Efficient and Flexible First-Order Optimization Algorithms

Sadeghi, Hamed

2022

Document Version:
Publisher's PDF, also known as Version of record

Efficient and Flexible First-Order Optimization Algorithms

Hamed Sadeghi

LUND
UNIVERSITY

Department of Automatic Control

PhD Thesis TFRT-1139
ISBN 978-91-8039-468-0 (print)
ISBN 978-91-8039-467-3 (web)
ISSN 0280–5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Media-Tryck.
Lund 2022

To my wonderful children:
***Mohammadhossein** and **Mahya**.*

Abstract

Optimization problems occur in many areas in science and engineering. When the optimization problem at hand is of large-scale, the computational cost of the optimization algorithm is a main concern. *First-order optimization algorithms*—in which updates are performed using only gradient or subgradient of the objective function—have low per-iteration computational cost, which make them suitable for tackling large-scale optimization problems. Even though the per-iteration computational cost of these methods is reasonably low, the number of iterations needed for finding a solution—especially if medium or high accuracy is needed—can in practice be very high; as a result, the overall computational cost of using these methods would still be high.

This thesis focuses on one of the most widely used first-order optimization algorithms, namely, the *forward–backward splitting* algorithm, and attempts to improve its performance. To that end, this thesis proposes novel first-order optimization algorithms which all are built upon the forward–backward method. An important feature of the proposed methods is their *flexibility*. Using the flexibility of the proposed algorithms along with the *safeguarding* notion, this thesis provides a framework through which many new and *efficient* optimization algorithms can be developed.

To improve efficiency of the forward–backward algorithm, two main approaches are taken in this thesis. In the first one, a technique is proposed to adjust the point at which the forward–backward operator is evaluated. This is done through including additive terms—which are called *deviations*—in the input argument of the forward–backward operator. The deviations then, in order to have a convergent algorithm, have to satisfy a safeguard condition at each iteration. Incorporating deviations provides great flexibility to the algorithm and paves the way for designing new and improved forward–backward-based methods. A few instances of employing this flexibility to derive new algorithms are presented in the thesis.

In the second proposed approach, a globally (and potentially slow) convergent algorithm can be combined with a fast and locally convergent one to form an efficient optimization scheme. The role of the globally convergent method is to ensure convergence of the overall scheme. The fast local algorithm's role is to speed up the convergence; this is done by switching from the globally convergent algorithm to

the local one whenever it is safe, i.e., when a safeguard condition is satisfied. This approach, which allows for combining different global and local algorithms within its framework, can result in fast and globally convergent optimization schemes.

Acknowledgements

I would like to begin by expressing my immense gratitude and regards to my supervisor, Pontus Giselsson. Thank you Pontus, for your fantastic job on guiding me through my research and studies, for your continuous support, for so many fruitful meetings and discussions, and for always having a positive attitude; I learned a lot from you and enjoyed working under your supervision. Next, I would like to deliver my deep respect and appreciation to Anders Rantzer, my former supervisor and current co-advisor, who gave me the opportunity of pursuing a PhD at the Automatic Control department. Thank you Anders for guiding me through my studies and research and also for giving me the freedom to change my research direction. I want to thank my co-advisor, Sebastian Banert. I am eminently grateful to you, for all the meetings and scientific discussions we had, for your support during the past few years, and particularly for your detailed and constructive comments on the manuscripts.

I also would like to express my appreciation to Anton Cervin for handling my ISP meetings and yearly employee development sessions; it was joyful working with you as a TA. Bo Bernahrdsson, I am grateful to you for your input on my manuscripts prior to my preparatory seminar. Mahdi Ghazaei Ardakani, Carolina Bergeling, and Richard Pates, my sincere thanks to you for helping me, particularly when I started my PhD journey. I also deliver my gratitude to the administrative staff, Eva Westin, Mika Nishimura, Monika Rasmusson, and Cecilia Edelborg, for making the department run smoothly. Thanks to all my other former and present colleagues, faculty members, administrative staff, and research engineers at the department for making such a positive, vivid, and friendly workplace. It was a pleasure to be a member of the Automatic Control department for the last years.

A big thanks to my family for their continuous support and encouragement throughout my life and in all my highs and lows. And finally, I would like to express my heartfelt emotions to my lovely children, Mohammadhossein and Mahya; having you beside me has always been a source of joy, energy, and motivation; love you!

Financial Support
This work was partially supported by the Wallenberg AI, Autonomous Systems, and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Contents

1

Introduction

Mathematical Optimization, which is a branch of applied mathematics, appears in many areas in science and engineering such as artificial intelligence [Le et al., 2011; Sra et al., 2012], statistics [Everitt, 2012], finance [Gilli et al., 2019], control [Darup et al., 2019; Giselsson and Rantzer, 2014], and transportation [Perea-Lopez et al., 2003; Yin, 2002], to mention a few. Due to the advancements in providing computational power, methods for large-scale mathematical optimization play a vital role in many applications involving massive amounts of data. Nowadays, mathematical optimization is viewed as a pivotal element in many modern machine learning and data science applications.

Mathematical optimization can be used to effectively answer decision-making or prediction questions; whether it is tuning the weights of a deep neural network to find a predictive model, route planning for logistic trucks, or devising the relative weights of a selection of equities in an investment portfolio to maximize the return while maintaining a tolerable risk level.

A mathematical optimization problem consists of an *objective (loss) function* and *decision or optimization variables* (or simply *variables*) which are possibly subject to some *constraints*. The objective function—which we wish to either minimize or maximize—provides a means to quantitatively measure the performance of a system under study. The objective function itself can consist of a combination of various quantitative measures of performance and depends on particular attributes or properties of the underlying system. Each of theses attributes can be represented by a decision variable. The decision variables are often confined to take values from certain ranges or sets. Such restrictions on the decision variables are called *constraints*. For instance, in a deep learning based classifier we may have a categorical cross entropy error as the objective function, the weights (and biases) of the network as variables, and enforcing the absolute values of the weights to be less than a certain value as the constraint; in route planning for logistic trucks, the objective function can be overall fuel consumption of the fleet of trucks, fuel consumption and travelling time of each truck can be seen as decision variables, and restricting the permissible fuel consumption of each logistic truck to a full tank may be considered as constraints; in an investment portfolio optimization, we may consider the

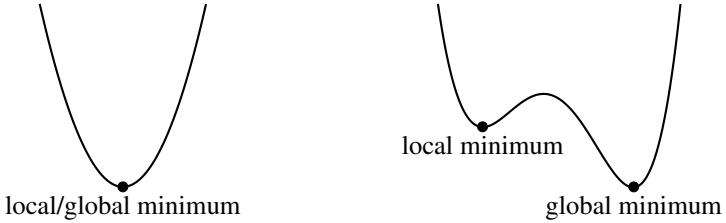

Figure 1.1: (*left*) a convex function where local minimum is a global minimum as well; (*right*) a non-convex function with a local minimum that is not a global minimum.

expected return (profit) as the objective function, the weight (or number of units) of each equity as the decision variables, and having the weighted sum of the monetary value dedicated to each equity less than or equal to the total asset (capital) under management as a constraint.

The goal in an optimization problem is to find the decision variables that optimize the objective function while satisfying the constraints. Such a variable is called a solution to the optimization problem. Depending on the formulation of mathematical optimization problems, they can be categorized as *convex* or *non-convex* optimization problems. For convex optimization problems, any local minimum (optimum) is a global minimum as well; however, this is not the case for non-convex problems (a global minimum is a point at which the objective function value is less than or equal to that of all other points; whereas, the function value at a local minimum needs to be merely less than or equal to that of its neighboring points; see Fig. 1.1). The theory around convex optimization is vast and well-developed, as a result, a convex optimization problem can often be solved to global optimality.

After formulating the optimization problem, an optimization algorithm can be used to solve it. Most optimization algorithms are iterative. Given an initial guess of where an optimal point is, iterative optimization algorithms generate a sequence of iterates (points) that, under some conditions, converges to a solution of the optimization problem. There exist a variety of optimization algorithms that can be used for solving an optimization problem. However, some may exhibit better performance compared to the others. In fact, each class of optimization algorithms is tailored towards particular categories of optimization problems. Selecting unsuitable algorithms for solving the problem at hand could result in a poor performance and the algorithm may even fail to find a solution.

Many optimization algorithms for solving convex optimization problems can be categorized as first- or second-order methods. A first-order algorithm uses merely the first-order information, i.e., the information on the gradient or—in case of non-differentiablity—subgradient of the objective function. On the other hand, second-order algorithms require also second-order information, i.e., information on the second derivative (Hessian) of the objective function. Therefore, generally speak-

ing, first-order optimization methods have lower computational complexity, which makes them more suitable and efficient for solving large-scale optimization problems; however, smaller problems can be solved very efficiently using second-order algorithms.

Monotone operator theory is an area of nonlinear analysis that has applications in many fields such as partial differential equations, variational inequalities, mathematical economics and, in particular, optimization [Borwein, 2010; Combettes, 2018; Minty, 1969; Ryu and Boyd, 2016]. Many convex optimization problems can be cast as finding a zero of a sum of monotone operators. The framework of monotone operators provides a unifying tool for derivation, analysis, and understanding of a variety of first-order convex optimization algorithms.

In this thesis, the focus is on first-order algorithms for convex optimization problems which are viewed through the framework of monotone operators theory. This framework is used to both develop families of novel optimization algorithms and improve performance of some of the existing first-order optimization methods.

1.1 Outline

The thesis consists of two main parts. The first part provides a background for non-experts in the field and the second part contains the collection of papers that is the main part of the thesis.

The remainder of this chapter, as well as chapter 2, are dedicated to introducing basic notions and definitions along with some relevant fundamental algorithms. Chapter 3 lists the papers that are included in the thesis along with the contributions of the authors of each paper

1.2 Notations and basic definitions

In this section, after introducing the notations, we present some fundamental notions related to *convex analysis* followed by basic concepts from *set-valued analysis*.

Notation

The set of real numbers and the d-dimensional Euclidean space are respectively denoted by $\mathbb{R}$ and $\mathbb{R}^d$. The extended real line is defined as $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$. $\mathscr{H}$ and $\mathscr{K}$ denote real Hilbert spaces that are equipped with inner products and induced norms, which we denote by $\langle x, y\rangle$ and $\|x\| = \sqrt{\langle x, x\rangle}$, respectively.

The adjoint operator of a linear bounded operator $L : \mathscr{H} \to \mathscr{K}$ is denoted by L^* satisfying $\langle Lx, y\rangle = \langle x, L^*y\rangle$ for all $x \in \mathscr{H}$ and $y \in \mathscr{K}$. A linear and bounded operator $L : \mathscr{H} \to \mathscr{H}$ is called self-adjoint if $\langle Lx, y\rangle = \langle x, Ly\rangle$ for all $x, y \in \mathscr{H}$. A linear, bounded, and self-adjoint operator $M : \mathscr{H} \to \mathscr{H}$ is said to be *strongly positive* if there exists some $c > 0$ such that $\langle x, Mx\rangle \geq c\|x\|^2$ for all $x \in \mathscr{H}$. We

denote the set of linear, bounded, self-adjoint, and strongly positive operators on $\mathscr{H}$ by $\mathscr{M}(\mathscr{H})$.

Convex analysis

A set $S \subseteq \mathscr{H}$ is *convex* if $x, y \in S$ implies $\theta x + (1-\theta)y \in S$ for all $\theta \in [0,1]$. The empty set $\emptyset$, singletons (sets that contain only one element), and $\mathbb{R}^d$ are examples of convex sets. For the convex set S, the line segment connecting any two points in S, lies within S; see Fig. 1.2.

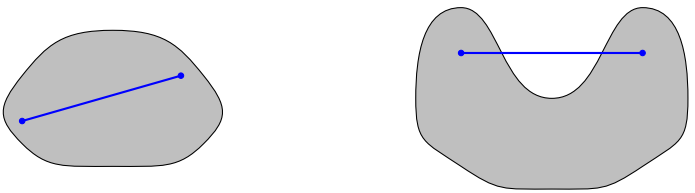

Figure 1.2: (*left*) a convex set; (*right*) a non-convex set.

The *(effective) domain* of a function $f : \mathscr{H} \to \overline{\mathbb{R}}$ is defined as

$$\operatorname{dom}(f) := \{x \in \mathscr{H} : f(x) < +\infty\}.$$

A function $f : \mathscr{H} \to \overline{\mathbb{R}}$ is proper if its effective domain is nonempty. The *epigraph* of a function $f : \mathscr{H} \to \overline{\mathbb{R}}$ is the set of points above its graph, which is formally defined as

$$\operatorname{epi}(f) := \{(x, \alpha) \in \mathscr{H} \times \mathbb{R} : f(x) \leq \alpha\}.$$

A function is said to be *closed* or *lower semi-continuous* if its epigraph is a closed set. A function $f : \mathscr{H} \to \overline{\mathbb{R}}$ is a *convex function* if for all $x, y \in \mathscr{H}$ and $\theta \in [0,1]$

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y).$$

Alternatively, we say a function is convex if its epigraph is a convex set; see Fig. 1.3.

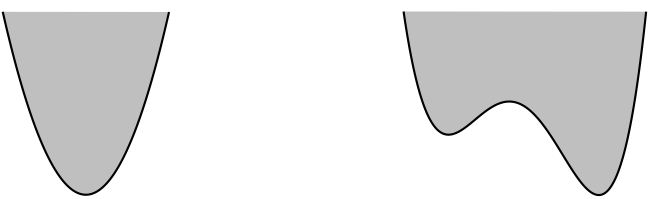

Figure 1.3: (*left*) epigraph of a convex function; (*right*) epigraph of a non-convex function.

Set-valued analysis and operators

Let $\mathscr{X}$, $\mathscr{Y}$, and $\mathscr{Z}$ be nonempty subsets of the real Hilbert space $\mathscr{H}$, and let $2^{\mathscr{X}}$ denote the power set—the set of all subsets—of $\mathscr{X}$. An operator T is said to be a *mapping* or an *operator* from $\mathscr{X}$ to $\mathscr{Y}$ if it maps every point in $\mathscr{X}$ to a point $Tx = T(x)$ in $\mathscr{Y}$. This is denoted as $T : \mathscr{X} \to \mathscr{Y}$.

In what follows we present some basic notions that are frequently used throughout the thesis.

The set-valued operator $A : \mathscr{X} \to 2^{\mathscr{Y}}$ maps every point $x \in \mathscr{X}$ to a (potentially empty) set $Ax \subseteq \mathscr{Y}$. For a given set $S \subseteq \mathscr{X}$, we define $A(S) := \bigcup_{x \in S} Ax$. The *graph* of an operator $A : \mathscr{X} \to 2^{\mathscr{Y}}$ is defined by

$$\operatorname{gra}(A) := \{(x,u) \in \mathscr{X} \times \mathscr{Y} : u \in Ax\}.$$

An operator is uniquely characterized by its graph. The domain and range of an operator $A : \mathscr{X} \to 2^{\mathscr{Y}}$ are respectively defined by

$$\begin{aligned} \operatorname{dom}(A) &:= \{x \in \mathscr{X} : Ax \neq \emptyset\}, \\ \operatorname{ran}(A) &:= A(\mathscr{X}). \end{aligned}$$

Scaling, summation, and composition are other notions that are frequently used for operators. Given $A : \mathscr{X} \to 2^{\mathscr{Y}}$, $B : \mathscr{X} \to 2^{\mathscr{Y}}$, and $\lambda \in \mathbb{R}$, we define $A + \lambda B$ as

$$\operatorname{gra}(A + \lambda B) := \{(x, y_1 + \lambda y_2) \in \mathscr{X} \times \mathscr{Y} : y_1 \in Ax, y_2 \in Bx\}.$$

Given $A : \mathscr{X} \to 2^{\mathscr{Y}}$ and $C : \mathscr{Y} \to 2^{\mathscr{Z}}$, the composition $C \circ A$ is defined via its graph as

$$\operatorname{gra}(C \circ A) := \{(x,z) \in \mathscr{X} \times \mathscr{Z} : \exists y \text{ such that } y \in Ax, z \in Cy\},$$

or alternatively, it can be defined as $(C \circ A)x = C(Ax) = \bigcup_{y \in Ax} Cy$ for all $x \in \mathscr{H}$.

The *identity operator* is denoted by Id and defined as

$$\operatorname{Id} := \{(x,x) : x \in \mathscr{H}\}.$$

The inverse operator of $A : \mathscr{X} \to 2^{\mathscr{Y}}$ is denoted by $A^{-1}\mathscr{Y} \to 2^{\mathscr{X}}$ and defined via its graph as

$$\operatorname{gra}(A^{-1}) := \{(u,x) \in \mathscr{Y} \times \mathscr{X} : (x,u) \in \operatorname{gra}(A)\}.$$

Observe that, by this definition, we have $(A^{-1})^{-1} = A$, and thus, $\operatorname{dom}(A) = \operatorname{ran}(A^{-1})$ and $\operatorname{ran}(A) = \operatorname{dom}(A^{-1})$. The *zero set* of an operator $A : \mathscr{X} \to 2^{\mathscr{Y}}$ is defined as

$$\operatorname{zer}(A) = A^{-1}(0) := \{x \in \mathscr{X} : 0 \in Ax\}.$$

A mapping $T : \mathscr{H} \to \mathscr{H}$ is said to be L-Lipschitz continuous (with $L > 0$) if for all $x, y \in \mathscr{H}$

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

An operator $T : \mathscr{H} \to \mathscr{H}$ is nonexpansive, if it is 1-Lipschitz continuous. If the mapping $T : \mathscr{H} \to \mathscr{H}$ is Lipschitz continuous with Lipschitz constant $L < 1$, we say that it is a *contractive mapping* or a *contraction*. We say that an operator $T : \mathscr{H} \to \mathscr{H}$ is α-averaged with $\alpha \in (0,1)$ if

$$T = (1-\alpha)\,\mathrm{Id} + \alpha N$$

for some nonexpansive operator N.

2

Background

This chapter, besides collecting some basic concepts and definitions that are needed for the rest of the thesis, describes our approach towards solving convex optimization problems. More specifically, some notions and definitions from the areas of *fixed-point theory*, *convex analysis*, and *theory of monotone operators* which this thesis relies on, are presented. This chapter merely touches upon the surface of these subjects and to the extent that provides sufficient background for the rest of the thesis. For a technically in-depth treatment of these subjects, interested readers are referred to textbooks on convex optimization, monotone operator theory, and iterative methods, for instance [Bauschke and Combettes, 2017; Boyd and Vandenberghe, 2004; Kelley, 1999; Rockafellar, 1970; Ryu and Yin, 2022].

2.1 Fixed-point iterations

We say $x \in \mathscr{H}$ is a *fixed point* of a mapping $T : \mathscr{H} \to \mathscr{H}$ if $x = Tx$. The set of fixed points of T is denoted by $\operatorname{fix}(T)$ and defined as

$$\operatorname{fix}(T) = \{x \in \mathscr{H} : x = Tx\}.$$

Observe that the fixed-point set of T is identical to the zero set of its *fixed-point residual mapping* $\operatorname{Id} - T$, that is, $\operatorname{fix}(T) = \operatorname{zer}(\operatorname{Id} - T)$.

The literature on iterative methods for finding a fixed point of a mapping $T : \mathscr{H} \to \mathscr{H}$ is rich and extensive [Berinde and Takens, 2007; Kelley, 1995; Kelley, 1999]. In this section, we consider Picard, Krasnosel'skiĭ–Mann, and Halpern iterations, which are widely used iterations in optimization frameworks.

Picard iteration

Given $x_0 \in \mathscr{H}$ and a mapping T,

$$x_{n+1} = Tx_n, \qquad \text{for all } n \in \mathbb{N}$$

is called a *fixed-point iteration* which sometimes is referred to as the *Picard iteration*. This algorithm, in general, even for nonexpansive T is not guaranteed to converge. A classical example to show that is the two-dimensional rotation operator

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

with $\theta \neq 2k\pi$ where k is an integer number. For R_θ, given $x_0 \neq 0$, its only fixed point $x^\star = 0$ cannot be found using the Picard iteration. Nevertheless, if $\operatorname{fix}(T)$ is nonempty, under some suitable assumptions on T, the Picard iteration is guaranteed to find a fixed point $x^\star \in \operatorname{fix}(T)$. More specifically, the Picard iteration is guaranteed to weakly converge to a fixed point of the mapping T if it is either a contractive or an averaged mapping [Ryu and Boyd, 2016]. Note that contractivity is considered a strong assumption in many optimization settings, however, averagedness is a milder assumption and is quite common.

Krasnosel'skiĭ–Mann iteration

Given $x_0 \in \mathscr{H}$ and a nonexpansive operator T, the algorithm of *Krasnosel'skiĭ–Mann* is given as

$$x_{n+1} = (1-\alpha_n)x_n + \alpha_n T x_n, \qquad \text{for all } n \in \mathbb{N}$$

where $\alpha_n \in [0,1]$. If $\operatorname{fix}(T)$ is nonempty, under some appropriate assumptions on $(\alpha_n)_{n\in\mathbb{N}}$, the sequence of iterates generated by this algorithm converges weakly to a fixed point of T [Bauschke and Combettes, 2017, Theorem 5.15]. The Krasnosel'skiĭ–Mann iteration can be viewed as a generalization to the Picard iteration, as with $\alpha_n = 1$ it reduces to the Picard iteration.

Observe that it is possible to find a fixed point of the nonexpansive operator T by first α-averaging it—with $\alpha \in (0,1)$—and then, using the Picard iteration on the resulting averaged operator. This approach is equivalent to using T in the Krasnosel'skiĭ–Mann iteration with $\alpha_n = \alpha$ and for all $n \in \mathbb{N}$.

Halpern iteration

Given $x_0 \in \mathscr{H}$ and the nonexpansive operator T, the *Halpern iteration* reads as

$$x_{n+1} = (1-\alpha_n)x_0 + \alpha_n T x_n, \qquad \text{for all } n \in \mathbb{N}$$

where $\alpha_n \in (0,1)$. If $\operatorname{fix}(T)$ is nonempty, under some suitable assumptions on $(\alpha_n)_{n\in\mathbb{N}}$, Halpern iteration converges strongly to a $x^\star \in \operatorname{fix}(T)$ [Bauschke and Combettes, 2017, Theorem 30.1].

2.2 Convex optimization

An optimization problem of the form

$$\underset{x\in S}{\text{minimize}} \quad f_0(x) \tag{2.1}$$

where $f_0(x) : \mathscr{H} \to \overline{\mathbb{R}}$ is a convex function and $S \subseteq \mathscr{H}$ is a convex set is called a *convex optimization problem*. The function f is called the *objective function* or the *cost function* and x is referred to as the *optimization variable* or the *decision variable*. We refer to the set S as the *feasible set* or the *constraint set* of the problem. If the set S is the whole Hilbert space $\mathscr{H}$, then the problem is said to be *unconstrained*, otherwise, the problem is *constrained*. A point $x^\star \in S$ is a *solution* to the optimization problem (2.1), if $f_0(x^\star) = p^\star$ where

$$p^\star = \inf\{f_0(x) : x \in S\}.$$

The set of all solutions to the optimization problem (2.1) is called its *solution set*.

In this thesis, we consider a formulation of optimization problems that is slightly different compared to the one given by (2.1). This formulation is known as the *composite form* and is introduced in the following section.

Composite form of optimization problems

In this thesis, we consider optimization problems of the form

$$\underset{x \in \mathscr{H}}{\text{minimize}} \quad f(x) + g(x), \tag{2.2}$$

where $f : \mathscr{H} \to \mathbb{R}$ is a convex and *smooth function* and $g : \mathscr{H} \to \overline{\mathbb{R}}$ is a convex and potentially non-smooth function. The function f is said to be β-*smooth*, if its gradient ∇f is β-Lipschitz continuous. This formulation of convex optimization problems is more common in the setting of first-order optimization algorithms. Note that the optimization problem given in (2.1) can be cast in the form of (2.2) as well. For that, we can encode the constraint $x \in S$ into the formulation of problem (2.2) using the notion of *indicator function*. The indicator function of a set $S \in \mathscr{H}$ is denoted by $\iota_S : \mathscr{H} \to \overline{\mathbb{R}}$ and defined as

$$\iota_S(x) = \begin{cases} 0, & x \in S, \\ \infty, & x \notin S. \end{cases}$$

Note that if the set S is convex, then ι_S is a convex function. Therefore, setting $f(x) = f_0(x)$ and $g(x) = \iota_S(x)$ in the optimization problem (2.2) makes it equivalent to problem (2.1).

The optimality condition of problem (2.2) can be defined based on the notion of *subdifferential* operator. The *subdifferential* of a proper function $f : \mathscr{H} \to \overline{\mathbb{R}}$ at a point $x \in \mathscr{H}$ is denoted by $\partial f(x)$ and defined as

$$\partial f(x) = \{s \in \mathscr{H} : f(y) \geq f(x) + \langle s, y - x \rangle, \quad \text{for all } y \in \mathscr{H}\}.$$

Observe that by definition, ∂f is a set-valued operator, which is defined from $\mathscr{H}$ to its power-set $2^{\mathscr{H}}$. For a differentiable function f, $\partial f(x)$ is equal to $\{\nabla f(x)\}$, and thus, it is a single-valued operator.

By Fermat's rule [Bauschke and Combettes, 2017, Theorem 16.3], a point $x^\star$ is a solution to the optimization problem (2.2) if and only if it satisfies the optimality condition

$$0 \in \partial(f+g)(x). \tag{2.3}$$

2.3 Monotone inclusions

In this section, after introducing some notions, the general definition of a monotone inclusion problem and an approach to solve it are presented. Next, a specific class of monotone inclusion problems that is studied in this thesis, a standard approach for solving it, and examples of its applications are presented.

DEFINITION 1—MONOTONE OPERATOR
An operator $A\colon \mathscr{H} \to 2^{\mathscr{H}}$ is said to be *monotone* if for all $(x,u),(y,v) \in \operatorname{gra}(A)$,

$$\langle u-v, x-y\rangle \geq 0. \qquad \square$$

For instance, the identity operator and the subdifferential of a proper function are monotone operators.

DEFINITION 2—MAXIMAL MONOTONICTY
A monotone operator $A\colon \mathscr{H} \to 2^{\mathscr{H}}$ is *maximally monotone* if there exists no monotone operator $B\colon \mathscr{H} \to 2^{\mathscr{H}}$ such that $\operatorname{gra}(B)$ properly contains $\operatorname{gra}(A)$. $\square$

A monotone operator $A\colon \mathscr{H} \to 2^{\mathscr{H}}$ is maximal if and only if $\operatorname{ran}(\operatorname{Id}+A) = \mathscr{H}$. This is known as Minty's theorem [Minty, 1962]. For example, the subdifferential of a convex, closed, and proper function is maximally monotone.

DEFINITION 3—COCOERCIVE OPERATOR
An operator $T\colon \mathscr{H} \to \mathscr{H}$ is $\frac{1}{\beta}$-cocoercive with respect to the metric $\|\cdot\|_M$ with $\beta > 0$ and $M \in \mathscr{M}(\mathscr{H})$, if for all $x, y \in \mathscr{H}$,

$$\langle Tx-Ty, x-y\rangle \geq \frac{1}{\beta}\|Tx-Ty\|^2_{M^{-1}}. \qquad \square$$

Since cocoercive operators are monotone and Lipschitz continuous [Giselsson, 2021], they are maximally monotone, as well [Bauschke and Combettes, 2017, Corollary 20.28]. When the cocoercivity is considered with respect to the canonical norm, i.e., when $M = \operatorname{Id}$, we do not mention the underlying metric; that is, it is merely said that the operator at hand is $\frac{1}{\beta}$-cocoercive (note that cocercivity with respect to one metric implies cocoercivity with respect to other metrics, as well). For instance, the gradient of a smooth convex function is a cocoercive operator. In fact, a convex differentiable function f is β-smooth if and only if its gradient ∇f is $\frac{1}{\beta}$-cocoercive. This result is recognized as the Ballion–Haddad theorem [Bauschke and Combettes, 2017, Corollary 18.17].

DEFINITION 4—PROXIMAL OPERATOR
Given a function $f\colon \mathscr{H} \to \overline{\mathbb{R}}$, the proximal operator of f at $z \in \mathscr{H}$ is defined as

$$\operatorname{prox}_f(z) = \operatorname*{arg\,min}_{x\in\mathscr{H}} \left\{ f(x) + \frac{1}{2}\|z-x\|^2 \right\}. \qquad \square$$

For a convex, closed, and proper function f, the arg min uniquely exists, and thus, the proximal operator is single-valued.

DEFINITION 5—RESOLVENT OPERATOR
The operator $(\mathrm{Id}+\gamma A)^{-1}$ is called the *resolvent operator* of A and is denoted by $J_{\gamma A}$. $\square$

The resolvent of a maximally monotone operator A is single-valued and $\frac{1}{2}$-averaged. In addition, if A is maximally monotone its resolvent has a full domain, that is, $\operatorname{dom}(J_{\gamma A}) = \operatorname{ran}(\mathrm{Id}+\gamma A) = \mathscr{H}$. For a convex, closed, and proper function f we have $\operatorname{prox}_{\gamma f} = J_{\gamma \partial f}$.

Monotone inclusion and proximal point algorithm

The problem of finding $x \in \mathscr{H}$ such that

$$0 \in Ax, \tag{2.4}$$

where $A\colon \mathscr{H} \to 2^{\mathscr{H}}$ is a monotone operator, is called a *monotone inclusion* problem. This type of problem is closely related to convex optimization problems. For instance, the optimality condition of problem (2.2) is a monotone inclusion problem of the form above with $A = \partial(f+g)$. Therefore, the techniques that are used to solve monotone inclusions, can be utilized to find solutions to convex optimization problems.

The problem of finding zeros of the maximally monotone operator A can be cast as the problem of finding fixed points of an associated mapping. For that, letting $\gamma > 0$ and $x \in \mathscr{H}$, from the monotone inclusion (2.4), we obtain

$$\begin{aligned} 0 \in Ax \quad &\iff \quad 0 \in \gamma Ax \\ &\iff \quad x \in x + \gamma Ax = (\mathrm{Id}+\gamma A)x \\ &\iff \quad x = (\mathrm{Id}+\gamma A)^{-1}x = J_{\gamma A}(x) \end{aligned}$$

where in the first equivalence, both sides of the inclusion are scaled by γ; in the second equivalence, x is added to both sides; and in the last one, the single-valued operator $(\mathrm{Id}+\gamma A)^{-1}$ is applied to both sides of the inclusion.

As seen above, the zeros of A are the fixed points of the operator $J_{\gamma A}$, hence, in order to find a solution to the inclusion problem (2.4), one can solve the associated

fixed-point problem $x = J_{\gamma A}x$, and any solution of this fixed-point problem would be a zero of A. Since $J_{\gamma A}$ is averaged, it is guaranteed that the Picard iteration,

$$x_{n+1} = J_{\gamma A}(x_n),$$

converges to a fixed-point of $J_{\gamma A}$, if such a point exists.

In the iterative algorithm resulting from the Picard iteration, the step size γ has to be a fixed positive value. However, it was shown in [Rockafellar, 1976], that we can allow for a varying (iteration dependent) step size as in

$$x_{n+1} = J_{\gamma_n A}(x_n), \tag{2.5}$$

where $\gamma_n \geq \varepsilon > 0$ ($n \in \mathbb{N}$) is the step-size. This algorithm is known as the *proximal point algorithm*. Given $x_0 \in \mathscr{H}$, if $\operatorname{fix}(J_{\gamma_n A})$ is nonempty, the sequence of iterates generated by the proximal point algorithm converges weakly to a point in $\operatorname{zer}(A)$ [Rockafellar, 1976].

EXAMPLE 1
We are interested in finding a minimizer of the non-smooth function $g : \mathscr{H} \to \overline{\mathbb{R}}$ which is convex, closed, and proper. An optimality condition for the underlying optimization problem is given by $0 \in \partial g(x)$. Since g is convex, closed, and proper its subdifferential operator is maximally monotone; hence, this optimality condition is equivalent to the monotone inclusion (2.4) with $A = \partial g$. Substituting this in (2.5) and using $J_{\gamma \partial f} = \operatorname{prox}_{\gamma f}$, yield

$$x_{n+1} = \operatorname{prox}_{\gamma_n g}(x_n). \qquad \square$$

If the solution set of the underlying optimization problem is nonempty, provided that $\gamma_n \geq \varepsilon > 0$, for all $n \in \mathbb{N}$, the sequence of iterates generated by the proximal point algorithm converges weakly to a minimizer of g.

Forward–backward splitting

Let us consider the problem of finding zeros of the operator $A + C$, that is, the problem of finding $x \in \mathscr{H}$ such that

$$0 \in Ax + Cx, \tag{2.6}$$

where A is maximally monotone and C is a $\frac{1}{\beta}$-cocoercive operator with $\beta > 0$. Since the operator C is maximally monotone and has full domain, i.e., $\operatorname{dom}(C) = \mathscr{H}$, by [Bauschke and Combettes, 2017, Corollary 25.5], the operator $A + C$ is maximally monotone. Hence, this problem can basically be solved using proximal point algorithm. However, in many cases, evaluation of $J_{\gamma(A+C)}$ is computationally expensive which rules out applicability of the proximal point algorithm for finding roots of

$A+C$. In that case, we can use *operator splitting algorithms*. In simple words, splitting techniques enable us to decompose the maximally monotone operator at hand to separate operators such that the resulting algorithm utilizes the decomposed operators at different steps of the algorithm, each of which can be evaluated at a reduced computational cost. Below, it is shown how to arrive at the forward–backward operator associated with problem (2.6).

Let $\gamma > 0$ and $x \in \mathscr{H}$. Then, from the monotone inclusion (2.6), we obtain

$$\begin{aligned} 0 \in Ax + Cx \quad &\iff \quad 0 \in \gamma Ax + \gamma Cx \\ &\iff \quad -\gamma Cx \in \gamma Ax \\ &\iff \quad x - \gamma Cx \in (\mathrm{Id} + \gamma A)x \\ &\iff \quad x = (\mathrm{Id} + \gamma A)^{-1}(\mathrm{Id} - \gamma C)x = J_{\gamma A}(\mathrm{Id} - \gamma C)x \end{aligned} \tag{2.7}$$

where in the first equivalence, both sides of the inclusion are scaled by the positive coefficient γ; in the second and third equivalence, $-\gamma Cx$ and x are added to both sides of the inclusion; and in the last one, the single-valued operator $(\mathrm{Id}+\gamma A)^{-1}$ is applied to both sides of the inclusion. As seen, the zero set of $A+C$ is equal to the fixed-point set of the mapping $J_{\gamma A}(\mathrm{Id}-\gamma C)$. The mapping $J_{\gamma A}(\mathrm{Id}-\gamma C)$ is called the *forward–backward operator*. With $\gamma \in (0, \frac{2}{\beta})$, the forward–backward operator is averaged [Bauschke and Combettes, 2017, Proposition 26.1]. Therefore, we can use the Picard iteration to find a zero of $A+C$. The resulting iterative method reads as $x_{n+1} = J_{\gamma A}(\mathrm{Id}-\gamma C)x_n$, where γ is the step-size. This algorithm was introduced in [Bruck Jr, 1975; Lions and Mercier, 1979] and is known as the *forward–backward splitting algorithm*. To get its varying step-size variant, similar to the proximal point algorithm, it is possible to replace γ with an iteration dependent step-size γ_n. Then, we obtain

$$x_{n+1} = J_{\gamma_n A}(\mathrm{Id}-\gamma_n C)x_n. \tag{2.8}$$

If $\operatorname{zer}(A+C)$ is nonempty and $\gamma_n \in [\varepsilon, \frac{2}{\beta} - \varepsilon]$ with small enough $\varepsilon > 0$, the sequence of iterates generated by this algorithm converges weakly to a point in $\operatorname{zer}(A+C)$ [Combettes, 2004].

The following example exhibits an application of the approach described above for solving convex optimization problems.

EXAMPLE 2—PROXIMAL–GRADIENT METHOD
Consider the following problem

$$\underset{x \in \mathscr{H}}{\text{minimize}} \quad f(x) + g(x)$$

where $f : \mathscr{H} \to \mathbb{R}$ is a smooth convex function and $g : \mathscr{H} \to \overline{\mathbb{R}}$ is a convex, closed, proper, and potentially non-smooth function. Then, $x \in \mathscr{H}$ is a solution, if and only if, it satisfies the optimality condition $0 \in \partial(f+g)x$. This optimality condition can

be cast as $0 \in \nabla f(x) + \partial g(x)$. As ∇f is cocoercive and ∂g is maximally monotone, the monotone inclusion (2.6) with $A = \partial g$ and $C = \nabla f$ is equivalent to the desired optimality condition. Substituting these in (2.10), we get

$$x_{n+1} = \operatorname{prox}_{\gamma_n g}(x_n - \gamma_n \nabla f(x_n)).$$

This algorithm is recognized as the *proximal–gradient algorithm*. Given $x_0 \in \mathscr{H}$ and $\gamma_n \in [\varepsilon, \frac{2}{\beta} - \varepsilon]$ ($n \in \mathbb{N}$) for small enough $\varepsilon > 0$, the sequence of the generated iterates of this algorithm converges weakly to a minimizer of the underlying optimization problem, as long as such a point exists.

Observe that setting $g = 0$ in the proximal–gradient algorithm, results in

$$x_{n+1} = x_n - \gamma_n \nabla f(x_n),$$

which is the well known *gradient (descent) algorithm*. □

Preconditioned forward-backward algorithm

Let us consider the problem of finding $x \in \mathscr{H}$ such that

$$0 \in Ax + Cx, \tag{2.9}$$

where A is maximally monotone and C is a $\frac{1}{\beta}$-cocoercive operator ($\beta > 0$) with respect to the metric $\|\cdot\|_M$ ($M \in \mathscr{M}(\mathscr{H})$). To find an associated operator whose fixed-points are solutions to (2.9), we can take similar steps as in (2.7). However, in the third equivalence in (2.7), instead of adding x, Mx has to be added to both sides of the equivalence. Hence, the resulting operator would be of the form $(M + \gamma A)^{-1}(M - \gamma C)$ and is known as the *preconditioned forward–backward mapping*. An appropriate choice of the preconditioning M can improve the rate of convergence compared to the non-preconditioned case. In addition, if due to a particular structure in A the evaluation of $(I + \gamma A)^{-1}$ is expensive, the preconditioning can be used to make the evaluation of $(M + \gamma A)^{-1}$ computationally cheaper.

Given $x_0 \in \mathscr{H}$ and $M \in \mathscr{M}(\mathscr{H})$, the *preconditioned forward–backward algorithm* reads as

$$x_{n+1} = (M + \gamma_n A)^{-1}(M - \gamma_n C)x_n, \tag{2.10}$$

where $\gamma_n > 0$ is the step-size. Given $\gamma_n \in [\varepsilon, \frac{2}{\beta} - \varepsilon]$, for all $n \in \mathbb{N}$, with small enough $\varepsilon > 0$, and provided that $\operatorname{zer}(A + C)$ is nonempty, the sequence of iterates generated by this algorithm converges weakly to a point in $\operatorname{zer}(A + C)$ [Combettes and Vũ, 2014]. With $M = \mathrm{Id}$, the algorithm (2.10) reduces to the standard forward–backward algorithm (2.8).

In the rest of this section, we study a family of monotone inclusion problems that are not of the form (2.9) but can be transformed to that form using a primal-dual trick.

Monotone inclusions involving composition with linear operator

Let us consider the problem of finding $x \in \mathscr{H}$ such that

$$0 \in Ax + L^*BLx + Cx, \tag{2.11}$$

where $A : \mathscr{H} \to 2^{\mathscr{H}}$ and $B : \mathscr{K} \to 2^{\mathscr{K}}$ are maximally monotone operators, $L : \mathscr{H} \to \mathscr{K}$ is a linear and bounded operator, and $C : \mathscr{H} \to \mathscr{H}$ is $\frac{1}{\beta}$-cocoercive. Let us assume that due to the presence of L and L^* there is no easy way to evaluate the resolvent of $A + L^*BL$; thus, we cannot directly use the forward–backward method to solve this problem. However, it is possible to reformulate the monotone inclusion (2.11) such that the preconditioned forward–backward splitting can be used to find its zeros. To that end, by introducing an auxiliary variable $\mu \in B(Lx)$, that is called a *dual variable*, the monotone inclusion (2.11) can be cast as

$$0 \in \mathscr{A}w + \mathscr{C}w, \tag{2.12}$$

where $w = (x, \mu) \in \mathscr{H} \times \mathscr{K}$ and (with slight abuse of notation)

$$\mathscr{A} = \begin{bmatrix} A & L^* \\ -L & B^{-1} \end{bmatrix}, \qquad \mathscr{C} = \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix},$$

where $\mathscr{A}$ is maximally monotone by [Bauschke and Combettes, 2017, Proposition 26.32] and $\mathscr{C}$ is $1/\beta$-cocoercive with respect to the metric $\|\cdot\|_M$, with

$$M = \begin{bmatrix} I & -\tau L^* \\ -\tau L & \tau\sigma^{-1} I \end{bmatrix}.$$

Given $\sigma, \tau > 0$ such that $\sigma\tau\|L\|^2 < 1$, the operator M is strongly positive. With this translation, the resulting monotone inclusion (2.12) can be solved using the preconditioned forward–backward algorithm (2.10). Inserting $\mathscr{A}$, $\mathscr{C}$, and M into this algorithm, after some simplifications, the following algorithm is obtained:

$$\begin{aligned} x_{n+1} &= J_{\tau A}\left(x_n - \tau L^*\mu_n - \tau C x_n\right), \\ \mu_{n+1} &= J_{\sigma B^{-1}}\left(\mu_n + \sigma L(2x_{n+1} - x_n)\right). \end{aligned} \tag{2.13}$$

This algorithm is the basic form of the Condat–Vũ algorithm [Condat, 2013; Vũ, 2013]. Given $(x_0, \mu_0) \in \mathscr{H} \times \mathscr{K}$ and $\sigma, \tau > 0$ such that $\sigma\tau\|L\|^2 < 1$, the sequence $(x_n)_{n\in\mathbb{N}}$ converges weakly to a point in $\operatorname{zer}(A + L^*BL + C)$.

In what follows an application of the algorithm above in solving convex optimization problems is presented.

In some optimization problems, the objective function consists of two nonsmooth convex functions one of which has its input argument mapped by a linear and bounded operator L. Hence, due to presence of L, evaluation of the proximal operator of this function might be expensive. This case is covered in Example 3.

This example uses the notion of *conjugate function*; the conjugate function of a convex, closed, and proper function $h\colon \mathscr{H} \to \overline{\mathbb{R}}$, is denoted by $h^*\colon \mathscr{H} \to \overline{\mathbb{R}}$ and, for all $s \in \mathscr{H}$, defined as

$$h^*(s) := \sup_{x \in \mathscr{H}} \{\langle s, x\rangle - h(x)\}.$$

Given $\operatorname{prox}_h(\cdot)$, the proximal operator of h^*, for all $z \in \mathscr{H}$, can be found using

$$\sigma \operatorname{prox}_{\sigma^{-1}h}(z/\sigma) + \operatorname{prox}_{\sigma h^*}(z) = z,$$

where $\sigma > 0$. This identity is known as the *extended Moreau decomposition* [Beck, 2017].

EXAMPLE 3—PRIMAL–DUAL HYBRID GRADIENT METHOD

We are interested in solving

$$\underset{x \in \mathscr{H}}{\text{minimize}} \quad g(x) + h(Lx) \tag{2.14}$$

where $g : \mathscr{H} \to \overline{\mathbb{R}}$ and $h : \mathscr{K} \to \overline{\mathbb{R}}$ are convex, closed, proper, and potentially non-smooth functions and $L : \mathscr{H} \to \mathscr{K}$ is a bounded and linear operator. Under some assumptions [Bauschke and Combettes, 2017, Theorem 27.2], the optimality condition of this problem can be given by

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

A point $x \in \mathscr{H}$ is a solution to the optimization problem under study if and only if it satisfies the optimality condition above. Since ∂g and ∂h are maximally monotone and L is a linear and bounded operator, the monotone inclusion (2.11) with $A = \partial g$, $B = \partial h$, and $Cx = 0$ for all $x \in \mathscr{H}$ is equivalent to the desired optimality condition. Inserting these into (2.13), we get the iteration

$$\begin{aligned} x_{n+1} &= \operatorname{prox}_{\tau g}(x_n - \tau L^* \mu_n), \\ \mu_{n+1} &= \operatorname{prox}_{\sigma h^*}(\mu_n + \sigma L(2x_{n+1} - x_n)), \end{aligned}$$

which is known as the primal–dual hybrid gradient method or the Chambolle–Pock algorithm [Chambolle and Pock, 2011]. Given the initial point $(x_0, \mu_0) \in \mathscr{H} \times \mathscr{K}$ and $\sigma, \tau > 0$ such that $\sigma\tau\|L\|^2 < 1$, the sequence $(x_n)_{n\in\mathbb{N}}$ generated by this algorithm converges weakly to a solution, if one exists, of the optimization problem at hand. □

2.4 Overview of papers

There are many ways to modify the algorithms presented in Section 2.3 to achieve a better performance [Beck and Teboulle, 2009; Chambolle and Pock, 2011; Combettes and Vũ, 2014; Condat, 2013; d'Aspremont et al., 2021; Giselsson et al., 2016;

Kim, 2021; Nesterov, 1983; Themelis and Patrinos, 2019; Vũ, 2013; Walker and Ni, 2011; Zhang et al., 2020]. The focus of this thesis is to find new methods to improve performance of these algorithms. In this thesis, two general approaches are taken towards that end: a *direct approach* or the approach of *nested scheme* [d'Aspremont et al., 2021].

Here a simple example is used to illustrate how the *direct approach* works. Let T_{fb} be the forward–backward operator. Then, given $x_0 \in \mathscr{H}$, the standard forward–backward algorithm is of the form $x_{n+1} = T_{\mathrm{fb}}(x_n)$. Now, in a direct approach, one would alter the point at which T_{fb} is evaluated to change its convergence pattern. To that end, given $y_0 \in \mathscr{H}$, an iteration of the following form can be used

$$\begin{aligned} x_n &= T_{\mathrm{fb}}(y_n), \\ y_{n+1} &= \text{update equation of } y_n. \end{aligned}$$

The update rule of y_n can, for instance, be defined based on a linear combination of the past iterates. If the update rule is devised appropriately, this strategy can result in performance improvement. This approach have been used in various acceleration techniques for optimization problems (Nesterov's accelerated gradient method [Nesterov, 1983] and FISTA [Beck and Teboulle, 2009]), monotone inclusions (accelerated proximal point method [Kim, 2021]), or fixed-point iterations (Anderson acceleration [Walker and Ni, 2011]).

The general idea of *nested scheme* is to combine two optimization algorithms. One as the outer iteration inside which the other algorithm is run as the inner loop. The nested optimization scheme switches between the two algorithms based on some rule. If the switching strategy is designed appropriately, combining the algorithms in this way can enhance the performance. This approach have been used in acceleration techniques for optimization problems and fixed-point iterations [Giselsson et al., 2016; Lin et al., 2015; Scieur et al., 2017; Scieur et al., 2016; Themelis and Patrinos, 2019; Zhang et al., 2020].

In what follows, a concise overview of the papers that are included in the thesis is presented. In Papers I-III, a direct approach is utilized to develop new algorithms, and in Paper IV, a nested scheme is used for that purpose.

Paper I

This paper presents a weakly convergent extension to the standard forward–backward splitting method to solve the monotone inclusion (2.9). In this paper, the direct approach is used to improve performance of the standard forward–backward algorithm. A simplified version of the proposed algorithm is given by

$$\begin{aligned} p_n &= (\mathrm{Id} + \tfrac{1}{\beta}A)^{-1} \circ (\mathrm{Id} - \tfrac{1}{\beta}C)(y_n) \\ y_{n+1} &= p_n - u_n + u_{n+1} \end{aligned}$$

where u_n is a *deviation vector*. The only requirement on the deviation vector to guarantee convergence is that its norm has to be bounded by a quantity that can be

computed at each iteration. This approach gives great flexibility to the algorithm and paves the way for designing new and improved forward–backward-based algorithms, while retaining global convergence guarantees. For instance, using the proposed algorithm, variations of the primal–dual hybrid gradient algorithm and the Krasnosel'skiĭ–Mann iteration that incorporate deviations are presented.

Paper II

This paper is an extension to the work done in Paper I and presents a novel variation of the standard forward–backward splitting algorithm to solve the monotone inclusion problem (2.9). This algorithm, in addition to including *deviation vectors*, incorporates iterates from the past. The past information is incorporated in different places of the algorithm in the form of momentum-like terms. Several new algorithms can be derived based on the proposed method. For instance, for a particular choice of the deviations and the parameters of the algorithm, it reduces to the Halpern iteration and the accelerated proximal point method that both converge as $\mathcal{O}(\frac{1}{n^2})$ in squared norm of the fixed-point residual.

Paper III

This paper showcases an application of the algorithm proposed in Paper I. The proposed algorithm uses the main algorithm of Paper I as the basis and combines it with the extrapolation technique used in Anderson acceleration to improve local convergence. In particular, the extrapolation technique of Anderson acceleration is used to select a direction for the deviation vector whose norm is bounded by a quantity that is computable based on available information. Combining these two methods leads to a fast and globally convergent algorithm.

Paper IV

With the goal of alleviating the slow convergence rate of first-order optimization algorithms, this paper proposes a framework for combining a family of stochastic variance reduced algorithms such as SVRG with a fast locally convergent method like Anderson acceleration, using a nested scheme. The variance reduced optimization method is used as the outer loop whose role is to ensure global convergence of the whole scheme. Within the outer loop, an iterative algorithm is used which has the role of accelerating the convergence of the optimization scheme. The scheme switches between the two based on a specific safeguarding condition. As a result of such a combination, the resulting optimization scheme can exhibit good performance while guaranteeing global convergence.

3

Publications

In what follows, a list of the papers that form the main part of the thesis is presented, along with a statement on the contributions of the authors for each paper.

Paper I

Sadeghi, H., S. Banert, and P. Giselsson (2021). *Forward–backward splitting with deviations for monotone inclusions*. arXiv: 2112.00776v1 [math.OC].

The general idea of this paper was proposed by S. Banert. Most of the results were derived through collaboration between the authors. Implementations, numerical experiments, and preparation of the manuscript were conducted by H. Sadeghi.

Paper II

Sadeghi, H., S. Banert, and P. Giselsson (2022). *Incorporating history and deviations in forward–backward splitting*. arXiv: 2208.05498 [math.OC].

H. Sadeghi contributed with the majority of the work including the general idea of the work, derivation of the results, and writing of the manuscript. Part of the results were found through an idea from S. Banert. S. Banert and P. Giselsson helped revising the manuscript.

Paper III

Sadeghi, H., S. Banert, and P. Giselsson (2021). *Dwifob: a dynamically weighted inertial forward–backward algorithm for monotone inclusions*. arXiv: 2203.00028 [math.OC].

H. Sadeghi contributed with the majority of the work including the idea of the work, implementations, and writing of the manuscript. S. Banert and P. Giselsson helped revising the manuscript.

Paper IV

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

The general idea of the paper was proposed by P. Giselsson. The results were found by H. Sadeghi in collaboration with P. Giselsson. Implementations, numerical evaluations, and writing of the manuscript were carried out by H. Sadeghi.

Bibliography

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

Beck, A. (2017). *First-order methods in optimization*. SIAM. DOI: 10.1137/1.9781611974997.

Beck, A. and M. Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". *SIAM journal on imaging sciences* **2**:1, pp. 183–202. DOI: 10.1137/080716542.

Berinde, V. and F. Takens (2007). *Iterative approximation of fixed points*. Vol. 1912. Berlin: Springer. DOI: https://doi.org/10.1007/978-3-540-72234-2.

Borwein, J. M. (2010). "Fifty years of maximal monotonicity". *Optimization Letters* **4**:4, pp. 473–490. DOI: 10.1007/s11590-010-0178-x.

Boyd, S. P. and L. Vandenberghe (2004). *Convex optimization*. Cambridge university press. DOI: 10.1017/CBO9780511804441.

Bruck Jr, R. E. (1975). "An iterative solution of a variational inequality for certain monotone operators in hilbert space". *Bulletin of the American Mathematical Society* **81**:5, pp. 890–892. DOI: bams/1183537239.

Chambolle, A. and T. Pock (2011). "A first-order primal–dual algorithm for convex problems with applications to imaging". *Journal of Mathematical Imaging and Vision* **40**:1, pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

Combettes, P. L. (2018). "Monotone operator theory in convex optimization". *Mathematical Programming* **170**:1, pp. 177–206. DOI: 10.1007/s10107-018-1303-3.

Combettes, P. L. (2004). "Solving monotone inclusions via compositions of nonexpansive averaged operators". *Optimization* **53**:5-6, pp. 475–504. DOI: 10.1080/02331930412331327157.

Combettes, P. L. and B. C. Vũ (2014). "Variable metric forward–backward splitting with applications to monotone inclusions in duality". *Optimization* **63**:9, pp. 1289–1318. DOI: 10.1080/02331934.2012.733883.

Condat, L. (2013). "A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms". *Journal of Optimization Theory and Applications* **158**:2, pp. 460–479. DOI: 10.1007/s10957-012-0245-9.

d'Aspremont, A., D. Scieur, A. Taylor, et al. (2021). "Acceleration methods". *Foundations and Trends® in Optimization* **5**:1-2, pp. 1–245. DOI: 10.1561/2400000036.

