI: 10.1287/moor.25.2.214.12222.

[35] Solodov, M. V. and Svaiter, B. F. “A unified framework for some inexact proximal point algorithms”. In: Numerical Functional Analysis and Optimization 22.7–8 (2001), pp. 1013–1035. DOI: 10.1081/NFA-100108320.

[36] Themelis, A. and Patrinos, P. “SuperMann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators”. In: IEEE Transactions on Automatic Control 64.12 (2019), pp. 4875–4890. DOI: 10.1109/TAC.2019.2906393.

[37] Tseng, P. “A modified forward–backward splitting method for maximal monotone mappings”. In: SIAM Journal on Control and Optimization 38.2 (2000), pp. 431–446. DOI: 10.1137/S0363012998338806.

[38] Villa, S. et al. “Accelerated and inexact forward–backward algorithms”. In: SIAM Journal on Optimization 23.3 (2013), pp. 1607–1633. DOI: 10.1137/110844805.

[39] Vu, B. C. “A splitting algorithm for dual monotone inclusions involving cocoercive operators”. In: Advances in Computational Mathematics 38.3 (2013), pp. 667–681. DOI: 10.1007/s10444-011-9254-8.
[40] Zhang, J., O’Donoghue, B., and Boyd, S. “Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations”. In: *SIAM Journal on Optimization* 30.4 (2020), pp. 3170–3197. DOI: 10.1137/18M1232772.