Darup, M. S., G. Book, and P. Giselsson (2019). "Towards real-time admm for linear mpc". In: *2019 18th European Control Conference (ECC)*, pp. 4276–4282. DOI: 10.23919/ECC.2019.8796239.

Everitt, B. (2012). *Introduction to optimization methods and their application in statistics*. Springer science & business media. DOI: 10.1007/978-94-009-3153-4.

Gilli, M., D. Maringer, and E. Schumann (2019). *Numerical methods and optimization in finance*. Academic Press. DOI: 10.1016/C2017-0-01621-X.

Giselsson, P. (2021). "Nonlinear forward-backward splitting with projection correction". *SIAM Journal on Optimization* **31**:3, pp. 2199–2226. DOI: 10.1137/20M1345062.

Giselsson, P., M. Fält, and S. Boyd (2016). "Line search for averaged operator iteration". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 1015–1022. DOI: 10.1109/CDC.2016.7798401.

Giselsson, P. and A. Rantzer (2014). "Generalized accelerated gradient methods for distributed mpc based on dual decomposition". In: *Distributed model predictive control made easy*. Springer, pp. 309–325. DOI: 10.1007/978-94-007-7006-5_19.

Kelley, C. T. (1995). *Iterative methods for linear and nonlinear equations*. SIAM. DOI: 10.1137/1.9781611970944.

Kelley, C. T. (1999). *Iterative methods for optimization*. SIAM. DOI: 10.1137/1.9781611970920.

Kim, D. (2021). "Accelerated proximal point method for maximally monotone operators". *Mathematical Programming* **190**:1, pp. 57–87. DOI: 10.1007/s10107-021-01643-0.

Le, Q. V., J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng (2011). "On optimization methods for deep learning". In: *ICML*. DOI: 10.5555/3104482.3104516.

Lin, H., J. Mairal, and Z. Harchaoui (2015). "A universal catalyst for first-order optimization". *Advances in neural information processing systems* **28**. DOI: 10.5555/2969442.2969617.

Lions, P.-L. and B. Mercier (1979). "Splitting algorithms for the sum of two nonlinear operators". *SIAM Journal on Numerical Analysis* **16**:6, pp. 964–979. DOI: 10.1137/0716071.

Minty, G. J. (1962). "Monotone (nonlinear) operators in hilbert space". *Duke mathematical journal* **29**:3, pp. 341–346. DOI: 10.1215/S0012-7094-62-02933-2.

Minty, G. J. (1969). "On some aspects of the theory of monotone operators". *Theory and Applications of Monotone Operators (Proc. NATO Advanced Study Inst., Venice, 1968)*, pp. 67–82.

Nesterov, Y. E. (1983). "A method for solving the convex programming problem with convergence rate o (1/k$\hat{2}$)". In: *Dokl. akad. nauk Sssr*. Vol. 269, pp. 543–547. URL: https://vsokolov.org/courses/750/2018/files/nesterov.pdf.

Perea-Lopez, E., B. E. Ydstie, and I. E. Grossmann (2003). "A model predictive control strategy for supply chain optimization". *Computers & Chemical Engineering* **27**:8-9, pp. 1201–1218. DOI: 10.1016/S0098-1354(03)00047-4.

Rockafellar, R. T. (1970). *Convex analysis*. Vol. 18. Princeton university press. DOI: 10.1515/9781400873173.

Rockafellar, R. T. (1976). "Monotone operators and the proximal point algorithm". *SIAM journal on control and optimization* **14**:5, pp. 877–898. DOI: 10.1137/0314056.

Ryu, E. K. and S. Boyd (2016). "Primer on monotone operator methods". *Appl. Comput. Math* **15**:1, pp. 3–43. URL: https://web.stanford.edu/~boyd/papers/pdf/monotone_primer.pdf.

Ryu, E. K. and W. Yin (2022). *Large-scale convex optimization via monotone operators*. Cambridge University Press. URL: https://large-scale-book.mathopt.com/.

Scieur, D., F. Bach, and A. d'Aspremont (2017). "Nonlinear acceleration of stochastic algorithms". *Advances in Neural Information Processing Systems* **30**. URL: https://proceedings.neurips.cc/paper/2017/file/fca0789e7891cbc0583298a238316122-Paper.pdf.

Scieur, D., A. d'Aspremont, and F. Bach (2016). "Regularized nonlinear acceleration". *Advances In Neural Information Processing Systems* **29**. URL: https://proceedings.neurips.cc/paper/2016/file/bbf94b34eb32268ada57a3be5062fe7d-Paper.pdf.

Sra, S., S. Nowozin, and S. J. Wright (2012). *Optimization for machine learning*. Mit Press. URL: https://mitpress.mit.edu/9780262537766/optimization-for-machine-learning/.

Themelis, A. and P. Patrinos (2019). "Supermann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators". *IEEE Transactions on Automatic Control* **64**:12, pp. 4875–4890. DOI: 10.1109/TAC.2019.2906393.

Vũ, B. C. (2013). "A splitting algorithm for dual monotone inclusions involving cocoercive operators". *Advances in Computational Mathematics* **38**:3, pp. 667–681. DOI: 10.1007/s10444-011-9254-8.

Walker, H. F. and P. Ni (2011). "Anderson acceleration for fixed-point iterations". *SIAM Journal on Numerical Analysis* **49**:4, pp. 1715–1735. DOI: 10.1137/10078356X.

Yin, Y. (2002). "Multiobjective bilevel optimization for transportation planning and management problems". *Journal of Advanced Transportation* **36**:1, pp. 93–105. DOI: 10.1002/atr.5670360106.

Zhang, J., B. O'Donoghue, and S. Boyd (2020). "Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations". *SIAM Journal on Optimization* **30**:4, pp. 3170–3197. DOI: 10.1137/18M1232772.

Paper I

Forward–Backward Splitting with Deviations for Monotone Inclusions

Hamed Sadeghi Sebastian Banert Pontus Giselsson

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'skiĭ–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.

Available on arXiv.

1. Introduction

Forward–backward (FB) splitting [Bruck, 1975; Lions and Mercier, 1979; Passty, 1979] has been extensively used to solve structured monotone inclusion problems of finding $x \in \mathscr{H}$ such that

$$0 \in Ax + Cx, \tag{I.1}$$

where $A\colon \mathscr{H} \to 2^{\mathscr{H}}$ is a maximally monotone operator, $C\colon \mathscr{H} \to \mathscr{H}$ is a cocoercive operator, and $\mathscr{H}$ 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

$$x_{n+1} = (\mathrm{Id} + \gamma_n A)^{-1} \circ (\mathrm{Id} - \gamma_n C) x_n, \tag{I.2}$$

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 $f\colon \mathscr{H} \to \mathbb{R}$ be a convex, differentiable function whose gradient is Lipschitz continuous and $g\colon \mathscr{H} \to \mathbb{R} \cup \{\pm\infty\}$ be a proper, convex and lower semicontinuous function, and let $A = \partial g$ (the subdifferential of g) and $C = \nabla f$. Then, (I.1) is the problem of finding a minimizer of $f + g$, and (I.2) describes the *proximal gradient method* [Combettes and Pesquet, 2011].

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

$$\begin{aligned} p_n &= (\mathrm{Id} + \tfrac{1}{\beta} A)^{-1} \circ (\mathrm{Id} - \tfrac{1}{\beta} C)(x_n + u_n) \\ x_{n+1} &= p_n - u_n \end{aligned} \tag{I.3}$$

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 (I.3) reduces to the standard FB step in (I.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*

$$\|u_n\|^2 \leq \tfrac{1-\varepsilon}{4} \|p_{n-1} - x_{n-1} + u_{n-1}\|^2, \tag{I.4}$$

where $\varepsilon \in [0, 1)$ 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 [Nocedal and Wright, 2006, Chapter 3] ensure a sufficient decrease in the objective function value, and trust-region methods [Nocedal and Wright, 2006, Chapter 4] are based on a quadratic model having sufficient accuracy within a given radius. Recently, a norm condition similar to (I.4) has been combined with a deep-learning approach to speed up the convergence [Banert et al., 2021]. Even for monotone operators, line-search

strategies with safeguarding have been developed, see [Tseng, 2000, Eq. (2.4)] for an example. In contrast to line search, (I.4) does not require to compute (and possibly reject) several steps per iteration. For other examples of safeguarding, see [Giselsson et al., 2016; Sadeghi and Giselsson, 2021; Themelis and Patrinos, 2019; Zhang et al., 2020].

Our main algorithm (Algorithm 1) is more general than (I.3). It uses two deviation vectors and a slightly more involved safeguard condition. A similar algorithm with deviation vectors has been proposed in [Banert et al., 2021] 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 [Chambolle and Pock, 2011] and Condat–Vũ [Condat, 2013; Vũ, 2013] methods that both are preconditioned FB methods [He and Yuan, 2012]. To facilitate the derivation of these special cases, we derive our algorithm with explicit preconditioning, such as in [Chouzenoux et al., 2013; Combettes and Vũ, 2012; Giselsson, 2021; Giselsson and Boyd, 2015; Giselsson and Boyd, 2014a; Giselsson and Boyd, 2014b; Pock and Chambolle, 2011; Raguet and Landrieu, 2015].

Our algorithm is also related to inexact FB methods, which are studied in the framework of monotone inclusions [Raguet et al., 2013; Solodov and Svaiter, 2000; Solodov and Svaiter, 2001; Vũ, 2013] and in a convex optimization setting [Condat, 2013; Schmidt et al., 2011; Villa et al., 2013]. 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'skiĭ–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 [Polyak, 1964] momentum direction (see [Sadeghi et al., 2022a] for another novel usage of the deviations in a primal–dual setting). The resulting algorithm bears similarities with the inertial FB methods [Alvarez, 2000; Alvarez and Attouch, 2001; Attouch and Cabot, 2019; Cholamjiak et al., 2018; Lorenz and Pock, 2015] 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 [Lorenz and Pock, 2015].

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 [Condat, 2013; Raguet et al., 2013; Vũ, 2013; Villa et al., 2013], 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 [Banert et al., 2021], our work considers more general monotone inclusion problems. Hence, we immediately obtain the algorithms of Chambolle–Pock [Chambolle and Pock, 2011] and Krasnosel'skiĭ–Mann with deviations as special cases. Moreover, our convergence result is slightly stronger than [Banert et al., 2021, 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 [Polyak, 1964]—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 $\mathbb{R}$; $\mathscr{H}$ and $\mathscr{K}$ 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 $M\colon \mathscr{H} \to \mathscr{H}$ is said to be *strongly positive* if there exists some $c > 0$ such that $\langle x, Mx\rangle \geq c\|x\|^2$ for all $x \in \mathscr{H}$. We use the notation $\mathscr{M}(\mathscr{H})$ to denote the set of linear, self-adjoint, strongly positive operators on $\mathscr{H}$. For $M \in \mathscr{M}(\mathscr{H})$ and for all $x, y \in \mathscr{H}$, 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

$$\langle x,y\rangle \leq \frac{\omega}{2}\|x\|_M^2 + \frac{1}{2\omega}\|y\|_{M^{-1}}^2$$

holds for all $x,y \in \mathscr{H}$, $\omega > 0$, and $M \in \mathscr{M}(\mathscr{H})$. Hence, with the same variables,

$$\|x+y\|_M^2 = \|x\|_M^2 + \|y\|_M^2 + 2\langle x,My\rangle \leq (1+\omega)\|x\|_M^2 + \frac{1+\omega}{\omega}\|y\|_M^2.$$

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

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

Let $M \in \mathscr{M}(\mathscr{H})$. An operator $T\colon \mathscr{H} \to \mathscr{H}$ is said to be

(i) *L-Lipschitz continuous* ($L \geq 0$) w.r.t. $\|\cdot\|_M$ if

$$\|Tx - Ty\|_{M^{-1}} \leq L\|x-y\|_M \qquad \text{for all } x,y \in \mathscr{H};$$

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

$$\langle Tx - Ty, x-y\rangle \geq \frac{1}{\beta}\|Tx-Ty\|_{M^{-1}}^2 \qquad \text{for all } x,y \in \mathscr{H};$$

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

(iv) *firmly nonexpansive* if

$$\|Tx-Ty\|^2 + \|(\mathrm{Id}-T)x - (\mathrm{Id}-T)y\|^2 \leq \|x-y\|^2 \qquad \text{for all } x,y \in \mathscr{H}.$$

By the Cauchy–Schwarz inequality, a $\frac{1}{\beta}$-cocoercive operator is β-Lipschitz continuous. The *resolvent* of a maximally monotone operator $A\colon \mathscr{H} \to 2^{\mathscr{H}}$ is denoted by $J_{\gamma A}\colon \mathscr{H} \to \mathscr{H}$ and defined as $J_{\gamma A} := (\mathrm{Id} + \gamma A)^{-1}$. $J_{\gamma A}$ has full domain, is firmly nonexpansive [Bauschke and Combettes, 2017, Corollary 23.8], and is single-valued.

3. Forward–backward splitting with deviations

We consider structured monotone inclusion problems of the form

$$0 \in Ax + Cx, \tag{I.5}$$

that satisfy the following assumptions.

ASSUMPTION 1 Assume that $\beta > 0$,

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

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

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

Observe that, as a cocoercive operator is maximally monotone [Bauschke and Combettes, 2017, Corollary 20.28], and since C has a full domain, the operator $A+C$ is maximally monotone [Bauschke and Combettes, 2017, Corollary 25.5].

We present and prove convergence for the following extended variant of FB splitting for solving (I.5).

Algorithm 1 Forward–backward splitting with deviations

Input: initial point $x_0 \in \mathscr{H}$, the sequences $(\zeta_n)_{n\in\mathbb{N}}$, $(\lambda_n)_{n\in\mathbb{N}}$, and $(\gamma_n)_{n\in\mathbb{N}}$ as per Assumption 2, and the metric $\|\cdot\|_M$ with $M \in \mathscr{M}(\mathscr{H})$.
set: $u_0 = v_0 = 0$
for $n = 0, 1, 2, \ldots$ **do**
  $y_n = x_n + u_n$
  $z_n = x_n + \frac{(1-\lambda_n)\gamma_n\beta}{2-\lambda_n\gamma_n\beta} u_n + v_n$
  $p_n = (M + \gamma_n A)^{-1}(Mz_n - \gamma_n C y_n)$
  $x_{n+1} = x_n + \lambda_n(p_n - z_n)$
  choose u_{n+1} and v_{n+1} such that

$$\frac{\lambda_{n+1}\gamma_{n+1}\beta}{2-\lambda_{n+1}\gamma_{n+1}\beta}\|u_{n+1}\|_M^2 + \frac{\lambda_{n+1}(2-\lambda_{n+1}\gamma_{n+1}\beta)}{4-2\lambda_{n+1}-\gamma_{n+1}\beta}\|v_{n+1}\|_M^2 \leq \zeta_n \ell_n^2 \tag{I.6}$$

  is satisfied, where

$$\ell_n^2 = \frac{\lambda_n(4-2\lambda_n-\gamma_n\beta)}{2}\left\| p_n - x_n + \frac{\lambda_n\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n - \frac{2(1-\lambda_n)}{4-2\lambda_n-\gamma_n\beta}v_n \right\|_M^2 \tag{I.7}$$

end for

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

ASSUMPTION 2 Choose $\varepsilon \in \left(0, \min\left(1, \frac{4}{3+\beta}\right)\right)$, and assume that, for all $n \in \mathbb{N}$, the following hold:

(i) $0 \leq \zeta_n \leq 1-\varepsilon$;
(ii) $\varepsilon \leq \gamma_n \leq \frac{4-3\varepsilon}{\beta}$; and
(iii) $\varepsilon \leq \lambda_n \leq 2 - \frac{\gamma_n\beta}{2} - \frac{\varepsilon}{2}$. □

The sequence $(\zeta_n)_{n\in\mathbb{N}}$ relates the norm of the deviation vector (u_{n+1}, v_{n+1}) in (I.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 (I.6) and (I.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

$$4 - 2\lambda_n - \gamma_n\beta \geq \varepsilon \tag{I.8}$$

and

$$2 - \lambda_n\gamma_n\beta \geq 2 - \left(2 - \frac{\gamma_n\beta}{2} - \frac{\varepsilon}{2}\right)\gamma_n\beta = \frac{\varepsilon\gamma_n\beta}{2} + 2\left(1 - \frac{\gamma_n\beta}{2}\right)^2 \geq \frac{\varepsilon^2\beta}{2}. \tag{I.9}$$

Algorithm 1 handles the evaluation of C and A in step 6 differently than the standard FB method (I.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 $\mathscr{H}$ with non-empty interior (if $\ell_n^2 > 0$ in step 8). Hence, the degrees of freedom in the parameter choice are determined by the dimension of $\mathscr{H}$. It is important to note that the upper bound ℓ_n^2, as it is seen from (I.7), is computable at the time of selecting u_{n+1} and v_{n+1}. See [Sadeghi et al., 2022b] 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 (I.6) is already satisfied, and Algorithm 1 reduces to the relaxed preconditioned FB iteration

$$\begin{aligned} p_n &= (M + \gamma_n A)^{-1}(Mx_n - \gamma_n Cx_n), \\ x_{n+1} &= x_n + \lambda_n(p_n - x_n). \end{aligned}$$

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

EXAMPLE 2
With $M = \mathrm{Id}$, $\gamma_n = \frac{1}{\beta}$, $\lambda_n = 1$, $v_n = u_n$, and $\zeta_n = 1 - \varepsilon$ ($n \in \mathbb{N}$), we recover the simplified version from (I.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

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

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

with the norm condition

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

EXAMPLE 4—FORWARD ITERATION WITH DEVIATIONS
With $Ax = \{0\}$ for all $x \in \mathscr{H}$, $v_n = 0$, and $\gamma_n = 2/\beta$ for all $n \in \mathbb{N}$, Algorithm 1 simplifies to the iteration

$$\begin{aligned} y_n &= x_n + u_n, \\ x_{n+1} &= x_n - \tfrac{2\lambda_n}{\beta}M^{-1}Cy_n \end{aligned}$$

with the norm condition

$$\frac{\lambda_{n+1}}{1-\lambda_{n+1}}\|u_{n+1}\|_M^2 \leq \zeta_n\lambda_n(1-\lambda_n)\left\|\frac{1}{1-\lambda_n}u_n - \frac{2}{\beta}M^{-1}Cy_n\right\|_M^2 \qquad \square$$

EXAMPLE 5—BACKWARD ITERATION WITH DERIVATIONS
With $Cx = 0$ for all $x \in \mathscr{H}$ and $u_n = 0$ for all $n \in \mathbb{N}$, Algorithm 1 simplifies to the iteration

$$\begin{aligned} p_n &= (M+\gamma_n A)^{-1}M(x_n + v_n), \\ x_{n+1} &= x_n + \lambda_n(p_n - x_n - v_n). \end{aligned}$$

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

$$\frac{\lambda_{n+1}}{2-\lambda_{n+1}}\|v_{n+1}\|_M^2 \leq \zeta_n\lambda_n(2-\lambda_n)\left\|p_n - x_n - \frac{1-\lambda_n}{2-\lambda_n}v_n\right\|_M^2. \qquad \square$$

REMARK 1 Many works exist that allow for error terms in FB algorithms [Condat, 2013; Raguet et al., 2013; Vũ, 2013; Villa et al., 2013]. Convergence is often based on a summability argument so that any summable sequence of errors is allowed. The strength of our condition (I.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. $\square$

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 $n\in\mathbb{N}$, $(p_n,\Delta_n)\in \operatorname{gra}(A+C)$, where

$$\Delta_n := \frac{Mz_n - Mp_n}{\gamma_n} - (Cy_n - Cp_n).$$

Moreover,

$$\begin{aligned}\|\Delta_n\|_{M^{-1}} &\leq \frac{1}{2\gamma_n}\|2(z_n-p_n)-\beta\gamma_n(y_n-p_n)\|_M + \frac{\beta}{2}\|y_n-p_n\|_M \\ &= \frac{1}{2\gamma_n}\left\|(2-\beta\gamma_n)(x_n-p_n) - \frac{\lambda_n\gamma_n\beta(2-\gamma_n\beta)}{2-\lambda_n\gamma_n\beta}u_n + 2v_n\right\|_M \\ &\quad + \frac{\beta}{2}\|x_n-p_n+u_n\|_M. \end{aligned} \tag{I.10}$$

□

Before we prove Lemma 1, note that the right-hand side of (I.10) only contains data that is computed in Algorithm 1, whereas evaluating Δ_n requires the knowledge of Cp_n. Therefore, (I.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, (I.10) reduces to

$$\|\Delta_n\| \leq \beta\|y_n-p_n\| = \beta\left\|\left(\mathrm{Id}-\left(\mathrm{Id}+\tfrac{1}{\beta}A\right)^{-1}\circ\left(\mathrm{Id}-\tfrac{1}{\beta}C\right)\right)y_n\right\|.$$

Hence, the right-hand side of (I.10) plays the role of a residual for the iteration in Algorithm 1.

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

$$\frac{Mz_n - Mp_n}{\gamma_n} - Cy_n \in Ap_n, \tag{I.11}$$

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

Furthermore, we have

$$\begin{aligned}
\|\Delta_n\|_{M^{-1}}^2 &- \left(\frac{1}{2\gamma_n}\|2(z_n-p_n)-\beta\gamma_n(y_n-p_n)\|_M+\frac{\beta}{2}\|y_n-p_n\|_M\right)^2 \\
&= \frac{1}{\gamma_n^2}\|z_n-p_n\|_M^2+\|Cy_n-Cp_n\|_{M^{-1}}^2-\frac{2}{\gamma_n}\langle z_n-p_n,Cy_n-Cp_n\rangle \\
&\quad -\frac{1}{4\gamma_n^2}\|2(z_n-p_n)-\beta\gamma_n(y_n-p_n)\|_M^2-\frac{\beta^2}{4}\|y_n-p_n\|_M^2 \\
&\quad -\frac{\beta}{2\gamma_n}\|2(z_n-p_n)-\beta\gamma_n(y_n-p_n)\|_M\|y_n-p_n\|_M \\
&= \|Cy_n-Cp_n\|_{M^{-1}}^2-\frac{2}{\gamma_n}\langle z_n-p_n,Cy_n-Cp_n\rangle \\
&\quad -\frac{\beta^2}{2}\|y_n-p_n\|_M^2+\frac{\beta}{\gamma_n}\langle z_n-p_n,y_n-p_n\rangle_M \\
&\quad -\frac{\beta}{2\gamma_n}\|2(z_n-p_n)-\beta\gamma_n(y_n-p_n)\|_M\|y_n-p_n\|_M \\
&= \|Cy_n-Cp_n\|_{M^{-1}}^2-\beta\langle y_n-p_n,Cy_n-Cp_n\rangle \\
&\quad +\frac{1}{\gamma_n}\left\langle 2(z_n-p_n)-\beta\gamma_n(y_n-p_n),\frac{\beta}{2}M(y_n-p_n)-(Cy_n-Cp_n)\right\rangle \\
&\quad -\frac{\beta}{2\gamma_n}\|2(z_n-p_n)-\beta\gamma_n(y_n-p_n)\|_M\|y_n-p_n\|_M.
\end{aligned} \tag{I.12}$$

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

$$\|y_n-p_n\|_M \geq \frac{2}{\beta}\left\|Cy_n-Cp_n-\frac{\beta}{2}M(y_n-p_n)\right\|_{M^{-1}}. \tag{I.13}$$

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

LEMMA 2 (LYAPUNOV INEQUALITY) Suppose that Assumption 1 and Assumption 2 hold. Let $(x_n)_{n\in\mathbb{N}}$, $(u_n)_{n\in\mathbb{N}}$, $(v_n)_{n\in\mathbb{N}}$, $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ be sequences generated by Algorithm 1 and $x^\star$ be an arbitrary point in $\operatorname{zer}(A+C)$. Then,

$$\|x_{n+1}-x^\star\|_M^2+\ell_n^2 \leq \|x_n-x^\star\|_M^2+\tfrac{\lambda_n\gamma_n\beta}{2-\lambda_n\gamma_n\beta}\|u_n\|_M^2+\tfrac{\lambda_n(2-\lambda_n\gamma_n\beta)}{4-2\lambda_n-\gamma_n\beta}\|v_n\|_M^2 \tag{I.14}$$

and

$$\|x_{n+1}-x^\star\|_M^2+\ell_n^2 \leq \|x_n-x^\star\|_M^2+\zeta_{n-1}\ell_{n-1}^2 \tag{I.15}$$

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

$$\frac{Mz_n - Mp_n}{\gamma_n} - Cy_n \in Ap_n. \tag{I.16}$$

Since $x^\star \in \operatorname{zer}(A+C)$, we also have

$$-Cx^\star \in Ax^\star. \tag{I.17}$$

Using (I.16), (I.17), and the monotonicity of A gives

$$0 \leq \left\langle \frac{Mz_n - Mp_n}{\gamma_n} - Cy_n + Cx^\star, p_n - x^\star \right\rangle. \tag{I.18}$$

By the $1/\beta$-cocoercivity of C w.r.t. $\|\cdot\|_M$ we have

$$\tfrac{1}{\beta}\|Cy_n - Cx^\star\|_{M^{-1}}^2 \leq \langle Cy_n - Cx^\star, y_n - x^\star\rangle. \tag{I.19}$$

Adding (I.18) and (I.19) yields

$$0 \leq \left\langle \frac{Mz_n - Mp_n}{\gamma_n}, p_n - x^\star \right\rangle + \langle Cy_n - Cx^\star, y_n - p_n\rangle - \tfrac{1}{\beta}\|Cy_n - Cx^\star\|_{M^{-1}}^2.$$

Then, from step 7 in Algorithm 1, we substitute $z_n - p_n = \frac{1}{\lambda_n}(x_n - x_{n+1})$ to obtain

$$\begin{aligned}
0 &\leq \tfrac{1}{\gamma_n\lambda_n}\langle x_n - x_{n+1}, p_n - x^\star\rangle_M + \langle Cy_n - Cx^\star, y_n - p_n\rangle - \tfrac{1}{\beta}\|Cy_n - Cx^\star\|_{M^{-1}}^2 \\
&= \tfrac{1}{2\gamma_n\lambda_n}\Big(\|x_n - x^\star\|_M^2 + \|x_{n+1} - p_n\|_M^2 - \|x_n - p_n\|_M^2 - \|x_{n+1} - x^\star\|_M^2\Big) \\
&\qquad + \langle Cy_n - Cx^\star, y_n - p_n\rangle - \tfrac{1}{\beta}\|Cy_n - Cx^\star\|_{M^{-1}}^2 \\
&\leq \tfrac{1}{2\gamma_n\lambda_n}\Big(\|x_n - x^\star\|_M^2 + \|x_{n+1} - p_n\|_M^2 - \|x_n - p_n\|_M^2 - \|x_{n+1} - x^\star\|_M^2\Big) \\
&\qquad + \tfrac{\beta}{4}\|y_n - p_n\|_M^2
\end{aligned}$$

where we use the identity $2\langle a-b, c-d\rangle_M = \|a-d\|_M^2 + \|b-c\|_M^2 - \|a-c\|_M^2 - \|b-d\|_M^2$ for all $a,b,c,d \in \mathscr{H}$ and Young's inequality. Multiplying both sides of the last inequality by $2\gamma_n\lambda_n$ and reordering the terms yield

$$\begin{aligned}
&\|x_{n+1} - x^\star\|_M^2 - \|x_n - x^\star\|_M^2 \\
&\quad \leq \|x_{n+1} - p_n\|_M^2 - \|x_n - p_n\|_M^2 + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2 \\
&\quad = \|x_n - p_n + \lambda_n(p_n - z_n)\|_M^2 - \|x_n - p_n\|_M^2 + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2 \\
&\quad = \lambda_n^2\|p_n - z_n\|_M^2 + 2\lambda_n\langle x_n - p_n, p_n - z_n\rangle_M + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2 \\
&\quad = -\lambda_n(2-\lambda_n)\|p_n - z_n\|_M^2 + 2\lambda_n\langle p_n - z_n, x_n - z_n\rangle_M \\
&\qquad\quad + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2,
\end{aligned} \tag{I.20}$$

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

$$\begin{aligned}
\ell_n^2 &= \Big(\lambda_n(2-\lambda_n) - \tfrac{\lambda_n\gamma_n\beta}{2}\Big)\Big\|p_n - x_n + \tfrac{\lambda_n\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n + \tfrac{2(1-\lambda_n)}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\|_M^2 \qquad \text{(I.21)}\\
&= \lambda_n(2-\lambda_n)\|p_n - z_n\|_M^2 + \lambda_n(2-\lambda_n)\Big\|\tfrac{\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n - \tfrac{2-\gamma_n\beta}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\|_M^2\\
&\quad + 2\lambda_n(2-\lambda_n)\Big\langle p_n - z_n, \tfrac{\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n - \tfrac{2-\gamma_n\beta}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\rangle_M\\
&\quad - \tfrac{\lambda_n\gamma_n\beta}{2}\|p_n - y_n\|_M^2 - \tfrac{\lambda_n\gamma_n\beta}{2}\Big\|\tfrac{2}{2-\lambda_n\gamma_n\beta}u_n + \tfrac{2(1-\lambda_n)}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\|_M^2\\
&\quad - \lambda_n\gamma_n\beta\Big\langle p_n - y_n, \tfrac{2}{2-\lambda_n\gamma_n\beta}u_n + \tfrac{2(1-\lambda_n)}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\rangle_M.
\end{aligned}$$

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

$$\begin{aligned}
&2\lambda_n\Big\langle p_n - z_n, x_n - z_n + (2-\lambda_n)\Big(\tfrac{\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n - \tfrac{2-\gamma_n\beta}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big)\Big\rangle_M\\
&\quad = 2\lambda_n\Big\langle p_n - z_n, \Big(\tfrac{\gamma_n\beta(2-\lambda_n)}{2-\lambda_n\gamma_n\beta} - \tfrac{(1-\lambda_n)\gamma_n\beta}{2-\lambda_n\gamma_n\beta}\Big)u_n - \Big(1 + \tfrac{(2-\lambda_n)(2-\gamma_n\beta)}{\gamma_n\beta-2(2-\lambda_n)}\Big)v_n\Big\rangle_M\\
&\quad = 2\lambda_n\Big\langle p_n - z_n, \tfrac{\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n + \tfrac{(1-\lambda_n)\gamma_n\beta}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\rangle_M
\end{aligned}$$

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

$$\begin{aligned}
&\|x_{n+1} - x^\star\|_M^2 - \|x_n - x^\star\|_M^2 + \ell_n^2\\
&\qquad \le 2\lambda_n\gamma_n\beta\Big\langle y_n - z_n, \tfrac{1}{2-\lambda_n\gamma_n\beta}u_n + \tfrac{(1-\lambda_n)}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\rangle_M\\
&\qquad\quad + \lambda_n(2-\lambda_n)\Big\|\tfrac{\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n - \tfrac{2-\gamma_n\beta}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\|_M^2\\
&\qquad\quad - 2\lambda_n\gamma_n\beta\Big\|\tfrac{1}{2-\lambda_n\gamma_n\beta}u_n + \tfrac{(1-\lambda_n)}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\|_M^2.
\end{aligned} \qquad \text{(I.22)}$$

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

$$\begin{aligned}
&\|x_{n+1} - x^\star\|_M^2 - \|x_n - x^\star\|_M^2 + \ell_n^2\\
&\qquad \le 2\lambda_n\gamma_n\beta\Big\langle \tfrac{1-\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n - \tfrac{\gamma_n\beta-3+\lambda_n}{\gamma_n\beta-2(2-\lambda_n)}v_n, \tfrac{1}{2-\lambda_n\gamma_n\beta}u_n + \tfrac{(1-\lambda_n)}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\rangle_M\\
&\qquad\quad + \lambda_n(2-\lambda_n)\Big\|\tfrac{\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n - \tfrac{2-\gamma_n\beta}{\gamma_n\beta-2(2-\lambda_n)}v_n\Big\|_M^2.
\end{aligned}$$

In order to verify (I.14), it suffices to check the coefficients of $\|u_n\|_M^2$, $\|v_n\|_M^2$, and $\langle u_n, v_n\rangle_M$ on the right-hand side. This results in

$$
\begin{aligned}
&\|x_{n+1}-x^\star\|_M^2-\|x_n-x^\star\|_M^2+\ell_n^2\\
&\quad\le \tfrac{2\lambda_n\gamma_n\beta(1-\gamma_n\beta)+\lambda_n\gamma_n^2\beta^2(2-\lambda_n)}{(2-\lambda_n\gamma_n\beta)^2}\|u_n\|_M^2\\
&\qquad+\tfrac{-2\lambda_n\gamma_n\beta(\gamma_n\beta-3+\lambda_n)(1-\lambda_n)+\lambda_n(2-\lambda_n)(2-\gamma_n\beta)^2}{(\gamma_n\beta-2(2-\lambda_n))^2}\|v_n\|_M^2\\
&\qquad+\tfrac{2\lambda_n\gamma_n\beta(1-\gamma_n\beta)(1-\lambda_n)-2\lambda_n\gamma_n\beta(\gamma_n\beta-3+\lambda_n)-2\lambda_n\gamma_n\beta(2-\lambda_n)(2-\gamma_n\beta)}{(2-\lambda_n\gamma_n\beta)(\gamma_n\beta-2(2-\lambda_n))}\langle u_n,v_n\rangle_M\\
&\quad=\tfrac{\lambda_n\gamma_n\beta}{2-\lambda_n\gamma_n\beta}\|u_n\|_M^2+\tfrac{\lambda_n(-2+\lambda_n\gamma_n\beta)}{(\gamma_n\beta-2(2-\lambda_n))}\|v_n\|_M^2,
\end{aligned}
$$

showing (I.14). Finally, (I.15) follows from inserting (I.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 Assumption 1 and Assumption 2 hold. Let the sequences $(x_n)_{n\in\mathbb{N}}$, $(u_n)_{n\in\mathbb{N}}$, $(v_n)_{n\in\mathbb{N}}$, and $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ be generated by Algorithm 1. Then, the following hold:

(i) The sequence $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ 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 $x^\star\in\operatorname{zer}(A+C)$, the sequence $(\|x_n-x^\star\|_M)_{n\in\mathbb{N}}$ converges.
(iii) The sequence $(x_n)_{n\in\mathbb{N}}$ converges weakly to a point in $\operatorname{zer}(A+C)$. □

Proof. We start by proving Theorem 1 (i) via a telescoping argument for (I.15). To this end, let $N\in\mathbb{N}$. We sum (I.15) for $n=1,2,\ldots,N$ to obtain

$$
\|x_{N+1}-x^\star\|_M^2+\ell_N^2+\sum_{n=1}^{N-1}(1-\zeta_n)\ell_n^2\le\|x_1-x^\star\|_M^2+\zeta_0\ell_0^2.
$$

Then, rearranging the terms gives

$$
\begin{aligned}
\sum_{n=1}^{N}(1-\zeta_n)\ell_n^2&\le\|x_1-x^\star\|_M^2-\|x_{N+1}-x^\star\|_M^2-\zeta_N\ell_N^2\\
&\le\|x_1-x^\star\|_M^2+\zeta_0\ell_0^2.
\end{aligned}
$$

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 $\zeta_n\le 1-\varepsilon$ from Assumption 2, implies that

$$
\ell_n^2\to 0 \tag{I.23}
$$

as $n \to \infty$. Then, (I.6) implies that $u_n \to 0$ and $v_n \to 0$ as $n \to \infty$. This proves Theorem 1 (i).

The proof of Theorem 1 (ii) follows from the property that (I.15) defines a Lyapunov function: since $\zeta_n \leq 1$, we get from (I.15) that

$$\|x_{n+1} - x^\star\|_M^2 + \ell_n^2 \leq \|x_n - x^\star\|_M^2 + \ell_{n-1}^2,$$

i.e., the sequence $\left(\|x_n - x^\star\|_M^2 + \ell_{n-1}^2\right)_{n\in\mathbb{N}}$ is nonincreasing. As it is also nonnegative, it is convergent, say $\|x_n - x^\star\|_M^2 + \ell_{n-1}^2 \to \ell_{x^\star} \geq 0$ as $n \to \infty$. Moreover, $\ell_n^2 \to 0$ by Theorem 1 (i) as $n \to \infty$, so $\|x_n - x^\star\|_M^2 \to \ell_{x^\star}$, proving Theorem 1 (ii).

For the proof of Theorem 1 (iii), recall that $(p_n, \Delta_n) \in \operatorname{gra}(A+C)$ for all $n \in \mathbb{N}$ by Lemma 1. Now, by (I.8), we have $\frac{\lambda_n(4-2\lambda_n-\gamma_n\beta)}{2} \geq \varepsilon^2/2$ for all $n \in \mathbb{N}$. By this and $\ell_n \to 0$ as $n \to \infty$, we have that

$$p_n - x_n + \tfrac{\lambda_n\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n + \tfrac{2(\lambda_n-1)}{4-2\lambda_n-\gamma_n\beta}v_n \to 0.$$

Next, from $u_n \to 0$ and $v_n \to 0$, together with (I.8) and (I.9), we conclude that $p_n - x_n \to 0$ as $n \to \infty$. Then, by Lemma 1

$$\|\Delta_n\|_{M^{-1}} \leq \frac{1}{2\gamma_n}\left\|(2-\beta\gamma_n)(x_n-p_n) - \frac{\lambda_n\gamma_n\beta(2-\gamma_n\beta)}{2-\lambda_n\gamma_n\beta}u_n + 2v_n\right\|_M + \frac{\beta}{2}\|x_n - p_n + u_n\|_M,$$

hence, $\Delta_n \to 0$ as $n \to \infty$.

Now, from Theorem 1 (ii), we know that $\left(\|x_n - x^\star\|_M^2\right)_{n\in\mathbb{N}}$ 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 $x_{k_n} \rightharpoonup x_{\mathrm{wc}}^\star \in \mathscr{H}$ as $n \to \infty$. By the arguments above, we have $p_{k_n} \rightharpoonup x_{\mathrm{wc}}^\star$ and $\Delta_{k_n} \to 0$. Therefore, $(x_{\mathrm{wc}}^\star, 0) \in \operatorname{gra}(A+C)$ by the weak–strong closedness of $\operatorname{gra}(A+C)$ [Bauschke and Combettes, 2017, Proposition 20.38]. Then, Theorem 1 (iii) follows from [Bauschke and Combettes, 2017, 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 $M \in \mathscr{M}(\mathscr{H})$. A mapping $T : \mathscr{H} \to 2^{\mathscr{H}}$ is called M-metrically subregular at $\bar{x}$ for $\bar{y}$ if $(\bar{x}, \bar{y}) \in \operatorname{gra}(T)$ and there exists a $\kappa \geq 0$ along with neighborhoods $\mathscr{U}$ of $\bar{x}$ and $\mathscr{V}$ of $\bar{y}$ such that

$$\operatorname{dist}_M(x, T^{-1}(\bar{y})) \leq \kappa \operatorname{dist}_{M^{-1}}(\bar{y}, T(x) \cap \mathscr{V}) \tag{I.24}$$

for all $x \in \mathscr{U}$. □

This definition is equivalent to that in [Dontchev and Rockafellar, 2009], 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 (I.24) in terms of the distance of $T(x)$ from the input vector $\bar{y}$. For a detailed discussion on this subject, see [Dontchev and Rockafellar, 2009].

THEOREM 2 (LINEAR CONVERGENCE) Consider the monotone inclusion problem (I.5) and suppose that Assumption 1 and Assumption 2 hold, that $A+C$ is M-metrically subregular at all $x^\star \in \operatorname{zer}(A+C)$ for 0, and that either $\mathscr{H}$ is finite-dimensional or that in Definition 1 the neighborhood $\mathscr{U}$ at all $x^\star \in \operatorname{zer}(A+C)$ is the whole space $\mathscr{H}$. Then, there exists $0 \leq q < 1$ such that the following statements hold.

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

$$\operatorname{dist}_M^2(x_{n+1}, \operatorname{zer}(A+C)) + (1-\delta)\ell_n^2 \\ \leq q\big(\operatorname{dist}_M^2(x_n, \operatorname{zer}(A+C)) + (1-\delta)\ell_{n-1}^2\big)$$

for all $n \geq 1$;

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

Proof. We start by proving (i). Let $x^\star \in \operatorname{zer}(A+C)$ be the weak cluster point of the sequences $(x_n)_{n\in\mathbb{N}}$ and $(p_n)_{n\in\mathbb{N}}$ according to Theorem 1. From the metric subregularity of $A+C$ at x^* for 0, we get $\kappa \geq 0$ and neighborhoods $\mathscr{U}$ of x^* and $\mathscr{V}$ of 0 such that

$$\operatorname{dist}_M(x, \operatorname{zer}(A+C)) \leq \kappa \operatorname{dist}_{M^{-1}}(0, (A+C)(x) \cap \mathscr{V}) \tag{I.25}$$

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

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

$n \geq n_0$. Setting $x = p_n$ in (I.25) hence gives

$$\begin{aligned}
&\operatorname{dist}_M(p_n, \operatorname{zer}(A+C)) \\
&\quad \leq \kappa \operatorname{dist}_{M^{-1}}(0, (A+C)(p_n) \cap \mathscr{V}) \\
&\quad \leq \kappa \|\Delta_n\|_{M^{-1}} \\
&\quad \leq \frac{\kappa}{2\gamma_n} \left\| (2-\beta\gamma_n)(x_n - p_n) - \frac{\lambda_n \gamma_n \beta (2-\gamma_n\beta)}{2-\lambda_n\gamma_n\beta} u_n + 2v_n \right\|_M \\
&\qquad + \frac{\beta\kappa}{2} \|x_n - p_n + u_n\|_M
\end{aligned} \tag{I.26}$$

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

$$\|x_{n+1} - x^\star\|_M^2 + \ell_n^2 \leq \|x_n - x^\star\|_M^2 + \zeta_{n-1}\ell_{n-1}^2. \tag{I.27}$$

Now, set $x_n^\star := \Pi^M_{\operatorname{zer}(A+C)}(x_n)$. Then, from (I.27), we get

$$\begin{aligned}
\operatorname{dist}_M^2(x_{n+1}, \operatorname{zer}(A+C)) + \ell_n^2 &\leq \|x_{n+1} - x_n^\star\|_M^2 + \ell_n^2 \\
&\leq \|x_n - x_n^\star\|_M^2 + \zeta_{n-1}\ell_{n-1}^2 \\
&= \operatorname{dist}_M^2(x_n, \operatorname{zer}(A+C)) + \zeta_{n-1}\ell_{n-1}^2.
\end{aligned} \tag{I.28}$$

Next, we will estimate both sides of (I.26) in terms of $\operatorname{dist}_M^2(x_n, \operatorname{zer}(A+C))$, ℓ_n^2, and ℓ_{n-1}^2. Let $p_n^* := \Pi^M_{\operatorname{zer}(A+C)}(p_n)$. Then, since $\Pi_{\operatorname{zer}(A+C)}$ is the projection onto a convex set w.r.t. the M-induced metric, [Bauschke and Combettes, 2017, Theorem 3.16] yields

$$\begin{aligned}
&\operatorname{dist}_M^2(p_n, \operatorname{zer}(A+C)) \\
&\qquad \geq \|p_n - p_n^*\|_M^2 - 2\langle x_n^* - x_n, p_n^* - x_n^* \rangle_M \\
&\qquad = \|p_n - p_n^*\|_M^2 - 2\langle x_n^* - x_n, p_n^* - p_n \rangle_M - 2\langle x_n^* - x_n, p_n - x_n^* \rangle_M \\
&\qquad = \|p_n - p_n^* - x_n^* + x_n\|_M^2 - \|x_n^* - x_n\|_M^2 - 2\langle x_n^* - x_n, p_n - x_n^* \rangle_M \\
&\qquad \geq \|x_n^* - x_n\|_M^2 - 2\langle x_n^* - x_n, p_n - x_n \rangle_M \\
&\qquad \geq \tfrac{1}{2}\|x_n^* - x_n\|_M^2 - 2\|p_n - x_n\|_M^2,
\end{aligned}$$

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

$$\begin{aligned}
&\tfrac{1}{2}\operatorname{dist}_M^2(x_n, \operatorname{zer}(A+C)) \\
&\quad\leq \left(\frac{\kappa}{2\gamma_n} \left\| (2-\beta\gamma_n)(x_n-p_n) - \frac{\lambda_n\gamma_n\beta(2-\gamma_n\beta)}{2-\lambda_n\gamma_n\beta} u_n + 2v_n \right\|_M \right. \\
&\qquad \left. + \frac{\beta\kappa}{2}\|x_n - p_n + u_n\|_M \right)^2 + 2\|p_n - x_n\|_M^2 \qquad \text{(I.29)} \\
&\quad\leq \frac{\kappa^2}{2\gamma_n^2} \left\| (2-\beta\gamma_n)(x_n-p_n) - \frac{\lambda_n\gamma_n\beta(2-\gamma_n\beta)}{2-\lambda_n\gamma_n\beta} u_n + 2v_n \right\|_M^2 \\
&\qquad + \frac{\beta^2\kappa^2}{2}\|x_n - p_n + u_n\|_M^2 + 2\|p_n - x_n\|_M^2,
\end{aligned}$$

where we used Young's inequality in the last step. It remains to estimate the right-hand side of (I.29) in terms of ℓ_n^2 and ℓ_{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 $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ be generated by Algorithm 1 under Assumption 2, and let $(\mathfrak{a}_n)_{n\in\mathbb{N}}$, $(\mathfrak{b}_n)_{n\in\mathbb{N}}$, and $(\mathfrak{c}_n)_{n\in\mathbb{N}}$ be bounded sequences of real numbers. Then there exist $c_1, c_2 > 0$ (which do not depend on n) such that

$$\|\mathfrak{a}_n(p_n - x_n) + \mathfrak{b}_n u_n + \mathfrak{c}_n v_n\|_M^2 \leq c_1\ell_n^2 + c_2\ell_{n-1}^2. \qquad \square$$

Proof. The assertion is proven by repeatedly applying Young's inequality and subsequently using the norm condition (I.6):

$$\begin{aligned}
&\|\mathfrak{a}_n(p_n - x_n) + \mathfrak{b}_n u_n + \mathfrak{c}_n v_n\|_M^2 \\
&\quad= \left\| \mathfrak{a}_n \left(p_n - x_n + \frac{\lambda_n\gamma_n\beta}{2-\lambda_n\gamma_n\beta} u_n - \frac{2(1-\lambda_n)}{4-2\lambda_n-\gamma_n\beta} v_n \right) \right. \\
&\qquad \left. + \left(\mathfrak{b}_n - \frac{\lambda_n\gamma_n\beta\mathfrak{a}_n}{2-\lambda_n\gamma_n\beta} \right) u_n + \left(\mathfrak{c}_n + \frac{2\mathfrak{a}_n(1-\lambda_n)}{4-2\lambda_n-\gamma_n\beta} \right) v_n \right\|_M^2 \\
&\quad\leq 2\mathfrak{a}_n^2 \left\| p_n - x_n + \frac{\lambda_n\gamma_n\beta}{2-\lambda_n\gamma_n\beta} u_n - \frac{2(1-\lambda_n)}{4-2\lambda_n-\gamma_n\beta} v_n \right\|_M^2 \\
&\qquad + 2\left\| \left(\mathfrak{b}_n - \frac{\lambda_n\gamma_n\beta\mathfrak{a}_n}{2-\lambda_n\gamma_n\beta} \right) u_n + \left(\mathfrak{c}_n + \frac{2\mathfrak{a}_n(1-\lambda_n)}{4-2\lambda_n-\gamma_n\beta} \right) v_n \right\|_M^2 \\
&\quad\leq \frac{4\mathfrak{a}_n^2}{\lambda_n(4-2\lambda_n-\gamma_n\beta)} \ell_n^2 \\
&\qquad + 4\left(\mathfrak{b}_n - \frac{\lambda_n\gamma_n\beta\mathfrak{a}_n}{2-\lambda_n\gamma_n\beta} \right)^2 \|u_n\|_M^2 + 4\left(\mathfrak{c}_n + \frac{2\mathfrak{a}_n(1-\lambda_n)}{4-2\lambda_n-\gamma_n\beta} \right)^2 \|v_n\|_M^2
\end{aligned}$$

$$\leq \frac{4\mathfrak{a}_n^2}{\lambda_n(4-2\lambda_n-\gamma_n\beta)}\ell_n^2 + 4\mathfrak{d}_n\zeta_{n-1}\ell_{n-1}^2$$

with

$$\mathfrak{d}_n := \max\left\{ \frac{2-\lambda_n\gamma_n\beta}{\lambda_n\gamma_n\beta}\left(\mathfrak{b}_n - \frac{\lambda_n\gamma_n\beta\mathfrak{a}_n}{2-\lambda_n\gamma_n\beta}\right)^2, \right.$$
$$\left. \frac{4-2\lambda_n-\gamma_n\beta}{\lambda_n(2-\lambda_n\gamma_n\beta)}\left(\mathfrak{c}_n + \frac{2\mathfrak{a}_n(1-\lambda_n)}{4-2\lambda_n-\gamma_n\beta}\right)^2 \right\}.$$

It is straightforward to show, by using Assumption 2, that $\frac{4\mathfrak{a}_n^2}{\lambda_n(4-2\lambda_n-\gamma_n\beta)}$ and $4\mathfrak{d}_n\zeta_{n-1}$ 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 (I.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

$$\tfrac{1}{2}\operatorname{dist}_M^2(x_n, \operatorname{zer}(A+C)) \leq c_1\ell_n^2 + c_2\ell_{n-1}^2.$$

Multiplying this with any $\delta' > 0$ and adding (I.28) gives

$$\begin{aligned}\operatorname{dist}_M^2(x_{n+1}, \operatorname{zer}(A+C)) &+ \big(1-\delta' c_1\big)\ell_n^2 \\ &\leq \left(1-\tfrac{\delta'}{2}\right)\operatorname{dist}_M^2(x_n, \operatorname{zer}(A+C)) + \big(\zeta_{n-1} + \delta' c_2\big)\ell_{n-1}^2 \\ &\leq \left(1-\tfrac{\delta'}{2}\right)\operatorname{dist}_M^2(x_n, \operatorname{zer}(A+C)) + \big(1-\varepsilon + \delta' c_2\big)\ell_{n-1}^2.\end{aligned}$$

Choosing (for example) δ' as the smaller of the two solutions to

$$\left(1-\tfrac{\delta'}{2}\right)\big(1-\delta' c_1\big) = \big(1-\varepsilon+\delta' c_2\big),$$

namely

$$\delta' = \frac{1+2c_1+2c_2}{2c_1} - \sqrt{\frac{(1+2c_1+2c_2)^2}{4c_1^2} - \frac{2\varepsilon}{c_1}}, \tag{I.30}$$

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

$$\|x_{n+1}-x_n\|_M^2 = \lambda_n^2\left\|p_n - x_n - \frac{(1-\lambda_n)\gamma_n\beta}{2-\lambda_n\gamma_n\beta}u_n - v_n\right\|_M^2 \leq c_1'\ell_n + c_2'\ell_{n-1} \tag{I.31}$$

for all $n \geq 1$. From Item (i), we get $\delta > 0$ and $0 \leq q < 1$ such that

$$\operatorname{dist}_M^2(x_{n+1}, \operatorname{zer}(A+C)) + (1-\delta)\ell_n^2 \leq q\big(\operatorname{dist}_M^2(x_n, \operatorname{zer}(A+C)) + (1-\delta)\ell_{n-1}^2\big)$$

for all $n \geq 1$. Repeatedly applying this relation gives

$$\begin{aligned}\ell_n^2 &\leq \frac{1}{1-\delta}\big(\operatorname{dist}_M^2(x_{n+1}, \operatorname{zer}(A+C)) + (1-\delta)\ell_n^2\big) \\ &\leq \frac{q^n}{1-\delta}\big(\operatorname{dist}_M^2(x_1, \operatorname{zer}(A+C)) + (1-\delta)\ell_0^2\big).\end{aligned}$$

Inserting into (I.31) and taking square roots on both sides yields

$$\|x_{n+1} - x_n\|_M \leq q^{n/2}\sqrt{\frac{c_1' + c_2'/q}{1-\delta}\big(\operatorname{dist}_M^2(x_1, \operatorname{zer}(A+C)) + (1-\delta)\ell_0^2\big)}.$$

Let us choose $m > n \geq 1$ and apply the triangle inequality,

$$\begin{aligned}\|x_m - x_n\|_M &\leq \sum_{k=n}^{m-1} \|x_{k+1} - x_k\|_M \\ &\leq \sum_{k=n}^{m-1} q^{k/2}\sqrt{\frac{c_1' + c_2'/q}{1-\delta}\big(\operatorname{dist}_M^2(x_1, \operatorname{zer}(A+C)) + (1-\delta)\ell_0^2\big)} \\ &\leq \sum_{k=n}^{\infty} q^{k/2}\sqrt{\frac{c_1' + c_2'/q}{1-\delta}\big(\operatorname{dist}_M^2(x_1, \operatorname{zer}(A+C)) + (1-\delta)\ell_0^2\big)} \\ &= q^{n/2}\frac{1}{1-\sqrt{q}}\sqrt{\frac{c_1' + c_2'/q}{1-\delta}\big(\operatorname{dist}_M^2(x_1, \operatorname{zer}(A+C)) + (1-\delta)\ell_0^2\big)}\end{aligned} \tag{I.32}$$

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 (I.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 $x \in \mathscr{H}$ such that

$$0 \in Ax + L^*B(Lx) + Cx \tag{I.33}$$

under the following assumption.

ASSUMPTION 3 We assume that

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

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

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

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

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

Problem (I.33) can be translated to a primal–dual problem [He and Yuan, 2012]: $x \in \mathscr{H}$ is a solution to (I.33) if and only if there exists $\mu \in B(Lx)$ (the *dual variable*) such that

$$\begin{aligned} 0 &\in Ax + L^*\mu + Cx, \\ 0 &\in -Lx + B^{-1}\mu. \end{aligned} \tag{I.34}$$

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

$$0 \in \mathscr{A}w + \mathscr{C}w, \tag{I.35}$$

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

$$\mathscr{A} = \begin{bmatrix} A & L^* \\ -L & B^{-1} \end{bmatrix}, \qquad \mathscr{C} = \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix}. \tag{I.36}$$

The operator $\mathscr{A}$ is maximally monotone by [Bauschke and Combettes, 2017, Proposition 26.32] and $\mathscr{C}$ is $1/\beta$-cocoercive with respect to the metric $\|\cdot\|_M$, with

$$M = \begin{bmatrix} I & -\tau L^* \\ -\tau L & \tau\sigma^{-1} I \end{bmatrix} \tag{I.37}$$

where $\sigma, \tau > 0$ such that $\sigma\tau\|L\|^2 < 1$.

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

Algorithm 2

Input: $(x_0,\mu_0)\in\mathscr{H}\times\mathscr{K}$, 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 $\sigma\tau\|L\|^2<1$.
set: $u_{x,0}=v_{x,0}=0$, $v_{\mu,0}=0$.
for $n=0,1,2,\ldots$ **do**
  $\tilde{x}_n=x_n+u_{x,n}$
  $\begin{bmatrix}\hat{x}_n\\ \hat{\mu}_n\end{bmatrix}=\begin{bmatrix}x_n\\ \mu_n\end{bmatrix}+\begin{bmatrix}\frac{(1-\lambda_n)\tau\beta}{2-\lambda_n\tau\beta}u_{x,n}+v_{x,n}\\ v_{\mu,n}\end{bmatrix}$
  $\begin{bmatrix}p_{x,n}\\ p_{\mu,n}\end{bmatrix}=\begin{bmatrix}J_{\tau A}\left(\hat{x}_n-\tau L^*\hat{\mu}_n-\tau C\tilde{x}_n\right)\\ J_{\sigma B^{-1}}\left(\hat{\mu}_n+\sigma L(2p_{x,n}-\hat{x}_n)\right)\end{bmatrix}$
  $\begin{bmatrix}x_{n+1}\\ \mu_{n+1}\end{bmatrix}=\begin{bmatrix}x_n\\ \mu_n\end{bmatrix}+\lambda_n\left(\begin{bmatrix}p_{x,n}\\ p_{\mu,n}\end{bmatrix}-\begin{bmatrix}\hat{x}_n\\ \hat{\mu}_n\end{bmatrix}\right)$
  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

$$\begin{aligned}\tfrac{\lambda_{n+1}\tau\beta}{2-\lambda_{n+1}\tau\beta}\left\|u_{x,n+1}\right\|^2+\tfrac{\lambda_{n+1}(2-\lambda_{n+1}\tau\beta)}{4-2\lambda_{n+1}-\tau\beta}\left\|\begin{bmatrix}v_{x,n+1}\\ v_{\mu,n+1}\end{bmatrix}\right\|_M^2\\ \leq\zeta_n\tfrac{\lambda_n(4-2\lambda_n-\tau\beta)}{2}\left\|\begin{bmatrix}p_{x,n}\\ p_{\mu,n}\end{bmatrix}-\begin{bmatrix}x_n\\ \mu_n\end{bmatrix}+\tfrac{\lambda_n\tau\beta}{2-\lambda_n\tau\beta}\begin{bmatrix}u_{x,n}\\ 0\end{bmatrix}\right.\\ \left.-\tfrac{2(1-\lambda_n)}{4-2\lambda_n-\tau\beta}\begin{bmatrix}v_{x,n}\\ v_{\mu,n}\end{bmatrix}\right\|_M^2\end{aligned}\tag{I.38}$$

end for

COROLLARY 1 Consider monotone inclusions (I.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 $\operatorname{zer}(A+L^*BL+C)$. □

Proof. In Algorithm 1, replace A by $\mathscr{A}$ and C by $\mathscr{C}$ as devised by (I.36), and substitute (x_n,μ_n) in place of x_n, and also set $p_n=(p_{x,n},p_{\mu,n})$, $y_n=(\tilde{x}_n,\mu_n)$, $z_n=(\hat{x}_n,\hat{\mu}_n)$, $u_n=(u_{x,n},0)$, $v_n=(v_{x,n},v_{\mu,n})$, M as is in (I.37), and $\gamma_n=\tau$ ($n\in\mathbb{N}$). These changes, along with the update formula

$$\begin{aligned}p_n=(p_{x,n},p_{\mu,n})&=(M+\tau\mathscr{A})^{-1}(Mz_n-\tau\mathscr{C}y_n)\\&=\begin{bmatrix}I+\tau A & 0\\ -2\tau L & \tau\sigma^{-1}I+\tau B^{-1}\end{bmatrix}^{-1}\begin{bmatrix}\hat{x}_n-\tau L^*\hat{\mu}_n-\tau C\tilde{x}_n\\ -\tau L\hat{x}_n+\tau\sigma^{-1}\hat{\mu}_n\end{bmatrix}\\&=\begin{bmatrix}(I+\tau A)^{-1}(\hat{x}_n-\tau L^*\hat{\mu}_n-\tau C\tilde{x}_n)\\ (I+\sigma B^{-1})^{-1}(\hat{\mu}_n+\sigma L(2p_{x,n}-\hat{x}_n))\end{bmatrix}\end{aligned}$$

$$= \begin{bmatrix} J_{\tau A}\left(\hat{x}_n - \tau L^* \hat{\mu}_n - \tau C \tilde{x}_n\right) \\ J_{\sigma B^{-1}}\left(\hat{\mu}_n + \sigma L(2p_{x,n} - \hat{x}_n)\right) \end{bmatrix},$$

result in Algorithm 2. Therefore, Algorithm 2 is a special instance of Algorithm 1; and the corollary is an immediate consequence of Theorem 1. □

REMARK 3 In Algorithm 2, it might be expected that we get $\tilde{\mu}_n = \mu_n + u_{\mu,n}$, which is the dual counterpart of $\tilde{x}_n = x_n + u_{x,n}$, but we do not. That is because the corresponding part of $\tilde{\mu}_n$ of the operator $\mathscr{C}$ in (I.36), i.e. its second column, is zero, and thus, there is no need to define the dual counterpart of $\tilde{x}_n$. □

REMARK 4 In Algorithm 2, letting all deviations $u_{x,n}$, $v_{x,n}$, $v_{\mu,n}$ ($n \in \mathbb{N}$) be zero and $\lambda_n = 1$ give

$$\begin{aligned} x_{n+1} &= J_{\tau A}\left(x_n - \tau L^* \mu_n - \tau C x_n\right), \\ \mu_{n+1} &= J_{\sigma B^{-1}}\left(\mu_n + \sigma L(2x_{n+1} - x_n)\right). \end{aligned}$$

This is the Condat–Vũ algorithm in its basic form [Condat, 2013; Vũ, 2013], which, with $C = 0$, reduces to the basic form of the Chambolle–Pock primal–dual method [Chambolle and Pock, 2011]. □

REMARK 5 By letting $C = 0$, $\beta = 0$, and $u_{x,n} = 0$ for all $n \in \mathbb{N}$ in Algorithm 2, we arrive at a Chambolle–Pock method with deviations and the condition (I.38) reduces to

$$\left\| \begin{bmatrix} v_{x,n+1} \\ v_{\mu,n+1} \end{bmatrix} \right\|_M^2 \leq \zeta_n \tfrac{(2-\lambda_{n+1})(2-\lambda_n)\lambda_n}{\lambda_{n+1}} \left\| \begin{bmatrix} p_{x,n} \\ p_{\mu,n} \end{bmatrix} - \begin{bmatrix} x_n \\ \mu_n \end{bmatrix} - \tfrac{1-\lambda_n}{2-\lambda_n} \begin{bmatrix} v_{x,n} \\ v_{\mu,n} \end{bmatrix} \right\|_M^2. \qquad \square$$

5.2 Krasnosel'skiĭ–Mann iteration with deviations

Consider the fixed-point problem

$$x = Tx, \tag{I.39}$$

where $T : \mathscr{H} \to \mathscr{H}$ is a nonexpansive operator. Then, by [Bauschke and Combettes, 2017, Remark 4.34, Corollary 23.9], there is a maximally monotone operator $A : \mathscr{H} \to 2^{\mathscr{H}}$ for which $J_{\gamma A} = \frac{1}{2}\mathrm{Id} + \frac{1}{2}T$, with $\gamma > 0$. This correspondence suggests that Algorithm 1 can be used to solve (I.39). Letting $C = 0$, $\beta = 0$, $M = \mathrm{Id}$, and $u_n = 0$ for all $n \in \mathbb{N}$ in Algorithm 1, results in Algorithm 3, that can be used to solve problem (I.39). Weak convergence of Algorithm 3 is shown in Corollary 2.

COROLLARY 2 Consider the fixed-point problem (I.39); suppose that its solution set is nonempty and let $J_{\gamma A} = \frac{1}{2}\mathrm{Id} + \frac{1}{2}T$. 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

Input: $x_0 \in \mathscr{H}$, 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.
set: $v_0 = 0$
for $n = 0, 1, \ldots$ **do**
  $z_n = x_n + v_n$
  $p_n = \frac{1}{2}(\mathrm{Id}+T)(x_n + v_n)$
  $x_{n+1} = (1-\lambda_n)x_n + \lambda_n(p_n - v_n)$
  choose v_{n+1} such that

$$\|v_{n+1}\|^2 \leq \zeta_n \tfrac{\lambda_n(2-\lambda_n)(2-\lambda_{n+1})}{\lambda_{n+1}} \left\| p_n - x_n + \tfrac{\lambda_n - 1}{2-\lambda_n} v_n \right\|^2 \tag{I.40}$$

end for

Setting $v_n = 0$ for all $n \in \mathbb{N}$ in Algorithm 3 results in

$$x_{n+1} = (1 - \tfrac{\lambda_n}{2})x_n + \tfrac{\lambda_n}{2} T(x_n),$$

which is the standard Krasnosel'skiĭ–Mann iteration [Bauschke and Combettes, 2017, 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 (I.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 [Alvarez, 2000; Alvarez and Attouch, 2001; Attouch and Cabot, 2019; Cholamjiak et al., 2018; Lorenz and Pock, 2015] 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 (I.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 computationally expensive. However, by scrutinizing Algorithm 4, it is observed that some of the

Algorithm 4

1: **Input:** $(x_0, \mu_0) \in \mathscr{H} \times \mathscr{K}$, 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: $$\begin{bmatrix} p_{x,n} \\ p_{\mu,n} \end{bmatrix} = \begin{bmatrix} J_{\tau A}\left(\hat{x}_n - \tau L^* \hat{\mu}_n\right) \\ J_{\sigma B^{-1}}\left(\hat{\mu}_n + \sigma L(2p_{x,n} - \hat{x}_n)\right) \end{bmatrix}$$
6: $$\begin{bmatrix} x_{n+1} \\ \mu_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ \mu_n \end{bmatrix} + \lambda_n \left(\begin{bmatrix} p_{x,n} \\ p_{\mu,n} \end{bmatrix} - \begin{bmatrix} \hat{x}_n \\ \hat{\mu}_n \end{bmatrix} \right)$$
7: choose a_{n+1} such that

$$a_{n+1}^2 \left\| \begin{bmatrix} x_{n+1} - x_n \\ \mu_{n+1} - \mu_n \end{bmatrix} \right\|_M^2 \leq \zeta_n \frac{\lambda_n(2-\lambda_n)(2-\lambda_{n+1})}{\lambda_{n+1}} \left\| \begin{bmatrix} p_{x,n} - x_n \\ p_{\mu,n} - \mu_n \end{bmatrix} + \frac{\lambda_n - 1}{2 - \lambda_n} a_n \begin{bmatrix} x_n - x_{n-1} \\ \mu_n - \mu_{n-1} \end{bmatrix} \right\|_M^2 \tag{I.41}$$

8: **end for**

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 (I.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

$$\begin{aligned} L\hat{x}_n &= Lx_n + b_n(Lx_n - Lx_{n-1}), \\ L^*\hat{\mu}_n &= L^*\mu_n + b_n(L^*\mu_n - L^*\mu_{n-1}), \\ Lx_{n+1} &= Lx_n + \lambda_n(Lp_{x,n} - L\hat{x}_n), \\ L^*\mu_{n+1} &= L^*\mu_n + \lambda_n(L^*p_{\mu,n} - L^*\hat{\mu}_n), \end{aligned} \tag{I.42}$$

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 $Lp_{x,n}$ 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, $Lp_{x,n}$ and $L^* p_{\mu,n}$. The rest of the L and L^* evaluations can be done efficiently by exploiting previously computed quantities and (I.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	Lx_0, $L^*\mu_0$, $Lp_{x,0}$, $L^* p_{\mu,0}$	Lx_1, $L^*\mu_1$
1	$Lp_{x,1}$, $L^* p_{\mu,1}$	$L\hat{x}_2$, Lx_2, $L^*\hat{\mu}_2$, $L^*\mu_2$
2	$Lp_{x,2}$, $L^* p_{\mu,2}$	$L\hat{x}_3$, Lx_3, $L^*\hat{\mu}_3$, $L^*\mu_3$

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 (I.42).

7. Numerical experiments

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

$$\underset{x}{\text{minimize}}\ f(Lx) + g(x), \tag{I.43}$$

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

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

where $\mathbf{0} = (0, \ldots, 0)^T$, $\mathbf{1} = (1, \ldots, 1)^T$, $x = (\omega, b)$ 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 $x^\star$ is a solution to (I.43) if and only if it satisfies

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

This holds, since f and g are proper, closed, and convex functions with full domains, and thus, ∂f and ∂g are maximally monotone and L is a linear operator [Bauschke and Combettes, 2017, Proposition 16.42]. This monotone inclusion problem is an

instance of (I.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) [Chambolle and Pock, 2011], and to the inertial primal–dual algorithm of Lorenz–Pock (LP) [Lorenz and Pock, 2015]. In all experiments, we set the primal and the dual step-sizes to $\tau = \sigma = 0.99/\|L\|$, the regularization parameter of problem (I.43) to $\xi = 0.1$, and ζ_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 [Chang and Lin, 2011] which has 145 samples and 5 features. The solution $(x^\star, \mu^\star)$ is found by running the standard Chambolle–Pock algorithm until the residual gets smaller than 10^{-15}.

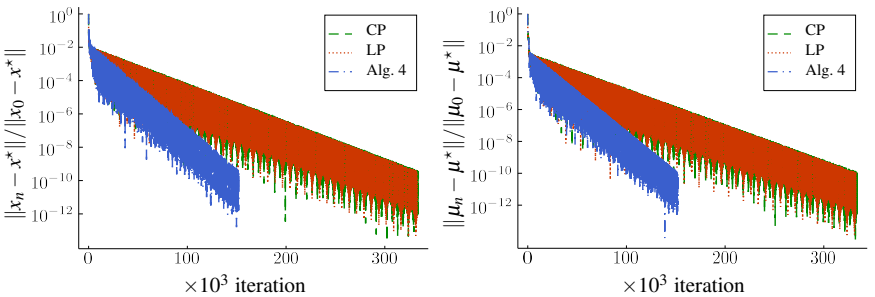

Figure 1: Distance to the solution vs. iteration number for the l_1-norm regularized SVM (I.43) with $\xi = 0.1$, on the *liver disorders* data-set [Chang and Lin, 2011] with 145 samples and 5 features. Solved using Chambolle–Pock primal–dual algorithm (CP), Lorenz–Pock inertial primal–dual method (LP), and Algorithm 4 with $\lambda = 1.0$. The primal and dual step-sizes are set to $\tau = \sigma = 0.99/\|L\|$ for all algorithms.

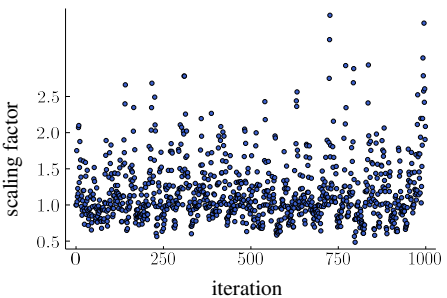

Figure 2: Scaling factor a_n of Algorithm 4 in the experiment shown in Fig. 1 vs. iteration number for the first 1000 iterations.

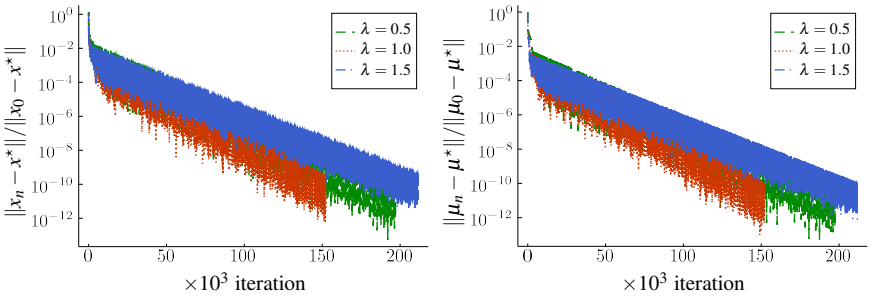

Figure 3: Distance to the solution vs. iteration number for the l_1-norm regularized SVM (I.43) with $\xi = 0.1$, on the *liver disorders* data-set [Chang and Lin, 2011] with 145 samples and 5 features. Solved using Algorithm 4 for some values of λ with $\tau = \sigma = 0.99/\|L\|$.

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

$$\mathscr{A} = \begin{bmatrix} \partial g & L^* \\ -L & \partial f^* \end{bmatrix}$$

is metrically subregular at any point in the solution set of the problem for 0, see [Latafat et al., 2019, 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 λ 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

Alvarez, F. (2000). "On the minimizing property of a second order dissipative system in Hilbert spaces". *SIAM Journal on Control and Optimization* **38**:4, pp. 1102–1119. DOI: 10.1137/s0363012998335802.

Alvarez, F. and H. Attouch (2001). "An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping". *Set-Valued Analysis* **9**:1/2, pp. 3–11. DOI: 10.1023/a:1011253113155.

Attouch, H. and A. Cabot (2019). "Convergence of a relaxed inertial forward–backward algorithm for structured monotone inclusions". *Applied Mathematics & Optimization* **80**:3, pp. 547–598. DOI: 10.1007/s00245-019-09584-z.

Banert, S., J. Rudzusika, O. Oktem, and J. Adler (2021). *Accelerated forward–backward optimization using deep learning*. arXiv: 2105 . 05210v1 [math.OC].

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

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

Chambolle, A. and T. Pock (2011). "A first-order primal–dual algorithm for convex problems with applications to imaging". *Journal of Mathematical Imaging and Vision* **40**:1, pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

Chang, C.-C. and C.-J. Lin (2011). "LIBSVM: a library for support vector machines". *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**:3, pp. 1–27. DOI: 10.1145/1961189.1961199.

Cholamjiak, W., P. Cholamjiak, and S. Suantai (2018). "An inertial forward–backward splitting method for solving inclusion problems in Hilbert spaces". *Journal of Fixed Point Theory and Applications* **20**:1. DOI: 10.1007/s11784-018-0526-5.

Chouzenoux, E., J.-C. Pesquet, and A. Repetti (2013). "Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function". *Journal of Optimization Theory and Applications* **162**:1, pp. 107–132. DOI: 10.1007/s10957-013-0465-7.

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

Combettes, P. L. and B. C. Vũ (2012). "Variable metric forward–backward splitting with applications to monotone inclusions in duality". *Optimization* **63**:9, pp. 1289–1318. DOI: 10.1080/02331934.2012.733883.

Condat, L. (2013). "A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms". *Journal of Optimization Theory and Applications* **158**:2, pp. 460–479. DOI: 10.1007/s10957-012-0245-9.

Dontchev, A. L. and R. T. Rockafellar (2009). *Implicit functions and solution mappings. A view from variational analysis*. Springer Monographs in Mathematics. Springer. DOI: 10.1007/978-0-387-87821-8.

Giselsson, P. (2021). "Nonlinear forward–backward splitting with projection correction". *SIAM Journal on Optimization* **31**:3, pp. 2199–2226. DOI: 10.1137/20M1345062.

Giselsson, P. and S. Boyd (2015). "Metric selection in fast dual forward–backward splitting". *Automatica* **62**, pp. 1–10. DOI: 10.1016/j.automatica.2015.09.010.

Giselsson, P. and S. P. Boyd (2014a). "Diagonal scaling in douglas–rachford splitting and admm". In: *53rd IEEE Conference on Decision and Control*. IEEE, pp. 5033–5039. DOI: 10.1109/CDC.2014.7040175.

Giselsson, P. and S. P. Boyd (2014b). "Preconditioning in fast dual gradient methods". In: *53rd IEEE Conference on Decision and Control*. IEEE, pp. 5040–5045. DOI: 10.1109/CDC.2014.7040176.

Giselsson, P., M. Fält, and S. Boyd (2016). "Line search for averaged operator iteration". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 1015–1022. DOI: 10.1109/CDC.2016.7798401.

He, B. and X. Yuan (2012). "Convergence analysis of primal–dual algorithms for a saddle-point problem: from contraction perspective". *SIAM Journal on Imaging Sciences* **5**:1, pp. 119–149. DOI: 10.1137/100814494.

Latafat, P., N. Freris, and P. Patrinos (2019). "A new randomized block-coordinate primal–dual proximal algorithm for distributed optimization". *IEEE Transactions on Automatic Control* **64**:10, pp. 4050–4065. DOI: 10.1109/TAC.2019.2906924.

Lions, P. L. and B. Mercier (1979). "Splitting algorithms for the sum of two nonlinear operators". *SIAM Journal on Numerical Analysis* **16**:6, pp. 964–979. DOI: 10.1137/0716071.

Lorenz, D. A. and T. Pock (2015). "An inertial forward–backward algorithm for monotone inclusions". *Journal of Mathematical Imaging and Vision* **51**:2, pp. 311–325. DOI: 10.1007/s10851-014-0523-2.

Nocedal, J. and S. J. Wright (2006). *Numerical optimization*. 2nd ed. Springer Series in Operations Research and Financial Engineering. Springer. DOI: 10.1007/978-0-387-40065-5.

Passty, G. B. (1979). "Ergodic convergence to a zero of the sum of monotone operators in Hilbert space". *Journal of Mathematical Analysis and Applications* **72**:2, pp. 383–390. DOI: 10.1016/0022-247x(79)90234-8.

Pock, T. and A. Chambolle (2011). "Diagonal preconditioning for first order primal–dual algorithms in convex optimization". In: *2011 International Conference on Computer Vision*. IEEE, pp. 1762–1769. DOI: 10.1109/ICCV.2011.6126441.

Polyak, B. T. (1964). "Some methods of speeding up the convergence of iteration methods". *USSR Computational Mathematics and Mathematical Physics* **4**:5, pp. 1–17. DOI: 10.1016/0041-5553(64)90137-5.

Raguet, H., J. Fadili, and G. Peyré (2013). "A generalized forward–backward splitting". *SIAM Journal on Imaging Sciences* **6**:3, pp. 1199–1226. DOI: 10.1137/120872802.

Raguet, H. and L. Landrieu (2015). "Preconditioning of a generalized forward–backward splitting and application to optimization on graphs". *SIAM Journal on Imaging Sciences* **8**:4, pp. 2706–2739. DOI: 10.1137/15m1018253.

Sadeghi, H., S. Banert, and P. Giselsson (2022a). *Dwifob: a dynamically weighted inertial forward–backward algorithm for monotone inclusions*. arXiv: 2203.00028 [math.OC].

Sadeghi, H., S. Banert, and P. Giselsson (2022b). *Incorporating history and deviations in forward–backward splitting*. arXiv: 2208.05498 [math.OC].

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

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

Solodov, M. V. and B. F. Svaiter (2000). "An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions". *Mathematics of Operations Research* **25**:2, pp. 214–230. DOI: 10.1287/moor.25.2.214.12222.

Solodov, M. V. and B. F. Svaiter (2001). "A unified framework for some inexact proximal point algorithms". *Numerical Functional Analysis and Optimization* **22**:7–8, pp. 1013–1035. DOI: 10.1081/NFA-100108320.

Themelis, A. and P. Patrinos (2019). "Supermann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators". *IEEE Transactions on Automatic Control* **64**:12, pp. 4875–4890. DOI: 10.1109/TAC.2019.2906393.

Tseng, P. (2000). "A modified forward–backward splitting method for maximal monotone mappings". *SIAM Journal on Control and Optimization* **38**:2, pp. 431–446. DOI: 10.1137/S0363012998338806.

Villa, S., S. Salzo, L. Baldassarre, and A. Verri (2013). "Accelerated and inexact forward–backward algorithms". *SIAM Journal on Optimization* **23**:3, pp. 1607–1633. DOI: 10.1137/110844805.

Vũ, B. C. (2013). "A splitting algorithm for dual monotone inclusions involving cocoercive operators". *Advances in Computational Mathematics* **38**:3, pp. 667–681. DOI: 10.1007/s10444-011-9254-8.

Zhang, J., B. O'Donoghue, and S. Boyd (2020). "Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations". *SIAM Journal on Optimization* **30**:4, pp. 3170–3197. DOI: 10.1137/18M1232772.

Paper II

Incorporating History and Deviations in Forward–Backward Splitting

Hamed Sadeghi Sebastian Banert Pontus Giselsson

Abstract

We propose a novel variation of the forward–backward splitting method for solving structured monotone inclusions that incorporates past iterates as well as two deviation vectors into the update equations. The deviation vectors bring a great flexibility to the algorithm and can be chosen arbitrarily as long as they jointly satisfy a norm condition. The method is derived from a Lyapunov analysis from which we conclude convergence rates for various quantities. For a specific choice of the parameters and the deviations, our algorithm reduces to the Halpern iteration and the accelerated proximal point method that both converge as $\mathcal{O}(\frac{1}{n^2})$ in squared norm of the fixed-point residual.

Available on arXiv.

1. Introduction

In this work, we consider the problem of finding x in the real Hilbert space $\mathscr{H}$ such that

$$0 \in Ax + Cx \tag{II.1}$$

where $A\colon \mathscr{H} \to 2^{\mathscr{H}}$ is a maximally monotone operator with $2^{\mathscr{H}}$ denoting the power-set of $\mathscr{H}$, and $C\colon \mathscr{H} \to \mathscr{H}$ is a cocoercive operator. This monotone inclusion has optimization problems [Eckstein, 1989; Raguet and Landrieu, 2015], convex-concave saddle-point problems [Chambolle and Pock, 2011], and variational inequalities [Attouch et al., 2011; Chen and Rockafellar, 1997; Tseng, 2000] as special cases.

Forward–backward (FB) splitting [Bruck, 1975; Lions and Mercier, 1979; Passty, 1979] has been broadly used to find solutions of the monotone inclusion problem (II.1). The FB splitting iteration is given by

$$x_{n+1} = (\mathrm{Id} + \gamma A)^{-1}(\mathrm{Id} - \gamma C)x_n, \tag{II.2}$$

where $\gamma > 0$ is a step-size parameter. The gradient method, the proximal point algorithm [Rockafellar, 1976], and the proximal-gradient method [Combettes and Pesquet, 2011] are some widely used special instances of the FB splitting method.

Several attempts have been made to improve the convergence of the FB splitting algorithm by incorporating information from the previous iterations. The heavy-ball method [Polyak, 1964], the inertial proximal point algorithm [Alvarez, 2000; Alvarez and Attouch, 2001], and inertial FB algorithms [Apidopoulos et al., 2020; Attouch and Cabot, 2020; Attouch and Peypouquet, 2016; Attouch et al., 2018; Beck and Teboulle, 2009; Chambolle and Dossal, 2015; Cholamjiak et al., 2018; Lorenz and Pock, 2015] are a few instances that fuse previous information into the current iteration by including a momentum term into the algorithm.

In this paper, we propose an extension to the standard FB splitting algorithm (II.2) to solve the monotone inclusion problem (II.1). In our algorithm, the past information is incorporated in two ways. We use momentum-like terms to construct two extrapolated/deviated points which are fed to the FB operator as its input arguments. In addition, our proposed algorithm has a relaxation step in which the momentum-like terms are included. As a result of fusing past information, our proposed algorithm attains a sublinear rate of convergence of $o(\frac{1}{n^2})$, which is faster than nominal FB splitting that achieves $o(\frac{1}{n})$.

Our algorithm, in its general form, in addition to incorporating iterates from the past, embodies two *deviation vectors*. These deviations can be seen as adjustable parameters of the algorithm that have the same dimension as the underlying space of the problem; hence, providing the algorithm with a great flexibility. This flexibility can be utilized to control the trajectory of the iterates and improve the convergence of the algorithm. Each iteration of the algorithm is safeguarded by requiring the

deviations to jointly satisfy a *safeguard condition* to ensure convergence. Unlike the safeguard conditions in [Giselsson et al., 2016; Sadeghi and Giselsson, 2021; Themelis and Patrinos, 2019; Zhang et al., 2020] that select between a globally convergent and locally fast method, our safeguard condition limits the size of the deviations such that the trajectory of iterates obtained from the algorithm is controlled in all iterations. The deviations in this work have the same role as the ones in [Banert et al., 2021; Sadeghi et al., 2021a; Sadeghi et al., 2021b], which in fact are special instances of our algorithm.

Our Lyapunov analysis of the algorithm is based on using the monotonicity inequality of A and the cocoercivity inequality of C between the last iterate and a solution as well as between the last two points generated by the algorithm. (Such inequalities are referred to as *interpolation conditions* in the terminology of performance estimation (PEP), see for instance [Ryu et al., 2020; Taylor et al., 2017b; Taylor et al., 2017a].) This is in contrast to the analysis in [Sadeghi et al., 2021b] that only uses these inequalities between the last iterate and a solution. The use of the extra inequalities paves the way for deriving a Lyapunov analysis from which we obtain convergence rates of order $o(\frac{1}{n^2})$, which is not attainable for the algorithm given in [Sadeghi et al., 2021b].

A simplified version of our algorithm is given by

$$\begin{aligned} p_n &= (\mathrm{Id}+\gamma A)^{-1}(\mathrm{Id}-\gamma C)y_n, \\ y_{n+1} &= y_n + \tfrac{n}{n+2}(y_n - y_{n-1}) \\ &\quad + \tfrac{(n+1)(4-\gamma\beta)}{2n+4}\big(p_n - y_n - \tfrac{n}{n+1}(p_{n-1}-y_{n-1})\big) \end{aligned} \tag{II.3}$$

where C is $\frac{1}{\beta}$-cocoercive and $0 < \gamma < \frac{4}{\beta}$, where, similar to what has been shown in NOFOB [Giselsson, 2019] and AFBA [Latafat and Patrinos, 2017], the upper step-size bound is larger as compared to nominal FB splitting. This algorithm is a novel inertial-type FB scheme that incorporates past data into its iteration's update equation. It has the Halpern iteration studied in [Lieder, 2021] and the accelerated proximal point method [Kim, 2021] as special instances. Note that algorithm (II.3), which itself is a simplified instance of our proposed algorithm, extends the accelerated proximal point (backward) method to the forward-backward setting for monotone inclusion problems. For this case, we show that the sequence of fixed-point residuals of the algorithm, $y_n - p_n$, converges to zero with a rate of $\mathcal{O}(\frac{1}{n})$.

The paper is organized as follows. In Section 2, we provide some basic definitions along with the notations used throughout the paper. Section 3, in addition to the formal statement of the problem under study, presents our proposed algorithm along with examples demonstrating some special instances of our algorithm. In Section 4, we provide our convergence analysis. In Section 5, we derive several further special instances of our algorithm, two of which lead to the Halpern iteration. We conclude the paper in Section 6 by presenting some deferred results/proofs.

2. Preliminaries

Throughout the paper, the set of real numbers is denoted by $\mathbb{R}$; $\mathscr{H}$ denotes a real Hilbert space that is equipped with an inner product and induced norm, which are denoted by $\langle\cdot,\cdot\rangle$ and $\|\cdot\| = \sqrt{\langle\cdot,\cdot\rangle}$, respectively. A self-adjoint bounded operator $M\colon \mathscr{H} \to \mathscr{H}$ is said to be *strongly positive* if there exists some $c > 0$ such that $\langle x, Mx\rangle \geq c\|x\|^2$ for all $x \in \mathscr{H}$. The notation $\mathscr{M}(\mathscr{H})$ is used to denote the set of linear, self-adjoint, strongly positive operators on $\mathscr{H}$. For $M \in \mathscr{M}(\mathscr{H})$ and for all $x, y \in \mathscr{H}$, 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.

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

Let $M \in \mathscr{M}(\mathscr{H})$. An operator $T\colon \mathscr{H} \to \mathscr{H}$ is said to be

(i) *L-Lipschitz continuous* ($L \geq 0$) w.r.t. $\|\cdot\|_M$ if

$$\|Tx - Ty\|_{M^{-1}} \leq L\|x - y\|_M \qquad \text{for all } x, y \in \mathscr{H};$$

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

$$\langle Tx - Ty, x - y\rangle \geq \frac{1}{\beta}\|Tx - Ty\|^2_{M^{-1}} \qquad \text{for all } x, y \in \mathscr{H};$$

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

By the Cauchy–Schwarz inequality, a $\frac{1}{\beta}$-cocoercive operator is β-Lipschitz continuous.

Let $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ be sequences of real numbers and let $b_n \neq 0$ for all $n \in \mathbb{N}$. we use the notation $a_n \in \mathscr{O}(b_n)$ if there exists $c_0 > 0$ such that $|a_n| \leq c_0|b_n|$ for sufficiently large n; and we say $a_n \in o(b_n)$ if and only if

$$\lim_{n\to\infty} \tfrac{a_n}{b_n} = 0;$$

and we use the notation $a_n \in \Omega(b_n)$ if there exists some $c_1 > 0$ such that $|a_n| \geq c_1|b_n|$ for sufficiently large n.

3. Proposed algorithm

We consider structured monotone inclusion problems of the form

$$0 \in Ax + Cx, \tag{II.4}$$

that satisfy the following assumption.

ASSUMPTION 1 Let $\beta > 0$ and assume that

(i) $A\colon \mathscr{H} \to 2^{\mathscr{H}}$ is maximally monotone,

(ii) $C\colon \mathscr{H} \to \mathscr{H}$ is $\frac{1}{\beta}$-cocoercive with respect to $\|\cdot\|_M$ with $M \in \mathscr{M}(\mathscr{H})$, and

(iii) the solution set $\operatorname{zer}(A+C) := \{x \in \mathscr{H} : 0 \in Ax + Cx\}$ is nonempty. □

As the operator C has a full domain and a cocoercive operator is maximally monotone [Bauschke and Combettes, 2017, Corollary 20.28], the operator $A+C$ is maximally monotone as well [Bauschke and Combettes, 2017, Corollary 25.5].

We present a variant of FB splitting with deviations for solving problem (II.4) in Algorithm 1.

For the FB step to be implementable and the safeguarding step in (II.5) to be satisfied for some u_{n+1} and v_{n+1}, we require for all $n \in \mathbb{N}$ that γ_n, λ_n, θ_n, $\hat{\theta}_n$, and $\tilde{\theta}_n$ are strictly positive and the parameters ζ_n, μ_n, α_n, and $\bar{\alpha}_n$ are non-negative. If these requirements are met, one trivial choice that satisfies the safeguard condition (II.5) is $u_n = v_n = 0$. For the convergence analysis, there are further requirements on some of these parameters that are discussed in Section 4.

The FB step of Algorithm 1 allows the points y_n and z_n to be different, which is in contrast to standard FB splitting (II.2) and the inertial FB method (II.3). The points y_n and z_n are constructed based on a linear combination of the last iterate, some momentum-like terms, and the deviations u_n and v_n. We have a great flexibility in choosing the direction of the deviations u_n and v_n, however, they are confined to a subset of $\mathscr{H}$ which is defined by the condition (II.5). At the step of selecting the deviations, all the quantities involved in the right-hand side of (II.5) are computable. The deviations can be viewed as design parameters of the algorithm, and thus, the flexibility provided by them can be used to control the trajectory of the algorithm with the aim of improving convergence of the algorithm.

In spite of the similarities between Algorithm 1 and the FB splitting with deviations algorithm of [Sadeghi et al., 2021b], there are several differences. First, in the update equations of y_n and z_n, momentum-like terms—$\alpha_n(y_{n-1} - x_n)$ for y_n update and $\alpha_n(y_{n-1} - x_n)$ and $\bar{\alpha}_n(z_{n-1} - p_{n-1})$ for z_n update—are included. Second, the relaxation step of the algorithm, step 8, is equipped with an additional relaxation-like term. Third, the safeguard condition includes an additional non-negative expression on the right-hand side—compared to that of [Sadeghi et al., 2021b]—which could potentially lead to a larger upper-bound on the norm of the deviations. These differences are rooted from including additional monotonicity and cocoercivity inequalities—so-called interpolation conditions in the terminology of [Taylor et al., 2017b; Ryu et al., 2020; Taylor et al., 2017a]—in our analysis compared to the analysis of [Sadeghi et al., 2021b]. Notably, beside using inequalities between the points of the last iterations and a solution, which are the only interpolations used in [Sadeghi et al., 2021b], we use inequalities also between the points generated in the last two iterations of our algorithm. This gives an extra degree of

Algorithm 1

1: **Input:** initial point $x_0 \in \mathscr{H}$; the strictly positive sequences $(\gamma_n)_{n\in\mathbb{N}}$ and $(\lambda_n)_{n\in\mathbb{N}}$; the non-negative sequences $(\zeta_n)_{n\in\mathbb{N}}$ and $(\mu_n)_{n\in\mathbb{N}}$; and the metric $\|\cdot\|_M$ with $M \in \mathscr{M}(\mathscr{H})$.

2: Given the input parameters, for all $n \in \mathbb{N}$, define:

(i) $\alpha_n := \frac{\mu_n}{\lambda_n+\mu_n}$;
(ii) $\bar{\alpha}_n := \frac{\gamma_n\mu_n}{\gamma_{n-1}(\lambda_n+\mu_n)}$;
(iii) $\theta_n := (4-\gamma_n\beta)(\lambda_n+\mu_n) - 2\lambda_n^2$;
(iv) $\hat{\theta}_n := 2\lambda_n + 2\mu_n - \gamma_n\beta\lambda_n^2$;
(v) $\bar{\theta}_n := \lambda_n + \mu_n - \lambda_n^2$;
(vi) $\tilde{\theta}_n := (\lambda_n+\mu_n)\gamma_n\beta$.

3: **set:** $z_0 = y_0 = x_0$ and $u_0 = v_0 = 0$

4: **for** $n = 0, 1, 2, \ldots$ **do**

5: $y_n = x_n + \alpha_n(y_{n-1} - x_n) + u_n$

6: $z_n = x_n + \alpha_n(p_{n-1} - x_n) + \bar{\alpha}_n(z_{n-1} - p_{n-1}) + \frac{\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n}u_n + v_n$

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

8: $x_{n+1} = x_n + \lambda_n(p_n - z_n) + \bar{\alpha}_n\lambda_n(z_{n-1} - p_{n-1})$

9: choose u_{n+1} and v_{n+1} such that

$$\frac{\lambda_{n+1}+\mu_{n+1}}{\zeta_{n+1}}\left(\frac{\tilde{\theta}_{n+1}}{\hat{\theta}_{n+1}}\|u_{n+1}\|_M^2 + \frac{\hat{\theta}_{n+1}}{\theta_{n+1}}\|v_{n+1}\|_M^2\right) \leq \ell_n^2 \tag{II.5}$$

is satisfied, where

$$\begin{aligned}\ell_n^2 = {} & \tfrac{\theta_n}{2}\left\|p_n - x_n + \alpha_n(x_n - p_{n-1}) + \tfrac{\gamma_n\beta\lambda_n^2}{\hat{\theta}_n}u_n - \tfrac{2\bar{\theta}_n}{\theta_n}v_n\right\|_M^2 \\ & + 2\mu_n\gamma_n\left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1}-p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1}\right\rangle_M \\ & + \tfrac{\mu_n\gamma_n\beta}{2}\|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2\end{aligned} \tag{II.6}$$

10: **end for**

freedom in our algorithm represented by the parameter μ_n that comes from how much of these extra interpolation conditions that are used in the analysis. This addition allows us to arrive at convergence rate of $\ell_n^2 \in o(\frac{1}{\theta_n})$ and in particular, by letting λ_n grow linearly with n—which makes θ_n grow quadratically—a rate of $\ell_n^2 \in o(\frac{1}{n^2})$. Such rates are not achievable in [Sadeghi et al., 2021b] as setting μ_n to zero takes us back to the algorithm of [Sadeghi et al., 2021b].

We end this section by presenting some simplified variations of our algorithm as special instances.

EXAMPLE 1
With $\mu_n = 0$ and $u_n = v_n = 0$ for all $n \in \mathbb{N}$, the safeguard condition (II.5) is always satisfied, and Algorithm 1 reads

$$\begin{aligned} p_n &= (M + \gamma_n A)^{-1}(M - \gamma_n C)x_n \\ x_{n+1} &= x_n + \lambda_n(p_n - x_n) \end{aligned}$$

which is the relaxed preconditioned variant of FB splitting. If we further choose $M = \mathrm{Id}$ and $\lambda_n = 1$ for all $n \in \mathbb{N}$, we recover the standard FB splitting (II.2). □

EXAMPLE 2
With $\gamma_n = \gamma$ and $u_n = v_n = 0$ for all $n \in \mathbb{N}$, the safeguard condition is already satisfied, we get $y_n = z_n$, and Algorithm 1, after eliminating x_n, can be simplified to

$$\begin{aligned} p_n &= (M + \gamma A)^{-1}(M - \gamma C)y_n \\ y_{n+1} &= y_n + \tfrac{(1-\alpha_{n+1})\alpha_n}{1-\alpha_n}(y_n - y_{n-1}) \\ &\quad + \lambda_n(1 - \alpha_{n+1})(p_n - y_n - \alpha_n(p_{n-1} - y_{n-1})). \end{aligned}$$

This algorithm is a novel inertial-type FB algorithm. □

EXAMPLE 3
With $\gamma_n = \gamma$ and

$$v_n = \tfrac{(2-\gamma\beta)(\lambda_n+\mu_n)}{\hat{\theta}_n} u_n,$$

for all $n \in \mathbb{N}$, we get $z_n = y_n$ and Algorithm 1 reduces to

$$\begin{aligned} y_n &= x_n + \alpha_n(y_{n-1} - x_n) + u_n \\ p_n &= (M + \gamma_n A)^{-1}(M - \gamma_n C)y_n \\ x_{n+1} &= x_n + \lambda_n(p_n - y_n) + \alpha_n\lambda_n(y_{n-1} - p_{n-1}) \end{aligned}$$

where the safeguard condition (II.5) reads as

$$\begin{aligned} \tfrac{(\lambda_{n+1}+\mu_{n+1})^2}{\zeta_{n+1}\theta_{n+1}}\|u_{n+1}\|_M^2 \leq \tfrac{\theta_n}{4}\Big\|p_n - x_n + \alpha_n(x_n - p_{n-1}) - \tfrac{\theta_n - 2(\lambda_n+\mu_n)}{\theta_n}u_n\Big\|_M^2 \\ + \mu_n\gamma_n\Big\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1}-p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1}\Big\rangle_M \\ + \tfrac{\mu_n\gamma_n\beta}{4}\|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2 \end{aligned}$$

In Section 5.2, we see that this algorithm, with a slightly tighter safeguard condition, has the Halpern iteration [Lieder, 2021] and the accelerated proximal point method [Kim, 2021] as special cases. □

EXAMPLE 4
With $\mu_n = 0$ for all $n \in \mathbb{N}$, the algorithm reduces to the forward–backward splitting with deviations [Sadeghi et al., 2021b]. If we further let

$$v_n = \tfrac{2-\gamma_n\beta}{4-\gamma_n\beta-2\lambda_n} u_n,$$

for all $n \in \mathbb{N}$, we get $z_n = y_n$ and Algorithm 1 is simplified to

$$\begin{aligned} y_n &= x_n + u_n \\ p_n &= (M+\gamma_n A)^{-1}(M-\gamma_n C)y_n \\ x_{n+1} &= x_n + \lambda_n(p_n - y_n) \end{aligned}$$

and the safeguard condition (II.5) reduces to

$$\|u_{n+1}\|_M^2 \leq \tfrac{\zeta_{n+1}\lambda_n(4-\gamma_n\beta-2\lambda_n)(4-\gamma_{n+1}\beta-2\lambda_{n+1})}{4\lambda_{n+1}} \left\| p_n - x_n - \tfrac{2-\gamma_n\beta-2\lambda_n}{4-\gamma_n\beta-2\lambda_n} u_n \right\|_M^2.$$

This algorithm, is Algorithm 1 of [Sadeghi et al., 2021a] which itself is a special instance of the forward–backward splitting with deviations algorithm [Sadeghi et al., 2021b]. □

4. Convergence analysis

In this section, we present our Lyapunov-based convergence analysis of Algorithm 1. In Theorem 1, we define a quantity V_n—which we call a Lyapunov function—based on the iterates generated by Algorithm 1, and present an identity that establishes a relation between V_{n+1} and V_n. In Theorem 2, under a set of assumptions on the the parameters of Algorithm 1, we introduce an inequality—which we refer to as a Lyapunov inequality—from which, several useful results can be deduced. Then, in Theorem 3, we use this inequality along with some assumptions on the parameters to draw conclusions on the rate of convergence of the algorithm, as well as, on the summability and convergence of some sequences of the terms that appear in the Lyapunov function/inequality. This theorem is followed by results that address two particularly important corner cases.

The proof of our first theorem is lengthy and only based on algebraic manipulations and is therefore deferred to Section 6.

THEOREM 1 Suppose that Assumption 1 holds. Let $x^\star$ be an arbitrary point in $\operatorname{zer}(A+C)$ and $V_0 = \|x_0 - x^\star\|_M^2$, and based on the iterates generated by Algorithm 1, for all $n \in \mathbb{N}$, let

$$V_{n+1} := \|x_{n+1} - x^\star\|_M^2 + 2\lambda_{n+1}\gamma_{n+1}\alpha_{n+1}\phi_n + \ell_n^2, \tag{II.7}$$

where

$$\phi_n := \left\langle \tfrac{z_n - p_n}{\gamma_n}, p_n - x^\star \right\rangle_M + \tfrac{\beta}{4}\|y_n - p_n\|_M^2, \tag{II.8}$$

and ℓ_n^2 given by (II.6). Then,

$$V_{n+1} + 2\gamma_n(\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + \ell_{n-1}^2 = V_n + (\lambda_n + \mu_n)\left(\frac{\tilde{\theta}_n}{\hat{\theta}_n}\|u_n\|_M^2 + \frac{\hat{\theta}_n}{\theta_n}\|v_n\|_M^2\right)$$

holds for all $n \in \mathbb{N}$. □

The identity relation provided in Theorem 1 becomes more insightful if we know that all its constituent terms are non-negative. In that case, one can immediately conclude, for instance, that $(V_n)_{n\in\mathbb{N}}$ is a non-increasing sequence. Non-negativity of these terms relies on the joint selection of the parameter sequences $(\zeta_n)_{n\in\mathbb{N}}$, $(\gamma_n)_{n\in\mathbb{N}}$, $(\mu_n)_{n\in\mathbb{N}}$, and $(\lambda_n)_{n\in\mathbb{N}}$ according, for instance, to the following assumption.

ASSUMPTION 2 Choose $\varepsilon_0, \varepsilon_1 \in [0,1)$ and $\varepsilon \in \left(0, \min\left(1, \frac{2}{\beta}\right)\right)$, and assume that, for all $n \in \mathbb{N}$, $\mu_n \geq 0$ and the following hold:

(i) $0 \leq \zeta_n \leq 1 - \varepsilon_0$;

(ii) $\frac{2-\sqrt{4-2\beta\varepsilon}}{\beta} \leq \gamma_n \leq \frac{2+\sqrt{4-2\beta\varepsilon}}{\beta}$;

(iii) $\varepsilon \leq \gamma_n\lambda_n \leq \gamma_{n-1}\lambda_{n-1} + 2\gamma_n(1 - \frac{\gamma_n\beta}{4}) - \varepsilon$; and

(iv) $\varepsilon_1 + \gamma_n\lambda_n - \gamma_{n-1}\lambda_{n-1} \leq \frac{\gamma_n\lambda_n^2}{\lambda_n+\mu_n} \leq \frac{(4-\gamma_n\beta)\gamma_n\lambda_n^2}{2\lambda_n^2+\varepsilon}$. □

REMARK 1 Our convergence analysis entails that the parameter sequences $(\theta_n)_{n\in\mathbb{N}}$, $\left(\hat{\theta}_n\right)_{n\in\mathbb{N}}$, and $\left(\tilde{\theta}_n\right)_{n\in\mathbb{N}}$ to be lower-bounded by a positive constant. This follows by Assumption 2 since then, $\tilde{\theta}_n \geq \beta\varepsilon$, $\theta_n = (4-\gamma_n\beta)(\lambda_n+\mu_n) - 2\lambda_n^2 \geq \varepsilon$, and

$$\begin{aligned}\hat{\theta}_n &= 2(\lambda_n+\mu_n) - \gamma_n\beta\lambda_n^2 \geq 2(\lambda_n+\mu_n) - \tfrac{1}{2}\gamma_n\beta((4-\gamma_n\beta)(\lambda_n+\mu_n) - \varepsilon) \\ &= \tfrac{1}{2}(\lambda_n+\mu_n)(2-\gamma_n\beta)^2 + \tfrac{1}{2}\gamma_n\beta\varepsilon \geq \varepsilon\left(1 - \sqrt{1 - \tfrac{1}{2}\beta\varepsilon}\right).\end{aligned}$$ □

In the following result, we introduce a so-called Lyapunov inequality that is the foundation of the rest of the results in this section.

THEOREM 2 Suppose that Assumption 1 and Assumption 2 hold. Let $x^\star$ be an arbitrary point in $\operatorname{zer}(A+C)$, and the sequences $\left(\ell_n^2\right)_{n\in\mathbb{N}}$, $(V_n)_{n\in\mathbb{N}}$, and $(\phi_n)_{n\in\mathbb{N}}$ be

constructed in terms of the iterates obtained from Algorithm 1, as per (II.6)–(II.8), respectively. Further, for all $n \in \mathbb{N}$, let

$$\begin{aligned}\varphi_n := &\left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1} - p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1} \right\rangle_M - \tfrac{1}{\beta}\|Cy_n - Cy_{n-1}\|^2_{M^{-1}} \\ &+ \langle Cy_n - Cy_{n-1}, y_n - y_{n-1} - (p_n - p_{n-1})\rangle.\end{aligned} \tag{II.9}$$

Then, for all $n \in \mathbb{N}$,

(i) $\varphi_n \geq 0$, and

$$\begin{aligned}&\varphi_n + \tfrac{\beta}{4}\left\| \tfrac{2}{\beta} M^{-1}(Cy_n - Cy_{n-1}) + p_n - p_{n-1} - y_n + y_{n-1} \right\|^2_M \\ &\quad = \left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1} - p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1} \right\rangle_M + \tfrac{\beta}{4}\|p_n - p_{n-1} - y_n + y_{n-1}\|^2_M;\end{aligned}$$

(ii) ℓ^2_n, ϕ_n, and V_n are non-negative;
(iii) and the following inequality holds

$$V_{n+1} + 2\gamma_n(\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + (1 - \zeta_n)\ell^2_{n-1} \leq V_n. \qquad \square$$

Proof. It follows from step 7 of Algorithm 1 that

$$\frac{Mz_n - Mp_n}{\gamma_n} - Cy_n \in Ap_n. \tag{II.10}$$

From (II.10) and montonicity of A we get

$$0 \leq \left\langle \tfrac{Mz_n - Mp_n}{\gamma_n} - Cy_n - \tfrac{Mz_{n-1} - Mp_{n-1}}{\gamma_{n-1}} + Cy_{n-1}, p_n - p_{n-1} \right\rangle. \tag{II.11}$$

From $\frac{1}{\beta}$-cocoercivity of C w.r.t. $\|\cdot\|_M$, we have

$$0 \leq \langle Cy_n - Cy_{n-1}, y_n - y_{n-1}\rangle - \tfrac{1}{\beta}\|Cy_n - Cy_{n-1}\|^2_{M^{-1}}. \tag{II.12}$$

Adding (II.11) and (II.12), yields

$$\begin{aligned}0 \leq &\left\langle \tfrac{Mz_n - Mp_n}{\gamma_n} - Cy_n - \tfrac{Mz_{n-1} - Mp_{n-1}}{\gamma_{n-1}} + Cy_{n-1}, p_n - p_{n-1} \right\rangle \\ &+ \langle Cy_n - Cy_{n-1}, y_n - y_{n-1}\rangle - \tfrac{1}{\beta}\|Cy_n - Cy_{n-1}\|^2_{M^{-1}} \\ = &\left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1} - p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1} \right\rangle_M - \tfrac{1}{\beta}\|Cy_n - Cy_{n-1}\|^2_{M^{-1}} \\ &+ \langle Cy_n - Cy_{n-1}, y_n - y_{n-1} - (p_n - p_{n-1})\rangle = \varphi_n.\end{aligned}$$

Then, from the equality above, we have

$$\varphi_n = \left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1} - p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1} \right\rangle_M + \tfrac{\beta}{4}\|p_n - p_{n-1} - y_n + y_{n-1}\|^2_M$$

$$-\frac{\beta}{4}\left\|\tfrac{2}{\beta}M^{-1}(Cy_n-Cy_{n-1})+p_n-p_{n-1}-y_n+y_{n-1}\right\|_M^2, \tag{II.13}$$

where we used

$$\langle s,t\rangle-\tfrac{1}{\delta}\|s\|_{M^{-1}}^2=\tfrac{\delta}{4}\|t\|_M^2-\tfrac{\delta}{4}\left\|\tfrac{2}{\delta}M^{-1}s-t\right\|_M^2$$

for all $t,s\in\mathscr{H}$. Rearranging the terms in (II.13) gives the desired relation.

For Theorem 2 (ii), due to Remark 1 and Assumption 2 and by construction of ℓ_n^2 as per (II.6), and given $\varphi_n\geq 0$, it is evident that, for all $n\in\mathbb{N}$, $\ell_n^2\geq 0$. By $x^\star\in\operatorname{zer}(A+C)$, we have

$$-Cx^\star\in Ax^\star \tag{II.14}$$

From (II.10) and (II.14), and montonicity of A we get

$$0\leq\left\langle\tfrac{Mz_n-Mp_n}{\gamma_n}-Cy_n+Cx^\star,p_n-x^\star\right\rangle. \tag{II.15}$$

From $\frac{1}{\beta}$-cocoercivity of C w.r.t. $\|\cdot\|_M$, we have

$$0\leq\langle Cy_n-Cx^\star,y_n-x^\star\rangle-\tfrac{1}{\beta}\|Cy_n-Cx^\star\|_{M^{-1}}^2. \tag{II.16}$$

Define

$$\begin{aligned}\widehat{\phi}_n:=&\left\langle\tfrac{Mz_n-Mp_n}{\gamma_n}-Cy_n+Cx^\star,p_n-x^\star\right\rangle\\&+\langle Cy_n-Cx^\star,y_n-x^\star\rangle-\tfrac{1}{\beta}\|Cy_n-Cx^\star\|_{M^{-1}}^2\\=&\left\langle\tfrac{z_n-p_n}{\gamma_n},p_n-x^\star\right\rangle_M+\langle Cy_n-Cx^\star,y_n-p_n\rangle-\tfrac{1}{\beta}\|Cy_n-Cx^\star\|_{M^{-1}}^2,\end{aligned}$$

which is constructed by adding (II.15) and (II.16), and thus, $\widehat{\phi}_n\geq 0$ by construction. Then, from (II.8) we have

$$\begin{aligned}\widehat{\phi}_n&=\left\langle\tfrac{z_n-p_n}{\gamma_n},p_n-x^\star\right\rangle_M+\langle Cy_n-Cx^\star,y_n-p_n\rangle-\tfrac{1}{\beta}\|Cy_n-Cx^\star\|_{M^{-1}}^2\\&=\left\langle\tfrac{z_n-p_n}{\gamma_n},p_n-x^\star\right\rangle_M+\tfrac{\beta}{4}\|y_n-p_n\|_M^2\\&\quad-\tfrac{\beta}{4}\left\|\tfrac{2}{\beta}M^{-1}(Cy_n-Cx^\star)+p_n-y_n\right\|_M^2,\end{aligned} \tag{II.17}$$

where in the last equality we used

$$\langle s,t\rangle-\tfrac{1}{\delta}\|s\|_{M^{-1}}^2=\tfrac{\delta}{4}\|t\|_M^2-\tfrac{\delta}{4}\left\|\tfrac{2}{\delta}M^{-1}s-t\right\|_M^2$$

for all $t,s\in\mathscr{H}$. Rearranging the terms in (II.17) gives

$$\begin{aligned}0&\leq\widehat{\phi}_n+\tfrac{\beta}{4}\left\|\tfrac{2}{\beta}M^{-1}(Cy_n-Cx^\star)+p_n-y_n\right\|_M^2\\&=\left\langle\tfrac{z_n-p_n}{\gamma_n},p_n-x^\star\right\rangle_M+\tfrac{\beta}{4}\|y_n-p_n\|_M^2=\phi_n,\end{aligned} \tag{II.18}$$

and thus, $\phi_n \geq 0$. Additionally, since $\ell_n^2 \geq 0$ and the coefficients of ϕ_n in (II.7) are non-negative by Assumption 2 (ii) and Assumption 2 (iii), $V_n \geq 0$ by construction.

For Theorem 2 (iii), by Theorem 1, we have

$$\begin{aligned} V_{n+1} + \ell_{n-1}^2 + 2\gamma_n(\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1})\phi_n \\ = V_n + (\lambda_n + \mu_n)\left(\tfrac{\tilde{\theta}_n}{\hat{\theta}_n}\|u_n\|_M^2 + \tfrac{\hat{\theta}_n}{\theta_n}\|v_n\|_M^2\right). \end{aligned}$$

Using this equality and (II.5) gives

$$V_{n+1} + 2\gamma_n(\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + \ell_{n-1}^2 \leq V_n + \zeta_n\ell_{n-1}^2.$$

Due to Assumption 2 (i), moving $\zeta_n\ell_{n-1}^2$ to the other side gives the desired result. This concludes the proof. □

REMARK 2 As seen in the proof of Theorem 2 (i), the interpolation conditions between the points obtained from the last two iterations of the algorithm, mentioned before, are in fact added to construct the condition $\varphi_n \geq 0$. Having this interpolation in our analysis allows us to bring the adjustable parameter μ_n into the algorithm. This parameter enters into the convergence analysis as the coefficient of the interpolation condition $\varphi_n \geq 0$. As we see later in this section, the rate results are obtained for our algorithm are all rooted into the presence of this parameter. □

REMARK 3 Note that from Assumption 2, one can potentially choose to fix λ_n, let it be growing with $\Omega(n)$, or select any other variation in between these two. If λ_n is selected to be fixed for all $n \in \mathbb{N}$, then, μ_n would be bounded from above and below by Assumption 2 (iv). On the other hand, if one choose $(\lambda_n)_{n\in\mathbb{N}}$ to be growing, then $(\mu_n)_{n\in\mathbb{N}}$ has to be growing as well. In particular, by Assumption 2 (iv), if the relaxation parameter $(\lambda_n)_{n\in\mathbb{N}}$ is chosen to be increasing as n increases, then $(\mu_n)_{n\in\mathbb{N}}$ must be growing with $\Omega(\lambda_n^2)$. □

Before stating the main convergence result, we present the following lemma on boundedness of some sequences of coefficients.

LEMMA 1 Consider the quantities defined in Algorithm 1 and suppose that Assumption 2 holds. Given $\beta > 0$, the sequences $\left(\frac{\tilde{\theta}_n}{\hat{\theta}_n}\right)_{n\in\mathbb{N}}$, $\left(\frac{\hat{\theta}_n}{\theta_n}\right)_{n\in\mathbb{N}}$, $\left(\frac{\theta_n}{2\lambda_n^2}\right)_{n\in\mathbb{N}}$, $\left(\frac{(2-\gamma_n\beta)(\lambda_n+\mu_n)}{\theta_n}\right)_{n\in\mathbb{N}}$, and $\left(\frac{\lambda_n+\mu_n}{\lambda_n^2}\right)_{n\in\mathbb{N}}$ are bounded. □

Proof. By the assumption, we know that all the quantities in the denominator of the sequences are lower-bounded by some positive constants. Thus, the only way that these sequences can be unbounded is that the absolute value of their numerator grow towards infinity with a rate faster than the rate of growth of their associated denominators. We show that this can not be the case. The only way that the absolute value of the numerators of these sequences can grow is to let $(\lambda_n)_{n\in\mathbb{N}}$ be increasing. Let

us make that assumption. Then, according to Assumption 2, the sequence $(\mu_n)_{n\in\mathbb{N}}$ grows with $\Omega(\lambda_n^2)$ (see Remark 3). By definition, the sequences $(\theta_n)_{n\in\mathbb{N}}$, $\left(\tilde{\theta}_n\right)_{n\in\mathbb{N}}$, and $\left(\hat{\theta}_n\right)_{n\in\mathbb{N}}$ grow linearly with μ_n, and thus, they are increasing with $\Omega(\lambda_n^2)$ as well. Hence, we see that if the numerators of these sequences are increasing, their associated denominators would be growing with the same rate. Therefore, none of these sequences are unbounded. □

The following convergence theorem, which is based on Theorem 2, excludes the edge cases of Assumption 2 where the constants ε_0 and ε_1 can be chosen as zero. We consider two corner cases corresponding to these choices after the theorem below.

THEOREM 3 Suppose that Assumption 1 and Assumption 2 hold. Let $x^\star$ be an arbitrary point in $\operatorname{zer}(A+C)$, and the sequences $\left(\ell_n^2\right)_{n\in\mathbb{N}}$, $(V_n)_{n\in\mathbb{N}}$, $(\phi_n)_{n\in\mathbb{N}}$, and $(\varphi_n)_{n\in\mathbb{N}}$ be constructed in terms of the iterates obtained from Algorithm 1, as per (II.6)-(II.9), respectively. Then, provided that $\varepsilon_0, \varepsilon_1 > 0$, the following hold:

(i) the sequences $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ and $(\phi_n)_{n\in\mathbb{N}}$ are summable, $(V_n)_{n\in\mathbb{N}}$ is convergent, and $(x_n)_{n\in\mathbb{N}}$ is a bounded sequence;
(ii) the sequences $(\lambda_n u_n)_{n\in\mathbb{N}}$ and $(\lambda_n v_n)_{n\in\mathbb{N}}$ are convergent to zero with a rate of $o(1)$;
(iii) $x_{n+1} - x_n \to 0$ as $n \to \infty$;
(iv) the sequences

$$\left(\left\|p_n - x_n + \alpha_n(x_n - p_{n-1}) + \tfrac{\gamma_n\beta\lambda_n^2}{\hat{\theta}_n}u_n - \tfrac{2\bar{\theta}_n}{\theta_n}v_n\right\|_M^2\right)_{n\in\mathbb{N}}$$

and

$$\left(\left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1} - p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1}\right\rangle_M + \tfrac{\beta}{4}\|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2\right)_{n\in\mathbb{N}}$$

converge to zero with the rates of $o(\frac{1}{\theta_n})$ and $o(\frac{1}{\mu_n})$, respectively;
(v) the sequences $(\varphi_n)_{n\in\mathbb{N}}$ and

$$\left(\left\|p_n - y_n + \tfrac{2}{\beta}M^{-1}Cy_n - \left(p_{n-1} - y_{n-1} + \tfrac{2}{\beta}M^{-1}Cy_{n-1}\right)\right\|_M^2\right)_{n\in\mathbb{N}}$$

converge to zero with a rate of $o(\frac{1}{\mu_n})$;
(vi) the sequences $(\phi_n)_{n\in\mathbb{N}}$ and $\left(p_n - y_n + \frac{2}{\beta}M^{-1}Cy_n\right)_{n\in\mathbb{N}}$ are respectively convergent to zero and $Cx^\star$ with a rate of $\mathcal{O}(\frac{1}{\lambda_n})$. □

Proof. To show Theorem 3 (i), we use

$$V_{n+1} + 2\gamma_n(\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + (1 - \zeta_n)\ell_{n-1}^2 \leq V_n,$$

from Theorem 2 (iii). The sequences $\left(\ell_n^2\right)_{n\in\mathbb{N}}$, $(V_n)_{n\in\mathbb{N}}$, and $(\phi_n)_{n\in\mathbb{N}}$ are non-negative by Theorem 2 (ii). Additionally, by Assumption 2 (i) and Assumption 2 (iv) respectively, the quantities $1-\zeta_n$ and $\lambda_n - \bar{\alpha}_n\lambda_{n+1}$ are non-negative for all $n\in\mathbb{N}$; and thus, the quantity $2\gamma_n(\lambda_n-\bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + (1-\zeta_n)\ell_{n-1}^2$ is non-negative for all $n\in\mathbb{N}$. Therefore, by [Bauschke and Combettes, 2017, Lemma 5.31] the sequence $(V_n)_{n\in\mathbb{N}}$ converges and the sequence

$$\left(2\gamma_n(\lambda_n-\bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + (1-\zeta_n)\ell_{n-1}^2\right)_{n\in\mathbb{N}}$$

is summable. Moreover, for all $n\in\mathbb{N}$, we have $\liminf_{n\to\infty}\gamma_n > 0$ by Assumption 2 (ii), $1-\zeta_n \geq \varepsilon_0 > 0$ by Assumption 2 (i), and $\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1} \geq \varepsilon_1 > 0$ by Assumption 2 (iv), hence, summability of the sequence above implies that $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ and $(\phi_n)_{n\in\mathbb{N}}$ are summable; thus, $\ell_n^2\to 0$ and $\phi_n\to 0$ as $n\to\infty$. Since V_n is convergent and its constituent terms in

$$V_n = \|x_n - x^\star\|_M^2 + \ell_{n-1}^2 + 2\lambda_n\gamma_n\alpha_n\phi_{n-1},$$

are all non-negative, the sequence $\left(\|x_n-x^\star\|_M^2\right)_{n\in\mathbb{N}}$ converges which implies that the sequence $(x_n)_{n\in\mathbb{N}}$ is bounded.

To show Theorem 3 (ii), note that due to (II.5) and Assumption 2 (i), the summability of $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ implies summability of

$$\left(\frac{\lambda_{n+1}+\mu_{n+1}}{\lambda_{n+1}^2}\left(\frac{\tilde{\theta}_{n+1}}{\hat{\theta}_{n+1}}\|\lambda_{n+1}u_{n+1}\|_M^2 + \frac{\hat{\theta}_{n+1}}{\theta_{n+1}}\|\lambda_{n+1}v_{n+1}\|_M^2\right)\right)_{n\in\mathbb{N}}. \tag{II.19}$$

Hence, as, for all $n\in\mathbb{N}$, by Remark 1 and Lemma 1 the coefficients in the expression above are strictly positive and bounded, the sequences $(\lambda_n u_n)_{n\in\mathbb{N}}$ and $(\lambda_n v_n)_{n\in\mathbb{N}}$ must be convergent to zero with a rate of $o(1)$.

For Theorem 3 (iii), note that since $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ is summable and it is comprised of two positive terms, its constituent terms must be summable. Therefore, from (II.6), the sequence

$$\left(\frac{\theta_n}{2}\left\|p_n - x_n + \alpha_n(x_n-p_{n-1}) + \frac{\gamma_n\beta\lambda_n^2}{\hat{\theta}_n}u_n - \frac{2\bar{\theta}_n}{\theta_n}v_n\right\|_M^2\right)_{n\in\mathbb{N}}$$

is summable. Next, we use Lemma 2 to replace the expression inside the norm above by Lemma 2 (iii). Then, by taking the factor $\frac{1}{\lambda_n}$ out of the norm, we get

$$\left(\frac{\theta_n}{2\lambda_n^2}\left\|x_{n+1} - x_n + \frac{\tilde{\theta}_n}{\hat{\theta}_n}\lambda_n u_n + \frac{(2-\gamma_n\beta)(\lambda_n+\mu_n)}{\theta_n}\lambda_n v_n\right\|_M^2\right)_{n\in\mathbb{N}},$$

which is a summable sequence too. Since all the coefficients in the expression above are bounded by Lemma 1, by Theorem 3 (ii), we get $x_{n+1}-x_n\to 0$ as $n\to\infty$.

For Theorem 3 (iv), since $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ is summable, it is convergent to zero with $o(1)$. As a result, its constituent terms which are non-negative, are convergent to zero at least with the same rate. Recalling the definition of ℓ_n^2 from (II.6) gives the desired result.

Theorem 2 (i) along with Theorem 3 (iv) immediately imply the assertion of Theorem 3 (v).

For the proof of Theorem 3 (vi), from (II.18) in proof of Theorem 2, we have

$$\phi_n = \widehat{\phi}_n + \frac{\beta}{4}\left\|\frac{2}{\beta}M^{-1}(Cy_n - Cx^\star) + p_n - y_n\right\|_M^2 \tag{II.20}$$

Thus, due to the summability of $(\phi_n)_{n\in\mathbb{N}}$ from Theorem 3 (i), and since the terms in the identity relation above are non-negative, both of them are convergent to zero, which implies that $p_n - y_n + \frac{2}{\beta}M^{-1}Cy_n \to Cx^\star$ as $n \to \infty$. Moreover, from Theorem 2 we have

$$V_{n+1} = \|x_{n+1} - x^\star\|_M^2 + \ell_n^2 + 2\lambda_{n+1}\gamma_{n+1}\alpha_{n+1}\phi_n \leq V_0.$$

Due to the fact that α_n and γ_n are bounded for all $n \in \mathbb{N}$, from this we see that $(\phi_n)_{n\in\mathbb{N}}$, as well as its constituent terms, are convergent to zero with $\mathcal{O}(\frac{1}{\lambda_n})$. This completes the proof. □

Next, we present a convergence result for one of the corner cases mentioned before in which we assume that $(\gamma_n\lambda_n)_{n\in\mathbb{N}}$ is an increasing sequence, that $\varepsilon_1 = 0$ in Assumption 2 and that the lower-bound of Assumption 2 (iv) holds with equality, which implies that Assumption 2 (iv) reads as

$$\begin{aligned} \mu_n &= \frac{\gamma_{n-1}\lambda_{n-1}\lambda_n}{\gamma_n\lambda_n - \gamma_{n-1}\lambda_{n-1}}, \\ \varepsilon &\leq \gamma_n\lambda_n - \gamma_{n-1}\lambda_{n-1}, \end{aligned} \tag{II.21}$$

for all $n \in \mathbb{N}$ and with $\lambda_{-1} = 0$. Based on this assumption, as proven in the theorem below, the $(\theta_n)_{n\in\mathbb{N}}$ grows quadratically with n, and as a consequence of Theorem 3 (iv), we conclude a convergence rate of $o(\frac{1}{n^2})$, at the cost of loosing summability of $(\phi_n)_{n\in\mathbb{N}}$ an boundedness of $(x_n)_{n\in\mathbb{N}}$.

THEOREM 4 Suppose that Assumption 1 and Assumption 2 (i)-Assumption 2 (iii) along with the assumptions given by (II.21) hold. Let $x^\star$ be an arbitrary point in $\operatorname{zer}(A+C)$, and the sequences $\left(\ell_n^2\right)_{n\in\mathbb{N}}$, $(V_n)_{n\in\mathbb{N}}$, $(\phi_n)_{n\in\mathbb{N}}$, and $(\varphi_n)_{n\in\mathbb{N}}$ be constructed in terms of the iterates obtained from Algorithm 1, as per (II.6)-(II.9), respectively. Then, provided that $\varepsilon_0 > 0$, the following hold:

(i) the sequence $\left(\ell_n^2\right)_{n\in\mathbb{N}}$ is summable, and $(V_n)_{n\in\mathbb{N}}$ is a convergent sequence;
(ii) the sequences $(u_n)_{n\in\mathbb{N}}$, and $(v_n)_{n\in\mathbb{N}}$ are convergent to zero with a rate of $o(\frac{1}{n})$;

(iii) $x_{n+1} - x_n \to 0$ as $n \to \infty$;
(iv) the sequences

$$\left(\left\| p_n - x_n + \alpha_n(x_n - p_{n-1}) + \tfrac{\gamma_n \beta \lambda_n^2}{\hat{\theta}_n} u_n - \tfrac{2\bar{\theta}_n}{\theta_n} v_n \right\|_M^2 \right)_{n\in\mathbb{N}}$$

and

$$\left(\left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1} - p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1} \right\rangle_M + \tfrac{\beta}{4} \|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2 \right)_{n\in\mathbb{N}}$$

converge to zero with a rate of $o(\frac{1}{n^2})$;
(v) the sequences $(\varphi_n)_{n\in\mathbb{N}}$ and

$$\left(\left\| p_n - y_n + \tfrac{2}{\beta} M^{-1} C y_n - \left(p_{n-1} - y_{n-1} + \tfrac{2}{\beta} M^{-1} C y_{n-1} \right) \right\|_M^2 \right)_{n\in\mathbb{N}}$$

converge to zero with a rate of $o(\frac{1}{n^2})$;
(vi) the sequences $(\phi_n)_{n\in\mathbb{N}}$ and $\left(p_n - y_n + \frac{2}{\beta} M^{-1} C y_n \right)_{n\in\mathbb{N}}$ are respectively convergent to zero and $Cx^\star$ with a rate of $\mathcal{O}(\frac{1}{n})$. □

Proof. From the assumption given by (II.21), $(\lambda_n)_{n\in\mathbb{N}}$ grows with $\Omega(n)$ and $(\mu_n)_{n\in\mathbb{N}}$ grows with $\Omega(n^2)$. By the definition of θ_n and from (II.21), we obtain

$$\begin{aligned}
\theta_n &= (4 - \gamma_n\beta)(\lambda_n + \mu_n) - 2\lambda_n^2 \\
&= (4 - \gamma_n\beta)\left(\lambda_n + \frac{\gamma_{n-1}\lambda_{n-1}\lambda_n}{\gamma_n\lambda_n - \gamma_{n-1}\lambda_{n-1}} \right) - 2\lambda_n^2 \\
&= \frac{(4 - \gamma_n\beta)\gamma_n\lambda_n^2 - 2(\gamma_n\lambda_n - \gamma_{n-1}\lambda_{n-1})\lambda_n^2}{\gamma_n\lambda_n - \gamma_{n-1}\lambda_{n-1}} \\
&= \frac{\lambda_n^2}{\gamma_n\lambda_n - \gamma_{n-1}\lambda_{n-1}} ((4 - \gamma_n\beta)\gamma_n + 2\gamma_{n-1}\lambda_{n-1} - 2\gamma_n\lambda_n).
\end{aligned}$$

Due to this and by assumption (II.21) and Assumption 2 (iii), we conclude that $(\theta_n)_{n\in\mathbb{N}}$ grows with $\Omega(n^2)$.

Given the growth rates of $(\lambda_n)_{n\in\mathbb{N}}$, $(\mu_n)_{n\in\mathbb{N}}$, and $(\theta_n)_{n\in\mathbb{N}}$ as above, it is straightforward to obtain the asserted results from Theorem 3. □

Note that in the theorem above,we cannot guarantee summability of $(\phi_n)_{n\in\mathbb{N}}$ as by (II.21), the coefficient of ϕ_n in the Lyapunov inequality is zero, that is $\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1} = 0$ for all $n \in \mathbb{N}$. As a consequence, we cannot coclude that

$$\left(\|x_n - x^\star\|_M^2 \right)_{n\in\mathbb{N}}$$

is convergent, and thus, our argument on boundedness of $(x_n)_{n\in\mathbb{N}}$ no longer holds.

Finally, we consider the corner case with $\varepsilon_0 = 0$ and $\zeta_n = 1$ for all $n \in \mathbb{N}$ and equality in the lower bound in Assumption 2 (iv) as in Theorem 4, which gives (II.21). To get meaningful summability results, we tighten the safeguard condition (5) to

$$(\lambda_{n+1} + \mu_{n+1}) \left(\frac{\tilde{\theta}_{n+1}}{\hat{\theta}_{n+1}} \|u_{n+1}\|_M^2 + \frac{\hat{\theta}_{n+1}}{\theta_{n+1}} \|v_{n+1}\|_M^2 \right) \leq \tilde{\ell}_n^2, \tag{II.22}$$

where $\tilde{\ell}_n^2 := \ell_n^2 - \hat{\ell}_n^2$ for all $n \in \mathbb{N}$, with ℓ_n^2 given by (II.6) and

$$\begin{aligned} \hat{\ell}_n^2 := 2\mu_n\gamma_n \left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1} - p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1} \right\rangle_M \\ + \tfrac{\mu_n\gamma_n\beta}{2} \|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2. \end{aligned} \tag{II.23}$$

Note that ,for all $n \in \mathbb{N}$, by Theorem 2 (i) and Assumption 2, $\hat{\ell}_n^2$ is non-negative and due to Remark 1 and by definition, $\tilde{\ell}_n^2$ is non-negative, as well. With these choices, the algorithm attains a convergence rate of $o(\frac{1}{n^2})$ for the sequence $(\varphi_n)_{n\in\mathbb{N}}$, which as per Theorem 2 (i), is a combination of monotonicity and cocoercivity inequalities for two consecutive iterates.

We will show in Section 5.3 that Algorithm 1 with this setting, for some specific choices of the parameters, leads to the Halpern iteration studied in [Lieder, 2021] and we recover the same convergence rate for the fixed-point residual from our rate result on $\left(\ell_n^2\right)_{n\in\mathbb{N}}$.

THEOREM 5 Suppose that Assumption 1 and Assumption 2 (i)-Assumption 2 (iii) along with the assumptions given by (II.21) hold. Let $x^\star$ be an arbitrary point in $\operatorname{zer}(A + C)$, and $(\phi_n)_{n\in\mathbb{N}}$, $(V_n)_{n\in\mathbb{N}}$, and $\left(\hat{\ell}_n^2\right)_{n\in\mathbb{N}}$ be respectively constructed as per (II.7), (II.8), and (II.23) by the iterates obtained from Algorithm 1 with its safeguard condition modified to (II.22). Then, the following hold:

(i) for all $n \in \mathbb{N}$

$$V_{n+1} + \hat{\ell}_{n-1}^2 \leq V_n.$$

(ii) the sequence $(V_n)_{n\in\mathbb{N}}$ is convergent and $\left(\hat{\ell}_n^2\right)_{n\in\mathbb{N}}$ is summable;

(iii) the sequences

$$\left(\left\| p_n - x_n + \alpha_n(x_n - p_{n-1}) + \tfrac{\gamma_n\beta\lambda_n^2}{\hat{\theta}_n} u_n - \tfrac{2\bar{\theta}_n}{\theta_n} v_n \right\|_M^2 \right)_{n\in\mathbb{N}}$$

and

$$\left(\left\langle \tfrac{z_n - p_n}{\gamma_n} - \tfrac{z_{n-1} - p_{n-1}}{\gamma_{n-1}}, p_n - p_{n-1} \right\rangle_M + \tfrac{\beta}{4} \|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2 \right)_{n\in\mathbb{N}}$$

converge to zero with rates of $\mathcal{O}(\frac{1}{n^2})$ and $o(\frac{1}{n^2})$, respectively;

(iv) the sequences $(\varphi_n)_{n\in\mathbb{N}}$ and

$$\left(\left\|p_n - y_n + \tfrac{2}{\beta}M^{-1}Cy_n - \left(p_{n-1} - y_{n-1} + \tfrac{2}{\beta}M^{-1}Cy_{n-1}\right)\right\|_M^2\right)_{n\in\mathbb{N}}$$

converge to zero with a rate of $o(\frac{1}{n^2})$;

(v) the sequences $(\phi_n)_{n\in\mathbb{N}}$ and $\left(p_n - y_n + \frac{2}{\beta}M^{-1}Cy_n\right)_{n\in\mathbb{N}}$ are respectively convergent to zero and $Cx^\star$ with a rate of $\mathcal{O}(\frac{1}{n})$. □

Proof. To prove Theorem 5 (i), from Theorem 1 we have

$$\begin{aligned} V_{n+1} + 2\gamma_n(\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + \ell_{n-1}^2 \\ = V_n + (\lambda_n + \mu_n)\left(\tfrac{\tilde{\theta}_n}{\hat{\theta}_n}\|u_n\|_M^2 + \tfrac{\hat{\theta}_n}{\theta_n}\|v_n\|_M^2\right). \end{aligned}$$

Note that by (II.21), $\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1}$ is zero for all $n \in \mathbb{N}$. This gives

$$V_{n+1} + \ell_{n-1}^2 = V_n + (\lambda_n + \mu_n)\left(\tfrac{\tilde{\theta}_n}{\hat{\theta}_n}\|u_n\|_M^2 + \tfrac{\hat{\theta}_n}{\theta_n}\|v_n\|_M^2\right).$$

Substituting $\ell_n^2 = \hat{\ell}_n^2 + \tilde{\ell}_n^2$, we obtain

$$V_{n+1} + \hat{\ell}_{n-1}^2 + \tilde{\ell}_{n-1}^2 = V_n + (\lambda_n + \mu_n)\left(\tfrac{\tilde{\theta}_n}{\hat{\theta}_n}\|u_n\|_M^2 + \tfrac{\hat{\theta}_n}{\theta_n}\|v_n\|_M^2\right).$$

Using the new condition on deviations, given by (II.22), yields the desired inequality.

For the proof of Theorem 5 (ii), observe that the sequences $(V_n)_{n\in\mathbb{N}}$ and $(\hat{\ell}_n^2)_{n\in\mathbb{N}}$ are non-negative by Theorem 2 (i) and by construction, respectively. By Theorem 5 (i), the sequence $(V_n)_{n\in\mathbb{N}}$ is nonincreasing and, by [Bauschke and Combettes, 2017, Lemma 5.31], it is convergent and $(\hat{\ell}_n^2)_{n\in\mathbb{N}}$ is summable.

The proofs of Theorem 5 (iii)-Theorem 5 (v) follow from very similar arguments given in the proofs of Theorem 3 (iv) and Theorem 3 (vi), respectively. This completes the proof. □

5. Special cases

In this section, we introduce some special instances of Algorithm 1 by giving specific choices of the parameters and the deviations.

5.1 Vanishing deviations

By letting $u_n = v_n = 0$ for all $n \in \mathbb{N}$, Algorithm 1 reduces to

$$y_n = x_n + \alpha_n(y_{n-1} - x_n)$$

$$
\begin{aligned}
z_n &= x_n + \alpha_n(p_{n-1} - x_n) + \bar{\alpha}_n(z_{n-1} - p_{n-1}) \\
p_n &= (M + \gamma_n A)^{-1}(Mz_n - \gamma_n C y_n) \\
x_{n+1} &= x_n + \lambda_n(p_n - z_n) + \bar{\alpha}_n\lambda_n(z_{n-1} - p_{n-1}).
\end{aligned}
$$

The algorithm given in Example 2 is a special case of this algorithm. To show that, after eliminating x_n from the algorithm above, we obtain

$$
\begin{aligned}
y_n &= z_n + \bar{\alpha}_n(p_{n-1} - z_{n-1}) + \alpha_n(y_{n-1} - p_{n-1}), \\
p_n &= (M + \gamma_n A)^{-1}(Mz_n - \gamma_n C y_n), \\
z_{n+1} &= z_n + \tfrac{1-\alpha_{n+1}}{1-\alpha_n}(\alpha_n z_n - \bar{\alpha}_n z_{n-1}) \\
&\quad + (\lambda_n - \alpha_{n+1}\lambda_n - \bar{\alpha}_{n+1} + \alpha_{n+1})(p_n - z_n) \\
&\quad - \lambda_n\bar{\alpha}_n(1 - \alpha_{n+1})(p_{n-1} - z_{n-1}) + \tfrac{(1-\alpha_{n+1})(\bar{\alpha}_n - \alpha_n)}{1-\alpha_n} p_{n-1}.
\end{aligned}
\tag{II.24}
$$

Setting $\gamma_n = \gamma$ gives $y_n = z_n$ and $\bar{\alpha}_n = \alpha_n$, for all $n \in \mathbb{N}$. Substituting these into (II.24) yields the algorithm of Example 2. The algorithms above are convergent by Theorem 3 with a rate of $o(\frac{1}{\mu_n})$.

REMARK 4 In the case that the deviations are identically zero, unlike Remark 1, we do not require $\hat{\theta}_n$ to be strictly positive for all $n \in \mathbb{N}$. In fact, for our convergence analysis to hold, it suffices to have $\theta_n \geq 0$ for all $n \in \mathbb{N}$; as in that case, the safeguard condition (II.5) is already satisfied. Therefore, we can drop the ε from the upper-bounds of Assumption 2 (iii) and Assumption 2 (iv). In particular, the condition on λ_n can be restated as

$$
\gamma_n\lambda_n \leq \gamma_{n-1}\lambda_{n-1} + 2\gamma_n\left(1 - \frac{\gamma_n\beta}{4}\right). \tag{II.25}
$$

□

In what follows, we present a special instance of algorithm (II.24), using

$$
\lambda_n = 1 + \frac{\gamma_{n-1}}{\gamma_n}\lambda_{n-1},
$$

and $\gamma_n\beta \leq 2$, for all $n \in \mathbb{N}$, which satisfy (II.21) and (II.25). Substituting μ_n from (II.21) and definitions of α_n and $\bar{\alpha}_n$ from Algorithm 1 along with the choice of λ_n above into (II.24), we obtain

$$
\begin{aligned}
y_n &= z_n + \tfrac{\gamma_n(\lambda_n - 1)}{\gamma_{n-1}\lambda_n}(p_{n-1} - z_{n-1}) + \tfrac{\lambda_n - 1}{\lambda_n}(y_{n-1} - p_{n-1}), \\
p_n &= (M + \gamma_n A)^{-1}(Mz_n - \gamma_n C y_n), \\
z_{n+1} &= \tfrac{\gamma_{n+1}\lambda_n}{\gamma_n\lambda_n + \gamma_{n+1}} z_n + \tfrac{\gamma_n\lambda_n}{\gamma_n\lambda_n + \gamma_{n+1}} p_n - \tfrac{(\lambda_n - 1)\gamma_{n+1}}{\gamma_n\lambda_n + \gamma_{n+1}} p_{n-1}.
\end{aligned}
\tag{II.26}
$$

According to Theorem 4, this algorithm is convergent with a rate of $o(\frac{1}{n^2})$.

5.2 Parallel deviations

In this section, we consider Algorithm 1 with the alternative safeguard condition given by (II.22), and with the particular choice of the deviations as

$$u_n = \tfrac{\bar{\alpha}_n \theta_n}{2(\lambda_n+\mu_n)}(p_{n-1}-z_{n-1}), \qquad v_n = \tfrac{\bar{\alpha}_n \theta_n (2-\gamma_n\beta)}{2\hat{\theta}_n}(p_{n-1}-z_{n-1}). \tag{II.27}$$

Having these parallel deviations along with the assumption of $\gamma_n = \gamma$, implies $y_n = z_n$ for all $n \in \mathbb{N}$. Then, by substitution of the specified parameters and deviations into Algorithm 1, and after eliminating x_n, we obtain

$$\begin{aligned} p_n &= (M+\gamma A)^{-1}(M-\gamma C)y_n \\ y_{n+1} &= y_n + \tfrac{\alpha_n(1-\alpha_{n+1})}{1-\alpha_n}(y_n - y_{n-1}) \\ &\quad + \Big(\lambda_n(1-\alpha_{n+1}) + \tfrac{\alpha_{n+1}\theta_{n+1}}{2(\lambda_{n+1}+\mu_{n+1})}\Big)(p_n - y_n) \\ &\quad - \Big(\tfrac{(1-\alpha_{n+1})\alpha_n\theta_n}{2(1-\alpha_n)(\lambda_n+\mu_n)} + \alpha_n\lambda_n(1-\alpha_{n+1})\Big)(p_{n-1}-y_{n-1}), \end{aligned} \tag{II.28}$$

where the parameters must be chosen such that

$$\bar{\alpha}_{n+1}^2 \theta_{n+1} \leq \theta_n \tag{II.29}$$

is satisfied at each iteration, as the safeguard condition (II.22) reduces to this condition in the specified setting. This means that, we do not need to evaluate any norms to ensure convergence of the algorithm.

By letting $\lambda_n = (1+n)\sigma$, for all $n \in \mathbb{N}$, with σ being a positive constant such that Assumption 2 (iii) and (II.21) are satisfied, regardless of the value of σ, (II.28) becomes

$$\begin{aligned} p_n &= (M+\gamma A)^{-1}(M-\gamma C)y_n, \\ y_{n+1} &= \tfrac{\gamma\beta(1+n)}{4+2n}y_n + \tfrac{n(2-\gamma\beta)}{4+2n}y_{n-1} + \tfrac{(1+n)(4-\gamma\beta)}{4+2n}p_n - \tfrac{n(4-\gamma\beta)}{4+2n}p_{n-1}, \end{aligned} \tag{II.30}$$

which is an alternative presentation of the algorithm given by (II.3). Observe that with the set of selected parameters that led to this algorithm, the condition (II.29) is always satisfied.

The following result shows that algorithm (II.30) is convergent, with respect to the norm of the fixed-point residual, with a rate of $\mathcal{O}(\frac{1}{n})$.

PROPOSITION 1 Consider Algorithm 1 with the modified safeguard condition given by (II.22) and let the deviations be selected as per (II.27). Given $y_0 = x_0 \in \mathscr{H}$ and $\beta > 0$, let $x^\star$ be an arbitrary point in $\operatorname{zer}(A+C)$, and for all $n \in \mathbb{N}$, $\lambda_n = (1+n)(1-\frac{\gamma\beta}{4})$. Then, Algorithm 1 reduces to the algorithm given by (II.30) which converges as

$$\|p_n - y_n\|_M^2 \leq \frac{1}{\left(1-\frac{\gamma\beta}{4}\right)^2 (n+1)^2}\|y_0 - x^\star\|_M^2. \qquad \square$$

Proof. Using the choices made by assumption, from Theorem 5 and the definition of V_n in (II.7), we obtain

$$\tilde{\ell}_n^2 = \left(1 - \tfrac{\gamma\beta}{4}\right)^2 (n+1)^2 \|p_n - y_n\|_M^2 \leq V_{n+1} \leq V_0 = \|x_0 - x^\star\|_M^2 = \|y_0 - x^\star\|_M^2.$$

This concludes the proof. □

By letting $Cx = 0$ for all $x \in \mathscr{H}$, $M = \mathrm{Id}$ and $\beta = 0$, we arrive at the accelerated proximal point method [Kim, 2021] and the convergence rate results found in [Kim, 2021] can be recovered by Proposition 1. This means that this special case of our general Algorithm 1 is a generalization of the accelerated proximal point method.

5.3 Halpern iteration

Given $y_0 \in \mathscr{H}$ and the nonexpansive operator $N : \mathscr{H} \to \mathscr{H}$, a special case of the Halpern iteration that is studied in [Lieder, 2021] is defined as

$$y_{n+1} = \tfrac{1}{n+2} y_0 + \tfrac{n+1}{n+2} N y_n, \tag{II.31}$$

for all $n \in \mathbb{N}$.

Prior to proceeding to the derivation of Halpern iteration from our algorithm, we give the following result. The proof is straightforward and is left to the reader.

PROPOSITION 2 Given a nonexpansive operator $N : \mathscr{H} \to \mathscr{H}$ and a positive constant $\beta > 0$, the operator $\frac{\beta}{2}(\mathrm{Id} - N)$ is $\frac{1}{\beta}$-cocoercive. □

In what follows, we show that the algorithm given by (II.31) is a special case of Algorithm 1. We derive the Halpern iteration from our algorithm in two different ways. We first use the algorithm outlined in (II.26) in Section 5.1, and then, utilize the algorithm presented in (II.30). For both cases, we show that the algorithm converges with $o(\frac{1}{n^2})$.

Alternative 1. In the algorithm given by (II.26), having $\gamma_n = \gamma$, for all $n \in \mathbb{N}$, implies that $y_n = z_n$ and $\lambda_n = 1 + n$; and simplifies the algorithm to

$$\begin{aligned} p_n &= (M + \gamma A)^{-1}(M - \gamma C) y_n, \\ y_{n+1} &= \tfrac{n+1}{n+2} y_n + \tfrac{n+1}{n+2} p_n - \tfrac{n}{n+2} p_{n-1}, \end{aligned}$$

which can alternatively be cast as

$$\begin{aligned} p_n &= (M + \gamma A)^{-1}(M - \gamma C) y_n, \\ y_{n+1} &= \frac{1}{n+2} y_0 + \frac{n+1}{n+2} p_n. \end{aligned} \tag{II.32}$$

Next, given the nonexpansive operator N, let us set $Ax = \{0\}$ for all $x \in \mathscr{H}$, $M = \mathrm{Id}$, $C = \frac{\beta}{2}(\mathrm{Id} - N)$, and $\gamma\beta = 2$. By Proposition 2, the operator C is $\frac{1}{\beta}$-cocercive; and hence, it can be inserted in place of C in (II.32). This gives

$$\begin{aligned} p_n &= y_n - \frac{2}{\beta}\left(\frac{\beta}{2}(\mathrm{Id} - N)\right)y_n = Ny_n \\ y_{n+1} &= \frac{1}{n+2}y_0 + \frac{n+1}{n+2}p_n \end{aligned} \tag{II.33}$$

which is the Halpern iteration as given by (II.31). Recalling Theorem 4, it is easy to verify that this algorithm converges, for instance in φ_n value, with a convergence rate of $o(\frac{1}{n^2})$.

Alternative 2. Similar to what is done above, given the nonexpansive operator N, and having the choices $Ax = \{0\}$ for all $x \in \mathscr{H}$, $M = \mathrm{Id}$, $C = \frac{\beta}{2}(\mathrm{Id} - N)$, and $\gamma_n\beta = 2$ for all $n \in \mathbb{N}$, invoking Proposition 2 and applying this setting, the algorithm presented in (II.30) becomes (II.33). Similar to the earlier alternative, from Theorem 5, one can verify that this algorithm converges, e.g., in the value of φ_n, with a rate of $o(\frac{1}{n^2})$.

In addition to the rate result given above, for the approach taken in the latter alternative, we can derive exactly the same rate as found in [Lieder, 2021]. This is given in the following result.

PROPOSITION 3 Consider Algorithm 1 with the modified safeguard condition given by (II.22) and let the deviations be selected as per (II.27). Given $y_0 \in \mathscr{H}$, $\beta > 0$, and the nonexpansive operator $N : \mathscr{H} \to \mathscr{H}$, let $Ax = \{0\}$ for all $x \in \mathscr{H}$, $M = \mathrm{Id}$, $C = \frac{\beta}{2}(\mathrm{Id} - N)$, $x^\star$ be an arbitrary fixed-point of the operator N, and for all $n \in \mathbb{N}$, $\gamma_n\beta = 2$ and $\lambda_n = \frac{1}{2}(1+n)$. Then, Algorithm 1 results in Halpern iteration (II.31) which converges as

$$\|Ny_n - y_n\|^2 \leq \frac{4}{(n+1)^2}\|y_0 - x^\star\|^2. \qquad \square$$

Proof. The fact that with given assumptions, Algorithm 1 becomes the Halpern-itaeration is already shown. Next, using the choices made by assumption, from Theorem 5 and the definition of V_n in (II.7), we obtain

$$\tilde{\ell}_n^2 = \frac{1}{4}(n+1)^2\|p_n - y_n\|^2 \leq V_{n+1} \leq V_0 = \|x_0 - x^\star\|^2 = \|y_0 - x^\star\|^2.$$

By substitution of $p_n = Ny_n$, we obtain the desired result. $\square$

We derived the Halpern iteration using two different approaches and verified that both converge in, for instance φ_n, with a rate of $o(\frac{1}{n^2})$. For the second alternative

we recover the same rate—in the norm of residual—as the one obtained by [Lieder, 2021]. However, to the best of our knowledge, this is not possible for the first alternative. This can be explained by the fact that we derived the same algorithm—Halpern iteration—assuming two different sets of parameters and deviations. In the setting that led to the second alternative, we used a tighter safeguard condition. This resulted in the two approaches having different underlying Lyapunov inequalities, and hence, their rates could be obtained for different quantities.

As a final remark, observe that with the choice of $N = 2(\mathrm{Id}+\gamma A)^{-1} - \mathrm{Id}$, the Halpern iteration [Lieder, 2021] for finding fixed-points of the nonexpansive operator N and the accelerated proximal point method [Kim, 2021] of finding the root of a maximally monotone operator A are equivalent, since, given the same initial point $x_0 \in \mathscr{H}$, they generate the same sequence of iterates. This was recently shown by [Ryu and Yin, 2021].

6. Deferred results and proofs

In what follows, we present some results that have been used in the previous sections along with the proof of Theorem 1 that was deferred to this section. Prior to that, we define the auxiliary parameter

$$\theta_n' := (2-\gamma_n\beta)\mu_n + 2\bar{\alpha}_n\bar{\theta}_n \tag{II.34}$$

which frequently appears throughout this section.

We begin by establishing some identities between the parameters defined in Algorithm 1. These identities are used several times in the proof of Theorem 1.

PROPOSITION 4 Consider the auxiliary parameters defined in step 2 of Algorithm 1. Then, for all $n \in \mathbb{N}$, the following identities hold

(i) $\theta_n = (2-\gamma_n\beta)\bar{\theta}_n + \hat{\theta}_n$;
(ii) $\theta_n = 2\bar{\theta}_n + (2-\gamma_n\beta)(\lambda_n+\mu_n)$;
(iii) $\lambda_n^2\theta_n = \hat{\theta}_n(\lambda_n+\mu_n) - 2\bar{\theta}_n^2$. □

Proof. For Proposition 4 (i), from definition of $\bar{\theta}_n$ and $\hat{\theta}_n$, we have

$$\begin{aligned}
\theta_n &= (2-\gamma_n\beta)\bar{\theta}_n + \hat{\theta}_n \\
&= (2-\gamma_n\beta)\left(\lambda_n+\mu_n-\lambda_n^2\right) + \left(2\lambda_n+2\mu_n-\gamma_n\beta\lambda_n^2\right) \\
&= (2-\gamma_n\beta)(\lambda_n+\mu_n) - 2\lambda_n^2 + \gamma_n\beta\lambda_n^2 + \left(2\lambda_n+2\mu_n-\gamma_n\beta\lambda_n^2\right) \\
&= (2-\gamma_n\beta)(\lambda_n+\mu_n) - 2\lambda_n^2 + 2(\lambda_n+\mu_n) \\
&= (4-\gamma_n\beta)(\lambda_n+\mu_n) - 2\lambda_n^2
\end{aligned}$$

which holds by definition of θ_n in Algorithm 1. For Proposition 4 (ii) we have

$$\begin{aligned}\theta_n &= 2\bar{\theta}_n + (2-\gamma_n\beta)(\lambda_n+\mu_n)\\ &= 2\big(\lambda_n+\mu_n-\lambda_n^2\big)+(2-\gamma_n\beta)(\lambda_n+\mu_n)\\ &= -2\lambda_n^2+(4-\gamma_n\beta)(\lambda_n+\mu_n).\end{aligned}$$

For Proposition 4 (iii), after moving all terms to the left-hand side of the equality we get

$$\begin{aligned}&\lambda_n^2\theta_n + 2\bar{\theta}_n^2 - \hat{\theta}_n(\lambda_n+\mu_n)\\ &\quad= \lambda_n^2\big((2-\gamma_n\beta)\bar{\theta}_n+\hat{\theta}_n\big)+2\bar{\theta}_n^2-\hat{\theta}_n(\lambda_n+\mu_n)\\ &\quad= \bar{\theta}_n\big(\lambda_n^2(2-\gamma_n\beta)+2\bar{\theta}_n\big)-\hat{\theta}_n(\lambda_n+\mu_n-\lambda_n^2)\\ &\quad= \bar{\theta}_n\big(2\lambda_n+2\mu_n-\gamma_n\beta\lambda_n^2\big)-\hat{\theta}_n\bar{\theta}_n = \bar{\theta}_n\hat{\theta}_n-\hat{\theta}_n\bar{\theta}_n\end{aligned}$$

where in the first equality θ_n is substituted by from Proposition 4 (i) and in the second and the third equality definitions of $\bar{\theta}_n$ and $\hat{\theta}_n$ are used, respectively. □

The next lemma provides alternative expressions for the term inside the first norm in (II.6).

LEMMA 2 Suppose that Assumption 1 holds and consider the sequences generated by Algorithm 1. Then, for all $n\in\mathbb{N}$, the following expressions represent the same vector

(i) $p_n-(1-\alpha_n)x_n-\alpha_n p_{n-1}+\frac{\gamma_n\beta\lambda_n^2}{\hat{\theta}_n}u_n-\frac{2\bar{\theta}_n}{\theta_n}v_n$;

(ii) $p_n-\frac{2\bar{\theta}_n}{\theta_n}z_n+\frac{\tilde{\theta}_n}{\theta_n}y_n-\frac{2\lambda_n}{\theta_n}x_n-\frac{\theta_n'}{\theta_n}p_{n-1}+\frac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n}z_{n-1}-\frac{\tilde{\theta}_n\alpha_n}{\theta_n}y_{n-1}$;

(iii) $\frac{1}{\lambda_n}(x_{n+1}-x_n)+\frac{\tilde{\theta}_n}{\hat{\theta}_n}u_n+\frac{(2-\gamma_n\beta)(\lambda_n+\mu_n)}{\theta_n}v_n$. □

Proof. We, first, show that Lemma 2 (ii) represents the same vector as Lemma 2 (i):

$$\begin{aligned}&p_n-\tfrac{2\bar{\theta}_n}{\theta_n}z_n+\tfrac{\tilde{\theta}_n}{\theta_n}y_n-\tfrac{2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'}{\theta_n}p_{n-1}+\tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n}z_{n-1}-\tfrac{\tilde{\theta}_n\alpha_n}{\theta_n}y_{n-1}\\ &\quad= p_n-\tfrac{2\bar{\theta}_n}{\theta_n}(z_n-\bar{\alpha}_n z_{n-1})+\tfrac{\tilde{\theta}_n}{\theta_n}(y_n-\alpha_n y_{n-1})-\tfrac{2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'}{\theta_n}p_{n-1}\\ &\quad= p_n-\tfrac{2\bar{\theta}_n}{\theta_n}\Big((1-\alpha_n)x_n+(\alpha_n-\bar{\alpha}_n)p_{n-1}+\tfrac{\tilde{\theta}_n\gamma_n\beta}{\hat{\theta}_n}u_n+v_n\Big)\\ &\qquad+\tfrac{\tilde{\theta}_n}{\theta_n}((1-\alpha_n)x_n+u_n)-\tfrac{2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'}{\theta_n}p_{n-1}\\ &\quad= p_n-\tfrac{2(1-\alpha_n)\bar{\theta}_n-(1-\alpha_n)\tilde{\theta}_n+2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'+2\bar{\theta}_n(\alpha_n-\bar{\alpha}_n)}{\theta_n}p_{n-1}\\ &\qquad+\tfrac{\hat{\theta}\tilde{\theta}_n-2\bar{\theta}_n^2\gamma_n\beta}{\theta_n\hat{\theta}_n}u_n-\tfrac{2\bar{\theta}_n}{\theta_n}v_n\end{aligned}$$

$$= p_n - (1-\alpha_n)x_n - \alpha_n p_{n-1} + \tfrac{\gamma_n\beta\lambda_n^2}{\hat{\theta}_n}u_n - \tfrac{2\bar{\theta}_n}{\theta_n}v_n$$

where the coefficients of the last equality are found as follows. The numerator of the coefficient of x_n reads

$$\begin{aligned}
&2(1-\alpha_n)\bar{\theta}_n - (1-\alpha_n)\tilde{\theta} + 2\lambda_n \\
&\quad = (1-\alpha_n)(\theta_n - (2-\gamma_n\beta)(\lambda_n+\mu_n)) \\
&\qquad - (1-\alpha_n)\gamma_n\beta(\lambda_n+\mu_n) + 2\lambda_n \\
&\quad = (1-\alpha_n)\theta_n - 2(1-\alpha_n)(\lambda_n+\mu_n) + 2\lambda_n \\
&\quad = (1-\alpha_n)\theta_n - 2\tfrac{\lambda_n}{\lambda_n+\mu_n}(\lambda_n+\mu_n) + 2\lambda_n = (1-\alpha_n)\theta_n
\end{aligned} \tag{II.35}$$

where in the first equality, $\bar{\theta}_n$ is substituted from Proposition 4 (ii), and $\tilde{\theta}_n$ and α_n are substituted by their definitions in Algorithm 1. The numerator of the coefficient of p_{n-1} is

$$\begin{aligned}
\theta_n' + 2\bar{\theta}_n(\alpha_n - \bar{\alpha}_n) &= (2-\gamma_n\beta)\mu_n + 2\bar{\alpha}_n\bar{\theta}_n + 2\bar{\theta}_n(\alpha_n - \bar{\alpha}_n) \\
&= (2-\gamma_n\beta)\mu_n + 2\bar{\theta}_n\alpha_n \\
&= (2-\gamma_n\beta)\alpha_n(\lambda_n+\mu_n) + 2\bar{\theta}_n\alpha_n = \alpha_n\theta_n
\end{aligned} \tag{II.36}$$

where in the first equality (II.34) is used, the third equality is obtained using the definition of α_n, and Proposition 4 (ii) is utilized in the last equality. For the numerator of u_n we get

$$\begin{aligned}
\hat{\theta}\tilde{\theta}_n - 2\bar{\theta}_n^2\gamma_n\beta &= \hat{\theta}\gamma_n\beta(\lambda_n+\mu_n) - 2\bar{\theta}_n^2\gamma_n\beta \\
&= \gamma_n\beta\left(\hat{\theta}_n(\lambda_n+\mu_n) - 2\bar{\theta}_n^2\right) = \gamma_n\beta\lambda_n^2\theta_n
\end{aligned} \tag{II.37}$$

where the first equality is obtained by substitution of the definition of $\tilde{\theta}_n$ from Algorithm 1, and in the last equality Proposition 4 (iii) is used.

Now, we show that Lemma 2 (ii) and Lemma 2 (iii) represent the same vector. Starting from Lemma 2 (ii), we have

$$\begin{aligned}
&p_n - \tfrac{2\bar{\theta}_n}{\theta_n}z_n + \tfrac{\tilde{\theta}_n}{\theta_n}y_n - \tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} + \tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n}z_{n-1} - \tfrac{\tilde{\theta}_n\alpha_n}{\theta_n}y_{n-1} \\
&\quad = p_n - \tfrac{2\bar{\theta}_n}{\theta_n}(z_n - \bar{\alpha}_n z_{n-1}) + \tfrac{\tilde{\theta}_n}{\theta_n}(y_n - \alpha_n y_{n-1}) - \tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} \\
&\quad = \tfrac{1}{\lambda_n}(x_{n+1} - x_n) + z_n + \bar{\alpha}_n(p_{n-1} - z_{n-1}) - \tfrac{2\bar{\theta}_n}{\theta_n}(z_n - \bar{\alpha}_n z_{n-1}) \\
&\qquad + \tfrac{\tilde{\theta}_n}{\theta_n}(y_n - \alpha_n y_{n-1}) - \tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} \\
&\quad = \tfrac{1}{\lambda_n}(x_{n+1} - x_n) + \tfrac{\theta_n - 2\bar{\theta}_n}{\theta_n}(z_n - \bar{\alpha}_n z_{n-1}) + \tfrac{\tilde{\theta}_n}{\theta_n}(y_n - \alpha_n y_{n-1}) - \tfrac{2\lambda_n}{\theta_n}x_n \\
&\qquad + \tfrac{\bar{\alpha}_n\theta_n - \theta_n'}{\theta_n}p_{n-1} \\
&\quad = \tfrac{1}{\lambda_n}(x_{n+1} - x_n) + \tfrac{\theta_n - 2\bar{\theta}_n}{\theta_n}\left((1-\alpha_n)x_n + (\alpha_n - \bar{\alpha}_n)p_{n-1} + \tfrac{\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n}u_n + v_n\right)
\end{aligned}$$

$$
\begin{aligned}
&\quad + \tfrac{\tilde{\theta}_n}{\theta_n}\big((1-\alpha_n)x_n + u_n\big) - \tfrac{2\lambda_n}{\theta_n}x_n + \tfrac{\bar{\alpha}\theta_n - \theta_n'}{\theta_n}p_{n-1} \\
&= \tfrac{1}{\lambda_n}(x_{n+1} - x_n) + \tfrac{\big(\theta_n - 2\bar{\theta}_n\big)\bar{\theta}_n\gamma_n\beta + \tilde{\theta}_n\hat{\theta}_n}{\theta_n\hat{\theta}_n}u_n + \tfrac{\theta_n - 2\bar{\theta}_n}{\theta_n}v_n \\
&\quad + \tfrac{\big(\theta_n - 2\bar{\theta}_n\big)(1-\alpha_n) + \tilde{\theta}_n(1-\alpha_n) - 2\lambda_n}{\theta_n}x_n + \tfrac{\big(\theta_n - 2\bar{\theta}_n\big)(\alpha_n - \bar{\alpha}_n) + \bar{\alpha}_n\theta_n - \theta_n'}{\theta_n}p_{n-1} \\
&= \tfrac{1}{\lambda_n}(x_{n+1} - x_n) + \tfrac{\tilde{\theta}_n}{\hat{\theta}_n}u_n + \tfrac{(2-\gamma_n\beta)(\lambda_n + \mu_n)}{\theta_n}v_n
\end{aligned}
$$

In the second equality, the definition of x_{n+1} from step 8 of Algorithm 1 is used. In the last equality, the coefficient of x_n is found to be $-\frac{1}{\lambda_n}$ by (II.35), the coefficient of p_{n-1} is zero by (II.36), the coefficient of v_n is found by Proposition 4 (ii), and for the coefficient of u_n we have

$$
\begin{aligned}
\big(\theta_n - 2\bar{\theta}_n\big)\bar{\theta}_n\gamma_n\beta + \tilde{\theta}_n\hat{\theta}_n &= \theta_n\bar{\theta}_n\gamma_n\beta - 2\bar{\theta}^2\gamma_n\beta + \tilde{\theta}_n\hat{\theta}_n \\
&= \theta_n\bar{\theta}_n\gamma_n\beta + \gamma_n\beta\lambda_n^2\theta_n \\
&= \theta_n\gamma_n\beta\big(\bar{\theta}_n + \lambda_n^2\big) = \theta_n\gamma_n\beta(\lambda_n + \mu_n)
\end{aligned}
$$

where the second equality is obtained by (II.37), and in the last equality the definition of $\bar{\theta}_n$ is used. This concludes the proof. □

6.1 Proof of Theorem 1

Proof. We define the following quantity

$$
\begin{aligned}
\Delta_n := V_{n+1} - V_n + 2\gamma_n(\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + \ell_{n-1}^2 \\
- (\lambda_n + \mu_n)\left(\tfrac{\tilde{\theta}_n}{\hat{\theta}_n}\|u_n\|_M^2 + \tfrac{\hat{\theta}_n}{\theta_n}\|v_n\|_M^2\right)
\end{aligned}
\tag{II.38}
$$

and prove the result by showing that, for all $n \in \mathbb{N}$, it is identical to zero. By substituting V_{n+1} and V_n in (II.38), we get

$$
\begin{aligned}
\Delta_n &= \|x_{n+1} - x^\star\|_M^2 + \ell_n^2 + 2\lambda_{n+1}\gamma_{n+1}\alpha_{n+1}\phi_n \\
&\quad - \|x_n - x^\star\|_M^2 - \ell_{n-1}^2 - 2\lambda_n\gamma_n\alpha_n\phi_{n-1} \\
&\quad + 2\gamma_n(\lambda_n - \bar{\alpha}_{n+1}\lambda_{n+1})\phi_n + \ell_{n-1}^2 - (\lambda_n + \mu_n)\left(\tfrac{\tilde{\theta}_n}{\hat{\theta}_n}\|u_n\|_M^2 + \tfrac{\hat{\theta}_n}{\theta_n}\|v_n\|_M^2\right) \\
&= \|x_{n+1} - x^\star\|_M^2 - \|x_n - x^\star\|_M^2 + \ell_n^2 - 2\lambda_n\gamma_n\alpha_n\phi_{n-1} + 2\gamma_n\lambda_n\phi_n \\
&\quad - (\lambda_n + \mu_n)\left(\tfrac{\tilde{\theta}_n}{\hat{\theta}_n}\|u_n\|_M^2 + \tfrac{\hat{\theta}_n}{\theta_n}\|v_n\|_M^2\right),
\end{aligned}
$$

where in the last equality we used $\gamma_n\bar{\alpha}_{n+1} = \gamma_{n+1}\alpha_{n+1}$. Next, substituting ℓ_n^2 from (II.6), and ϕ_{n-1} and ϕ_n from (II.8) on the right-hand side of the last equality above, yields

$$
\Delta_n = \|x_{n+1} - x^\star\|_M^2 - \|x_n - x^\star\|_M^2
$$

$$
\begin{aligned}
&+\tfrac{1}{2}\theta_n\Big\|p_n-x_n+\alpha_n(x_n-p_{n-1})+\tfrac{\gamma_n\beta\lambda_n^2}{\hat\theta_n}u_n-\tfrac{2\bar\theta_n}{\theta_n}v_n\Big\|_M^2\\
&+2\mu_n\gamma_n\Big\langle\tfrac{z_n-p_n}{\gamma_n}-\tfrac{z_{n-1}-p_{n-1}}{\gamma_{n-1}},p_n-p_{n-1}\Big\rangle_M\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_n-y_n-(p_{n-1}-y_{n-1})\|_M^2\\
&-2\lambda_n\gamma_n\alpha_n\left(\Big\langle\tfrac{z_{n-1}-p_{n-1}}{\gamma_{n-1}},p_{n-1}-x^\star\Big\rangle_M+\tfrac{\beta}{4}\|y_{n-1}-p_{n-1}\|_M^2\right)\\
&+2\lambda_n\gamma_n\left(\Big\langle\tfrac{z_n-p_n}{\gamma_n},p_n-x^\star\Big\rangle_M+\tfrac{\beta}{4}\|y_n-p_n\|_M^2\right)\\
&-(\lambda_n+\mu_n)\left(\tfrac{\tilde\theta_n}{\tilde\theta_n}\|u_n\|_M^2+\tfrac{\hat\theta_n}{\theta_n}\|v_n\|_M^2\right)\\
=\;&\|x_{n+1}-x^\star\|_M^2-\|x_n-x^\star\|_M^2+2\lambda_n\gamma_n\Big\langle\tfrac{z_n-p_n}{\gamma_n},p_n-x^\star\Big\rangle_M\\
&-2\lambda_n\gamma_n\alpha_n\Big\langle\tfrac{z_{n-1}-p_{n-1}}{\gamma_{n-1}},p_{n-1}-p_n+p_n-x^\star\Big\rangle_M\\
&+2\mu_n\gamma_n\Big\langle\tfrac{z_n-p_n}{\gamma_n}-\tfrac{z_{n-1}-p_{n-1}}{\gamma_{n-1}},p_n-p_{n-1}\Big\rangle_M\\
&+\tfrac{1}{2}\theta_n\Big\|p_n-x_n+\alpha_n(x_n-p_{n-1})+\tfrac{\gamma_n\beta\lambda_n^2}{\hat\theta_n}u_n-\tfrac{2\bar\theta_n}{\theta_n}v_n\Big\|_M^2\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_n-y_n-(p_{n-1}-y_{n-1})\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&+\tfrac{\lambda_n\gamma_n\beta}{2}\|y_n-p_n\|_M^2-(\lambda_n+\mu_n)\left(\tfrac{\tilde\theta_n}{\tilde\theta_n}\|u_n\|_M^2+\tfrac{\hat\theta_n}{\theta_n}\|v_n\|_M^2\right)\\
=\;&\|x_{n+1}-x^\star\|_M^2-\|x_n-x^\star\|_M^2\\
&+2\langle\lambda_n(z_n-p_n)-\bar\alpha_n\lambda_n(z_{n-1}-p_{n-1}),p_n-x^\star\rangle_M\\
&+2\Big\langle\mu_n(z_n-p_n)+(\bar\alpha_n\lambda_n-\tfrac{\gamma_n}{\gamma_{n-1}}\mu_n)(z_{n-1}-p_{n-1}),p_n-p_{n-1}\Big\rangle_M\\
&+\tfrac{1}{2}\theta_n\Big\|p_n-x_n+\alpha_n(x_n-p_{n-1})+\tfrac{\gamma_n\beta\lambda_n^2}{\hat\theta_n}u_n-\tfrac{2\bar\theta_n}{\theta_n}v_n\Big\|_M^2\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_n-y_n-(p_{n-1}-y_{n-1})\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&+\tfrac{\lambda_n\gamma_n\beta}{2}\|y_n-p_n\|_M^2-(\lambda_n+\mu_n)\left(\tfrac{\tilde\theta_n}{\tilde\theta_n}\|u_n\|_M^2+\tfrac{\hat\theta_n}{\theta_n}\|v_n\|_M^2\right).
\end{aligned}
$$

We define

$$\omega_n:=\bar\alpha_n\lambda_n-\tfrac{\gamma_n}{\gamma_{n-1}}\mu_n \tag{II.39}$$

and substitute it in the last equality above; and also from step 8 of Algorithm 1, we replace $\lambda_n(z_n-p_n)-\bar\alpha_n\lambda_n(z_{n-1}-p_{n-1})$ by x_n-x_{n+1}. Then, we get

$$
\begin{aligned}
\Delta_n=\;&\|x_{n+1}-x^\star\|_M^2-\|x_n-x^\star\|_M^2+2\langle x_n-x_{n+1},p_n-x^\star\rangle_M\\
&+2\langle\mu_n(z_n-p_n)+\omega_n(z_{n-1}-p_{n-1}),p_n-p_{n-1}\rangle_M\\
&+\tfrac{1}{2}\theta_n\Big\|p_n-x_n+\alpha_n(x_n-p_{n-1})+\tfrac{\gamma_n\beta\lambda_n^2}{\hat\theta_n}u_n-\tfrac{2\bar\theta_n}{\theta_n}v_n\Big\|_M^2\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_n-y_n-(p_{n-1}-y_{n-1})\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2
\end{aligned}
$$

$$
\begin{aligned}
&+ \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2 - (\lambda_n+\mu_n)\left(\tfrac{\tilde\theta_n}{\hat\theta_n}\|u_n\|_M^2 + \tfrac{\hat\theta_n}{\theta_n}\|v_n\|_M^2\right)\\
&= \|x_{n+1} - p_n\|_M^2 - \|x_n - p_n\|_M^2\\
&\quad + 2\langle \mu_n(z_n - p_n) + \omega_n(z_{n-1} - p_{n-1}), p_n - p_{n-1}\rangle_M\\
&\quad + \tfrac{1}{2}\theta_n\left\| p_n - x_n + \alpha_n(x_n - p_{n-1}) + \tfrac{\gamma_n\beta\lambda_n^2}{\hat\theta_n}u_n - \tfrac{2\bar\theta_n}{\theta_n}v_n\right\|_M^2\\
&\quad + \tfrac{\mu_n\gamma_n\beta}{2}\|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2 - \tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1} - p_{n-1}\|_M^2\\
&\quad + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2 - (\lambda_n+\mu_n)\left(\tfrac{\tilde\theta_n}{\hat\theta_n}\|u_n\|_M^2 + \tfrac{\hat\theta_n}{\theta_n}\|v_n\|_M^2\right)
\end{aligned}
$$

where in the last equality we used the identity $2\langle a-b, c-d\rangle_M + \|b-d\|_M^2 - \|a-d\|_M^2 = \|b-c\|_M^2 - \|a-c\|_M^2$ for all $a,b,c,d \in \mathscr{H}$. Now, inserting x_{n+1} from step 8 of Algorithm 1, yields

$$
\begin{aligned}
\Delta_n &= \|x_n - p_n + \lambda_n(p_n - z_n) + \lambda_n\bar\alpha_n(z_{n-1} - p_{n-1})\|_M^2 - \|x_n - p_n\|_M^2\\
&\quad + 2\langle \mu_n(z_n - p_n) + \omega_n(z_{n-1} - p_{n-1}), p_n - p_{n-1}\rangle_M\\
&\quad + \tfrac{1}{2}\theta_n\left\| p_n - x_n + \alpha_n(x_n - p_{n-1}) + \tfrac{\gamma_n\beta\lambda_n^2}{\hat\theta_n}u_n - \tfrac{2\bar\theta_n}{\theta_n}v_n\right\|_M^2\\
&\quad + \tfrac{\mu_n\gamma_n\beta}{2}\|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2 - \tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1} - p_{n-1}\|_M^2\\
&\quad + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2 - (\lambda_n+\mu_n)\left(\tfrac{\tilde\theta_n}{\hat\theta_n}\|u_n\|_M^2 + \tfrac{\hat\theta_n}{\theta_n}\|v_n\|_M^2\right)\\
&= \|\lambda_n(p_n - z_n) + \lambda_n\bar\alpha_n(z_{n-1} - p_{n-1})\|_M^2\\
&\quad + 2\langle x_n - p_n, \lambda_n(p_n - z_n) + \lambda_n\bar\alpha_n(z_{n-1} - p_{n-1})\rangle_M\\
&\quad + 2\langle \mu_n(z_n - p_n) + \omega_n(z_{n-1} - p_{n-1}), p_n - p_{n-1}\rangle_M\\
&\quad + \tfrac{1}{2}\theta_n\left\| p_n - x_n + \alpha_n(x_n - p_{n-1}) + \tfrac{\gamma_n\beta\lambda_n^2}{\hat\theta_n}u_n - \tfrac{2\bar\theta_n}{\theta_n}v_n\right\|_M^2\\
&\quad + \tfrac{\mu_n\gamma_n\beta}{2}\|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2 - \tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1} - p_{n-1}\|_M^2\\
&\quad + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2 - (\lambda_n+\mu_n)\left(\tfrac{\tilde\theta_n}{\hat\theta_n}\|u_n\|_M^2 + \tfrac{\hat\theta_n}{\theta_n}\|v_n\|_M^2\right).
\end{aligned}
$$

Next, using Lemma 2 and steps 5–6 of Algorithm 1, we replace the terms including u_n and v_n in terms of the iterates

$$
\begin{aligned}
\Delta_n &= \|\lambda_n(p_n - z_n) + \lambda_n\bar\alpha_n(z_{n-1} - p_{n-1})\|_M^2\\
&\quad + 2\langle x_n - p_n, \lambda_n(p_n - z_n) + \lambda_n\bar\alpha_n(z_{n-1} - p_{n-1})\rangle_M\\
&\quad + 2\langle \mu_n(z_n - p_n) + \omega_n(z_{n-1} - p_{n-1}), p_n - p_{n-1}\rangle_M\\
&\quad + \tfrac{\theta_n}{2}\left\| p_n - \tfrac{2\bar\theta_n}{\theta_n}z_n + \tfrac{\tilde\theta_n}{\theta_n}y_n - \tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} + \tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1} - \tfrac{\tilde\theta_n\alpha_n}{\theta_n}y_{n-1}\right\|_M^2\\
&\quad + \tfrac{\mu_n\gamma_n\beta}{2}\|p_n - y_n - (p_{n-1} - y_{n-1})\|_M^2 - \tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1} - p_{n-1}\|_M^2\\
&\quad + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n - p_n\|_M^2 - \tfrac{(\lambda_n+\mu_n)\tilde\theta_n}{\hat\theta_n}\|y_n - (1-\alpha_n)x_n - \alpha_n y_{n-1}\|_M^2
\end{aligned}
$$

$$- \tfrac{(\lambda_n+\mu_n)\hat{\theta}_n}{\theta_n} \Big\| z_n - \tfrac{\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n} y_n - \tfrac{(2-\gamma_n\beta)\lambda_n}{\hat{\theta}_n} x_n + (\bar{\alpha}_n - \alpha_n) p_{n-1} - \bar{\alpha}_n z_{n-1} + \tfrac{\alpha_n\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n} y_{n-1} \Big\|_M^2$$

where we used

$$\begin{aligned} v_n &= z_n - (1-\alpha_n)x_n + (\bar{\alpha}_n - \alpha_n)p_{n-1} - \bar{\alpha}_n z_{n-1} - \beta\,\tfrac{\bar{\theta}_n\gamma_n}{\hat{\theta}_n} u_n \\ &= z_n - \tfrac{\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n} y_n - \tfrac{(2-\gamma_n\beta)\lambda_n}{\hat{\theta}_n} x_n + (\bar{\alpha}_n - \alpha_n)p_{n-1} - \bar{\alpha}_n z_{n-1} + \tfrac{\alpha_n\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n} y_{n-1} \end{aligned}$$

which is obtained by substituting u_n from step 5 into step 6 of Algorithm 1. Next, we expand the terms on the right-hand side of the last equality above which include p_n. This yields

$$\begin{aligned} \Delta_n &= \lambda_n^2 \|p_n\|_M^2 + 2\big\langle p_n, \lambda_n^2(-z_n + \bar{\alpha}_n z_{n-1} - \bar{\alpha}_n p_{n-1})\big\rangle_M \\ &\quad + \lambda_n^2 \|z_n - \bar{\alpha}_n z_{n-1} + \bar{\alpha}_n p_{n-1}\|_M^2 \\ &\quad - 2\lambda_n \|p_n\|_M^2 + 2\langle p_n, \lambda_n(z_n + x_n + \bar{\alpha}_n p_{n-1} - \bar{\alpha}_n z_{n-1})\rangle_M \\ &\quad + 2\langle x_n, \lambda_n(-z_n - \bar{\alpha}_n p_{n-1} + \bar{\alpha}_n z_{n-1})\rangle_M \\ &\quad - 2\mu_n \|p_n\|_M^2 + 2\langle p_n, \mu_n z_n + (\mu_n - \omega_n)p_{n-1} + \omega_n z_{n-1}\rangle_M \\ &\quad + 2\langle \mu_n z_n + \omega_n(z_{n-1} - p_{n-1}), -p_{n-1}\rangle_M + \tfrac{1}{2}\theta_n \|p_n\|_M^2 \\ &\quad + 2\Big\langle p_n, \tfrac{\theta_n}{2}\Big(-\tfrac{2\bar{\theta}_n}{\theta_n} z_n + \tfrac{\bar{\theta}_n}{\theta_n} y_n - \tfrac{2\lambda_n}{\theta_n} x_n\Big)\Big\rangle_M \\ &\quad + 2\Big\langle p_n, \tfrac{\theta_n}{2}\Big(-\tfrac{\theta_n'}{\theta_n} p_{n-1} + \tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n} z_{n-1} - \tfrac{\bar{\theta}_n\alpha_n}{\theta_n} y_{n-1}\Big)\Big\rangle_M \\ &\quad + \tfrac{1}{2}\theta_n \Big\| -\tfrac{2\bar{\theta}_n}{\theta_n} z_n + \tfrac{\bar{\theta}_n}{\theta_n} y_n - \tfrac{2\lambda_n}{\theta_n} x_n - \tfrac{\theta_n'}{\theta_n} p_{n-1} + \tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n} z_{n-1} - \tfrac{\bar{\theta}_n\alpha_n}{\theta_n} y_{n-1}\Big\|_M^2 \\ &\quad + \tfrac{\mu_n\gamma_n\beta}{2}\|p_n\|_M^2 + 2\Big\langle p_n, \tfrac{\mu_n\gamma_n\beta}{2}(-y_n - (p_{n-1} - y_{n-1}))\Big\rangle_M \\ &\quad + \tfrac{\mu_n\gamma_n\beta}{2}\|-y_n - (p_{n-1} - y_{n-1})\|_M^2 - \tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1} - p_{n-1}\|_M^2 \\ &\quad + \tfrac{\lambda_n\gamma_n\beta}{2}\|p_n\|_M^2 - 2\Big\langle p_n, \tfrac{\lambda_n\gamma_n\beta}{2} y_n\Big\rangle_M + \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n\|_M^2 \\ &\quad - \tfrac{\bar{\theta}_n(\lambda_n+\mu_n)}{\hat{\theta}_n}\|y_n - (1-\alpha_n)x_n - \alpha_n y_{n-1}\|_M^2 \\ &\quad - \tfrac{\hat{\theta}_n(\lambda_n+\mu_n)}{\theta_n}\Big\| z_n - \tfrac{\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n} y_n - \tfrac{(2-\gamma_n\beta)\lambda_n}{\hat{\theta}_n} x_n + (\bar{\alpha}_n - \alpha_n)p_{n-1} - \bar{\alpha}_n z_{n-1} + \tfrac{\alpha_n\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n} y_{n-1}\Big\|_M^2 \\ &= \Big(\lambda_n^2 - \tfrac{(4-\gamma_n\beta)(\lambda_n+\mu_n)}{2} + \tfrac{1}{2}\theta_n\Big)\|p_n\|_M^2 \\ &\quad + 2\Big\langle p_n, (\lambda_n + \mu_n - \lambda_n^2 - \bar{\theta}_n)z_n + \Big(\tfrac{\bar{\theta}_n}{2} - \tfrac{(\lambda_n+\mu_n)\gamma_n\beta}{2}\Big)y_n\Big\rangle_M \\ &\quad + 2\Big\langle p_n, (\lambda_n - \lambda_n)x_n + \Big((1-\lambda_n)\lambda_n\bar{\alpha}_n + \mu_n - \omega_n - \tfrac{1}{2}\theta_n' - \tfrac{\mu_n\gamma_n\beta}{2}\Big)p_{n-1}\Big\rangle_M \end{aligned}$$

$$
\begin{aligned}
&+2\Big\langle p_n, \big(\lambda_n^2\bar{\alpha}_n - \lambda_n\bar{\alpha}_n + \omega_n + \bar{\theta}_n\bar{\alpha}_n\big)z_{n-1} + \Big(-\tfrac{1}{2}\tilde{\theta}_n\alpha_n + \tfrac{\mu_n\gamma_n\beta}{2}\Big)y_{n-1}\Big\rangle_M \\
&+\lambda_n^2\|z_n - \bar{\alpha}_n z_{n-1} + \bar{\alpha}_n p_{n-1}\|_M^2 + 2\langle x_n, \lambda_n(-z_n - \bar{\alpha}_n p_{n-1} + \bar{\alpha}_n z_{n-1})\rangle_M \\
&+2\langle \mu_n z_n + \omega_n(z_{n-1} - p_{n-1}), -p_{n-1}\rangle_M \\
&+\tfrac{1}{2}\theta_n\Big\|-\tfrac{2\bar{\theta}_n}{\theta_n}z_n + \tfrac{\tilde{\theta}_n}{\theta_n}y_n - \tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} + \tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n}z_{n-1} - \tfrac{\tilde{\theta}_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2 \\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|-y_n - (p_{n-1} - y_{n-1})\|_M^2 - \tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1} - p_{n-1}\|_M^2 \\
&+\tfrac{\lambda_n\gamma_n\beta}{2}\|y_n\|_M^2 - \tfrac{\tilde{\theta}_n(\lambda_n+\mu_n)}{\hat{\theta}_n}\|y_n - (1-\alpha_n)x_n - \alpha_n y_{n-1}\|_M^2 \\
&-\tfrac{\hat{\theta}_n(\lambda_n+\mu_n)}{\theta_n}\Big\|z_n - \tfrac{\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n}y_n - \tfrac{(2-\gamma_n\beta)\lambda_n}{\hat{\theta}_n}x_n \\
&\qquad\qquad + (\bar{\alpha}_n - \alpha_n)p_{n-1} - \bar{\alpha}_n z_{n-1} + \tfrac{\alpha_n\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n}y_{n-1}\Big\|_M^2.
\end{aligned}
$$

All terms involving p_n in this expression are identically zero since their coefficients become zero. This is for most terms straightforward to show by substituting θ_n, $\bar{\theta}_n$, $\tilde{\theta}_n$, θ_n', α_n, and $\bar{\alpha}_n$ defined in Algorithm 1 into the corresponding coefficients. We show this for two coefficients for which it is less obvious. For the coefficient of $\langle p_n, p_{n-1}\rangle_M$ we have

$$
\begin{aligned}
&-\lambda_n^2\bar{\alpha}_n + \lambda_n\bar{\alpha}_n + \mu_n - \omega_n - \tfrac{1}{2}\theta_n' - \tfrac{\mu_n\gamma_n\beta}{2} \\
&\quad= -\lambda_n^2\bar{\alpha}_n + \lambda_n\bar{\alpha}_n + \mu_n - \Big(\lambda_n\bar{\alpha}_n - \tfrac{\gamma_n}{\gamma_{n-1}}\mu_n\Big) - \tfrac{1}{2}\theta_n' - \tfrac{\mu_n\gamma_n\beta}{2} \\
&\quad= -\lambda_n^2\bar{\alpha}_n + \tfrac{\gamma_n}{\gamma_{n-1}}\mu_n + \tfrac{(2-\gamma_n\beta)\mu_n}{2} - \tfrac{1}{2}\theta_n' \\
&\quad= -\lambda_n^2\bar{\alpha}_n + (\lambda_n+\mu_n)\bar{\alpha}_n + \tfrac{(2-\gamma_n\beta)\mu_n}{2} - \tfrac{1}{2}\theta_n' \\
&\quad= \bar{\theta}_n\bar{\alpha}_n + \tfrac{(2-\gamma_n\beta)\mu_n}{2} - \tfrac{1}{2}\theta_n' = \tfrac{1}{2}\theta_n' - \tfrac{1}{2}\theta_n' = 0,
\end{aligned}
$$

where in the first equality ω_n is substituted from (II.39) and in the third equality the definition of $\bar{\alpha}_n$ is used. For the coefficient of $\langle p_n, z_{n-1}\rangle_M$

$$
\begin{aligned}
&\lambda_n^2\bar{\alpha}_n - \lambda_n\bar{\alpha}_n + \omega_n + \bar{\theta}_n\bar{\alpha}_n \\
&\quad= \lambda_n^2\bar{\alpha}_n - \lambda_n\bar{\alpha}_n + \lambda_n\bar{\alpha}_n - \tfrac{\gamma_n}{\gamma_{n-1}}\mu_n + \bar{\theta}_n\bar{\alpha}_n \\
&\quad= \lambda_n^2\bar{\alpha}_n - (\lambda_n+\mu_n)\bar{\alpha}_n + \bar{\theta}_n\bar{\alpha}_n = (\bar{\theta}_n - \bar{\theta}_n)\bar{\alpha}_n = 0.
\end{aligned}
$$

Next, for the terms containing z_n, we do a similar procedure of expanding, reordering, and recollecting the terms as we did for p_n. This gives

$$
\begin{aligned}
\Delta_n = &\lambda_n^2\|z_n - \bar{\alpha}_n z_{n-1} + \bar{\alpha}_n p_{n-1}\|_M^2 + 2\langle x_n, \lambda_n(-z_n - \bar{\alpha}_n p_{n-1} + \bar{\alpha}_n z_{n-1})\rangle_M \\
&+2\langle \mu_n z_n + \omega_n(z_{n-1} - p_{n-1}), -p_{n-1}\rangle_M \\
&+\tfrac{1}{2}\theta_n\Big\|-\tfrac{2\bar{\theta}_n}{\theta_n}z_n + \tfrac{\tilde{\theta}_n}{\theta_n}y_n - \tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} + \tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n}z_{n-1} - \tfrac{\tilde{\theta}_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2
\end{aligned}
$$

$$
\begin{aligned}
&+\tfrac{\mu_n\gamma_n\beta}{2}\|-y_n-(p_{n-1}-y_{n-1})\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&+\tfrac{\lambda_n\gamma_n\beta}{2}\|y_n\|_M^2-\tfrac{\tilde\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}\|y_n-(1-\alpha_n)x_n-\alpha_n y_{n-1}\|_M^2\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big\|z_n-\tfrac{\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_n-\tfrac{(2-\gamma_n\beta)\lambda_n}{\hat\theta_n}x_n\\
&\qquad\qquad+(\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big\|_M^2\\
=\;&\lambda_n^2\|z_n\|_M^2+2\big\langle z_n,\lambda_n^2(-\bar\alpha_n z_{n-1}+\bar\alpha_n p_{n-1})\big\rangle_M+\lambda_n^2\|-\bar\alpha_n z_{n-1}+\bar\alpha_n p_{n-1}\|_M^2\\
&+2\langle z_n,-\lambda_n x_n\rangle_M+2\langle x_n,\lambda_n(-\bar\alpha_n p_{n-1}+\bar\alpha_n z_{n-1})\rangle_M\\
&+2\langle z_n,-\mu_n p_{n-1}\rangle_M+2\langle p_{n-1},-\omega_n(z_{n-1}-p_{n-1})\rangle_M+\tfrac{2\bar\theta_n^2}{\theta_n}\|z_n\|_M^2\\
&+2\Big\langle z_n,-\bar\theta_n\Big(\tfrac{\tilde\theta_n}{\theta_n}y_n-\tfrac{2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'}{\theta_n}p_{n-1}+\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}-\tfrac{\tilde\theta_n\alpha_n}{\theta_n}y_{n-1}\Big)\Big\rangle_M\\
&+\tfrac12\theta_n\Big\|\tfrac{\tilde\theta_n}{\theta_n}y_n-\tfrac{2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'}{\theta_n}p_{n-1}+\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}-\tfrac{\tilde\theta_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|-y_n-(p_{n-1}-y_{n-1})\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&+\tfrac{\lambda_n\gamma_n\beta}{2}\|y_n\|_M^2-\tfrac{\tilde\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}\|y_n-(1-\alpha_n)x_n-\alpha_n y_{n-1}\|_M^2\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\|z_n\|_M^2+2\Big\langle z_n,-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big(-\tfrac{\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_n-\tfrac{(2-\gamma_n\beta)\lambda_n}{\hat\theta_n}x_n\Big)\Big\rangle_M\\
&+2\Big\langle z_n,-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big((\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big)\Big\rangle_M\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big\|-\tfrac{\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_n-\tfrac{(2-\gamma_n\beta)\lambda_n}{\hat\theta_n}x_n\\
&\qquad\qquad+(\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big\|_M^2\\
=\;&\tfrac{\lambda_n^2\theta_n+2\bar\theta_n^2-\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\|z_n\|_M^2+2\Big\langle z_n,\tfrac{\bar\theta_n\big(\gamma_n\beta(\lambda_n+\mu_n)-\tilde\theta_n\big)}{\theta_n}y_n\Big\rangle_M\\
&+2\Big\langle z_n,\tfrac{2\bar\theta_n\lambda_n-\lambda_n\theta_n+(2-\gamma_n\beta)(\lambda_n+\mu_n)\lambda_n}{\theta_n}x_n\Big\rangle_M\\
&+2\Big\langle z_n,\tfrac{\lambda_n^2\bar\alpha_n\theta_n-\mu_n\theta_n+\bar\theta_n\theta_n'-\hat\theta_n(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)}{\theta_n}p_{n-1}\Big\rangle_M\\
&+2\Big\langle z_n,\tfrac{\bar\alpha_n\big(\hat\theta_n(\lambda_n+\mu_n)-2\bar\theta_n^2-\lambda_n^2\theta_n\big)}{\theta_n}z_{n-1}+\tfrac{\alpha_n\bar\theta_n\big(\tilde\theta_n-\gamma_n\beta(\lambda_n+\mu_n)\big)}{\theta_n}y_{n-1}\Big\rangle_M\\
&+\lambda_n^2\|-\bar\alpha_n z_{n-1}+\bar\alpha_n p_{n-1}\|_M^2+2\langle x_n,\lambda_n(-\bar\alpha_n p_{n-1}+\bar\alpha_n z_{n-1})\rangle_M\\
&+2\langle p_{n-1},-\omega_n(z_{n-1}-p_{n-1})\rangle_M\\
&+\tfrac12\theta_n\Big\|\tfrac{\tilde\theta_n}{\theta_n}y_n-\tfrac{2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'}{\theta_n}p_{n-1}+\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}-\tfrac{\tilde\theta_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|-y_n-p_{n-1}+y_{n-1}\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&+\tfrac{\lambda_n\gamma_n\beta}{2}\|y_n\|_M^2-\tfrac{\tilde\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}\|y_n-(1-\alpha_n)x_n-\alpha_n y_{n-1}\|_M^2
\end{aligned}
$$

$$- \tfrac{\hat{\theta}_n(\lambda_n+\mu_n)}{\theta_n}\Big\| - \tfrac{\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n}y_n - \tfrac{(2-\gamma_n\beta)\lambda_n}{\hat{\theta}_n}x_n + (\bar{\alpha}_n-\alpha_n)p_{n-1} - \bar{\alpha}_n z_{n-1} + \tfrac{\alpha_n\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n}y_{n-1}\Big\|_M^2$$

Now, we show that all the coefficients of the terms containing z_n are identical to zero. The coefficients of $\|z_n\|_M^2$ and $\langle z_n, z_{n-1}\rangle_M$ are zero by Proposition 4 (iii).For the coefficient of $\langle z_n, x_n\rangle_M$ we have

$$2\bar{\theta}_n\lambda_n - \lambda_n\theta_n + (2-\gamma_n\beta)(\lambda_n+\mu_n)\lambda_n = \lambda_n\big(2\bar{\theta}_n + (2-\gamma_n\beta)(\lambda_n+\mu_n) - \theta_n\big)$$

which is identical to zero by Proposition 4 (ii). For the coefficient of $\langle z_n, p_{n-1}\rangle_M$ we have

$$\begin{aligned}
&\lambda_n^2\bar{\alpha}_n\theta_n - \mu_n\theta_n + \bar{\theta}_n\theta_n' - \hat{\theta}_n(\lambda_n+\mu_n)(\bar{\alpha}_n-\alpha_n)\\
&\quad= \lambda_n^2\bar{\alpha}_n\theta_n - \mu_n\theta_n + \bar{\theta}_n\big((2-\gamma_n\beta)\mu_n + 2\bar{\alpha}_n\bar{\theta}_n\big) - \hat{\theta}_n(\lambda_n+\mu_n)(\bar{\alpha}_n-\alpha_n)\\
&\quad= \bar{\alpha}_n\big(\lambda_n^2\theta_n + 2\bar{\theta}_n^2 - (\lambda_n+\mu_n)\hat{\theta}_n\big) - \mu_n\theta_n\\
&\qquad+ (2-\gamma_n\beta)\mu_n\bar{\theta}_n + (\lambda_n+\mu_n)\alpha_n\hat{\theta}_n\\
&\quad= \big(-\theta_n + (2-\gamma_n\beta)\bar{\theta}_n + \hat{\theta}_n\big)\mu_n = 0
\end{aligned}$$

where in the first equality, θ_n' is substituted and in the third equality Proposition 4 (iii) is used and in the last equality Proposition 4 (i) is used. Therefore, all terms containing z_n can be eliminated from Δ_n and we are left with

$$\begin{aligned}
\Delta_n = {}&\\
&\lambda_n^2\|-\bar{\alpha}_n z_{n-1} + \bar{\alpha}_n p_{n-1}\|_M^2 + 2\langle x_n, \lambda_n(-\bar{\alpha}_n p_{n-1} + \bar{\alpha}_n z_{n-1})\rangle_M\\
&+ 2\langle p_{n-1}, -\omega_n(z_{n-1}-p_{n-1})\rangle_M\\
&+ \tfrac{1}{2}\theta_n\Big\|\tfrac{\bar{\theta}_n}{\theta_n}y_n - \tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} + \tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n}z_{n-1} - \tfrac{\bar{\theta}_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2\\
&+ \tfrac{\mu_n\gamma_n\beta}{2}\|-y_n - p_{n-1} + y_{n-1}\|_M^2 - \tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&+ \tfrac{\lambda_n\gamma_n\beta}{2}\|y_n\|_M^2 - \tfrac{\tilde{\theta}_n(\lambda_n+\mu_n)}{\hat{\theta}_n}\|y_n - (1-\alpha_n)x_n - \alpha_n y_{n-1}\|_M^2\\
&- \tfrac{\hat{\theta}_n(\lambda_n+\mu_n)}{\theta_n}\Big\| - \tfrac{\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n}y_n - \tfrac{(2-\gamma_n\beta)\lambda_n}{\hat{\theta}_n}x_n\\
&\qquad\qquad + (\bar{\alpha}_n-\alpha_n)p_{n-1} - \bar{\alpha}_n z_{n-1} + \tfrac{\alpha_n\bar{\theta}_n\gamma_n\beta}{\hat{\theta}_n}y_{n-1}\Big\|_M^2\\
= {}&\lambda_n^2\|-\bar{\alpha}_n z_{n-1} + \bar{\alpha}_n p_{n-1}\|_M^2 + 2\langle x_n, \lambda_n(-\bar{\alpha}_n p_{n-1} + \bar{\alpha}_n z_{n-1})\rangle_M\\
&+ 2\langle p_{n-1}, -\omega_n(z_{n-1}-p_{n-1})\rangle_M + \tfrac{\tilde{\theta}_n^2}{2\theta_n}\|y_n\|_M^2\\
&+ 2\Big\langle y_n, \tfrac{\tilde{\theta}_n}{2}\Big(-\tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} + \tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n}z_{n-1} - \tfrac{\tilde{\theta}_n\alpha_n}{\theta_n}y_{n-1}\Big)\Big\rangle_M\\
&+ \tfrac{1}{2}\theta_n\Big\| - \tfrac{2\lambda_n}{\theta_n}x_n - \tfrac{\theta_n'}{\theta_n}p_{n-1} + \tfrac{2\bar{\theta}_n\bar{\alpha}_n}{\theta_n}z_{n-1} - \tfrac{\tilde{\theta}_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2
\end{aligned}$$

$$
\begin{aligned}
&+\tfrac{\mu_n\gamma_n\beta}{2}\|y_n\|_M^2+2\Big\langle y_n,\tfrac{\mu_n\gamma_n\beta}{2}(p_{n-1}-y_{n-1})\Big\rangle_M+\tfrac{\mu_n\gamma_n\beta}{2}\|p_{n-1}-y_{n-1}\|_M^2\\
&+\tfrac{\lambda_n\gamma_n\beta}{2}\|y_n\|_M^2-\tfrac{\tilde\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}\|y_n\|_M^2\\
&-2\Big\langle y_n,\tfrac{\tilde\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}((\alpha_n-1)x_n-\alpha_n y_{n-1})\Big\rangle_M\\
&-\tfrac{\tilde\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}\|(\alpha_n-1)x_n-\alpha_n y_{n-1}\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&-\tfrac{\bar\theta_n^2\gamma_n^2\beta^2(\lambda_n+\mu_n)}{\hat\theta_n\theta_n}\|y_n\|_M^2+2\Big\langle y_n,\tfrac{\bar\theta_n\gamma_n\beta(\lambda_n+\mu_n)}{\theta_n}\Big(-\tfrac{(2-\gamma_n\beta)\lambda_n}{\hat\theta_n}x_n\Big)\Big\rangle_M\\
&+2\Big\langle y_n,\tfrac{\bar\theta_n\gamma_n\beta(\lambda_n+\mu_n)}{\theta_n}\Big((\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big)\Big\rangle_M\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big\|-\tfrac{(2-\gamma_n\beta)\lambda_n}{\hat\theta_n}x_n+(\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big\|_M^2\\
=&\Big(\tfrac{\tilde\theta_n^2}{2\theta_n}+\tfrac{\mu_n\gamma_n\beta}{2}+\tfrac{\lambda_n\gamma_n\beta}{2}-\tfrac{\tilde\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}-\tfrac{\bar\theta_n^2\gamma_n^2\beta^2(\lambda_n+\mu_n)}{\hat\theta_n\theta_n}\Big)\|y_n\|_M^2\\
&+2\Big\langle y_n,\Big(-\tfrac{\lambda_n\tilde\theta_n}{\theta_n}+\tfrac{\tilde\theta_n(\lambda_n+\mu_n)(1-\alpha_n)}{\hat\theta_n}-\tfrac{\bar\theta_n\lambda_n\gamma_n\beta(\lambda_n+\mu_n)(2-\gamma_n\beta)}{\theta_n\hat\theta_n}\Big)x_n\Big\rangle_M\\
&+2\Big\langle y_n,\Big(-\tfrac{\tilde\theta_n\theta_n'}{2\theta_n}+\tfrac{\mu_n\gamma_n\beta}{2}+\tfrac{\bar\theta_n\gamma_n\beta(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)}{\theta_n}\Big)p_{n-1}\Big\rangle_M\\
&+2\Big\langle y_n,\Big(\tfrac{\tilde\theta_n\bar\theta_n\bar\alpha_n}{\theta_n}-\tfrac{\tilde\theta_n\bar\alpha_n\gamma_n\beta(\lambda_n+\mu_n)}{\theta_n}\Big)z_{n-1}\Big\rangle_M\\
&+2\Big\langle y_n,\Big(-\tfrac{\tilde\theta_n^2\alpha_n}{2\theta_n}-\tfrac{\mu_n\gamma_n\beta}{2}+\tfrac{\tilde\theta_n\alpha_n(\lambda_n+\mu_n)}{\hat\theta_n}+\tfrac{\alpha_n\bar\theta_n^2\gamma_n^2\beta^2(\lambda_n+\mu_n)}{\theta_n\hat\theta_n}\Big)y_{n-1}\Big\rangle_M\\
&+\lambda_n^2\|-\bar\alpha_n z_{n-1}+\bar\alpha_n p_{n-1}\|_M^2+2\langle x_n,\lambda_n(-\bar\alpha_n p_{n-1}+\bar\alpha_n z_{n-1})\rangle_M\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_{n-1}-y_{n-1}\|_M^2+2\langle p_{n-1},-\omega_n(z_{n-1}-p_{n-1})\rangle_M\\
&+\tfrac{1}{2}\theta_n\Big\|-\tfrac{2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'}{\theta_n}p_{n-1}+\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}-\tfrac{\tilde\theta_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2\\
&-\tfrac{\tilde\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}\|(\alpha_n-1)x_n-\alpha_n y_{n-1}\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big\|-\tfrac{(2-\gamma_n\beta)\lambda_n}{\hat\theta_n}x_n+(\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big\|_M^2
\end{aligned}
$$

Now, we show that all the coefficients of the terms containing y_n are identically zero. For the coefficient of $\|y_n\|_M^2$ we have

$$
\begin{aligned}
&\big(\theta_n\hat\theta_n\gamma_n\beta-2\theta_n\tilde\theta_n-2\bar\theta_n^2\gamma_n^2\beta^2\big)(\lambda_n+\mu_n)+\tilde\theta_n^2\hat\theta_n\\
&\quad=\big(\theta_n\gamma_n\beta(2\lambda_n+2\mu_n-\lambda_n^2\gamma_n\beta)-2\theta_n\gamma_n\beta(\lambda_n+\mu_n)-2\bar\theta_n^2\gamma_n^2\beta^2\big)(\lambda_n+\mu_n)\\
&\qquad+\tilde\theta_n^2\hat\theta_n\\
&\quad=-\big(\theta_n\lambda_n^2+2\bar\theta_n^2\big)(\lambda_n+\mu_n)\gamma_n^2\beta^2+(\lambda_n+\mu_n)^2\gamma_n^2\beta^2\hat\theta_n\\
&\quad=\big((\lambda_n+\mu_n)\hat\theta_n-\theta_n\lambda_n^2-2\bar\theta_n^2\big)(\lambda_n+\mu_n)\gamma_n^2\beta^2,
\end{aligned}
$$

which, by Proposition 4 (iii), is identical to zero. Now, for the coefficient of $\langle y_n, x_n \rangle_M$ we have

$$\begin{aligned}
&\tilde{\theta}_n \theta_n (\lambda_n + \mu_n)(1-\alpha_n) - \lambda_n \tilde{\theta}_n \hat{\theta}_n - \bar{\theta}_n \lambda_n \gamma_n \beta (\lambda_n + \mu_n)(2 - \gamma_n \beta) \\
&\quad = \big(\theta_n (\lambda_n + \mu_n)(1-\alpha_n) - \lambda_n \hat{\theta}_n - \bar{\theta}_n \lambda_n (2 - \gamma_n \beta)\big) \tilde{\theta}_n \\
&\quad = \big(\theta_n - \hat{\theta}_n - \bar{\theta}_n (2 - \gamma_n \beta)\big) \lambda_n \tilde{\theta}_n
\end{aligned}$$

which is identically zero by Proposition 4 (i). For the coefficient of $\langle y_n, p_{n-1} \rangle_M$ we have

$$\begin{aligned}
&\mu_n \gamma_n \beta \theta_n + 2\bar{\theta}_n \gamma_n \beta (\lambda_n + \mu_n)(\bar{\alpha}_n - \alpha_n) - \tilde{\theta}_n \theta_n' \\
&\quad = \mu_n \gamma_n \beta \theta_n + 2\bar{\theta}_n \tilde{\theta}_n (\bar{\alpha}_n - \alpha_n) - \tilde{\theta}_n \big((2 - \gamma_n \beta)\mu_n + 2\bar{\alpha}_n \bar{\theta}_n\big) \\
&\quad = \mu_n \gamma_n \beta \theta_n + 2\bar{\theta}_n \tilde{\theta}_n \bar{\alpha}_n - 2\bar{\theta}_n \tilde{\theta}_n \alpha_n - (2 - \gamma_n \beta)\mu_n \tilde{\theta}_n - 2\bar{\alpha}_n \bar{\theta}_n \tilde{\theta}_n \\
&\quad = \mu_n \gamma_n \beta \theta_n - 2\bar{\theta}_n \tilde{\theta}_n \alpha_n - (2 - \gamma_n \beta)\mu_n \tilde{\theta}_n \\
&\quad = \mu_n \gamma_n \beta \theta_n - 2\bar{\theta}_n \mu_n \gamma_n \beta - (2 - \gamma_n \beta)(\lambda_n + \mu_n)\mu_n \gamma_n \beta \\
&\quad = \mu_n \gamma_n \beta \big(\theta_n - 2\bar{\theta}_n - (2 - \gamma_n \beta)(\lambda_n + \mu_n)\big) \\
&\quad = \mu_n \gamma_n \beta \big(\theta_n - 2\big(\lambda_n + \mu_n - \lambda_n^2\big) - (2 - \gamma_n \beta)(\lambda_n + \mu_n)\big) \\
&\quad = \mu_n \gamma_n \beta \big(\theta_n + 2\lambda_n^2 - (4 - \gamma_n \beta)(\lambda_n + \mu_n)\big)
\end{aligned}$$

which is equal to zero by the definition of θ_n. The equivalence of the coefficient of $\langle y_n, z_{n-1} \rangle_M$ to zero follows from the definition of $\tilde{\theta}_n$. For the coefficient of $\langle y_n, y_{n-1} \rangle_M$ we have

$$\begin{aligned}
&2\alpha_n \bar{\theta}_n^2 \gamma_n^2 \beta^2 (\lambda_n + \mu_n) - \tilde{\theta}_n^2 \hat{\theta}_n \alpha_n + 2\theta_n \tilde{\theta}_n \alpha_n (\lambda_n + \mu_n) - \mu_n \gamma_n \beta \theta_n \hat{\theta}_n \\
&\quad = 2\alpha_n \bar{\theta}_n^2 \gamma_n \beta \tilde{\theta}_n - \tilde{\theta}_n^2 \hat{\theta}_n \alpha_n + 2\theta_n \tilde{\theta}_n \alpha_n (\lambda_n + \mu_n) - \alpha_n \tilde{\theta}_n \theta_n \hat{\theta}_n \\
&\quad = \alpha_n \tilde{\theta}_n \big(2\bar{\theta}_n^2 \gamma_n \beta - \tilde{\theta}_n \hat{\theta}_n + 2\theta_n (\lambda_n + \mu_n) - \theta_n \hat{\theta}_n\big) \\
&\quad = \alpha_n \tilde{\theta}_n \big(2\bar{\theta}_n^2 \gamma_n \beta - \tilde{\theta}_n \hat{\theta}_n + \theta_n \big(2(\lambda_n + \mu_n) - 2(\lambda_n + \mu_n) + \lambda_n^2 \gamma_n \beta\big)\big) \\
&\quad = \alpha_n \tilde{\theta}_n \big(2\bar{\theta}_n^2 \gamma_n \beta - \tilde{\theta}_n \hat{\theta}_n + \theta_n \lambda_n^2 \gamma_n \beta\big) \\
&\quad = \alpha_n \tilde{\theta}_n \big(2\bar{\theta}_n^2 \gamma_n \beta - \hat{\theta}_n \gamma_n \beta (\lambda_n + \mu_n) + \theta_n \lambda_n^2 \gamma_n \beta\big) \\
&\quad = \alpha_n \tilde{\theta}_n \gamma_n \beta \big(2\bar{\theta}_n^2 - \hat{\theta}_n (\lambda_n + \mu_n) + \theta_n \lambda_n^2\big)
\end{aligned}$$

which by Proposition 4 (iii) is identical to zero. Therefore, all the coefficients of the terms containing y_n are zero and we can eliminate those terms. The remaining terms are

$$\begin{aligned}
&\Delta_n = \\
&\quad \lambda_n^2 \| -\bar{\alpha}_n z_{n-1} + \bar{\alpha}_n p_{n-1} \|_M^2 + 2\langle x_n, \lambda_n (-\bar{\alpha}_n p_{n-1} + \bar{\alpha}_n z_{n-1}) \rangle_M \\
&\quad + \tfrac{\mu_n \gamma_n \beta}{2} \| p_{n-1} - y_{n-1} \|_M^2 + 2\langle p_{n-1}, -\omega_n (z_{n-1} - p_{n-1}) \rangle_M
\end{aligned}$$

$$
\begin{aligned}
&+\tfrac{1}{2}\theta_n\Big\|-\tfrac{2\lambda_n}{\theta_n}x_n-\tfrac{\theta_n'}{\theta_n}p_{n-1}+\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}-\tfrac{\bar\theta_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2\\
&-\tfrac{\bar\theta_n(\lambda_n+\mu_n)}{\hat\theta_n}\|(\alpha_n-1)x_n-\alpha_n y_{n-1}\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big\|-\tfrac{(2-\gamma_n\beta)\lambda_n}{\hat\theta_n}x_n+(\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big\|_M^2\\
=\;&\lambda_n^2\|-\bar\alpha_n z_{n-1}+\bar\alpha_n p_{n-1}\|_M^2+2\langle x_n,\lambda_n(-\bar\alpha_n p_{n-1}+\bar\alpha_n z_{n-1})\rangle_M\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_{n-1}-y_{n-1}\|_M^2+2\langle p_{n-1},-\omega_n(z_{n-1}-p_{n-1})\rangle_M\\
&+\tfrac{2\lambda_n^2}{\theta_n}\|x_n\|_M^2+2\Big\langle x_n,\lambda_n\Big(\tfrac{\theta_n'}{\theta_n}p_{n-1}-\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}+\tfrac{\bar\theta_n\alpha_n}{\theta_n}y_{n-1}\Big)\Big\rangle_M\\
&+\tfrac{1}{2}\theta_n\Big\|\tfrac{\theta_n'}{\theta_n}p_{n-1}-\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}+\tfrac{\bar\theta_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&-\tfrac{\bar\theta_n(\lambda_n+\mu_n)(1-\alpha_n)^2}{\hat\theta_n}\|x_n\|_M^2+2\Big\langle x_n,\tfrac{\bar\theta_n\alpha_n(\lambda_n+\mu_n)(\alpha_n-1)}{\hat\theta_n}y_{n-1}\Big\rangle_M\\
&-\tfrac{(\lambda_n+\mu_n)(2-\gamma_n\beta)^2\lambda_n^2}{\theta_n\hat\theta_n}\|x_n\|_M^2-\tfrac{\bar\theta_n\alpha_n^2(\lambda_n+\mu_n)}{\hat\theta_n}\|y_{n-1}\|_M^2\\
&+2\Big\langle x_n,\tfrac{\lambda_n(2-\gamma_n\beta)(\lambda_n+\mu_n)}{\theta_n}\Big((\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big)\Big\rangle_M\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big\|(\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big\|_M^2\\
=\;&\Big(\tfrac{2\lambda_n^2}{\theta_n}-\tfrac{\bar\theta_n(\lambda_n+\mu_n)(1-\alpha_n)^2}{\hat\theta_n}-\tfrac{(\lambda_n+\mu_n)(2-\gamma_n\beta)^2\lambda_n^2}{\theta_n\hat\theta_n}\Big)\|x_n\|_M^2\\
&+2\Big\langle x_n,\Big(\tfrac{\lambda_n\theta_n'}{\theta_n}+\tfrac{\lambda_n(2-\gamma_n\beta)(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)}{\theta_n}-\lambda_n\bar\alpha_n\Big)p_{n-1}\Big\rangle_M\\
&+2\Big\langle x_n,\Big(-\tfrac{2\lambda_n\bar\theta_n\bar\alpha_n}{\theta_n}-\tfrac{\lambda_n\bar\alpha_n(2-\gamma_n\beta)(\lambda_n+\mu_n)}{\theta_n}+\lambda_n\bar\alpha_n\Big)z_{n-1}\Big\rangle_M\\
&+2\Big\langle x_n,\Big(\tfrac{\lambda_n\bar\theta_n\alpha_n}{\theta_n}+\tfrac{\bar\theta_n\alpha_n(\lambda_n+\mu_n)(\alpha_n-1)}{\hat\theta_n}+\tfrac{\bar\theta_n\alpha_n\lambda_n\gamma_n\beta(2-\gamma_n\beta)(\lambda_n+\mu_n)}{\theta_n\hat\theta_n}\Big)y_{n-1}\Big\rangle_M\\
&+\lambda_n^2\|-\bar\alpha_n z_{n-1}+\bar\alpha_n p_{n-1}\|_M^2-\tfrac{\bar\theta_n\alpha_n^2(\lambda_n+\mu_n)}{\hat\theta_n}\|y_{n-1}\|_M^2\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_{n-1}-y_{n-1}\|_M^2+2\langle p_{n-1},-\omega_n(z_{n-1}-p_{n-1})\rangle_M\\
&+\tfrac{1}{2}\theta_n\Big\|\tfrac{\theta_n'}{\theta_n}p_{n-1}-\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}+\tfrac{\bar\theta_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big\|(\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big\|_M^2
\end{aligned}
$$

We want to show that all the coefficients of the terms containing x_n are zero. For the coefficient of $\|x_n\|_M^2$ we have

$$\begin{aligned}
&2\lambda_n^2\hat{\theta}_n - (\lambda_n+\mu_n)(2-\gamma_n\beta)^2\lambda_n^2 - \theta_n\tilde{\theta}_n(\lambda_n+\mu_n)(1-\alpha_n)^2 \\
&\quad= 2\lambda_n^2\hat{\theta}_n - (\lambda_n+\mu_n)(2-\gamma_n\beta)^2\lambda_n^2 - \theta_n\lambda_n^2\gamma_n\beta \\
&\quad= \lambda_n^2\Big(2\hat{\theta}_n - (\lambda_n+\mu_n)(2-\gamma_n\beta)^2 - \theta_n\gamma_n\beta\Big) \\
&\quad= \lambda_n^2\big(2\big(2\lambda_n+2\mu_n-\gamma_n\beta\lambda_n^2\big) - (\lambda_n+\mu_n)\big(4-4\gamma_n\beta+\gamma_n^2\beta^2\big) - \theta_n\gamma_n\beta\big) \\
&\quad= \lambda_n^2\Bigg(-2\gamma_n\beta\lambda_n^2 - (\lambda_n+\mu_n)\big(-4\gamma_n\beta+\gamma_n^2\beta^2\big) \\
&\qquad\qquad - \big((4-\gamma_n\beta)(\lambda_n+\mu_n) - 2\lambda_n^2\big)\gamma_n\beta\Bigg) \\
&\quad= \lambda_n^2\big(-2\gamma_n\beta\lambda_n^2 + 2\lambda_n^2\gamma_n\beta\big) = 0
\end{aligned} \tag{II.40}$$

where in the first equality $\tilde{\theta}_n$ and α_n and in the third equality $\hat{\theta}_n$ are substituted by their definition from Algorithm 1. For the coefficient of $\langle x_n, p_{n-1}\rangle$ we have

$$\begin{aligned}
&\lambda_n\theta_n' + \lambda_n(2-\gamma_n\beta)(\lambda_n+\mu_n)(\bar{\alpha}_n-\alpha_n) - \lambda_n\bar{\alpha}_n\theta_n \\
&\quad= \lambda_n\big((2-\gamma_n\beta)\mu_n + 2\bar{\alpha}_n\bar{\theta}_n\big) + \lambda_n(2-\gamma_n\beta)(\lambda_n+\mu_n)(\bar{\alpha}_n-\alpha_n) - \lambda_n\bar{\alpha}_n\theta_n \\
&\quad= 2\lambda_n\bar{\alpha}_n\bar{\theta}_n + (2-\gamma_n\beta)(\lambda_n+\mu_n)\lambda_n\bar{\alpha}_n - \lambda_n\bar{\alpha}_n\theta_n \\
&\qquad - (2-\gamma_n\beta)(\lambda_n+\mu_n)\lambda_n\alpha_n + (2-\gamma_n\beta)\lambda_n\mu_n \\
&\quad= \lambda_n\bar{\alpha}_n\big(2\bar{\theta}_n + (2-\gamma_n\beta)(\lambda_n+\mu_n) - \theta_n\big) \\
&\qquad - (2-\gamma_n\beta)\lambda_n\mu_n + (2-\gamma_n\beta)\lambda_n\mu_n
\end{aligned}$$

which by Proposition 4 (ii) is zero. For the coefficient of $\langle x_n, z_{n-1}\rangle$ we have

$$\begin{aligned}
&\lambda_n\bar{\alpha}_n\theta_n - 2\lambda_n\bar{\theta}_n\bar{\alpha}_n - \lambda_n\bar{\alpha}_n(2-\gamma_n\beta)(\lambda_n+\mu_n) \\
&\qquad= \lambda_n\bar{\alpha}_n\big(\theta_n - 2\bar{\theta}_n - (2-\gamma_n\beta)(\lambda_n+\mu_n)\big)
\end{aligned}$$

which is identically zero by Proposition 4 (ii). For the coefficient of $\langle x_n, y_{n-1}\rangle$ we have

$$\begin{aligned}
&\lambda_n\tilde{\theta}_n\alpha_n\hat{\theta}_n + \bar{\theta}_n\alpha_n\lambda_n\gamma_n\beta(2-\gamma_n\beta)(\lambda_n+\mu_n) - \theta_n\tilde{\theta}_n\alpha_n(\lambda_n+\mu_n)(1-\alpha_n) \\
&\quad= \lambda_n\tilde{\theta}_n\alpha_n\hat{\theta}_n + \bar{\theta}_n\alpha_n\lambda_n\tilde{\theta}_n(2-\gamma_n\beta) - \theta_n\tilde{\theta}_n\alpha_n\lambda_n \\
&\quad= \lambda_n\tilde{\theta}_n\alpha_n\big(\hat{\theta}_n + \bar{\theta}_n(2-\gamma_n\beta) - \theta_n\big)
\end{aligned}$$

which by Proposition 4 (i) is identically zero. Now, expanding all the remaining terms, reordering and recollecting them give

$$\Delta_n =$$

$$
\begin{aligned}
&\lambda_n^2\|-\bar\alpha_n z_{n-1}+\bar\alpha_n p_{n-1}\|_M^2-\tfrac{\tilde\theta_n\alpha_n^2(\lambda_n+\mu_n)}{\hat\theta_n}\|y_{n-1}\|_M^2\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_{n-1}-y_{n-1}\|_M^2+2\langle p_{n-1},-\omega_n(z_{n-1}-p_{n-1})\rangle_M\\
&+\tfrac12\theta_n\Big\|\tfrac{\theta_n'}{\theta_n}p_{n-1}-\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}+\tfrac{\tilde\theta_n\alpha_n}{\theta_n}y_{n-1}\Big\|_M^2-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}-p_{n-1}\|_M^2\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)}{\theta_n}\Big\|(\bar\alpha_n-\alpha_n)p_{n-1}-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big\|_M^2\\
&=\lambda_n^2\bar\alpha_n^2\|p_{n-1}\|_M^2+2\langle p_{n-1},-\lambda_n^2\bar\alpha_n^2 z_{n-1}\rangle_M+\lambda_n^2\bar\alpha_n^2\|z_{n-1}\|_M^2\\
&+\tfrac{\mu_n\gamma_n\beta}{2}\|p_{n-1}\|_M^2+2\Big\langle p_{n-1},-\tfrac{\mu_n\gamma_n\beta}{2}y_{n-1}\Big\rangle_M+\tfrac{\mu_n\gamma_n\beta}{2}\|y_{n-1}\|_M^2\\
&-\tfrac{\tilde\theta_n\alpha_n^2(\lambda_n+\mu_n)}{\hat\theta_n}\|y_{n-1}\|_M^2+2\omega_n\|p_{n-1}\|_M^2+2\langle p_{n-1},-\omega_n z_{n-1}\rangle_M\\
&+\tfrac{\theta_n'^2}{2\theta_n}\|p_{n-1}\|_M^2+2\Big\langle p_{n-1},\tfrac12\theta_n'\Big(-\tfrac{2\bar\theta_n\bar\alpha_n}{\theta_n}z_{n-1}+\tfrac{\tilde\theta_n\alpha_n}{\theta_n}y_{n-1}\Big)\Big\rangle_M\\
&+\tfrac{2\bar\theta_n^2\bar\alpha_n^2}{\theta_n}\|z_{n-1}\|_M^2+2\Big\langle z_{n-1},-\tfrac{\bar\theta_n\bar\alpha_n\tilde\theta_n\alpha_n}{\theta_n}y_{n-1}\Big\rangle_M+\tfrac{\tilde\theta_n^2\alpha_n^2}{2\theta_n}\|y_{n-1}\|_M^2\\
&-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|p_{n-1}\|_M^2+2\Big\langle p_{n-1},\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}y_{n-1}\Big\rangle_M-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}\|y_{n-1}\|_M^2\\
&-\tfrac{\hat\theta_n(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)^2}{\theta_n}\|p_{n-1}\|_M^2-\tfrac{\alpha_n^2\bar\theta_n^2\gamma_n^2\beta^2(\lambda_n+\mu_n)}{\hat\theta_n\theta_n}\|y_{n-1}\|_M^2\\
&-\tfrac{\hat\theta_n\bar\alpha_n^2(\lambda_n+\mu_n)}{\theta_n}\|z_{n-1}\|_M^2+2\Big\langle z_{n-1},\tfrac{\bar\alpha_n\alpha_n\bar\theta_n\gamma_n\beta(\lambda_n+\mu_n)}{\theta_n}y_{n-1}\Big\rangle_M\\
&+2\Big\langle p_{n-1},-\tfrac{\hat\theta_n(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)}{\theta_n}\Big(-\bar\alpha_n z_{n-1}+\tfrac{\alpha_n\bar\theta_n\gamma_n\beta}{\hat\theta_n}y_{n-1}\Big)\Big\rangle_M\\
&=\Big(\lambda_n^2\bar\alpha_n^2+\tfrac{\mu_n\gamma_n\beta}{2}+2\omega_n+\tfrac{\theta_n'^2}{2\theta_n}-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}-\tfrac{\hat\theta_n(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)^2}{\theta_n}\Big)\|p_{n-1}\|_M^2\\
&+2\Big\langle p_{n-1},\Big(-\lambda_n^2\bar\alpha_n^2-\omega_n-\tfrac{\bar\theta_n\theta_n'\bar\alpha_n}{\theta_n}+\tfrac{\hat\theta_n\bar\alpha_n(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)}{\theta_n}\Big)z_{n-1}\Big\rangle_M\\
&+2\Big\langle p_{n-1},\Big(-\tfrac{\mu_n\gamma_n\beta}{2}+\tfrac{\theta_n'\tilde\theta_n\alpha_n}{2\theta_n}+\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}-\tfrac{\alpha_n\bar\theta_n\gamma_n\beta(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)}{\theta_n}\Big)y_{n-1}\Big\rangle_M\\
&+\Big(\lambda_n^2\bar\alpha_n^2+\tfrac{2\bar\theta_n^2\bar\alpha_n^2}{\theta_n}-\tfrac{\hat\theta_n\bar\alpha_n^2(\lambda_n+\mu_n)}{\theta_n}\Big)\|z_{n-1}\|_M^2\\
&+2\Big\langle z_{n-1},\Big(-\tfrac{\bar\theta_n\bar\alpha_n\tilde\theta_n\alpha_n}{\theta_n}+\tfrac{\bar\alpha_n\alpha_n\bar\theta_n\gamma_n\beta(\lambda_n+\mu_n)}{\theta_n}\Big)y_{n-1}\Big\rangle_M\\
&+\Big(\tfrac{\mu_n\gamma_n\beta}{2}-\tfrac{\tilde\theta_n\alpha_n^2(\lambda_n+\mu_n)}{\hat\theta_n}+\tfrac{\tilde\theta_n^2\alpha_n^2}{2\theta_n}-\tfrac{\lambda_n\gamma_n\alpha_n\beta}{2}-\tfrac{\alpha_n^2\bar\theta_n^2\gamma_n^2\beta^2(\lambda_n+\mu_n)}{\hat\theta_n\theta_n}\Big)\|y_{n-1}\|_M^2
\end{aligned}
$$

We show that all the coefficients in the expression above are identically zero. Starting by the coefficient of $\|p_{n-1}\|_M^2$, we have

$$
\begin{aligned}
&2\theta_n\lambda_n^2\bar\alpha_n^2+\theta_n\mu_n\gamma_n\beta+4\theta_n\omega_n+\theta_n'^2-\theta_n\lambda_n\gamma_n\alpha_n\beta-2\hat\theta_n(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)^2\\
&\quad=2\theta_n\lambda_n^2\bar\alpha_n^2+\theta_n\mu_n\gamma_n\beta-4\theta_n\bar\alpha_n\mu_n+\big((2-\gamma_n\beta)\mu_n+2\bar\alpha_n\bar\theta_n\big)^2\\
&\qquad-\theta_n\lambda_n\gamma_n\alpha_n\beta-2\hat\theta_n(\lambda_n+\mu_n)(\bar\alpha_n-\alpha_n)^2
\end{aligned}
$$

$$
\begin{aligned}
&= 2\bar{\alpha}_n^2\big(\theta_n\lambda_n^2 + 2\bar{\theta}_n^2 - \hat{\theta}_n(\lambda_n+\mu_n)\big) + 4\mu_n\bar{\alpha}_n\big(-\theta_n + \bar{\theta}_n(2-\gamma_n\beta) + \hat{\theta}_n\big)\\
&\quad + \theta_n\mu_n\gamma_n\beta + (2-\gamma_n\beta)^2\mu_n^2 - \theta_n\lambda_n\gamma_n\alpha_n\beta - 2\hat{\theta}_n(\lambda_n+\mu_n)\alpha_n^2\\
&= \theta_n\mu_n\gamma_n\beta + (2-\gamma_n\beta)^2\mu_n^2 - \theta_n\lambda_n\gamma_n\alpha_n\beta - 2\hat{\theta}_n\mu_n\alpha_n\\
&= \theta_n\gamma_n\beta(\mu_n - \lambda_n\alpha_n) + (2-\gamma_n\beta)^2\mu_n^2 - 2\hat{\theta}_n\mu_n\alpha_n\\
&= \theta_n\gamma_n\beta\mu_n\alpha_n + (2-\gamma_n\beta)^2(\lambda_n+\mu_n)\alpha_n\mu_n - 2\hat{\theta}_n\mu_n\alpha_n\\
&= \mu_n\alpha_n\Big(\theta_n\gamma_n\beta + (2-\gamma_n\beta)^2(\lambda_n+\mu_n) - 2\hat{\theta}_n\Big)
\end{aligned}
$$

where in the first equality $\omega_n = -\bar{\alpha}_n\mu_n$ is used and θ_n' is substituted from (II.34), the third equality is attained from Proposition 4 (i) and Proposition 4 (iii), and the expression to the right-hand side of the last equality is identically zero by (II.40). For the coefficient of $\langle p_{n-1}, z_{n-1}\rangle_M$ we have

$$
\begin{aligned}
&-\lambda_n^2\bar{\alpha}_n^2\theta_n - \omega_n\theta_n - \bar{\theta}_n\theta_n'\bar{\alpha}_n + \hat{\theta}_n\bar{\alpha}_n(\lambda_n+\mu_n)(\bar{\alpha}_n-\alpha_n)\\
&\quad = -\lambda_n^2\bar{\alpha}_n^2\theta_n - \omega_n\theta_n - \bar{\theta}_n\bar{\alpha}_n\big((2-\gamma_n\beta)\mu_n + 2\bar{\alpha}_n\bar{\theta}_n\big)\\
&\qquad + \hat{\theta}_n\bar{\alpha}_n(\lambda_n+\mu_n)(\bar{\alpha}_n-\alpha_n)\\
&\quad = \bar{\alpha}_n^2\big(-\lambda_n^2\theta_n - 2\bar{\theta}_n^2 + \hat{\theta}_n(\lambda_n+\mu_n)\big) - \omega_n\theta_n - \bar{\theta}_n\bar{\alpha}_n\mu_n(2-\gamma_n\beta)\\
&\qquad - \hat{\theta}_n\bar{\alpha}_n(\lambda_n+\mu_n)\alpha_n\\
&\quad = \bar{\alpha}_n\mu_n\theta_n - \bar{\theta}_n\bar{\alpha}_n\mu_n(2-\gamma_n\beta) - \hat{\theta}_n\bar{\alpha}_n\mu_n\\
&\quad = \bar{\alpha}_n\mu_n\big(\theta_n - \bar{\theta}_n(2-\gamma_n\beta) - \hat{\theta}_n\big)
\end{aligned}
$$

which by Proposition 4 (i) is equal to zero. The third equality above is attained by using Proposition 4 (iii). For the coefficient of $\langle p_{n-1}, y_{n-1}\rangle_M$ we have

$$
\begin{aligned}
&\theta_n'\tilde{\theta}_n\alpha_n - \mu_n\gamma_n\beta\theta_n + \lambda_n\gamma_n\alpha_n\beta\theta_n - 2\alpha_n\bar{\theta}_n\gamma_n\beta(\lambda_n+\mu_n)(\bar{\alpha}_n-\alpha_n)\\
&\quad = \big((2-\gamma_n\beta)\mu_n + 2\bar{\alpha}_n\bar{\theta}_n\big)\tilde{\theta}_n\alpha_n - \mu_n\gamma_n\beta\theta_n + \lambda_n\gamma_n\alpha_n\beta\theta_n\\
&\qquad - 2\alpha_n\bar{\theta}_n\tilde{\theta}_n(\bar{\alpha}_n-\alpha_n)\\
&\quad = (2-\gamma_n\beta)\mu_n\tilde{\theta}_n\alpha_n - \mu_n\gamma_n\beta\theta_n + \lambda_n\gamma_n\beta\theta_n\alpha_n + 2\alpha_n\bar{\theta}_n\tilde{\theta}_n\alpha_n\\
&\quad = (2-\gamma_n\beta)\mu_n(\lambda_n+\mu_n)\gamma_n\beta\alpha_n - (\lambda_n+\mu_n)\alpha_n\gamma_n\beta\theta_n\\
&\qquad + \lambda_n\gamma_n\beta\theta_n\alpha_n + 2\alpha_n\bar{\theta}_n\mu_n\gamma_n\beta\\
&\quad = (2-\gamma_n\beta)\mu_n(\lambda_n+\mu_n)\gamma_n\beta\alpha_n - \mu_n\alpha_n\gamma_n\beta\theta_n + 2\alpha_n\bar{\theta}_n\mu_n\gamma_n\beta\\
&\quad = \mu_n\gamma_n\beta\alpha_n\big((2-\gamma_n\beta)(\lambda_n+\mu_n) - \theta_n + 2\bar{\theta}_n\big)
\end{aligned}
$$

which by Proposition 4 (ii) is identical to zero. For the coefficient of $\|z_{n-1}\|_M^2$, it is straightforward to see its equivalence to zero by Proposition 4 (iii). Likewise, the coefficient of $\langle z_{n-1}, y_{n-1}\rangle_M$ is identically zero by definition of $\tilde{\theta}_n$. The coefficient of $\|y_{n-1}\|_M^2$ is

$$
\mu_n\gamma_n\beta\hat{\theta}_n\theta_n - 2\theta_n\tilde{\theta}_n\alpha_n^2(\lambda_n+\mu_n) + \hat{\theta}_n\tilde{\theta}_n^2\alpha_n^2 - \lambda_n\gamma_n\alpha_n\beta\hat{\theta}_n\theta_n
$$

$$
\begin{aligned}
&\quad - 2\alpha_n^2\bar{\theta}_n^2\gamma_n^2\beta^2(\lambda_n+\mu_n) \\
&= (\lambda_n+\mu_n)\alpha_n\gamma_n\beta\hat{\theta}_n\theta_n - 2\theta_n\tilde{\theta}_n\alpha_n^2(\lambda_n+\mu_n) + \hat{\theta}_n\tilde{\theta}_n^2\alpha_n^2 \\
&\quad - \lambda_n\gamma_n\alpha_n\beta\hat{\theta}_n\theta_n - 2\alpha_n^2\bar{\theta}_n^2\gamma_n^2\beta^2(\lambda_n+\mu_n) \\
&= \mu_n\alpha_n\gamma_n\beta\hat{\theta}_n\theta_n - 2\theta_n\tilde{\theta}_n\alpha_n^2(\lambda_n+\mu_n) + \hat{\theta}_n\tilde{\theta}_n^2\alpha_n^2 - 2\alpha_n^2\bar{\theta}_n^2\gamma_n^2\beta^2(\lambda_n+\mu_n) \\
&= \mu_n\alpha_n\gamma_n\beta\hat{\theta}_n\theta_n - 2\theta_n\gamma_n\beta\mu_n\alpha_n(\lambda_n+\mu_n) + \hat{\theta}_n\tilde{\theta}_n\mu_n\gamma_n\beta\alpha_n - 2\alpha_n\mu_n\bar{\theta}_n^2\gamma_n^2\beta^2 \\
&= \mu_n\alpha_n\gamma_n\beta\big(\hat{\theta}_n\theta_n - 2\theta_n(\lambda_n+\mu_n) + \hat{\theta}_n\tilde{\theta}_n - 2\bar{\theta}_n^2\gamma_n\beta\big) \\
&= \mu_n\alpha_n\gamma_n\beta\big(\theta_n\big(\hat{\theta}_n - 2\lambda_n - 2\mu_n\big) + \hat{\theta}_n\gamma_n\beta(\lambda_n+\mu_n) - 2\bar{\theta}_n^2\gamma_n\beta\big) \\
&= \mu_n\alpha_n\gamma_n\beta\big(-\theta_n\lambda_n^2\gamma_n\beta + \hat{\theta}_n\gamma_n\beta(\lambda_n+\mu_n) - 2\bar{\theta}_n^2\gamma_n\beta\big) \\
&= \mu_n\alpha_n\gamma_n^2\beta^2\big(-\theta_n\lambda_n^2 + \hat{\theta}_n(\lambda_n+\mu_n) - 2\bar{\theta}_n^2\big)
\end{aligned}
$$

where by Proposition 4 (iii) is equivalent to zero. This concludes the proof. □

References

Alvarez, F. (2000). "On the minimizing property of a second order dissipative system in Hilbert spaces". *SIAM Journal on Control and Optimization* **38**:4, pp. 1102–1119. DOI: 10.1137/s0363012998335802.

Alvarez, F. and H. Attouch (2001). "An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping". *Set-Valued Analysis* **9**:1/2, pp. 3–11. DOI: 10.1023/a:1011253113155.

Apidopoulos, V., J.-F. Aujol, and C. Dossal (2020). "Convergence rate of inertial forward–backward algorithm beyond nesterov's rule". *Mathematical Programming* **180**:1, pp. 137–156. DOI: 10.1007/s10107-018-1350-9.

Attouch, H. and A. Cabot (2020). "Convergence of a relaxed inertial proximal algorithm for maximally monotone operators". *Mathematical Programming* **184**:1, pp. 243–287.

Attouch, H., Z. Chbani, J. Peypouquet, and P. Redont (2018). "Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity". *Mathematical Programming* **168**:1, pp. 123–175. DOI: 10.1007/s10107-016-0992-8.

Attouch, H., M.-O. Czarnecki, and J. Peypouquet (2011). "Coupling forward–backward with penalty schemes and parallel splitting for constrained variational inequalities". *SIAM Journal on Optimization* **21**:4, pp. 1251–1274. DOI: 10.1137/110820300.

Attouch, H. and J. Peypouquet (2016). "The rate of convergence of nesterov's accelerated forward-backward method is actually faster than $1/k\hat{2}$". *SIAM Journal on Optimization* **26**:3, pp. 1824–1834. DOI: 10.1137/15M1046095.

Banert, S., J. Rudzusika, O. Oktem, and J. Adler (2021). *Accelerated forward–backward optimization using deep learning*. arXiv: 2105.05210v1 [math.OC].

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

Beck, A. and M. Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". *SIAM journal on imaging sciences* **2**:1, pp. 183–202. DOI: 10.1137/080716542.

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

Chambolle, A. and C. Dossal (2015). "On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm"". *Journal of Optimization theory and Applications* **166**:3, pp. 968–982. DOI: 10.1007/s10957-015-0746-4.

Chambolle, A. and T. Pock (2011). "A first-order primal–dual algorithm for convex problems with applications to imaging". *Journal of Mathematical Imaging and Vision* **40**:1, pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

Chen, G. H.-G. and R. T. Rockafellar (1997). "Convergence rates in forward–backward splitting". *SIAM Journal on Optimization* **7**:2, pp. 421–444. DOI: 10.1137/S1052623495290179.

Cholamjiak, W., P. Cholamjiak, and S. Suantai (2018). "An inertial forward–backward splitting method for solving inclusion problems in Hilbert spaces". *Journal of Fixed Point Theory and Applications* **20**:1. DOI: 10.1007/s11784-018-0526-5.

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

Eckstein, J. (1989). *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis. Massachusetts Insitute of Technology. URL: http://hdl.handle.net/1721.1/14356.

Giselsson, P. (2019). *Nonlinear forward–backward splitting with projection correction*. arXiv: 1908.07449v3 [math.OC].

Giselsson, P., M. Fält, and S. Boyd (2016). "Line search for averaged operator iteration". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 1015–1022. DOI: 10.1109/CDC.2016.7798401.

Kim, D. (2021). "Accelerated proximal point method for maximally monotone operators". *Mathematical Programming* **190**:1, pp. 57–87. DOI: 10.1007/s10107-021-01643-0.

Latafat, P. and P. Patrinos (2017). "Asymmetric forward–backward–adjoint splitting for solving monotone inclusions involving three operators". *Computational Optimization and Applications* **68**:1, pp. 57–93. DOI: 10.1007/s10589-017-9909-6.

Lieder, F. (2021). "On the convergence rate of the halpern-iteration". *Optimization letters* **15**:2, pp. 405–418. DOI: 10.1007/s11590-020-01617-9.

Lions, P. L. and B. Mercier (1979). "Splitting algorithms for the sum of two nonlinear operators". *SIAM Journal on Numerical Analysis* **16**:6, pp. 964–979. DOI: 10.1137/0716071.

Lorenz, D. A. and T. Pock (2015). "An inertial forward–backward algorithm for monotone inclusions". *Journal of Mathematical Imaging and Vision* **51**:2, pp. 311–325. DOI: 10.1007/s10851-014-0523-2.

Passty, G. B. (1979). "Ergodic convergence to a zero of the sum of monotone operators in Hilbert space". *Journal of Mathematical Analysis and Applications* **72**:2, pp. 383–390. DOI: 10.1016/0022-247x(79)90234-8.

Polyak, B. T. (1964). "Some methods of speeding up the convergence of iteration methods". *USSR Computational Mathematics and Mathematical Physics* **4**:5, pp. 1–17. DOI: 10.1016/0041-5553(64)90137-5.

Raguet, H. and L. Landrieu (2015). "Preconditioning of a generalized forward–backward splitting and application to optimization on graphs". *SIAM Journal on Imaging Sciences* **8**:4, pp. 2706–2739. DOI: 10.1137/15m1018253.

Rockafellar, R. T. (1976). "Monotone operators and the proximal point algorithm". *SIAM journal on control and optimization* **14**:5, pp. 877–898. DOI: 10.1137/0314056.

Ryu, E. and W. Yin (2021). *Large-scale convex optimization via monotone operators*. URL: https://large-scale-book.%20mathopt.%20com/LSCOMO.%20pdf.(visited%20on%2003/2021).

Ryu, E. K., A. B. Taylor, C. Bergeling, and P. Giselsson (2020). "Operator splitting performance estimation: tight contraction factors and optimal parameter selection". *SIAM Journal on Optimization* **30**:3, pp. 2251–2271. DOI: 10.1137/19M1304854.

Sadeghi, H., S. Banert, and P. Giselsson (2021a). *Dwifob: a dynamically weighted inertial forward–backward algorithm for monotone inclusions*. arXiv: 2203.00028 [math.OC].

Sadeghi, H., S. Banert, and P. Giselsson (2021b). *Forward–backward splitting with deviations for monotone inclusions*. arXiv: 2112.00776 [math.OC].

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

Taylor, A. B., J. M. Hendrickx, and F. Glineur (2017a). "Exact worst-case performance of first-order methods for composite convex optimization". *SIAM Journal on Optimization* **27**:3, pp. 1283–1313. DOI: 10.1137/16M108104X.

Taylor, A. B., J. M. Hendrickx, and F. Glineur (2017b). "Performance estimation toolbox (pesto): automated worst-case analysis of first-order optimization methods". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, pp. 1278–1283. DOI: 10.1109/CDC.2017.8263832.

Themelis, A. and P. Patrinos (2019). "Supermann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators". *IEEE Transactions on Automatic Control* **64**:12, pp. 4875–4890. DOI: 10.1109/TAC.2019.2906393.

Tseng, P. (2000). "A modified forward–backward splitting method for maximal monotone mappings". *SIAM Journal on Control and Optimization* **38**:2, pp. 431–446. DOI: 10.1137/S0363012998338806.

Zhang, J., B. O'Donoghue, and S. Boyd (2020). "Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations". *SIAM Journal on Optimization* **30**:4, pp. 3170–3197. DOI: 10.1137/18M1232772.

Paper III

DWIFOB: A Dynamically Weighted Inertial Forward–Backward Algorithm for Monotone Inclusions

Hamed Sadeghi Sebastian Banert Pontus Giselsson

Abstract

We propose a novel *dynamically weighted inertial forward–backward* algorithm (DWIFOB) for solving structured monotone inclusion problems. The scheme exploits the globally convergent forward–backward algorithm with deviations in [Sadeghi et al., 2021] as the basis and combines it with the extrapolation technique used in Anderson acceleration to improve local convergence. We also present a globally convergent primal–dual variant of DWIFOB and numerically compare its performance to the primal–dual method of Chambolle–Pock and a Tikhonov regularized version of Anderson acceleration applied to the same problem. In all our numerical evaluations, the primal–dual variant of DWIFOB outperforms the Chambolle–Pock algorithm. Moreover, our numerical experiments suggest that our proposed method is much more robust than the regularized Anderson acceleration, which can fail to converge and be sensitive to algorithm parameters. These numerical experiments highlight that our method performs very well while still being robust and reliable.

Submitted (Available on arXiv).

1. Introduction

We consider structured monotone inclusion problems of the form

$$0 \in Ax + Cx, \tag{III.1}$$

where $A : \mathscr{H} \to 2^{\mathscr{H}}$ is a maximally monotone operator, $C : \mathscr{H} \to \mathscr{H}$ is a cocoercive operator, and $\mathscr{H}$ is a real Hilbert space. This fundamental problem emerges in many areas such as optimization [Eckstein, 1989; Raguet and Landrieu, 2015] and variational analysis [Attouch et al., 2011; Chen and Rockafellar, 1997; Tseng, 2000]. For instance, consider optimization problems of the form

$$\underset{x \in \mathscr{H}}{\text{minimize}} \quad f(x) + g(Lx) + h(x), \tag{III.2}$$

where $f : \mathscr{H} \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathscr{K} \to \mathbb{R} \cup \{+\infty\}$ are convex closed proper and potentially non-smooth functions, $h : \mathscr{H} \to \mathbb{R}$ is a convex differentiable function with β-Lipschitz continuous gradient, and $L : \mathscr{H} \to \mathscr{K}$ is a bounded linear operator. Given some constraint qualification, see for instance [Bauschke and Combettes, 2017, Corollary 16.48], a point $x^\star \in \mathscr{H}$ is a minimizer of (III.2) if and only if it satisfies the optimality condition

$$0 \in \partial f(x) + L^* \partial g(Lx) + \nabla h(x), \tag{III.3}$$

where L^* is the adjoint operator of L [Bauschke and Combettes, 2017, Theorem 16.3]. Therefore, one can solve the inclusion problem (III.3) in order to find a solution of the optimization problem (III.2). With a transformation to a primal–dual setting this problem can be reformulated as a monotone inclusion of the form (III.1), see Section 4.

Forward–backward (FB) splitting [Bruck, 1975; Lions and Mercier, 1979; Passty, 1979] has been widely used to solve structured monotone inclusions of the form (III.1). The FB splitting method is given by

$$x_{n+1} = (\mathrm{Id} + \gamma_n A)^{-1} \circ (\mathrm{Id} - \gamma_n C)(x_n),$$

where $\gamma_n > 0$ is a step-size parameter. It involves evaluating the operator C in a forward (explicit) step, followed by computing the resolvent of the operator A in a backward (implicit) step. The FB splitting has many well-known special instances, such as the gradient method, the proximal point algorithm [Rockafellar, 1976], and the proximal-gradient method [Combettes and Pesquet, 2011].

The inertial proximal point algorithm in [Alvarez, 2000; Alvarez and Attouch, 2001] improves convergence by exploiting previous information in a momentum term. By incorporating an additional cocoercive operator to the inertial proximal point algorithm, several variations of inertial FB algorithms have been proposed to solve monotone inclusions [Attouch and Cabot, 2020; Cholamjiak et al., 2018;

Lorenz and Pock, 2015]. These algorithms provide enhanced performance, but are limited to FB splitting algorithms.

Anderson acceleration [Anderson, 1965] is an acceleration scheme that is aimed at expediting the convergence of fixed-point iterations including the FB algorithm. This algorithm was originally developed to solve nonlinear integral equations and was later used to solve fixed-point problems [Fang and Saad, 2009; Walker and Ni, 2011]. Lately, Anderson acceleration has gained considerable attention in the optimization community [He et al., 2021; Ouyang et al., 2020; Sadeghi and Giselsson, 2021; Scieur et al., 2020; Zhang et al., 2020].

Local convergence of Anderson acceleration has been studied recently. For instance, the authors of [Toth and Kelley, 2015] showed that Anderson acceleration, if applied to a contractive fixed-point map, exhibits linear convergence provided that the coefficients in the linear combination remain bounded. Along the same line, it was shown in [Evans et al., 2020] that applying Anderson acceleration to a linearly convergent fixed-point iteration improves the convergence rate in the vicinity of a fixed point. Despite recent studies that investigate local convergence properties of Anderson acceleration, yet, to the best of our knowledge, no global convergence result for Anderson acceleration (and its regularized variants) has been reported in the literature.

Recently, the FB algorithm with deviations was proposed in [Sadeghi et al., 2021] to solve the inclusion problem (III.1). This algorithm uses two auxiliary terms—called *deviations*—which are added to the iterates in order to define extrapolated iterates. The algorithm uses a safeguarding *norm condition* in the form of an iteration-dependent constraint on the norm of the deviations that has to be satisfied at each iteration in order to guarantee convergence. As long as this norm constraint is satisfied, the deviations can be chosen freely and point in any direction. In [Sadeghi et al., 2021], one suggestion is to define the deviations along the momentum direction as $a_n(x_n - x_{n-1})$, which gives an inertial-type method. An upper bound to the momentum coefficient a_n is directly obtained by the norm condition.

In this work, inspired by the extrapolation technique of Anderson acceleration, we propose a method to generate the deviation vectors of [Sadeghi et al., 2021] by linearly combining multiple momentum terms. The aim is to construct a version of FB splitting that exhibits fast local convergence while maintaining global convergence of the algorithm, thanks to the norm condition. This is in contrast to Anderson acceleration and its regularized variants [Scieur et al., 2020; Shi et al., 2019] that are only locally convergent. We call our proposed algorithm *dynamically weighted inertial forward–backward* method (DWIFOB).

The notion of safeguarding has been used also in other works to ensure global convergence of nonlinear acceleration algorithms [Giselsson et al., 2016; Sadeghi and Giselsson, 2021; Themelis and Patrinos, 2019; Zhang et al., 2020]. These are hybrid methods that can select between a basic globally convergent and a locally fast converging method, as decided by a safeguarding condition in every iteration. Although having the same objective of achieving global convergence and fast local

convergence, these safeguarding conditions are completely different compared to what we use in DWIFOB.

Besides the DWIFOB scheme itself, we also propose a primal–dual version of the DWIFOB scheme which is derived by a direct translation of the DWIFOB algorithm into a primal–dual framework. We have compared the primal–dual DWIFOB algorithm with the Chambolle–Pock algorithm in numerical experiments, which show a significant advantage of our proposed method in both convergence rate and overall computational cost. Moreover, our numerical evaluations show that regularized Anderson acceleration, in addition to being only locally convergent, is very sensitive to variations in the choice of parameters, while DWIFOB is more robust to parameter selection with the significant added benefit of having global convergence guarantees. The aforementioned robustness and global convergence property along with fast local convergence make the DWIFOB algorithm well-performing and reliable.

The paper is outlined as follows. In Section 2, after presenting the notations and stating the problem under consideration, we review two algorithms that our algorithm is built upon. Section 3 describes our proposed DWIFOB algorithm and Section 4 extends the DWIFOB algorithm to the primal–dual setting and suggests a novel algorithm in this framework. Numerical evaluations are provided in Section 5 and concluding remarks are presented in Section 6.

2. Problem statement and preliminaries

In this section, we present our notation and state the monotone inclusion problem and the associated assumptions. We then briefly review two methods [Sadeghi et al., 2021; Walker and Ni, 2011] that can be used to solve the problem at hand. These methods come with their own sets of weaknesses and strengths. Our proposed method combines these two methods to benefit from their individual strengths and avoid their drawbacks.

2.1 Notation

Throughout the paper, $\mathbb{R}$ and $\mathbb{R}^d$ indicate the sets of real numbers and d-dimensional real column vectors respectively. Additionally, $\mathscr{H}$ and $\mathscr{K}$ denote real Hilbert spaces that are equipped with inner products $\langle\cdot,\cdot\rangle$ and induced norms $\|\cdot\| = \sqrt{\langle\cdot,\cdot\rangle}$. A linear, bounded, self-adjoint operator $M\colon \mathscr{H} \to \mathscr{H}$ is said to be *strongly positive* if there exists $\rho > 0$ such that $\langle x, Mx\rangle \geq \rho \|x\|^2$ for all $x \in \mathscr{H}$. We denote the set of such operators $\mathscr{M}(\mathscr{H})$. For $M \in \mathscr{M}(\mathscr{H})$, the *$M$-induced inner product* and *norm* are defined by $\langle x, y\rangle_M = \langle x, My\rangle$ and $\|x\|_M = \sqrt{\langle x, Mx\rangle}$ $(x, y \in \mathscr{H})$, respectively.

By $2^{\mathscr{H}}$, we denote the *power set* of $\mathscr{H}$. A map $A : \mathscr{H} \to 2^{\mathscr{H}}$ is characterized by its *graph* $\operatorname{gra}(A) = \{(x,u) \in \mathscr{H} \times \mathscr{H} : u \in Ax\}$. An operator $A : \mathscr{H} \to 2^{\mathscr{H}}$ is *monotone*, if $\langle u - v, x - y\rangle \geq 0$ for all $(x,u),(y,v) \in \operatorname{gra}(A)$. A monotone operator A is *maximally monotone* if there exists no monotone operator $B : \mathscr{H} \to 2^{\mathscr{H}}$ such

that $\operatorname{gra}(B)$ properly contains $\operatorname{gra}(A)$. The *zero-set* of the operator A is defined as $\operatorname{zer}(A) := \{x \in \mathscr{H} : 0 \in Ax\}$.

For $\beta > 0$, a single-valued operator $T\colon \mathscr{H} \to \mathscr{H}$ is said to be $\frac{1}{\beta}$-*cocoercive* with respect to $\|\cdot\|_M$ with $M \in \mathscr{M}(\mathscr{H})$ if

$$\langle Tx - Ty, x - y\rangle \geq \tfrac{1}{\beta}\|Tx - Ty\|^2_{M^{-1}} \qquad (\forall x, y \in \mathscr{H}).$$

2.2 Problem statement

We consider structured monotone inclusion problems of the form

$$0 \in Ax + Cx, \tag{III.4}$$

that satisfy the following assumption.

ASSUMPTION 1 Assume that

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

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

(iii) The solution set $\operatorname{zer}(A + C)$ is nonempty. □

This assumption implies that the operator $A + C$ is maximally monotone [Bauschke and Combettes, 2017, Corollary 25.5].

2.3 Forward–backward splitting with deviations

The FB algorithm with deviations is an extension of the standard FB algorithm and was introduced recently in [Sadeghi et al., 2021]. In its most general form, two additive terms—called deviations—are added to the basic FB method to form extrapolations to the iterate. The algorithm uses the extrapolated points in the evaluation of the forward and the backward steps. If the deviations are chosen wisely, this can exhibit an improved convergence compared to standard FB splitting. Algorithm 1 presents an instance of the FB algorithm with only one deviation vector.

To ensure convergence of Algorithm 1, the deviation u_{n+1} must satisfy the iteration-dependent norm bound in step 6 at each iteration [Sadeghi et al., 2021]. This bound is referred to as a *norm condition*. The requirements on the parameters λ_n, γ_n, and ζ_n are collected in Assumption 2.

ASSUMPTION 2 Choose $\varepsilon \in \left(0, \min\left\{1, \frac{4}{3+\beta}\right\}\right)$, and assume that, for all $n \in \mathbb{N}$, the following hold:

(i) $0 \leq \zeta_n \leq 1 - \varepsilon$;
(ii) $\varepsilon \leq \gamma_n \leq \frac{4-3\varepsilon}{\beta}$; and
(iii) $\varepsilon \leq \lambda_n \leq 2 - \frac{\gamma_n \beta}{2} - \frac{\varepsilon}{2}$. □

Algorithm 1

Input: $x_0 \in \mathscr{H}$; and the sequences $(\gamma_n)_{n\in\mathbb{N}}$, $(\lambda_n)_{n\in\mathbb{N}}$, and $(\zeta_n)_{n\in\mathbb{N}}$ according to Assumption 2; and the metric $\|\cdot\|_M$ with $M \in \mathscr{M}(\mathscr{H})$.
set: $y_0 = x_0$ and $u_0 = 0$.
for $n = 0, 1, 2, \ldots$ **do**
  $p_n = (M + \gamma_n A)^{-1} \circ (M - \gamma_n C)(y_n)$
  $x_{n+1} = x_n + \lambda_n(p_n - y_n)$
  choose u_{n+1} such that

$$\|u_{n+1}\|_M^2 \leq \zeta_n^2 \tfrac{\lambda_n(4-2\lambda_n-\gamma_n\beta)(4-2\lambda_{n+1}-\gamma_{n+1}\beta)}{4\lambda_{n+1}} \left\| p_n - x_n + \tfrac{2\lambda_n+\gamma_n\beta-2}{4-2\lambda_n-\gamma_n\beta} u_n \right\|_M^2$$

  $y_{n+1} = x_{n+1} + u_{n+1}$
end for

The following result, which is adopted from [Sadeghi et al., 2021], provides a convergence guarantee for the iterates that are obtained from Algorithm 1.

THEOREM 1 Consider the monotone inclusion problem (III.4) and suppose that Assumption 1 and Assumption 2 hold. Let $(x_n)_{n\in\mathbb{N}}$ be the sequence generated by Algorithm 1. Then, the sequence $(x_n)_{n\in\mathbb{N}}$ converges weakly to a point in $\operatorname{zer}(A + C)$. □

Proof. In the FB splitting with deviations [Sadeghi et al., 2021, Algorithm 1], set $z_n = y_n$. This gives the relation

$$v_n = \tfrac{2-\gamma_n\beta}{2-\lambda_n\gamma_n\beta} u_n$$

between u_n and v_n, which yields Algorithm 1. Therefore, Algorithm 1 is an instance of the FB splitting algorithm with deviations; consequently, Theorem 1 is a direct consequence of [Sadeghi et al., 2021, Theorem 1]. □

There is a great flexibility in the choice of deviation vector u_{n+1}. This flexibility has not been fully explored in [Sadeghi et al., 2021, Section 6], where only a simple momentum direction has been considered. Our proposed method is an instance of Algorithm 1 from [Sadeghi et al., 2021], where the deviations are chosen based on ideas from the extrapolation step of Anderson acceleration with the goal of improving local performance while benefiting from the global convergence properties of Algorithm 1.

2.4 Regularized Anderson acceleration

Consider the following fixed-point problem

$$\text{find } x \in \mathscr{H} \text{ such that } x = T(x), \tag{III.5}$$

where $T\colon \mathscr{H} \to \mathscr{H}$ is a nonexpansive mapping. One way to solve this problem is to use *Anderson acceleration* [Anderson, 1965; Walker and Ni, 2011]. Anderson acceleration is easy to implement and often improves the convergence of fixed-point iterations, particularly in their terminal phase of convergence, i.e., when close to a solution. However, Anderson acceleration (in its original form [Anderson, 1965; Walker and Ni, 2011]) suffers from numerical instability. This issue can, to some extent, be addressed by adding a Tikhonov regularization term to its inner least-squares problem. A regularized formulation of Anderson acceleration is given in Algorithm 2 [Scieur et al., 2020; Shi et al., 2019]. In spite of their popularity and benefits, there are not yet any global convergence results for the pure Anderson acceleration or its regularized variant, to the best of our knowledge.

Algorithm 2 Regularized Anderson acceleration

1: **Input:** $y_0 \in \mathscr{H}$; $m \geq 1$; and the regularization parameter ξ.
2: **for** $n = 0, 1, 2, \ldots$ **do**
3: $\quad m_n = \min\{m, n\}$
4: $\quad x_n = T(y_n)$
5: $\quad$ find $\alpha^{(n)} = (\alpha_0^{(n)}, \ldots, \alpha_{m_n}^{(n)})$ that solves

$$\begin{aligned} \underset{\alpha^{(n)} \in \mathbb{R}^{m_n+1}}{\text{minimize}} \quad & \left\| \mathcal{R}_n \alpha^{(n)} \right\|_2^2 + \xi \left\| \mathcal{R}_n^T \mathcal{R}_n \right\|_F \left\| \alpha^{(n)} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{1}^T \alpha^{(n)} = 1 \end{aligned}$$

$\quad$ where $\mathcal{R}_n = (r_{n-m_n}, \ldots, r_n)$ and $r_j = y_j - x_j$ for $j \in \{n - m_n, \ldots, n\}$
6: $\quad y_{n+1} = \sum_{i=0}^{m_n} \alpha_i^{(n)} x_{n-m_n+i}$
7: **end for**

Anderson acceleration is retrieved from Algorithm 2 by setting $\xi = 0$. The original formulation of Anderson acceleration [Anderson, 1965] is more general as it allows for the following damped (mixed) step to be taken

$$y_{n+1} = \mu_n \sum_{i=0}^{m_n} \alpha_i^{(n)} x_{n-m_n+i} + (1 - \mu_n) \sum_{i=0}^{m_n} \alpha_i^{(n)} y_{n-m_n+i},$$

instead of step 6, in which $\mu_n > 0$ is the damping (mixing) parameter. In this work, we consider the regularized variant of Anderson acceleration, given in Algorithm 2, and refer to it as RAA.

REMARK 1 Anderson acceleration (Algorithm 2 with $\xi = 0$) can be viewed as a quasi-Newton method [Eyert, 1996; Fang and Saad, 2009; Walker and Ni, 2011; Zhang et al., 2020]. To see this, first observe that the inner optimization problem of Anderson acceleration can be written as the following unconstrained least-squares

problem

$$\underset{\omega^{(n)}\in\mathbb{R}^{m_n}}{\text{minimize}} \quad \left\| r_n - \Delta\mathcal{R}_n \omega^{(n)} \right\|_2, \tag{III.6}$$

where $\Delta\mathcal{R}_n = (r_{n-m_n+1} - r_{n-m_n}, \ldots, r_n - r_{n-1})$ and $\omega^{(n)} = (\omega_0^{(n)}, \ldots, \omega_{m_n-1}^{(n)})$ with $\omega_i^{(n)} = \sum_{j=0}^{i} \alpha_j^{(n)}$ for $i \in \{0, \ldots, m_n - 1\}$. Then, defining $\Delta\mathcal{Y}_n = (y_{n-m_n+1} - y_{n-m_n}, \ldots, y_n - y_{n-1})$, the extrapolation step of AA can be cast as

$$y_{n+1} = y_n - G_n r_n$$

where $G_n = \mathrm{Id} + (\Delta\mathcal{Y}_n - \Delta\mathcal{R}_n)(\Delta\mathcal{R}_n^T \Delta\mathcal{R}_n)^{-1} \Delta\mathcal{R}_n^T$. In this framework, Anderson acceleration can be seen a quasi-Newton method where G_n is an approximate inverse Jacobian of $x - T(x)$ that minimizes $\|G_n - I\|_F$ subject to the inverse multi-secant condition $G_n \Delta\mathcal{Y}_n = \Delta\mathcal{R}_n$. □

3. Dynamically weighted inertial FB scheme

In this section, we present a dynamically weighted inertial forward–backward (DWIFOB) scheme to solve the problem introduced in Section 2.2. It is based on Algorithm 1 with a choice of deviation vectors inspired by RAA (Algorithm 2).

The DWIFOB scheme exploits a history of search directions similar to RAA to find a deviation vector, and it uses the norm condition in step 6 of Algorithm 1 to bound the norm of the deviation. This results in an algorithm that addresses the drawbacks of Algorithm 1 (slow local convergence) and RAA (no global convergence guarantee) and benefits from their favorable properties; namely, global convergence of Algorithm 1 and the often fast local convergence of RAA.

The convergence of DWIFOB follows from Theorem 1, that shows the convergence of Algorithm 1, of which DWIFOB is a special instance with a specific class of deviations.

COROLLARY 1 Consider the monotone inclusion problem (III.4) and suppose that Assumption 1 and Assumption 2 hold. Let $(x_n)_{n\in\mathbb{N}}$ be the sequence generated by Algorithm 3. Then, the sequence $(x_n)_{n\in\mathbb{N}}$ converges weakly to a point in the solution set $\mathrm{zer}(A + C)$. □

4. Primal–dual variant of DWIFOB

In this section, we consider a specific type of monotone inclusion problems that, after being translated to a primal–dual framework, can be efficiently tackled by DWIFOB. We propose a primal–dual algorithm based on Algorithm 3 for solving such problems.

Algorithm 3 DWIFOB

Input: $x_0 \in \mathscr{H}$; $m \geq 1$; the sequences $(\lambda_n)_{n\in\mathbb{N}}$, $(\gamma_n)_{n\in\mathbb{N}}$, and $(\zeta_n)_{n\in\mathbb{N}}$ as defined in Assumption 2; the regularization parameter ξ; the metric $\|\cdot\|_M$ with $M \in \mathscr{M}(\mathscr{H})$; and $\varepsilon \geq 0$.
set $y_0 = x_0$ and $u_0 = 0$.
for $n = 0, 1, 2, \ldots$ **do**
  $m_n = \min(m, n)$
  $p_n = (M + \gamma_n A)^{-1} \circ (M - \gamma_n C) y_n$
  $x_{n+1} = x_n + \lambda_n (p_n - y_n)$
  find $\alpha^{(n)} = (\alpha_0^{(n)}, \ldots, \alpha_{m_n}^{(n)})$ that solves

$$\begin{aligned} \underset{\alpha^{(n)} \in \mathbb{R}^{m_n+1}}{\text{minimize}} \quad & \left\| \mathcal{R}_n \alpha^{(n)} \right\|_2^2 + \xi \left\| \mathcal{R}_n^T \mathcal{R}_n \right\|_F \left\| \alpha^{(n)} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{1}^T \alpha^{(n)} = 1 \end{aligned}$$

  where $\mathcal{R}_n = (r_{n-m_n}, \ldots, r_n)$ and $r_j = x_{j+1} - y_j$
  $\widehat{u}_{n+1} = x_{n+1} - \sum_{i=0}^{m_n} \alpha_i^{(n)} x_{n-m_n+i+1}$
  $\ell_n^2 = \frac{\lambda_n (4 - 2\lambda_n - \gamma_n \beta)(4 - 2\lambda_{n+1} - \gamma_{n+1}\beta)}{4\lambda_{n+1}} \left\| p_n - x_n + \frac{2\lambda_n + \gamma_n \beta - 2}{4 - 2\lambda_n - \gamma_n \beta} u_n \right\|_M^2$
  $u_{n+1} = \zeta_n |\ell_n| \frac{\widehat{u}_{n+1}}{\varepsilon + \|\widehat{u}_{n+1}\|_M}$
  $y_{n+1} = x_{n+1} + u_{n+1}$
end for

Problem statement. We consider primal inclusion problems of finding $x \in \mathscr{H}$ such that

$$0 \in Ax + L^* B(Lx) + Cx \tag{III.7}$$

with the following assumptions.

ASSUMPTION 3 Assume that

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

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

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

(iv) $C : \mathscr{H} \to \mathscr{H}$ is a $\frac{1}{\beta}$-cocoercive operator with respect to the metric $\|\cdot\|$;

(v) The solution set $\operatorname{zer}(A + L^* B L + C)$ is nonempty. □

Translation to a primal–dual framework. The inclusion problem (III.7) can be translated to a primal–dual setting [He and Yuan, 2012] to get the inclusion problem

$$0 \in \mathscr{A} z + \mathscr{C} z \tag{III.8}$$

in which, with some abuse of notation,

$$\mathscr{A} = \begin{bmatrix} A & L^* \\ -L & B^{-1} \end{bmatrix} \qquad \mathscr{C} = \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix} \tag{III.9}$$

and $z := (x, \mu) \in \mathscr{H} \times \mathscr{K}$ is a primal–dual pair. It holds that x is a solution to (III.7) if and only if there exists some $\mu \in \mathscr{K}$ such that $z = (x, \mu)$ is a solution to (III.8).

In this setting, the operator $\mathscr{A}$ is a maximally monotone [Bauschke and Combettes, 2017, Proposition 26.32] and the operator $\mathscr{C}$ is $1/\beta$-cocoercive with respect to the norm $\|\cdot\|_M$, with

$$M = \begin{bmatrix} I & -\tau L^* \\ -\tau L & \tau\sigma^{-1} I \end{bmatrix}, \tag{III.10}$$

where $\tau > 0$ and $\sigma > 0$ are chosen such that $\sigma\tau\|L\|^2 < 1$, which ensures that M is strictly positive. Therefore, the inclusion problem (III.8) can be solved using the DWIFOB algorithm. Algorithm 4 describes our primal–dual DWIFOB algorithm which is derived by a straightforward application of DWIFOB to (III.8). With $C = 0$ and $m = 1$, this algorithm is equivalent to [Sadeghi et al., 2021, Algorithm 4], an inertial primal–dual algorithm.

The following is a result on weak convergence of the iterates generated by Algorithm 4. It is based on showing that Algorithm 4 is a special case of the weakly convergent Algorithm 1.

COROLLARY 2 Consider the monotone inclusion problem (III.7) under Assumption 3 and suppose that Assumption 2 holds. Then the sequence $(x_n)_{n\in\mathbb{N}}$ in Algorithm 4 converges weakly to a point in $\operatorname{zer}(A + L^*BL + C)$. □

Proof. Comparing Algorithm 4 with Algorithm 1, we set $p_n = (p_{x,n}, p_{\mu,n})$, $y_n = (\widehat{x}_n, \widehat{\mu}_n)$, define $\mathscr{A}$ and $\mathscr{C}$ as in (III.9), and let M be defined as in (III.10). Then, we have the following update

$$\begin{aligned} p_n = (p_{x,n}, p_{\mu,n}) &= (M + \tau\mathscr{A})^{-1}(My_n - \tau\mathscr{C}y_n) \\ &= \begin{bmatrix} I + \tau A & 0 \\ -2\tau L & \tau\sigma^{-1} I + \tau B^{-1} \end{bmatrix}^{-1} \begin{bmatrix} \widehat{x}_n - \tau L^*\widehat{\mu}_n - \tau C\widehat{x}_n \\ -\tau L\widehat{x}_n + \tau\sigma^{-1}\widehat{\mu}_n \end{bmatrix} \\ &= \begin{bmatrix} (I + \tau A)^{-1}(\widehat{x}_n - \tau L^*\widehat{\mu}_n - \tau C\widehat{x}_n) \\ (I + \sigma B^{-1})^{-1}(\widehat{\mu}_n + \sigma L(2p_{x,n} - \widehat{x}_n)) \end{bmatrix} \\ &= \begin{bmatrix} J_{\tau A}(\widehat{x}_n - \tau L^*\widehat{\mu}_n - \tau C\widehat{x}_n) \\ J_{\sigma B^{-1}}(\widehat{\mu}_n + \sigma L(2p_{x,n} - \widehat{x}_n)) \end{bmatrix}, \end{aligned}$$

Algorithm 4

1: **Input:** $(x_0,\mu_0)\in\mathscr{H}\times\mathscr{K}$; $m\geq 1$; the sequences $(\lambda_n)_{n\in\mathbb{N}}$ and $(\zeta_n)_{n\in\mathbb{N}}$ as defined in Assumption 2; the regularization parameter ξ; $\sigma>0,\tau>0$ such that $\sigma\tau\|L\|^2<1$; and $\varepsilon\geq 0$.
2: **set** $(\widehat{x}_0,\widehat{\mu}_0)=(x_0,\mu_0)$ and $(u_{x,0},u_{\mu,0})=(0,0)$
3: **for** $n=0,1,2,\ldots$ **do**
4: $\quad m_n=\min(m,n)$
5: $\quad p_{x,n}=J_{\tau A}(\widehat{x}_n-\tau L^*\widehat{\mu}_n-\tau C\widehat{x}_n)$
6: $\quad p_{\mu,n}=J_{\sigma B^{-1}}(\widehat{\mu}_n+\sigma L(2p_{x,n}-\widehat{x}_n))$
7: $\quad x_{n+1}=x_n+\lambda_n(p_{x,n}-\widehat{x}_n)$
8: $\quad \mu_{n+1}=\mu_n+\lambda_n(p_{\mu,n}-\widehat{\mu}_n)$
9: $\quad$ find $\alpha^{(n)}=(\alpha_0^{(n)},\ldots,\alpha_{m_n}^{(n)})$ that solves

$$\begin{aligned}&\underset{\alpha^{(n)}\in\mathbb{R}^{m_n+1}}{\text{minimize}} \quad \left\|\mathcal{R}_n\alpha^{(n)}\right\|_2^2+\xi\left\|\mathcal{R}_n^T\mathcal{R}_n\right\|_F\left\|\alpha^{(n)}\right\|_2^2\\ &\text{subject to} \quad \mathbf{1}^T\alpha^{(n)}=1\end{aligned}$$

$\quad$ where $\mathcal{R}_n=(r_{n-m_n},\ldots,r_n)$ where $r_j=(x_{j+1}-\widehat{x}_j,\mu_{j+1}-\widehat{\mu}_j)$
10: $\quad \begin{bmatrix}\widehat{u}_{x,n+1}\\ \widehat{u}_{\mu,n+1}\end{bmatrix}=\begin{bmatrix}x_{n+1}\\ \mu_{n+1}\end{bmatrix}-\sum_{i=0}^{m_n}\alpha_i^{(n)}\begin{bmatrix}x_{n-m_n+i+1}\\ \mu_{n-m_n+i+1}\end{bmatrix}$
11: $\quad \ell_n^2=\frac{\lambda_n(4-2\lambda_n-\tau\beta)(4-2\lambda_{n+1}-\tau\beta)}{4\lambda_{n+1}}\left\|\begin{bmatrix}p_{x,n}\\ p_{\mu,n}\end{bmatrix}-\begin{bmatrix}x_n\\ \mu_n\end{bmatrix}+\frac{2\lambda_n+\tau\beta-2}{4-2\lambda_n-\tau\beta}\begin{bmatrix}u_{x,n}\\ u_{\mu,n}\end{bmatrix}\right\|_M^2$
12: $\quad \begin{bmatrix}u_{x,n+1}\\ u_{\mu,n+1}\end{bmatrix}=\frac{\zeta_n|\ell_n|}{\varepsilon+\left\|(\widehat{u}_{x,n+1},\widehat{u}_{\mu,n+1})\right\|_M}\begin{bmatrix}\widehat{u}_{x,n+1}\\ \widehat{u}_{\mu,n+1}\end{bmatrix}$
13: $\quad \widehat{x}_{n+1}=x_{n+1}+u_{x,n+1}$
14: $\quad \widehat{\mu}_{n+1}=\mu_{n+1}+u_{\mu,n+1}$
15: **end for**

which gives the resolvent steps of Algorithm 4 (steps 5 and 6). Moreover, it is also straightforward to verify that, by substituting (x_{n+1},μ_{n+1}) in place of x_{n+1} in Algorithm 1, the relaxation steps of Algorithm 4 (steps 7 and 8) are equivalent to that of Algorithm 1. Additionally, with the devised choice of $u_{n+1}=(u_{x,n+1},u_{\mu,n+1})$ in Algorithm 4, the norm condition of Algorithm 1 holds. Therefore, since Algorithm 4 is a special instance of Algorithm 1 and due to equivalence of (III.7) and (III.8), a direct application of Theorem 1 concludes the proof. □

REMARK 2 For the choice of $\lambda_n=1$, $u_{x,n}=0$ and $u_{\mu,n}=0$ for all $n\in\mathbb{N}$ and $C=0$, Algorithm 4 reduces to the standard Chambolle–Pock iteration [Chambolle and Pock, 2011], that is

$$(x_{n+1},\mu_{n+1})=\begin{bmatrix}J_{\tau A}(x_n-\tau L^*\mu_n)\\ J_{\sigma B^{-1}}(\mu_n+\sigma L(2x_{n+1}-x_n))\end{bmatrix}.$$ □

4.1 Efficient evaluation of the M-induced norm

In Algorithm 4, we need to evaluate two M-induced norms per iteration, where M is given by (III.10). This means that, in addition to evaluating L and L^* in the resolvent steps, two extra evaluations each of L and L^* are needed due the M-induced norms. These extra evaluations can be computationally expensive, which would make the algorithm computationally inefficient. However, by utilizing a similar approach as in [Sadeghi et al., 2021, Section 6.1], the extra evaluations can be efficiently done by reusing some of the previous computations.

We next show that we only need to apply L and L^* once per iteration (except for the first) in Algorithm 4. Observe that, by applying the operator L on steps 7, 10, and 13 (after substitution of step 12) of Algorithm 4, we obtain the following relations

$$\begin{aligned} Lx_{n+1} &= Lx_n + \lambda_n(Lp_{x,n} - L\widehat{x}_n), \\ L\widehat{u}_{x,n+1} &= Lx_{n+1} - \sum_{i=0}^{m_n} \alpha_i^{(n)} Lx_{n-m_n+i+1}, \\ L\widehat{x}_{n+1} &= Lx_{n+1} + \frac{\zeta_n|\ell_n|}{\varepsilon + \left\|(\widehat{u}_{x,n+1}, \widehat{u}_{\mu,n+1})\right\|_M} L\widehat{u}_{x,n+1}. \end{aligned} \tag{III.11}$$

In these relations, for all $n > 0$, we only need to evaluate $Lp_{x,n}$. The rest of the quantities to the right-hand sides of the above relations are already computed and can be reused. This means that, in practice, we only need to only evaluate one of each L (for $Lp_{x,n}$) and L^* (for $L^*\widehat{\mu}_n$) at each iteration, except for the first. Therefore, since the most computationally expensive part of our algorithm often is evaluating L and L^*, exploiting this technique keeps the computational cost of our algorithm similar to that of the Chambolle–Pock method. However, in order to use this approach, one needs to store $m_n + 4$ vectors of the same dimension as the dual variable. Hence, in applications where storage is a bottleneck, using a large m_n might be restrictive.

Evaluation of the M-induced norm of, for instance, $\left\|(\widehat{u}_{x,n}, \widehat{u}_{\mu,n})\right\|_M$ can be done as

$$\left\|(\widehat{u}_{x,n}, \widehat{u}_{\mu,n})\right\|_M^2 = \|\widehat{u}_{x,n}\|^2 + \tfrac{\tau}{\sigma}\left\|\widehat{u}_{\mu,n}\right\|^2 - 2\tau\left\langle \widehat{u}_{\mu,n}, L\widehat{u}_{x,n}\right\rangle, \tag{III.12}$$

where $L\widehat{u}_{x,n}$ is already available from the stored set of quantities. The other M-induced norm in step 11 of Algorithm 4 can be computed in the same way as above without extra evaluations of L or L^*.

5. Numerical experiments

In this section, we evaluate the performance of the primal–dual variant of the DWIFOB algorithm and compare it with the Chambolle–Pock primal–dual method and RAA.

We consider a *support vector machine* (SVM) problem with l_1-norm regularization for classification of the form

$$\underset{(w,b)\in\mathbb{R}^d\times\mathbb{R}}{\text{minimize}} \sum_{i=1}^{N} \max\left(0, 1-\phi_i(w^T\theta_i+b)\right) + \delta\|w\|_1 \tag{III.13}$$

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, $\delta > 0$ is the regularization parameter, and $x = (w,b)$ with $b \in \mathbb{R}$ and $w \in \mathbb{R}^d$ is the decision variable. This problem can be reformulated as

$$\underset{x\in\mathbb{R}^{d+1}}{\text{minimize}}\ f(Lx) + g(x) \tag{III.14}$$

with

$$f(y) = \sum_{i=1}^{N} \max(0, 1-y_i), \qquad g(x) = \delta\|\omega\|_1, \qquad L = \begin{bmatrix} \phi_1\theta_1^T & \phi_1 \\ \vdots & \vdots \\ \phi_N\theta_N^T & \phi_N \end{bmatrix},$$

where $f, g\colon \mathbb{R}^{d+1} \to \mathbb{R}$ are proper, closed, and convex (and non-smooth) functions with full domain and L is a bounded linear operator. A point $x^\star \in \mathbb{R}^{d+1}$ solves problem (III.14) if and only if

$$0 \in L^*\partial f(Lx) + \partial g(x), \tag{III.15}$$

where ∂f and ∂g are the subdifferentials of f and g, respectively [Bauschke and Combettes, 2017, Proposition 16.42]. By [Bauschke and Combettes, 2017, Theorem 20.25], ∂f and ∂g are maximally monotone. Therefore, we solve the monotone inclusion (III.15) in order to find a solution to problem (III.14), which, by setting $A = \partial g$, $B = \partial f$, and $C = 0$, fits into the framework of problem (III.8). We use the following algorithms to solve the problem:

- Chambolle and Pock's primal–dual method (CP) [Chambolle and Pock, 2011];
- The primal–dual DWIFOB method in Algorithm 4 (Alg4);
- Regularized Anderson acceleration (RAA), Algorithm 2, [Scieur et al., 2020; Walker and Ni, 2011], applied to the fixed-point map of Chambolle–Pock, see Remark 2.

In the algorithms listed above, evaluating L and L^* in the resolvent steps and solving the least-squares problem, if there is one, are the computationally intensive parts. Since the Chambolle–Pock algorithm does not involve solving a least-squares problem, it has a cheaper per-iteration cost compared to the other algorithms. To

provide a fair comparison, we compare the methods using *scaled iterations*. Let C_{CP} and C_{alg} be the average per-iteration computational cost of the Chambolle–Pock method and one of the algorithms mentioned above ($\mathrm{alg} \in \{\mathrm{CP}, \mathrm{Alg4}, \mathrm{RAA}\}$), respectively. The scaled iteration is the iteration count scaled by the ratio $\frac{C_{\mathrm{alg}}}{C_{\mathrm{CP}}}$. The iteration costs C_{CP} and C_{alg} are numerically approximated by measuring the average per-iteration elapsed time of the individual algorithms. The benefits of using the notion of scaled iteration are two-fold. In addition to considering the relative per-iteration computational cost of the algorithms, it eliminates the impact of computational capacity/power of the platform that the algorithms are implemented on, which makes the results more reproducible.

The experiments are done using three different benchmark datasets; the *breast cancer* dataset with 683 samples and 10 features, the *sonar* dataset with 208 samples and 60 features, and *colon cancer* dataset with 62 samples and 2000 features, all from [Chang and Lin, 2011]. The numerical experiments are done on a laptop with a 1.4 GHz Quad-core Intel Core i5 processor with 16 GB of memory. The algorithms are implemented using the Julia programming language (Version 1.3.1).

In all experiments, the primal and the dual step-size parameters are chosen as $\tau = \sigma = 0.99/\|L\|^2$, $\zeta_n = 0.99$ for all $n \in \mathbb{N}$, $\varepsilon = 0$, and a fixed relaxation parameter $\lambda = 1.0$ for Algorithm 4 is used. Unless otherwise stated, the algorithms are initialized at $(x_0, \mu_0) = 0$. We report results from the numerical experiments in a sequence of figures. The M-induced distance to a solution is used as the convergence measure where the individual underlying solutions are found by running the standard Chambolle–Pock algorithm until $\|x_n - x_{n-1}\| \leq 10^{-15}$ and $\|\mu_n - \mu_{n-1}\| \leq 10^{-15}$. All algorithms that converge do so to the same solution. Moreover, all evaluations of L, L^*, and $\|\cdot\|_M$ are done using the proposed recursive method of Section 4.1, unless otherwise stated.

Figures 1 to 3 provide a comparison between the Chambolle–Pock method and Algorithm 4 for several memory size values using different datasets. The figures show that for the considered different values of the memory size m, Algorithm 4 outperforms the Chambolle–Pock method. It can also be seen that increasing the memory size m in Algorithm 4 improves the local convergence rate. However, by increasing m in Algorithm 4, the computational cost of solving the least-squares problem increases, while the computational cost of the resolvent steps is fixed. Therefore, it is expected that there is an optimal memory size beyond which increasing m degrades the performance (compared to the optimal one). This can be better seen in Fig. 4, which shows the number of scaled iterations until the M-scaled distance of (x_n, μ_n) to the solution is less than some value *tol*, against the memory size. It is seen that we get good performance for a wide range of memory sizes (typically $10 \leq m \leq 25$). It is also good to mention that even for small or large m, we still see a considerable improvement compared to the Chambolle–Pock method.

Figure 5 shows the impact of using direct evaluation of L, L^*, and $\|\cdot\|_M$ instead of the proposed recursive method of Section 4.1, on the convergence pattern of Al-

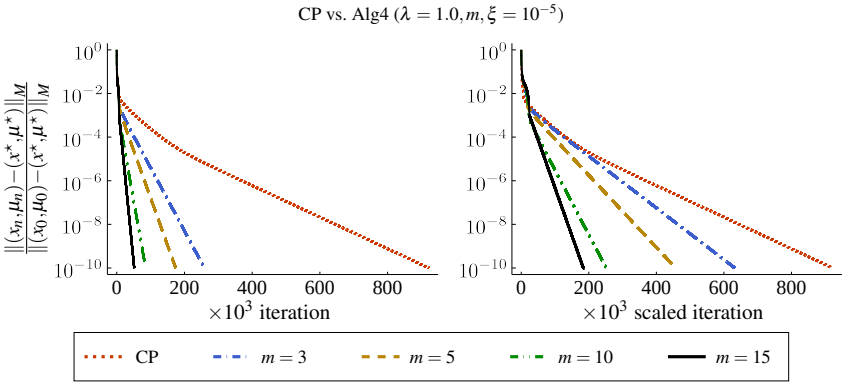

Figure 1: Normalized M-induced distance to the solution vs. iteration number (*left*) and scaled iteration number (*right*) for the l_1-norm regularized SVM, problem (III.13), with $\delta = 0.5$, on the *breast cancer* dataset [Chang and Lin, 2011] with 683 samples and 10 features. Solved using the Chambolle–Pock algorithm and Alg4 ($\lambda = 1.0$, m, $\xi = 10^{-5}$) for several memory sizes m, all with $\tau = \sigma = 0.99/\|L\|$.

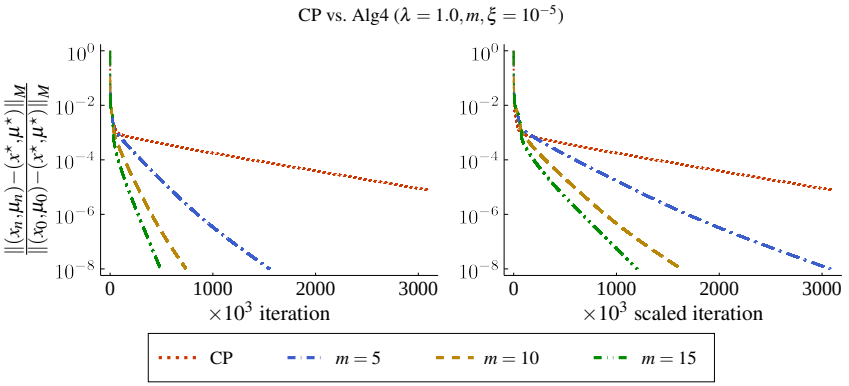

Figure 2: Normalized M-induced distance to the solution vs. iteration number (*left*) and scaled iteration number (*right*) for the l_1-norm regularized SVM, problem (III.13), with $\delta = 1.0$, on the *sonar* dataset [Chang and Lin, 2011] with 208 samples and 60 features. Solved using the Chambolle–Pock algorithm and Alg4 ($\lambda = 1.0$, m, $\xi = 10^{-5}$) for several memory sizes m, all with $\tau = \sigma = 0.99/\|L\|$.

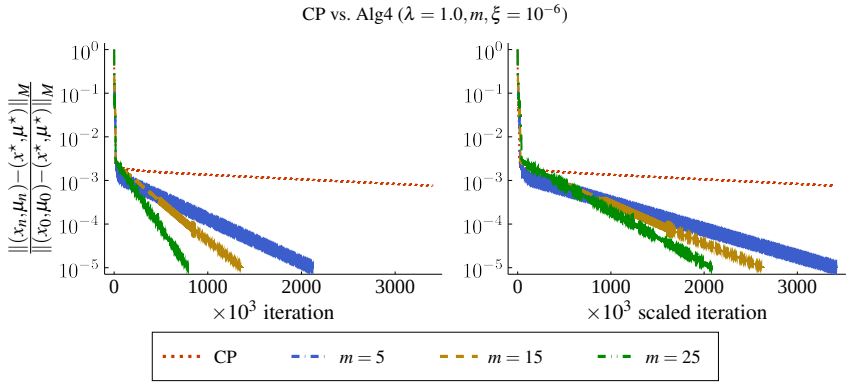

Figure 3: Normalized M-induced distance to the solution vs. iteration number (*left*) and scaled iteration number (*right*) for the l_1-norm regularized SVM, problem (III.13), with $\delta = 0.1$, on the *colon cancer* dataset [Chang and Lin, 2011] with 62 samples and 2000 features. Solved using the Chambolle–Pock algorithm and Alg4 ($\lambda = 1.0$, m, $\xi = 10^{-6}$) for several memory sizes, m, all with $\tau = \sigma = 0.99/\|L\|$.

gorithm 4. The experiment is done with the same setting as in the one reported in Fig. 3 for the case of $m = 25$. The top right plot shows that the suggested method of recursive evaluation of Algorithm 4 considerably decreases the overall computational cost, in this instance by about 30%. Additionally, it is observed that by using the suggested recursive evaluation of L, L^*, and $\|\cdot\|_M$, we might see some unexpected spikes in the plots, which are caused by accumulated errors due to recursive evaluations, while using the direct evaluation method does not result in such spikes. The bottom plot in Fig. 5 compares

$$V_n := \left\| \begin{bmatrix} x_{n+1} \\ \mu_{n+1} \end{bmatrix} - \begin{bmatrix} x^\star \\ \mu^\star \end{bmatrix} \right\|_M^2 + \lambda_n(2-\lambda_n) \left\| \begin{bmatrix} p_{x,n} \\ p_{\mu,n} \end{bmatrix} - \begin{bmatrix} x_n \\ \mu_n \end{bmatrix} + \frac{\lambda_n - 1}{2-\lambda_n} \begin{bmatrix} u_{x,n} \\ u_{\mu,n} \end{bmatrix} \right\|_M^2 \tag{III.16}$$

for the case of direct and recursive evaluation methods. According to [Sadeghi et al., 2021, Lemma 1] with exact evaluation of L, L^*, and M, this quantity should be decreasing, which is confirmed by the figure. However, this is not the case for the recursive evaluation method due to accumulated errors.

The results of experiments with the Chambolle–Pock method, Algorithm 4, and RAA are shown in Fig. 6. The plots on the left-hand side compare the Chambolle–Pock algorithm and Algorithm 4 and the plots on the right-hand side show the convergence of RAA versus the Chambolle–Pock algorithm. For these experiments, the algorithms are initialized far from the origin (at $(x_0, \mu_0) = 10^4 \times \mathbf{1}_{694}$, where $\mathbf{1}_{694}$ is a vector of ones with 694 elements). We see that RAA is not globally convergent;

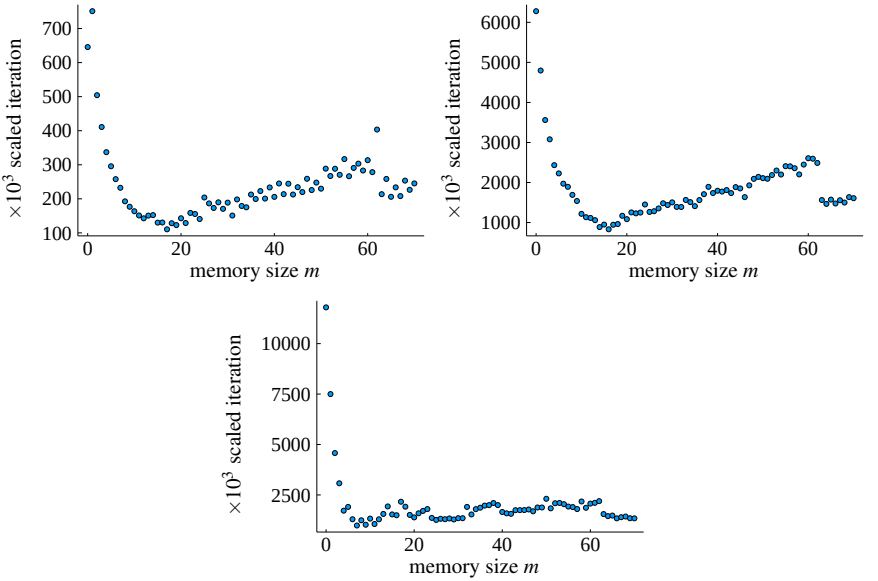

Figure 4: Number of scaled iterations until the normalized M-induced distance to the solution gets smaller than some value *tol* vs. memory size with the settings in the experiments of Fig. 1 (*tol*$= 10^{-8}$, *top left* panel), Fig. 2 (*tol*$= 10^{-6}$, *top right* panel), and Fig. 3 (*tol*$= 10^{-4}$, *bottom* panel); using Algorithm 4, where $m = 0$ corresponds to the Chambolle–Pock method and $m = 1$ corresponds to the inertial primal–dual method of [Sadeghi et al., 2021].

however, when it converges, it does so fast. It is also seen that RAA is really sensitive to parameter variations; and besides that, for it to perform well, there should be a reasonable match between the regularization parameter and its memory size (see the middle plot of RAA). On the other hand, Algorithm 4 is more robust against variations in parameters. These results suggest that Algorithm 4 is more reliable than RAA in the sense of robustness against variations in parameters and also predictability of its behavior.

The distances to a solution for RAA that do not converge to zero in Fig. 6 have not converged although they seem to have flat asymptotes. In fact, consecutive iterates differ a lot and the primal iterate inserted into the objective function (III.14) gives values that are several orders of magnitude larger than the optimal value, also at the end of the simulation. This rules out that the algorithm converges to a different solution (if it exists) than all the other methods do.

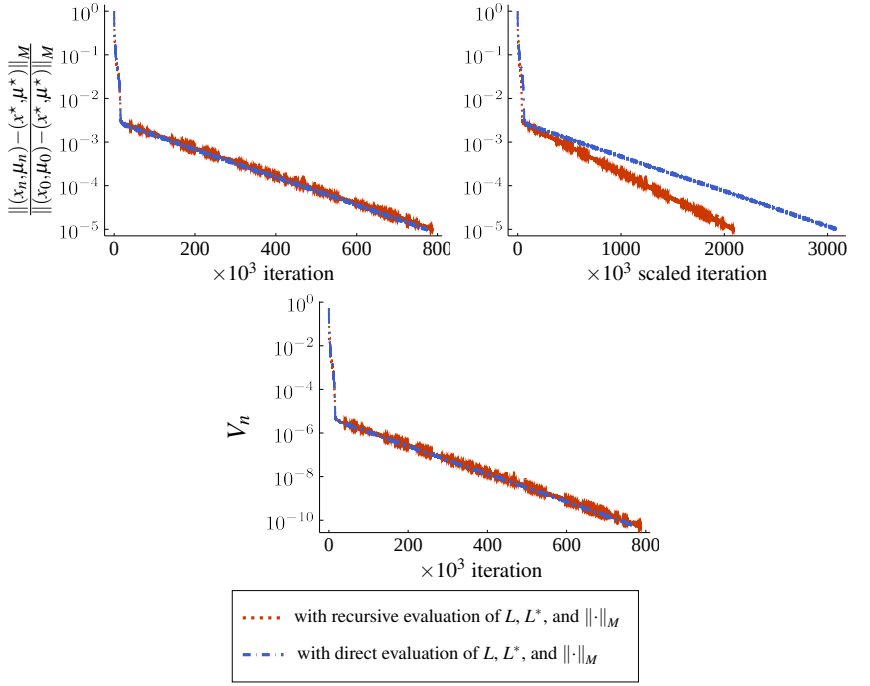

Figure 5: Comparing the impact of *recursive* and *direct* evaluation of L, L^*, and $\|\cdot\|_M$ on the convergence pattern of Alg4 ($\lambda = 1.0$, $m = 25$, $\xi = 10^{-6}$) for problem (III.13) with $\delta = 0.1$, on the *colon cancer* dataset [Chang and Lin, 2011]; *Top panels:* normalized M-induced distance to the solution vs. iteration number (*top left*) and scaled iteration number (*top right*); *bottom panel:* V_n (defined in (III.16)) vs. iteration number.

6. Conclusion

We have proposed a novel scheme to solve structured monotone inclusion problems. By combining a variant of FB splitting with deviations with an extrapolation technique similar to that of Anderson acceleration, we introduced the DWIFOB algorithm. Using the flexibility that the FB algorithm with deviations provides, we introduced a primal–dual variant of the DWIFOB algorithm. Numerical experiments on an l_1-norm regularized SVM problem showed that the primal–dual variant of the DWIFOB algorithm outperforms the Chambolle–Pock primal–dual method. Additionally, we compared the performance of the primal–dual variation of DWIFOB to the regularized Anderson acceleration on the same benchmark problem. The results showed that, in addition to only being locally (though fast) convergent, Anderson acceleration is very sensitive to the variations in choice of parameters while primal–

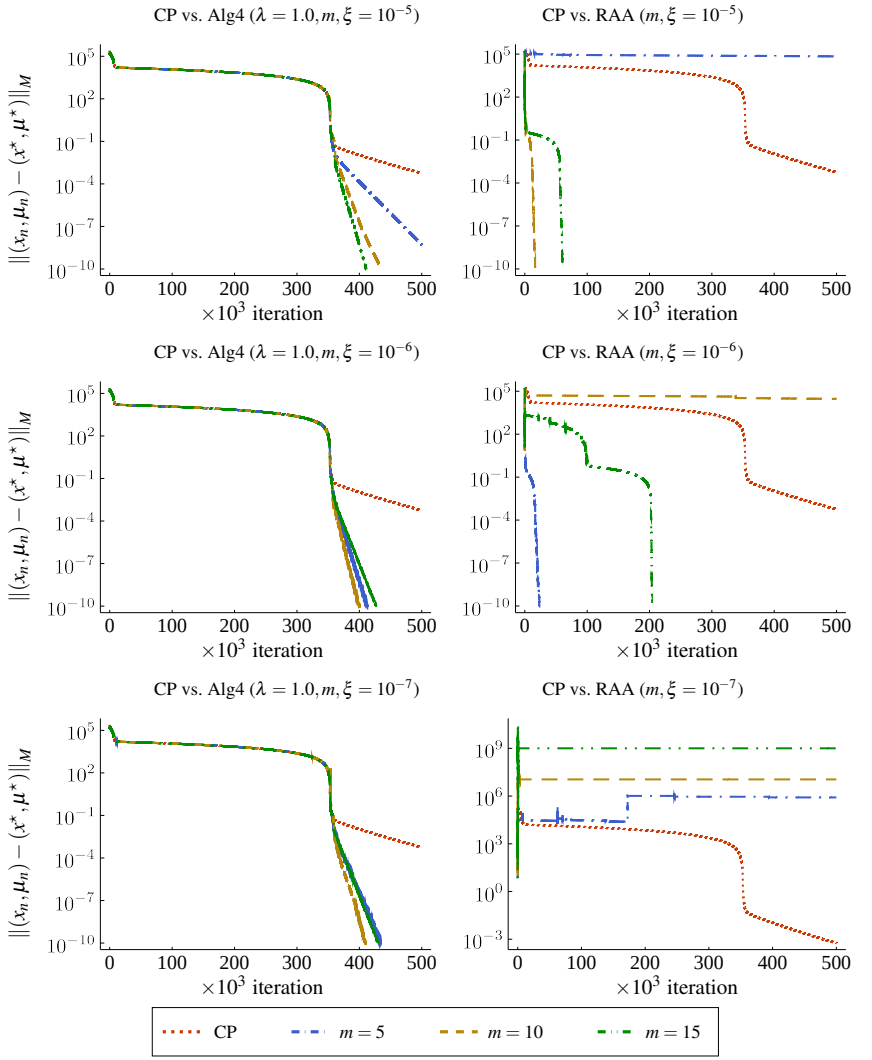

Figure 6: Normalized M-induced distance to the solution vs. iteration number for the l_1-norm regularized SVM problem (III.13) with $\delta = 0.5$ on the *breast cancer* dataset [Chang and Lin, 2011] with 683 samples and 10 features. Solved using the Chambolle–Pock algorithm, Alg4 ($\lambda = 1.0$, m, ξ) (*left-hand side* plots), and RAA (m, ξ) (*right-hand side* plots) for several memory sizes and Tikhonov regularization parameters, all with $\tau = \sigma = 0.99/\|L\|$. In this case, the initial point is set far from the origin, namely, at a distance of approximately 2.6×10^5 to the origin.

dual DWIFOB is much more robust against them. This makes the behavior of the DWIFOB algorithm more reliable and predictable.

Acknowledgement. The authors would like to thank Bo Bernhardsson (Department of Automatic Control, Lund University) 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

Alvarez, F. (2000). "On the minimizing property of a second order dissipative system in Hilbert spaces". *SIAM Journal on Control and Optimization* **38**:4, pp. 1102–1119. DOI: 10.1137/s0363012998335802.

Alvarez, F. and H. Attouch (2001). "An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping". *Set-Valued Analysis* **9**:1/2, pp. 3–11. DOI: 10.1023/a:1011253113155.

Anderson, D. G. (1965). "Iterative procedures for nonlinear integral equations". *Journal of the ACM* **12**:4, pp. 547–560. DOI: 10.1145/321296.321305.

Attouch, H. and A. Cabot (2020). "Convergence of a relaxed inertial proximal algorithm for maximally monotone operators". *Mathematical Programming* **184**:1, pp. 243–287.

Attouch, H., M.-O. Czarnecki, and J. Peypouquet (2011). "Coupling forward–backward with penalty schemes and parallel splitting for constrained variational inequalities". *SIAM Journal on Optimization* **21**:4, pp. 1251–1274. DOI: 10.1137/110820300.

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

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

Chambolle, A. and T. Pock (2011). "A first-order primal–dual algorithm for convex problems with applications to imaging". *Journal of Mathematical Imaging and Vision* **40**:1, pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

Chang, C.-C. and C.-J. Lin (2011). "LIBSVM: a library for support vector machines". *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**:3, pp. 1–27. DOI: 10.1145/1961189.1961199.

Chen, G. H.-G. and R. T. Rockafellar (1997). "Convergence rates in forward–backward splitting". *SIAM Journal on Optimization* **7**:2, pp. 421–444. DOI: 10.1137/S1052623495290179.

Cholamjiak, W., P. Cholamjiak, and S. Suantai (2018). "An inertial forward–backward splitting method for solving inclusion problems in Hilbert spaces". *Journal of Fixed Point Theory and Applications* **20**:1. DOI: 10.1007/s11784-018-0526-5.

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

Eckstein, J. (1989). *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis. Massachusetts Insitute of Technology. URL: http://hdl.handle.net/1721.1/14356.

Evans, C., S. Pollock, L. G. Rebholz, and M. Xiao (2020). "A proof that anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically)". *SIAM Journal on Numerical Analysis* **58**:1, pp. 788–810. DOI: 10.1137/19M1245384.

Eyert, V. (1996). "A comparative study on methods for convergence acceleration of iterative vector sequences". *Journal of Computational Physics* **124**:2, pp. 271–285. DOI: 10.1006/jcph.1996.0059.

Fang, H.-r. and Y. Saad (2009). "Two classes of multisecant methods for nonlinear acceleration". *Numerical Linear Algebra with Applications* **16**:3, pp. 197–221. DOI: 10.1002/nla.617.

Giselsson, P., M. Fält, and S. Boyd (2016). "Line search for averaged operator iteration". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 1015–1022. DOI: 10.1109/CDC.2016.7798401.

He, B. and X. Yuan (2012). "Convergence analysis of primal–dual algorithms for a saddle-point problem: from contraction perspective". *SIAM Journal on Imaging Sciences* **5**:1, pp. 119–149. DOI: 10.1137/100814494.

He, H., S. Zhao, Y. Xi, J. C. Ho, and Y. Saad (2021). *Solve minimax optimization by Anderson acceleration*. arXiv: 2110.02457v2 [cs.LG].

Lions, P. L. and B. Mercier (1979). "Splitting algorithms for the sum of two nonlinear operators". *SIAM Journal on Numerical Analysis* **16**:6, pp. 964–979. DOI: 10.1137/0716071.

Lorenz, D. A. and T. Pock (2015). "An inertial forward–backward algorithm for monotone inclusions". *Journal of Mathematical Imaging and Vision* **51**:2, pp. 311–325. DOI: 10.1007/s10851-014-0523-2.

Ouyang, W., Y. Peng, Y. Yao, J. Zhang, and B. Deng (2020). "Anderson acceleration for nonconvex ADMM based on Douglas–Rachford splitting". *Computer Graphics Forum* **39**:5, pp. 221–239. DOI: 10.1111/cgf.14081.

Passty, G. B. (1979). "Ergodic convergence to a zero of the sum of monotone operators in Hilbert space". *Journal of Mathematical Analysis and Applications* **72**:2, pp. 383–390. DOI: 10.1016/0022-247x(79)90234-8.

Raguet, H. and L. Landrieu (2015). "Preconditioning of a generalized forward–backward splitting and application to optimization on graphs". *SIAM Journal on Imaging Sciences* **8**:4, pp. 2706–2739. DOI: 10.1137/15m1018253.

Rockafellar, R. T. (1976). "Monotone operators and the proximal point algorithm". *SIAM journal on control and optimization* **14**:5, pp. 877–898. DOI: 10.1137/0314056.

Sadeghi, H., S. Banert, and P. Giselsson (2021). *Forward–backward splitting with deviations for monotone inclusions*. arXiv: 2112.00776v1 [math.OC].

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

Scieur, D., A. d'Aspremont, and F. Bach (2020). "Regularized nonlinear acceleration". *Mathematical Programming* **179**:1–2, pp. 47–83. DOI: 10.1007/s10107-018-1319-8.

Shi, W., S. Song, H. Wu, Y.-C. Hsu, C. Wu, and G. Huang (2019). "Regularized anderson acceleration for off-policy deep reinforcement learning". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/bb1443cc31d7396bf73e7858cea114e1-Paper.pdf.

Themelis, A. and P. Patrinos (2019). "Supermann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators". *IEEE Transactions on Automatic Control* **64**:12, pp. 4875–4890. DOI: 10.1109/TAC.2019.2906393.

Toth, A. and C. T. Kelley (2015). "Convergence analysis for Anderson acceleration". *SIAM Journal on Numerical Analysis* **53**:2, pp. 805–819. DOI: 10.1137/130919398.

Tseng, P. (2000). "A modified forward–backward splitting method for maximal monotone mappings". *SIAM Journal on Control and Optimization* **38**:2, pp. 431–446. DOI: 10.1137/S0363012998338806.

Walker, H. F. and P. Ni (2011). "Anderson acceleration for fixed-point iterations". *SIAM Journal on Numerical Analysis* **49**:4, pp. 1715–1735. DOI: 10.1137/10078356X.

Zhang, J., B. O'Donoghue, and S. Boyd (2020). "Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations". *SIAM Journal on Optimization* **30**:4, pp. 3170–3197. DOI: 10.1137/18M1232772.

Paper IV

Hybrid Acceleration Scheme for Variance Reduced Stochastic Optimization Algorithms

Hamed Sadeghi **Pontus Giselsson**

Abstract

Stochastic variance reduced optimization methods are known to be globally convergent while they suffer from slow local convergence, especially when moderate or high accuracy is needed. To alleviate this problem, we propose an optimization algorithm—which we refer to as a hybrid acceleration scheme—for a class of proximal variance reduced stochastic optimization algorithms. The proposed optimization scheme combines a fast locally convergent algorithm, such as a quasi–Newton method, with a globally convergent variance reduced stochastic algorithm, for instance SAGA or L–SVRG. Our global convergence result of the hybrid acceleration method is based on specific safeguard conditions that need to be satisfied for a step of the locally fast convergent method to be accepted.

We prove that the sequence of the iterates generated by the hybrid acceleration scheme converges almost surely to a solution of the underlying optimization problem. We also provide numerical experiments that show significantly improved convergence of the hybrid acceleration scheme compared to the basic stochastic variance reduced optimization algorithm.

Submitted (Available on arXiv).

1. Introduction

We consider convex finite–sum optimization problems of the form

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \;\; F(x)+g(x), \tag{IV.1}$$

where $F:\mathbb{R}^d\to\mathbb{R}$ is the average of convex and smooth functions $f_i:\mathbb{R}^d\to\mathbb{R}$, that is,

$$F(x)=\tfrac{1}{N}\sum_{i=1}^{N} f_i(x),$$

for all $x\in\mathbb{R}^d$ and $g:\mathbb{R}^d\to\mathbb{R}\cup\{\infty\}$ is a closed, convex, proper, and potentially non–smooth function that can be used as a regularization term or to model convex constraints. Such finite–sum optimization problems are common in machine learning and statistics where they are known as regularized empirical risk minimization problems [Schmidt et al., 2017; Teo et al., 2007].

One approach to solve the finite–sum optimization problem (IV.1) is to use the proximal–gradient method [Beck and Teboulle, 2009]. However, at each iteration, the proximal–gradient algorithm requires as many individual gradient evaluations as the number of component functions of the finite–sum, which can be computationally expensive. Another approach is to apply stochastic proximal–gradient descent [Nitanda, 2014; Rosasco et al., 2020], which requires only one gradient evaluation at each iteration, but, due to the variance in the estimation of the full gradient, suffers from sub–linear convergence rate, even in the strongly convex setting [Johnson and Zhang, 2013; Kovalev et al., 2020]. Several stochastic variance–reduced optimization algorithms such as SDCA [Shalev-Shwartz and Zhang, 2013], SVRG [Johnson and Zhang, 2013], and SAGA [Defazio et al., 2014], have been designed to reduce the gradient approximation variance. These methods have been shown to be practically efficient and achieve global (linear) convergence for (strongly) convex problems. However, their local convergence is often slow in practice.

To improve convergence, pre–determined data preconditioning [Li, 2017; Yang et al., 2016] or metric selection [Giselsson and Boyd, 2015] can be used. These are generic approaches that can be applied on top of acceleration schemes. However, finding the optimal or even a good metric is problem– and algorithm–dependent and might be computationally expensive. Quasi–Newton type methods, such as Anderson acceleration [Anderson, 1965; Walker and Ni, 2011] and limited–memory BFGS [Liu and Nocedal, 1989], instead find a suitable metric on the fly. Compared to stochastic optimization algorithms, these methods have higher per–iteration cost, but, often exhibit very fast local convergence. However, global convergence results are scarce for non–smooth problems, whereas some results exist for fully smooth problems [Rodomanov and Nesterov, 2021a; Rodomanov and Nesterov, 2021b; Rodomanov and Nesterov, 2021c].

In this paper, we provide a generic algorithm that combines a method with locally fast convergence (that will be called *acceleration method*) with a globally convergent proximal stochastic optimization algorithm (that will be called *basic method*). The key feature of the general algorithm is a set of safeguard conditions that decide if an acceleration step can be accepted while maintaining global convergence. If the safeguard conditions are not satisfied, a step of the basic method is taken. This results in a hybrid scheme that automatically selects between two different algorithms and benefits both from the global convergence properties of the basic method and the fast local convergence of the acceleration method. We refer to our proposed algorithm as the *hybrid acceleration scheme*.

The idea of a hybrid algorithm that selects between a globally convergent method and locally fast, but not globally, convergent method has been explored, e.g., in [Themelis and Patrinos, 2019; Zhang et al., 2020], whose selection criteria are extensions of the one in [Giselsson et al., 2016]. A key difference between our approach and [Themelis and Patrinos, 2019; Zhang et al., 2020] is that their methods are based on a deterministic basic method, while ours is based on a variance reduced stochastic method. This difference necessitates a completely different convergence analysis and enables for faster progress far from the solution in our finite–sum problem setting since our method takes advantage of that particular problem structure [Schmidt et al., 2017].

Due to the flexibility of our scheme, many different locally fast methods can be used. For instance; limited–memory BFGS (lBFGS) [Liu and Nocedal, 1989], Anderson acceleration [Anderson, 1965], and the class of *vector extrapolation methods* [Smith et al., 1987] to which, e.g., the regularized nonlinear acceleration [Scieur et al., 2016] and its stochastic counterpart [Scieur et al., 2017] belong.

We instantiate our hybrid method with two different local methods, namely limited–memory BFGS (lBFGS) [Liu and Nocedal, 1989] and Anderson acceleration [Anderson, 1965]. In our numerical experiments, we combine these methods with Loop–less SVRG [Kovalev et al., 2020] in our hybrid acceleration method. Our numerical experiments show that our hybrid acceleration scheme can exhibit significantly improved convergence compared to the basic stochastic optimization algorithm.

The paper is outlined as follows. In Section 2, we recall some basic definitions. Section 3 discusses the problem formulation and the link between deterministic and stochastic gradient methods and introduces the family of stochastic optimization algorithms that is considered in this work. In Section 4, the hybrid acceleration method is introduced and in Section 5, we prove its convergence. Numerical experiments are presented in Section 6 and concluding remarks are given in Section 7.

2. Preliminaries

The set of the real numbers and the d-dimensional Euclidean space are denoted by $\mathbb{R}$ and $\mathbb{R}^d$ respectively. For a symmetric positive definite matrix Γ and $x, y \in \mathbb{R}^d$, $\langle x, y\rangle$, $\|x\|$, and $\|x\|_\Gamma$ are the inner product, the induced norm, and the weighted norm $\|x\|_\Gamma := \sqrt{\langle x, \Gamma x\rangle}$ respectively. Moreover, the $d \times d$ identity matrix is denoted by I_d.

The notation $2^{\mathbb{R}^d}$ denotes the power set of $\mathbb{R}^d$. A map $A : \mathbb{R}^d \rightrightarrows 2^{\mathbb{R}^d}$ is characterized by its graph $\mathrm{gra}(A) = \{(x,u) \in \mathbb{R}^d \times \mathbb{R}^d : u \in Ax\}$. The operator A is monotone, if $\langle u - v, x - y\rangle \geq 0$ for all $(x,u),(y,v) \in \mathrm{gra}(A)$. A monotone operator A is maximally monotone if there exists no monotone operator $B : \mathbb{R}^d \rightrightarrows 2^{\mathbb{R}^d}$ such that $\mathrm{gra}(B)$ properly contains $\mathrm{gra}(A)$. A mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ is *L-Lipschitz continuous* if $\|T(x) - T(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^d$, and is nonexpansive if it is 1-Lipschitz continuous. Further $T : \mathbb{R}^d \to \mathbb{R}^d$ is

i) *firmly nonexpansive* if

$$\|x - y - (T(x) - T(y))\|^2 \leq \|x - y\|^2 - \|T(x) - T(y)\|^2 \qquad \forall x, y \in \mathbb{R}^d,$$

ii) $\frac{1}{L}$*-cocoercive* if

$$\langle T(x) - T(y), x - y\rangle \geq \tfrac{1}{L}\|T(x) - T(y)\|^2 \qquad \forall x, y \in \mathbb{R}^d.$$

For a mapping T, $\frac{1}{L}$-cocoercivity implies its L-Lipschitz continuity. The other direction does not hold in general. However, if the mapping is the gradient of a convex function, then its L-Lipschitz continuity and $\frac{1}{L}$-cocoercivity are equivalent [Bauschke and Combettes, 2017, Corollary 18.17]. A differentiable function $F : \mathbb{R}^d \to \mathbb{R}$ is said to be *L-smooth*, if its gradient is L-Lipschitz continuous.

The subdifferential of a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ at $x \in \mathbb{R}^d$ is denoted by $\partial f(x)$ and defined as

$$\partial f(x) = \{v \in \mathbb{R}^d : f(y) \geq f(x) + \langle v, y - x\rangle \text{ for all } y \in \mathbb{R}^d\}.$$

The proximal mapping of a closed, convex and proper function $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$, is defined as $\mathrm{prox}_{\lambda g}(v) = \underset{x}{\mathrm{argmin}} \left(g(x) + \frac{1}{2\lambda}\|x - v\|^2\right)$, where $\lambda > 0$.

The set of fixed–points of a mapping $\mathscr{T} : \mathbb{R}^d \to \mathbb{R}^d$, is denoted by $\mathrm{fix}(\mathscr{T})$ and defined as $\mathrm{fix}(\mathscr{T}) = \{x \in \mathbb{R}^d : x = \mathscr{T}x\}$. The zero–set of a map $\mathscr{R} : \mathbb{R}^d \to \mathbb{R}^d$ is indicated by $\mathrm{zer}(\mathscr{R})$ and given by $\mathrm{zer}(\mathscr{R}) = \{x \in \mathbb{R}^d : 0 = \mathscr{R}x\}$. It is evident that $\mathrm{fix}(\mathscr{T}) = \mathrm{zer}((\mathrm{Id} - \mathscr{T}))$, where $\mathrm{Id} - \mathscr{T}$ is the residual map of the operator $\mathscr{T}$.

3. Problem formulation and basic method

We are interested in solving the following convex optimization problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \;\; \frac{1}{N}\sum_{i=1}^{N} f_i(x) + g(x), \tag{IV.2}$$

under the following assumptions.

ASSUMPTION 1 We assume that

(i) For each $i \in \{1,\dots,N\}$, the function $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex, differentiable and L_i-smooth.

(ii) The function $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is convex, closed and proper.

(iii) The solution set of the problem is nonempty. □

The necessary and sufficient optimality condition for this problem is given by Fermat's rule as

$$0 \in \partial(F+g)(x) = \nabla F(x) + \partial g(x), \tag{IV.3}$$

where the equality holds since all f_i have full domain and g is proper [Bauschke and Combettes, 2017, Theorem 16.3 and Corollary 16.48]. This means that any $x^\star$ that satisfies the optimality condition (IV.3), is a solution to the associated optimization problem (IV.2). It is also known that fixed–points of the proximal–gradient operator, namely, the set $\{x \in \mathbb{R}^d : x = \text{prox}_{\lambda g}(x - \lambda\nabla F(x)), \lambda > 0\}$, are solutions of problem (IV.2). In fact, all solutions of the inclusion problem (IV.3), are fixed–points of the proximal–gradient operator or, equivalently, zeros of its residual mapping, which is given by

$$\mathscr{R}x = x - \text{prox}_{\lambda g}\left(x - \lambda\nabla F(x)\right),$$

for any $\lambda > 0$ [Parikh and Boyd, 2014, Section 4.2]. For $0 < \lambda < \frac{2}{L}$ with L being the smoothness modulus of F, iterating the proximal gradient mapping finds a solution of problem (IV.2) [Combettes and Pesquet, 2011].

The optimality condition (IV.3), can be reformulated in a primal–dual form by storing all gradients of component functions f_i. In that case, the optimality condition becomes

$$\begin{cases} 0 \in \partial g(x) + \frac{1}{N}\sum_{i=1}^{N} y_i \\ 0 = y_1 - \nabla f_1(x) \\ \quad\vdots \\ 0 = y_N - \nabla f_N(x) \end{cases}, \tag{IV.4}$$

where y_i denotes the i–th dual variable. This is clearly equivalent to (IV.3). Therefore, a primal–dual solution $z^\star := (x^\star, y_1^\star, \dots, y_N^\star)$ satisfies (IV.4), if and only if $x^\star$

satisfies (IV.3) and is a solution of (IV.2). It also holds that $z^\star$ satisfies (IV.4) if and only if it satisfies $\bar{\mathscr{R}} z^\star = 0$, where $\bar{\mathscr{R}}$ is the primal–dual residual mapping

$$\bar{\mathscr{R}} z := \begin{pmatrix} x - \operatorname{prox}_{\lambda g}\left(x - \frac{\lambda}{N}\sum_{i=1}^N y_i\right) \\ y_1 - \nabla f_1(x) \\ \vdots \\ y_N - \nabla f_N(x) \end{pmatrix}, \tag{IV.5}$$

in which $z = (x, y_1, \ldots, y_N)$ is the primal–dual variable. We record the equivalence between zeroes of $\bar{\mathscr{R}}$ and solutions to (IV.2), in Proposition 1 (with proof in Appendix A.1) and Lipschitz continuity of $\bar{\mathscr{R}}$ in Proposition 2 (with proof in Appendix A.2).

PROPOSITION 1 Given the residual map in (IV.5), the primal–dual point $z^\star = (x^\star, y_1^\star, \ldots, y_N^\star)$ satisfies $\bar{\mathscr{R}} z^\star = 0$, if and only if $x^\star$ solves (IV.2). Furthermore, for each index i, $y_i^\star$ is unique. □

PROPOSITION 2 Let for all $i \in \{1, \ldots, N\}$, $f_i(x)$ be L_i-smooth, then the primal–dual residual mapping, $\bar{\mathscr{R}}$ in (IV.5), is Lipschitz-continuous. □

In order to find zeros of $\bar{\mathscr{R}}$, one way is to form iterates based on (IV.5), and evaluate all y_i's (the full gradient) at each iteration, which would be similar to the proximal–gradient algorithm. However, when N is very large, a key challenge is the high per–iteration cost of N gradient evaluations which makes the algorithm very expensive. This gives rise to the idea of using a cheaply evaluable approximation of the true gradient instead, and randomly evaluate gradients of only one or some of f_i's at each iteration. The following gives such an approximation

$$\widehat{\nabla}_{i_k} F(x, y) \triangleq \tfrac{1}{N p_{i_k}}\left(\nabla f_{i_k}(x) - y_{i_k}\right) + \tfrac{1}{N}\sum_{i=1}^N y_i, \tag{IV.6}$$

in which i_k is an index randomly drawn from $\{1, \ldots, N\}$ based on some probability distribution and p_{i_k} is its associated probability. This stochastic approximation is based on the average of the dual variables which is modified by a correction term, $(\nabla f_{i_k}(x) - y_{i_k})/(N p_{i_k})$. The correction term is added in order to progressively improve the approximation by incorporating the latest gradient information and also to make $\widehat{\nabla}_{i_k} F(x, y)$ an unbiased estimate of the true gradient.

Using the approximation $\nabla F(x) \approx \widehat{\nabla}_{i_k} F(x, y)$, and inspired by the proximal gradient algorithm, a family of proximal stochastic optimization algorithms can be formulated as

$$\begin{aligned} x^{k+1} &= \operatorname{prox}_{\lambda g}\left(x^k - \lambda \widehat{\nabla}_{i_k} F(x^k, y^k)\right), \\ y_i^{k+1} &= y_i^k + \varepsilon_i^k (\nabla f_i(x^k) - y_i^k), \quad \forall i \in \{1, \ldots, N\}, \end{aligned} \tag{IV.7}$$

where k is the iteration counter, x^k is the primal variable, $y^k = (y_1^k, \ldots, y_N^k)$ with y_i^k being the i–th dual variable, $\lambda > 0$ is the step size, $\varepsilon_i^k \in \{0,1\}$ is a random binary variable that determines whether the i–th dual variable is to be updated at iteration k (the associated probability of $\varepsilon_i^k = 1$ is ρ_i), and $\widehat{\nabla}_{i_k} F(x^k, y^k)$ is the stochastic approximation of the true gradient that is defined in (IV.6). This approximation of the full gradient is unbiased since

$$\begin{aligned}\mathbb{E}_k(\widehat{\nabla}_{i_k} F(x^k, y^k)) &= \tfrac{1}{N}\sum_{i=1}^{N}(\nabla f_i(x^k) - y_i^k) + \tfrac{1}{N}\sum_{i=1}^{N} y_i^k \\ &= \tfrac{1}{N}\sum_{i=1}^{N}\nabla f_i(x^k) = \nabla F(x^k),\end{aligned}$$

in which, $\mathbb{E}_k$ denotes expected value operation given all available information up to step k. We refer to (IV.7) as the *basic method*. On the other hand, there are algorithms that use a biased estimation of the true gradient [Morin and Giselsson, 2019; Roux et al., 2012], but in this work we only consider the unbiased case. The algorithm in (IV.7) has been analyzed in [Davis, 2016] in the monotone operator setting and in [Morin and Giselsson, 2020] in the strongly convex setting.

The class of stochastic optimization algorithms (IV.7) has L–SVRG [Davis, 2016; Kovalev et al., 2020] and SAGA [Defazio et al., 2014] as special cases. The L–SVRG algorithm is extracted from (IV.7) with uniform sampling of $i_k \in \{1, \ldots, N\}$ and

$$\varepsilon_i^k = \begin{cases} 1 & \text{if } q < \rho \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in \{1, \ldots, N\},$$

where q is uniformly sampled from $[0,1]$ and $0 < \rho \leq 1$. Therefore, all dual variables are updated together and on average once every ρ^{-1} iterations. The algorithm in (IV.7) reduces to SAGA with i_k uniformly sampled from $\{1, \ldots, N\}$ and

$$\varepsilon_i^k = \begin{cases} 1 & \text{if } i_k = i \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in \{1, \ldots, N\}.$$

Therefore, for SAGA, at each iteration, only one of the dual variables is updated and the others remain unchanged.

4. Hybrid acceleration scheme

In this section, we introduce a novel hybrid strategy to accelerate local convergence of proximal stochastic optimization algorithms of the form (IV.7), in which the approximation of the true gradient and the update law of the dual variables, vary depending on the choice of basic method. The basic method (IV.7), is devised to

solve large–scale finite–sum optimization problems of the form (IV.2), and is globally convergent while it has slow local convergence. Therefore, in our acceleration scheme, they are combined with a locally fast convergent method. The proposed acceleration scheme is given in Algorithm 1 and discussed below.

Algorithm 1 General framework of the hybrid acceleration scheme

1: **Input:** initial point z^0, positive constants C, D, δ, merit function $V(.)$, acceleration algorithm $\mathscr{A}(.)$ and its memory size m (if needed), K_0, the basic method and its parameters, primal and dual probability distribution, the step size λ, $\Gamma = \text{blkdiag}(I_d, \frac{\lambda}{N\rho_1 L_1} I_d, \ldots, \frac{\lambda}{N\rho_N L_N} I_d)$, and the maximum permissible number of iterations k_{max}.
2: set $k = k_{\text{aa}} = 0$
3: **while** $k < k_{max}$ **do**
4: $\quad m_{k_{\text{aa}}} = \min\{m, k_{\text{aa}}\}$
5: $\quad$ find $z^+ = \mathscr{A}(z^k, z^{k_{\text{aa}}-1}, \ldots, z^{k_{\text{aa}}-m_{k_{\text{aa}}}})$ from acceleration algorithm
6: $\quad$ **if** $V(z^+) \leq \frac{CV(z^0)}{(k_{\text{aa}}+1)^{(1+\delta)}}$ **and** $\left\|z^+ - z^k\right\|_\Gamma \leq DV(z^k)$ **then**
7: $\quad\quad$ set $z^{k+1} = z^+$ and $k_{\text{aa}} \leftarrow k_{\text{aa}} + 1$
8: $\quad\quad$ $k \leftarrow k+1$ and $k_{\text{aa}} \leftarrow k_{\text{aa}} + 1$
9: $\quad$ **else**
10: $\quad\quad$ set $(\widetilde{x}^0, \widetilde{y}_1^0, \ldots, \widetilde{y}_N^0) = z^k$
11: $\quad\quad$ **for** $s = 0, 1, \cdots, K_0 - 1$ **do**

$$\begin{cases} \widetilde{x}^{s+1} = \text{prox}_{\lambda g}\left(\widetilde{x}^s - \lambda \widehat{\nabla}_{i_s} F(\widetilde{x}^s, \widetilde{y}^s)\right), \\ \widetilde{y}_i^{s+1} = \widetilde{y}_i^s + \varepsilon_i^s(\nabla f_i(\widetilde{x}^s) - \widetilde{y}_i^s), \quad \forall i \in \{1, \ldots, N\}. \end{cases}$$
$$z^{k+1} = (\widetilde{x}^{s+1}, \widetilde{y}_1^{s+1}, \ldots, \widetilde{y}_N^{s+1})$$
$$k \leftarrow k+1$$

12: $\quad\quad$ **end for**
13: $\quad$ **end if**
14: **end while**

Description of the algorithm. In order to initialize the scheme one needs to select (i) the parameters and probability distributions used in the basic method; (ii) an acceleration algorithm $\mathscr{A}(.)$ along with its associated parameters; and (iii) an initial point z^0. The acceleration algorithm $\mathscr{A}(.)$ can be algorithms such as lBFGS or Anderson acceleration that both store and use a history of past m iterates to find a next iterate. Then, the algorithm works as follows: at the beginning of each iteration the iterate from the acceleration algorithm, z^+, has to be computed. If z^+ satisfies some *safeguard conditions*, that we will discuss below, we set it as the true next iter-

ate, z^{k+1}, and the main counter of the loop, k, and also the acceleration algorithm's counter, k_{aa}, are increased by one; then, we proceed to the next iteration. Otherwise, K_0 steps of the basic method are performed in the inner loop of the algorithm. It is evident that K_0 can differ among different iterations of the outer loop of the scheme, but, we considered it as a constant for simplicity. The algorithm is to be run as above until the last iteration is reached or some termination criteria are met. Note that if the iterate from the acceleration algorithm is accepted, the basic method steps need not to be performed, that is, the basic method and the acceleration algorithm are not being run in parallel.

Safeguard conditions and merit function. For a nominal next iterate of the acceleration algorithm, z^+, to be accepted as the actual next iterate of the scheme, the following conditions have to be satisfied

$$V(z^+) \leq CV(z^0)(1+k_{\text{aa}})^{-(1+\delta)}, \tag{IV.8}$$

$$\left\|z^+ - z^k\right\|_\Gamma \leq DV(z^k), \tag{IV.9}$$

where δ, C and D are positive constants, $V : \mathbb{R}^{(N+1)d} \to \mathbb{R}$ is a merit function (that is discussed below), and

$$\Gamma = \text{blkdiag}(I_d, \tfrac{\lambda}{N\rho_1 L_1} I_d, \ldots, \tfrac{\lambda}{N\rho_N L_N} I_d).$$

The safeguard condition (IV.8) enforces the merit function to be convergent to zero. Condition (IV.9), is to ensure that the sequence $(\|z^{k+1} - z^k\|_\Gamma)_{k \in I_{\text{aa}}}$, where I_{aa} is the set of indices for which the next iterate is obtained from acceleration algorithm, is diminishing and finally convergent to zero.

Our convergence theory, that will be given in the next section, assumes that; i) the merit function outputs nonnegative values, ii) for any sequence $(z^k)_{k\in\mathbb{N}}$, the merit function is such that

$$V(z^k) \to 0 \qquad \Longrightarrow \qquad \left\|\bar{\mathscr{R}} z^k\right\| \to 0.$$

Therefore, a feasible choice for the merit function can be the following scaled l_2–norm of $\bar{\mathscr{R}} z^k$

$$V(z^k) = \left\|\bar{\mathscr{R}} z^k\right\|_\Gamma. \tag{IV.10}$$

For this choice of merit function, which we use in this work, both requirements on the merit function are met. Other options for the merit function could be the sum or maximum of the vector of the last p scaled l_2–norm of residuals.

5. Convergence results

In this section, we provide results on convergence of the basic method and the hybrid acceleration scheme. Before proceeding to convergence results, we summarize the

notations and the assumptions that are used in the theorems and their proofs. The proofs are given in the Appendix.

Notation. $\mathscr{X}^\star$ indicates the solution set of problem (IV.2), $z = (x, y_1, \ldots, y_N)$ denotes a primal–dual variable for (IV.5), $\bar{\mathscr{R}}$ is the primal–dual residual operator defined in (IV.5), $z^\star = (x^\star, y_1^\star, \ldots, y_N^\star)$ is an arbitrary point in the set of zeros of the primal–dual residual mapping with $y_i^\star = \nabla f_i(x^\star)$, p_i is primal sampling probability, ρ_i is the i–th dual variable update probability, $\lambda > 0$ is the step size, $\mathbb{E}_k$ denotes the expected value operator given all the information up to the k–th iteration, and $\Gamma = \text{blkdiag}(I_d, \frac{\lambda}{N\rho_1 L_1} I_d, \ldots, \frac{\lambda}{N\rho_N L_N} I_d)$. Moreover, $z^k = (x^k, y_1^k, \ldots, y_N^k)$ denotes the k–th primal–dual iterate; and $(z^k)_{k\in\mathbb{N}}$, $(x^k)_{k\in\mathbb{N}}$, and $(y_i^k)_{k\in\mathbb{N}}$ are the sequences of primal–dual-, primal-, and the i–th dual iterates, respectively.

The following is a result that is used in proof of Theorem 2. The proof can be found in Appendix A.3.

PROPOSITION 3 Under Assumption 1, almost sure (a.s.) convergence of $(z^k)_{k\in\mathbb{N}}$ to a $\bar{z} \in \text{zer}(\bar{\mathscr{R}})$, implies a.s. convergence of $(x^k)_{k\in\mathbb{N}}$ and $(y_i^k)_{k\in\mathbb{N}}$ to a $\bar{x} \in \mathscr{X}^\star$ and $\bar{y}_i = \nabla f_i(\bar{x})$ respectively. □

The result in Theorem 1 and its proof (given in Appendix A.5) share similarities with [Davis, 2016; Morin and Giselsson, 2020].

THEOREM 1 Let z^k be the k–th primal–dual iterate associated with the basic method iterates in (IV.7), then given Assumption 1, the following holds

$$\mathbb{E}_k \|z^{k+1} - z^\star\|_\Gamma^2 \leq \|z^k - z^\star\|_\Gamma^2 - \zeta_k,$$

with

$$\begin{aligned}\zeta_k = \sum_{i=1}^N \tfrac{\lambda}{N}\left(\tfrac{1}{L_i} - \tfrac{2\lambda}{Np_i}\right)\left(\|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2 + \|y_i^k - y_i^\star\|^2\right) \\ + \tfrac{2\lambda^2}{N^2}\|\sum_{i=1}^N (y_i^k - y_i^\star)\|^2 + \lambda^2\|\nabla F(x^k) - \nabla F(x^\star)\|^2 \\ + \mathbb{E}_k\|x^{k+1} - x^k + \lambda(\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star))\|^2.\end{aligned} \tag{IV.11}$$

Furthermore, if $0 < \lambda < \min_i\{\frac{Np_i}{2L_i}\}$, then $(z^k)_{k\in\mathbb{N}}$ converges a.s. to a random variable $\bar{z} \in \text{zer}(\bar{\mathscr{R}})$ and $(x^k)_{k\in\mathbb{N}}$ and $(y_i^k)_{k\in\mathbb{N}}$ converge a.s. to random variables $\bar{x} \in \mathscr{X}^\star$ and $\bar{y}_i = \nabla f_i(\bar{x})$ respectively. □

REMARK 1 In order to ensure a.s. convergence in Theorem 1, the coefficient of all terms in ζ_k must be positive. Then, from relation (IV.11), it is evident that for each i, $0 < \lambda < \frac{Np_i}{2L_i}$ must hold. Therefore, the smallest of these has to be set as the upper

bound of λ, that is, $0 < \lambda < \min_i\{\frac{Np_i}{2L_i}\}$. The largest upper bound of the step size is attained when we have Lipschitz probability distribution for primal sampling, namely, $p_i = \frac{L_i}{\sum_{i=1}^N L_i}$. □

The following result is on a.s. convergence of the sequence of iterates that are obtained from Algorithm 1. The proof is presented in Appendix A.6.

THEOREM 2 Suppose that Assumption 1 holds, that $0 < \lambda < \min_i\{\frac{Np_i}{2L_i}\}$, and that the merit function $V : \mathbb{R}^{(N+1)d} \to \mathbb{R}$ is nonnegative and such that for all sequences $(z^k)_{k\in\mathbb{N}}$ satisfying $V(z^k) \to 0$ we have $\|\bar{\mathscr{R}}z^k\| \to 0$. Then $(z^k)_{k\in\mathbb{N}}$ in Algorithm 1 converges a.s. to a random variable $\bar{z} \in \operatorname{zer}(\bar{\mathscr{R}})$. Moreover, $(x^k)_{k\in\mathbb{N}}$ and $(y_i^k)_{k\in\mathbb{N}}$ converge a.s. to random variables $\bar{x} \in \mathscr{X}^\star$ and $\bar{y}_i = \nabla f_i(\bar{x})$ respectively. □

6. Numerical experiments

We solve a regularized logistic regression problem for binary classification of the form

$$\underset{x=(w,b)}{\text{minimize}} \quad \sum_{i=1}^{N} \log(1+e^{\theta_i^T w+b}) - u_i(\theta_i^T w+b) + \tfrac{\xi}{2}\|w\|_2^2, \tag{IV.12}$$

where $\theta_i \in \mathbb{R}^d$ and $u_i \in \{0,1\}$ are training data and labels respectively, and $\xi > 0$ is a regularization parameter. The optimization problem variable is $x = (w,b)$ with $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$.

In the hybrid acceleration scheme, we use L–SVRG as the basic method and either Anderson acceleration or lBFGS as the acceleration algorithms. The following lists the algorithms that are used in the numerical experiments

- GD: Gradient descent method with fixed step size,
- L–SVRG: Loopless Stochastic Variance Reduced Gradient method,
- L–SVRG+AA: L–SVRG as the basic method combined with Anderson acceleration,
- L–SVRG+lBFGS: L–SVRG as the basic method combined with limited–memory BFGS.

In order to use Anderson acceleration as the acceleration algorithm in the hybrid acceleration scheme, an associated fixed–point mapping of problem (IV.12) is needed. Let $F(x)$ denote the objective function of problem (IV.12). Then, the associated mapping of the problem that is used by Anderson acceleration is given by

$$\mathscr{T}_{gd}(x) = x - \lambda\nabla F(x)$$

for $\lambda \in (0, 2/L)$, where L is smoothness modulus of F. On the other hand, since the objective function at hand has no non–smooth part, the lBFGS algorithm can also be utilized in the hybrid acceleration scheme to solve problem (IV.12). Unlike Anderson acceleration, lBFGS method does not need an associated fixed–point map of the problem, rather, it requires gradients of the objective function in order to find a solution. See Appendix B.1 and Appendix B.2 for descriptions of Anderson acceleration and lBFGS methods, respectively.

A rough approximation of the per–iteration count of floating point operations for the different algorithms are as follows

- $4Nd$ for gradient descent,
- $12Nd$ for L–SVRG,
- $4Nd + \frac{4}{3}m^3 + 2m^2d$ for Anderson acceleration,
- $4Nd + 2d^2 + 13md + \xi_{bt}4Nd$ for lBFGS,

where, N is the number of the component functions (which is the same as the number of samples in the training dataset), d is the dimension of the optimization problem variable, m is the size of memory stack for either Anderson acceleration or lBFGS and ξ_{bt} is a coefficient to include an approximate average cost for backtracking line search of lBFGS.

Numerical simulations are done using two datasets; *UCI Madelon* [Chang and Lin, 2011] with 2000 samples and 500 features, and *UCI Sonar* [Chang and Lin, 2011] with 208 samples and 60 features. In the numerical experiments, the regularization parameter in the objective function is set to $\xi = 0.01$, we used a memory size of $m = 5$ for both Anderson acceleration and lBFGS, the constants of the safeguard condition of the hybrid acceleration scheme is $C = D = 10^6$, and $\delta = 10^{-6}$, and the parameter K_0 is set to the number of samples of the associated dataset. Moreover, we used the merit function as is defined in (IV.10).

In Figure 1 and Figure 2, the left plots show relative value of objective function versus step number (which is basically equal to the total number of full gradient evaluations up to that step), and the right plot illustrates the relative value of objective function versus weighted iteration counts. The weighted iteration is intended to include a rough approximation of computational cost in such a way that different methods at each weighted iteration have roughly the same computational expense. Therefore, it provides a better comparison in terms of computational complexity among different algorithms. The simulation results show remarkable improvement in convergence rate and overall computational cost of the hybrid acceleration scheme compared to those of the basic method.

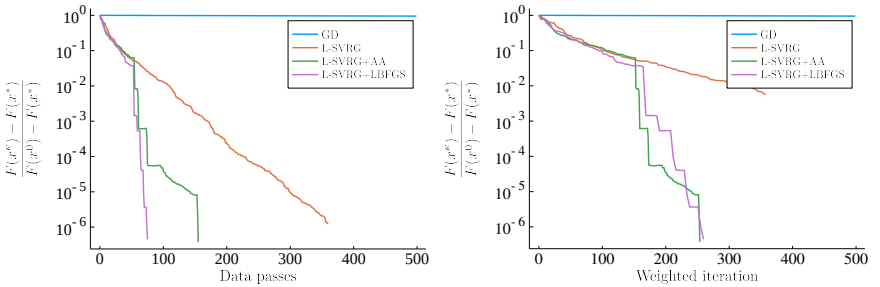

Figure 1: Normalized sub-optimality vs. number of passes over data (the plot to the left) and weighted iteration number (the plot to the right) for the logistic regression problem (IV.12), on *UCI Madelon* dataset (2000 samples, 500 features), solved using GD, L–SVRG, L–SVRG+AA and L–SVRG+lBFGS methods with regularization parameter $\xi = 0.01$.

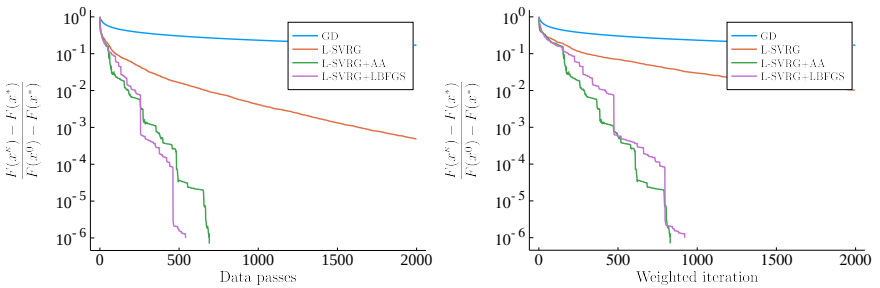

Figure 2: Normalized sub-optimality vs. number of passes over data (the plot to the left) and weighted iteration number (the plot to the right) for the logistic regression problem (IV.12), on *UCI Sonar* dataset (208 samples, 60 features), solved using GD, L–SVRG, L–SVRG+AA and L–SVRG+lBFGS methods with regularization parameter $\xi = 0.01$.

7. Conclusion

In this paper, we proposed and showed almost sure convergence of a hybrid acceleration scheme. It combines a globally convergent variance reduced stochastic gradient method—the basic method—with a fast locally convergent method—the acceleration method—to benefit from the strengths of both methods; global convergence of the basic method and fast local convergence of the acceleration method. Our numerical experiments show that our algorithm performs significantly better than the basic method in isolation, while preserving global convergence guarantees that the local acceleration methods lack.

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

Appendix A

In what follows, we provide the proofs of the propositions and the theorems that are not addressed in the body of the paper.

A.1 Proof of Proposition 1

From $\bar{\mathscr{R}} z^\star = 0$ and for any $\lambda > 0$ we have

$$\begin{cases} x^\star - \operatorname{prox}_{\lambda g}\left(x^\star - \frac{\lambda}{N}\sum_{i=1}^N y_i^\star\right) = 0 \\ y_1^\star = \nabla f_1(x^\star) \\ \quad\vdots \\ y_N^\star = \nabla f_N(x^\star) \end{cases} \iff \quad x^\star - \operatorname{prox}_{\lambda g}\left(x^\star - \lambda \nabla F(x^\star)\right) = 0$$

$$\iff \quad 0 \in \nabla F(x^\star) + \partial g(x^\star)$$

$$\iff \quad 0 \in \partial(F+g)(x^\star),$$

where the last equivalence holds due to f_i's and g having full domain [Bauschke and Combettes, 2017, Theorem 16.3 and Corollary 16.48]. Therefore, by Fermat's rule $x^\star$ is a solution of problem (IV.2).

Now suppose that $x_1^\star$ and $x_2^\star$ are two distinct solutions to the problem, that is

$$-\nabla F(x_1^\star) \in \partial g(x_1^\star),$$
$$-\nabla F(x_2^\star) \in \partial g(x_2^\star).$$

Then using the fact that ∂g is monotone and that each ∇f_i is $\frac{1}{L_i}$-cocoercive, we have

$$0 \geq \langle x_2^\star - x_1^\star, \nabla F(x_2^\star) - \nabla F(x_1^\star) \rangle \geq \sum_{i=1}^{N} \tfrac{1}{NL_i} \|\nabla f_i(x_2^\star) - \nabla f_i(x_1^\star)\|^2 \geq 0,$$

which gives that $y_i^\star = \nabla f_i(x_2^\star) = \nabla f_i(x_1^\star)$ for all i's. Hence it follows that $y_i^\star = \nabla f_i(x^\star)$ is unique.

A.2 Proof of Proposition 2

In the following proof, we use nonexpansiveness of the proximal operator and L_i-Lipschitz continuity of $\nabla f_i(x)$ for all i

$$\begin{aligned}
\left\|\bar{\mathscr{R}}\hat{z} - \bar{\mathscr{R}}z\right\|^2 &= \left\|\hat{x} - \operatorname{prox}_{\lambda g}\left(\hat{x} - \tfrac{\lambda}{N}\sum_{i=1}^{N}\hat{y}_i\right) - x + \operatorname{prox}_{\lambda g}\left(x - \tfrac{\lambda}{N}\sum_{i=1}^{N} y_i\right)\right\|^2 \\
&\quad + \sum_{i=1}^{N} \|\hat{y}_i - \nabla f_i(\hat{x}) - y_i + \nabla f_i(x)\|^2 \\
&\leq \left(\|\hat{x} - x\| + \left\|\hat{x} - \tfrac{\lambda}{N}\sum_{i=1}^{N}\hat{y}_i - \left(x - \tfrac{\lambda}{N}\sum_{i=1}^{N} y_i\right)\right\|\right)^2 \\
&\quad + \sum_{i=1}^{N} 2\left(\|\hat{y}_i - y_i\|^2 + \|\nabla f_i(\hat{x}) - \nabla f_i(x)\|^2\right) \\
&\leq 2\|\hat{x} - x\|^2 + 2\left\|\hat{x} - \tfrac{\lambda}{N}\sum_{i=1}^{N}\hat{y}_i - \left(x - \tfrac{\lambda}{N}\sum_{i=1}^{N} y_i\right)\right\|^2 \\
&\quad + \sum_{i=1}^{N} 2\left(\|\hat{y}_i - y_i\|^2 + \|\nabla f_i(\hat{x}) - \nabla f_i(x)\|^2\right) \\
&\leq 6\|\hat{x} - x\|^2 + 4\tfrac{\lambda^2}{N^2}\left\|\sum_{i=1}^{N}(\hat{y}_i - y_i)\right\|^2 \\
&\quad + \sum_{i=1}^{N} 2\left(\|\hat{y}_i - y_i\|^2 + L_i^2\|\hat{x} - x\|^2\right) \\
&\leq (6 + 2\sum_{i=1}^{N} L_i^2)\|\hat{x} - x\|^2 + (2 + 4\tfrac{\lambda^2}{N})\sum_{i=1}^{N}\|\hat{y}_i - y_i\|^2 \\
&\leq \bar{\alpha}\left(\|\hat{x} - x\|^2 + \sum_{i=1}^{N}\|\hat{y}_i - y_i\|^2\right) = \bar{\alpha}\|\hat{z} - z\|^2,
\end{aligned}$$

where

$$\bar{\alpha} = \max\left(6 + 2\sum_{i=1}^{N} L_i^2, 2 + 4\tfrac{\lambda^2}{N}\right).$$

The first inequality is given by the triangle inequality and nonexpansiveness of proximal operator for the first term and by Young's inequality for terms in the sum. The second and third inequalities is given by Young's inequality, and the fourth one is given by $\|a_1+\ldots+a_N\|_2^2 \leq N(\|a_1\|_2^2+\ldots+\|a_N\|_2^2)$. Therefore, $\bar{\mathscr{R}}$ is Lipschitz continuous.

A.3 Proof of Proposition 3

Using the definition of the primal–dual residual operator at $\bar{z}=(\bar{x},\bar{y}_1\ldots,\bar{y}_N)$

$$\bar{\mathscr{R}}\bar{z} \triangleq \begin{pmatrix} \bar{x}-\operatorname{prox}_{\lambda g}\left(\bar{x}-\frac{\lambda}{N}\sum_{i=1}^N \bar{y}_i\right) \\ \bar{y}_1-\nabla f_1(\bar{x}) \\ \vdots \\ \bar{y}_N-\nabla f_N(\bar{x}) \end{pmatrix} = 0$$

gives $\bar{y}_i=\nabla f_i(\bar{x})$ for all i. This in turn yields

$$\bar{x}-\operatorname{prox}_{\lambda g}\left(\bar{x}-\frac{\lambda}{N}\sum_{i=1}^N \nabla f_i(\bar{x})\right) = \bar{x}-\operatorname{prox}_{\lambda g}\left(\bar{x}-\lambda\nabla F(\bar{x})\right)=0,$$

which evidently means that $\bar{x}\in\mathscr{X}^\star$, i.e., $0\in\partial g(\bar{x})+\nabla F(\bar{x})$. Since z^k converges to $\bar{z}$ almost surely, x^k and y_i^k, respectively, converge to $\bar{x}$ and $\bar{y}_i=\nabla f_i(\bar{x})$ almost surely. This concludes the proof.

A.4 Lemmas for proof of Theorem 1

The following lemmas are needed in our proof of Theorem 1.

LEMMA A.1 Let $\bar{\mathscr{R}}$ be the primal–dual residual operator (IV.5), $(x^\star,y^\star)\in\operatorname{zer}(\bar{\mathscr{R}})$, f_i be convex and L_i-Lipschitz continuous, and $y_i^\star=\nabla f_i(x^\star)$ for all $i\in\{1,...,N\}$, then for the iterates given in (IV.7), the following bounds the variance of the primal variable

$$\begin{aligned}\mathbb{E}_k\|x^{k+1}-x^\star\|^2 \leq \|x^k-x^\star\|^2 &- \sum_{i=1}^N \tfrac{2\lambda}{NL_i}\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2 \\ &- \mathbb{E}_k\|x^{k+1}-x^k+\lambda(\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star))\|^2 \\ &+ \lambda^2\,\mathbb{E}_k\|\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star)\|^2.\end{aligned} \tag{IV.13}$$

□

Proof. Using firm nonexpansiveness of the proximal operator, we have

$$\|x^{k+1}-x^\star\|^2 = \left\|\operatorname{prox}_{\lambda g}\left(x^k-\lambda\widehat{\nabla}_{i_k}F(x^k,y^k)\right)-\operatorname{prox}_{\lambda g}\left(x^\star-\lambda\nabla F(x^\star)\right)\right\|^2$$

$$
\begin{aligned}
&\leq \left\| x^k - \lambda \widehat{\nabla}_{i_k} F(x^k, y^k) - (x^\star - \lambda \nabla F(x^\star)) \right\|^2 - \\
&\qquad \| x^k - \lambda \widehat{\nabla}_{i_k} F(x^k, y^k) - \operatorname{prox}_{\lambda g}(x^k - \lambda \widehat{\nabla}_{i_k} F(x^k, y^k)) \\
&\qquad\qquad - (x^\star - \lambda \nabla F(x^\star)) + \operatorname{prox}_{\lambda g}(x^\star - \lambda \nabla F(x^\star)) \|^2 \\
&= \|x^k - x^\star\|^2 - 2\lambda \langle x^k - x^\star, \widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star) \rangle \\
&\quad + \lambda^2 \|\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star)\|^2 \\
&\quad - \|x^{k+1} - x^k + \lambda(\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star))\|^2.
\end{aligned}
$$

The second equality above, is given by $x^\star = \operatorname{prox}_{\lambda g}(x^\star - \lambda \nabla F(x^\star))$ and by the primal update formula $x^{k+1} = \operatorname{prox}_{\lambda g}(x^k - \lambda \widehat{\nabla}_{i_k} F(x^k, y^k))$. Taking expected value conditioned on all available information up to step k, yields

$$
\begin{aligned}
\mathbb{E}_k \|x^{k+1} - x^\star\|^2 &\leq \|x^k - x^\star\|^2 - 2\lambda \langle x^k - x^\star, \nabla F(x^k) - \nabla F(x^\star) \rangle \\
&\qquad - \mathbb{E}_k \|x^{k+1} - x^k + \lambda(\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star))\|^2 \\
&\qquad\qquad + \lambda^2 \mathbb{E}_k \|\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star)\|^2 \\
&\leq \|x^k - x^\star\|^2 - 2\lambda \sum_{i=1}^N \tfrac{1}{NL_i} \|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2 \\
&\qquad - \mathbb{E}_k \|x^{k+1} - x^k + \lambda(\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star))\|^2 \\
&\qquad\qquad + \lambda^2 \mathbb{E}_k \|\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star)\|^2.
\end{aligned}
$$

In the first inequality, we used $\mathbb{E}_k \widehat{\nabla}_{i_k} F(x^k, y^k) = \nabla F(x^k)$ and the second inequality is given by cocoercivity of $\nabla f_i(x)$.

LEMMA A.2 Let $\bar{\mathscr{R}}$ be the primal–dual residual operator (IV.5), $(x^\star, y^\star) \in \operatorname{zer}(\bar{\mathscr{R}})$ and $y_i^\star = \nabla f_i(x^\star)$ for all $i \in \{1, ..., N\}$, then for the iterates given in (IV.7), the following holds:

$$
\begin{aligned}
\mathbb{E}_k \left(\sum_{i=1}^N \tfrac{\lambda}{N\rho_i L_i} \|y_i^{k+1} - y_i^\star\|^2 \right) &= \sum_{i=1}^N \tfrac{\lambda}{NL_i} \|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2 \\
&\qquad + \sum_{i=1}^N (1 - \rho_i) \tfrac{\lambda}{N\rho_i L_i} \|y_i^k - y_i^\star\|^2,
\end{aligned}
\tag{IV.14}
$$

where ρ_i is the probability of ε_i^k being 1 for y_i. □

Proof. By substitution of y_i^{k+1} from (IV.7) we get

$$
\mathbb{E}_k \left(\sum_{i=1}^N \tfrac{\lambda}{N\rho_i L_i} \|y_i^{k+1} - y_i^\star\|^2 \right) = \mathbb{E}_k \left(\sum_{i=1}^N \tfrac{\lambda}{N\rho_i L_i} \|y_i^k + \varepsilon_i^k (\nabla f_i(x^k) - y_i^k) - y_i^\star\|^2 \right)
$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} \tfrac{\lambda}{N\rho_i L_i}\, \mathbb{E}_k \left\| y_i^k + \varepsilon_i^k(\nabla f_i(x^k) - y_i^k) - y_i^\star \right\|^2 \\
&= \sum_{i=1}^{N} \tfrac{\lambda}{N\rho_i L_i} \left(\rho_i \|\nabla f_i(x^k) - y_i^\star\|^2 + (1-\rho_i)\|y_i^k - y_i^\star\|^2 \right) \\
&= \sum_{i=1}^{N} \tfrac{\lambda}{NL_i} \|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2 \\
&\qquad + \sum_{i=1}^{N} (1-\rho_i) \tfrac{\lambda}{N\rho_i L_i} \|y_i^k - y_i^\star\|^2.
\end{aligned}
$$

In the third equality we used the fact that the only random variable in the expression to the right of the second equality is $\varepsilon_i^k \in \{0,1\}$ and the probability of ε_i^k being 1 is assumed to be ρ_i.

LEMMA A.3 Let $\bar{\mathscr{R}}$ be the primal–dual residual operator of the problem, $(x^\star, y^\star) \in \operatorname{zer}(\bar{\mathscr{R}})$ and $y_i^\star = \nabla f_i(x^\star)$ for all $i \in \{1,\ldots,N\}$, then for the iterates given in (IV.7), the following gives the update variance bound:

$$
\begin{aligned}
\mathbb{E}_k \|\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star)\|^2 \leq \sum_{i=1}^{N} \tfrac{2}{N^2 p_i} \left(\|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2 + \|y_i^k - y_i^\star\|^2 \right) \\
- 2\|\tfrac{1}{N} \sum_{i=1}^{N} (y_i^k - y_i^\star)\|^2 - \|\nabla F(x^k) - \nabla F(x^\star)\|^2.
\end{aligned} \tag{IV.15}
$$

□

Proof. We start with the left-hand side of (IV.15). Using the identity $\mathbb{E}\|X\|^2 = \|\mathbb{E}X\|^2 + \mathbb{E}\|X - \mathbb{E}X\|^2$, gives

$$
\begin{aligned}
\mathbb{E}_k \|\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star)\|^2 = \|\nabla F(x^k) - \nabla F(x^\star)\|^2 \\
+ \mathbb{E}_k \|\widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^k)\|^2.
\end{aligned} \tag{IV.16}
$$

Now for the second term in the right-hand side, substitution of $\widehat{\nabla}_{i_k} F(x^k, y^k)$ yields

$$
\begin{aligned}
&\mathbb{E}_k \left\| \widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^k) \right\|^2 \\
&\qquad = \mathbb{E}_k \left\| \tfrac{1}{N p_{i_k}} (\nabla f_{i_k}(x^k) - y_{i_k}^k) + \tfrac{1}{N} \sum_{i=1}^{N} y_i^k - \nabla F(x^k) \right\|^2 \\
&\qquad = \mathbb{E}_k \left\| \tfrac{1}{N p_{i_k}} (\nabla f_{i_k}(x^k) - \nabla f_{i_k}(x^\star) + y_{i_k}^\star - y_{i_k}^k) \right.
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \tfrac{1}{N}\sum_{i=1}^{N} y_i^k - \tfrac{1}{N}\sum_{i=1}^{N} y_i^\star + \nabla F(x^\star) - \nabla F(x^k)\Big\|^2 \\
&\leq 2\,\mathbb{E}_k \left\| \tfrac{1}{N p_{i_k}} (\nabla f_{i_k}(x^k) - \nabla f_{i_k}(x^\star)) - (\nabla F(x^k) - \nabla F(x^\star)) \right\|^2 \\
&\quad + 2\,\mathbb{E}_k \left\| \tfrac{1}{N p_{i_k}} (y_{i_k}^k - y_{i_k}^\star) - (\tfrac{1}{N}\sum_{i=1}^{N} y_i^k - \tfrac{1}{N}\sum_{i=1}^{N} y_i^\star) \right\|^2 \\
&= 2\,\mathbb{E}_k \left\| \tfrac{1}{N p_{i_k}} (\nabla f_{i_k}(x^k) - \nabla f_{i_k}(x^\star)) \right\|^2 - 2\left\| \nabla F(x^k) - \nabla F(x^\star) \right\|^2 \\
&\quad + 2\,\mathbb{E}_k \left\| \tfrac{1}{N p_{i_k}} (y_{i_k}^k - y_{i_k}^\star) \right\|^2 - 2\left\| \tfrac{1}{N}\sum_{i=1}^{N} (y_i^k - y_i^\star) \right\|^2 \\
&= 2\sum_{i=1}^{N} \tfrac{1}{N^2 p_i} \left\| \nabla f_i(x^k) - \nabla f_i(x^\star) \right\|^2 - 2\left\| \nabla F(x^k) - \nabla F(x^\star) \right\|^2 \\
&\quad + 2\sum_{i=1}^{N} \tfrac{1}{N^2 p_i} \left\| y_i^k - y_i^\star \right\|^2 - 2\left\| \tfrac{1}{N}\sum_{i=1}^{N} (y_i^k - y_i^\star) \right\|^2 .
\end{aligned}
$$

The inequality above is given by Cauchy-Schwarz and Young's inequalities. The third equality is given by the identity $\mathbb{E}\,\|X - \mathbb{E}X\|^2 = \mathbb{E}\,\|X\|^2 - \|\,\mathbb{E}X\|^2$. Substituting in (IV.16) yields

$$
\begin{aligned}
&\mathbb{E}_k \left\| \widehat{\nabla}_{i_k} F(x^k, y^k) - \nabla F(x^\star) \right\|^2 \\
&\quad \leq \sum_{i=1}^{N} \tfrac{2}{N^2 p_i} \left\| \nabla f_i(x^k) - \nabla f_i(x^\star) \right\|^2 - \left\| \nabla F(x^k) - \nabla F(x^\star) \right\|^2 \\
&\qquad + \sum_{i=1}^{N} \tfrac{2}{N^2 p_i} \left\| y_i^k - y_i^\star \right\|^2 - 2\left\| \tfrac{1}{N}\sum_{i=1}^{N} (y_i^k - y_i^\star) \right\|^2 .
\end{aligned}
$$

A.5 Proof of Theorem 1

We first use the definition of Γ:

$$
\|z^k - z^\star\|_\Gamma^2 = \|x^k - x^\star\|^2 + \sum_{i=1}^{N} \tfrac{\lambda}{N\rho_i L_i} \|y_i^k - y_i^\star\|^2. \tag{IV.17}
$$

Then, adding (IV.13) to (IV.14) and reordering the terms, yield

$$
\begin{aligned}
&\mathbb{E}_k \|x^{k+1} - x^\star\|^2 + \mathbb{E}_k \left(\sum_{i=1}^{N} \tfrac{\lambda}{N\rho_i L_i} \|y_i^{k+1} - y_i^\star\|^2 \right) \\
&\quad \leq \|x^k - x^\star\|^2 - \sum_{i=1}^{N} \tfrac{2\lambda}{NL_i} \|\nabla f_i(x^k) - \nabla f_i(x^\star)\|^2
\end{aligned}
$$

$$
\begin{aligned}
&+\lambda^2\,\mathbb{E}_k\,\|\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star)\|^2\\
&-\mathbb{E}_k\,\|x^{k+1}-x^k+\lambda(\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star))\|^2\\
&+\sum_{i=1}^N\tfrac{\lambda}{NL_i}\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2+\sum_{i=1}^N(1-\rho_i)\tfrac{\lambda}{N\rho_iL_i}\|y_i^k-y_i^\star\|^2\\
=\;&\|x^k-x^\star\|^2+\sum_{i=1}^N\tfrac{\lambda}{N\rho_iL_i}\|y_i^k-y_i^\star\|^2-\sum_{i=1}^N\tfrac{2\lambda}{NL_i}\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2\\
&+\lambda^2\,\mathbb{E}_k\,\|\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star)\|^2\\
&-\mathbb{E}_k\,\|x^{k+1}-x^k+\lambda(\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star))\|^2\\
&+\sum_{i=1}^N\tfrac{\lambda}{NL_i}\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2-\sum_{i=1}^N\tfrac{\lambda}{NL_i}\|y_i^k-y_i^\star\|^2
\end{aligned}
$$

Now, we use (IV.17) and (IV.15) in the above inequality, which gives

$$
\begin{aligned}
\mathbb{E}_k\,\|z^{k+1}-z^\star\|_\Gamma^2\le\;&\|z^k-z^\star\|_\Gamma^2-\sum_{i=1}^N\tfrac{2\lambda}{NL_i}\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2-\lambda^2\|\nabla F(x^k)-\nabla F(x^\star)\|^2\\
&+\sum_{i=1}^N\tfrac{\lambda}{NL_i}\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2+\sum_{i=1}^N(\tfrac{2\lambda^2}{N^2p_i}-\tfrac{\lambda}{NL_i})\|y_i^k-y_i^\star\|^2\\
&+\sum_{i=1}^N\tfrac{2\lambda^2}{N^2p_i}\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2-\tfrac{2\lambda^2}{N^2}\|\sum_{i=1}^N(y_i^k-y_i^\star)\|^2\\
&-\mathbb{E}_k\,\|x^{k+1}-x^k+\lambda(\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star))\|^2\\
=\;&\|z^k-z^\star\|_\Gamma^2-\sum_{i=1}^N(\tfrac{\lambda}{NL_i}-\tfrac{2\lambda^2}{N^2p_i})\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2\\
&-\sum_{i=1}^N(\tfrac{\lambda}{NL_i}-\tfrac{2\lambda^2}{N^2p_i})\|y_i^k-y_i^\star\|^2-\tfrac{2\lambda^2}{N^2}\|\sum_{i=1}^N(y_i^k-y_i^\star)\|^2\\
&-\mathbb{E}_k\,\|x^{k+1}-x^k+\lambda(\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star))\|^2\\
&-\lambda^2\|\nabla F(x^k)-\nabla F(x^\star)\|^2\\
=\;&\|z^k-z^\star\|_\Gamma^2-\zeta_k
\end{aligned}
$$

where

$$
\begin{aligned}
\zeta_k=\;&\sum_{i=1}^N(\tfrac{\lambda}{NL_i}-\tfrac{2\lambda^2}{N^2p_i})\|\nabla f_i(x^k)-\nabla f_i(x^\star)\|^2\\
&+\sum_{i=1}^N(\tfrac{\lambda}{NL_i}-\tfrac{2\lambda^2}{N^2p_i})\|y_i^k-y_i^\star\|^2+\tfrac{2\lambda^2}{N^2}\|\sum_{i=1}^N(y_i^k-y_i^\star)\|^2
\end{aligned}
$$

$$+\mathbb{E}_k\|x^{k+1}-x^k+\lambda(\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star))\|^2+\lambda^2\|\nabla F(x^k)-\nabla F(x^\star)\|^2.$$

This proves the first part of the theorem. To show a.s. convergence of $(x^k)_{k\in\mathbb{N}}$ to a random variable in $\mathscr{X}^\star$, in view of [Combettes and Pesquet, 2015, Proposition 2.3], we need to show that the set of sequential cluster points of the sequence $(x^k)_{k\in\mathbb{N}}$ is a subset of $\mathscr{X}^\star$, then a.s. convergence of $(x^k)_{k\in\mathbb{N}}$ to an $\mathscr{X}^\star$-valued random variable will follow. In the following, all limits and convergences are to be considered to hold almost surely, also if it is not explicitly written.

We choose λ such that $0<\lambda<\min_i\{\frac{Np_i}{2L_i}\}$ holds. This choice of λ, enforces non-negativeness to all the coefficients in relation (IV.11); thus, we have $\zeta_k\geq 0$ for all $k\in\mathbb{N}$. Now using [Combettes and Pesquet, 2015, Proposition 2.3.i], we get that $(\zeta_k)_{k\in\mathbb{N}}$ is a.s. summable. It follows by a.s. summability of $(\zeta_k)_{k\in\mathbb{N}}$ that both $(y_i^k)_{k\in\mathbb{N}}$ and $(\nabla f_i(x^k))_{k\in\mathbb{N}}$ converge to $\nabla f_i(x^\star)$ almost surely. This in turn means that, as $k\to\infty$, $\widehat{\nabla}_{i_k}F(x^k,y^k)\to\nabla F(x^\star)$ almsot surely. Moreover, a.s. summability of $(\zeta_k)_{k\in\mathbb{N}}$ implies that $(\mathbb{E}_k(\|x^{k+1}-x^k+\lambda(\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star))\|^2))_{k\in\mathbb{N}}$ a.s. converges to zero as $k\to\infty$ and since $\widehat{\nabla}_{i_k}F(x^k,y^k)-\nabla F(x^\star)\to 0$, we have that $\mathbb{E}_k(\|x^{k+1}-x^k\|^2)\to 0$, which implies $x^{k+1}-x^k\to 0$ almost surely. Now, since the Euclidean space $\mathbb{R}^{(N+1)d}$ is separable and $\operatorname{zer}(\bar{\mathscr{R}})$ is closed, using [Combettes and Pesquet, 2015, Proposition 2.3.iii], for every $z^\star\in\operatorname{zer}(\bar{\mathscr{R}})$, the sequence $(\|z^k-z^\star\|)_{k\in\mathbb{N}}$ converges almost surely. Summability of $(\zeta_k)_{k\in\mathbb{N}}$ implies that $\|z^k-z^\star\|_\Gamma^2-\|x^k-x^\star\|^2\to 0$, and therefore, we infer that for every $x^\star\in\mathscr{X}^\star$, the sequence $(\|x^k-x^\star\|^2)_{k\in\mathbb{N}}$ is a.s. convergent, and therefore, the sequence $(x^k)_{k\in\mathbb{N}}$ is bounded. Boundedness of $(x^k)_{k\in\mathbb{N}}$ implies that it has at least one convergent subsequence. Denote this subsequence by $(x^{n_k})_{k\in\mathbb{N}}$. Now, from the optimality condition of the proximal operator we get

$$\begin{aligned}
0\in\lambda\partial g(x^{n_k+1})+(x^{n_k+1}-(x^{n_k}-\lambda\widehat{\nabla}_{i_k}F(x^{n_k},y^{n_k})))\Leftrightarrow\\
\lambda\nabla F(x^{n_k+1})-\lambda\nabla F(x^{n_k+1})\in\lambda\partial g(x^{n_k+1})+(x^{n_k+1}-(x^{n_k}-\lambda\widehat{\nabla}_{i_k}F(x^{n_k},y^{n_k})))\Leftrightarrow\\
u^{n_k}\in\partial g(x^{n_k+1})+\nabla F(x^{n_k+1})\Leftrightarrow\\
u^{n_k}\in\partial(g+F)(x^{n_k+1})\Leftrightarrow\\
(x^{n_k+1},u^{n_k})\in\operatorname{gra}(\partial(g+F))
\end{aligned}$$

where $u^{n_k}=\lambda^{-1}(x^{n_k}-x^{n_k+1})+\nabla F(x^{n_k+1})-\widehat{\nabla}_{i_k}F(x^{n_k},y^{n_k})$. As $n_k\to\infty$, $x^{n_k}-x^{n_k+1}\to 0$ and $\nabla F(x^{n_k+1})-\widehat{\nabla}_{i_k}F(x^{n_k},y^{n_k})\to 0$. Thus, $u^{n_k}\to 0$ almost surely. In the second to last equivalence above, since ∂F has full domain, we used the identity $\partial(g+F)=\partial g+\partial F$ by [Bauschke and Combettes, 2017, Corollary 16.48]. Let us assume that the subsequence converges to $\bar{x}$, that is $x^{n_k}\to\bar{x}$. Now by [Bauschke and Combettes, 2017, Corollary 25.5] since ∂F has full domain, $\partial(g+F)$ is maximally monotone. Using [Bauschke and Combettes, 2017, Proposition 20.37.ii], we get $(\bar{x},0)\in\operatorname{gra}(\partial(g+F))$ which implies that $0\in\partial g(\bar{x})+\nabla F(\bar{x})$. This clearly means that all sequential cluster points of $(x^k)_{k\in\mathbb{N}}$ belong to $\mathscr{X}^\star$. Now, invoking [Com-

bettes and Pesquet, 2015, Proposition 2.3.iv], implies that $(x^k)_{k\in\mathbb{N}}$ converges almost surely to a $\mathscr{X}^\star$-valued random variable. Invoking Proposition 1 concludes the proof.

A.6 Proof of Theorem 2

In the following proof, all the convergences and limits hold almost surely, even if it is not explicitly mentioned.

Let I_{bm} and I_{aa} be the sets of indices for which the next iterate is obtained by a basic method step and an acceleration algorithm step, respectively. These index sets satisfy $I_{\mathrm{bm}} \cap I_{\mathrm{aa}} = \emptyset$ and $I_{\mathrm{bm}} \cup I_{\mathrm{aa}} = \mathbb{N}$. Note that if the cardinality of I_{aa} is finite, after a finite number of steps, the algorithm will reduce to the basic method, which we know is convergent by Theorem 1. Therefore, we assume that $|I_{\mathrm{aa}}|$ is infinite.

From Theorem 1, for all $k \in I_{\mathrm{bm}}$ and all $z^\star \in \operatorname{zer}(\bar{\mathscr{R}})$ we have

$$\mathbb{E}_k \left\| z^{k+1} - z^\star \right\|_\Gamma^2 \leq \left\| z^k - z^\star \right\|_\Gamma^2 - \zeta_k, \tag{IV.18}$$

where $\zeta_k \geq 0$. Using the identity $\mathbb{E}\|X - \mathbb{E}X\|^2 = \mathbb{E}\|X\|^2 - \|\mathbb{E}X\|^2$, we have $\|\mathbb{E}X\|^2 \leq \mathbb{E}\|X\|^2$. Thus, from (IV.18) and $\zeta_k \geq 0$ for all $k \in I_{\mathrm{bm}}$ we have

$$\left(\mathbb{E}_k \left\| z^{k+1} - z^\star \right\|_\Gamma\right)^2 \leq \mathbb{E}_k \left\| z^{k+1} - z^\star \right\|_\Gamma^2 \leq \left\| z^k - z^\star \right\|_\Gamma^2 - \zeta_k \leq \left\| z^k - z^\star \right\|_\Gamma^2.$$

Therefore, for all $k \in I_{\mathrm{bm}}$ we have

$$\mathbb{E}_k \left\| z^{k+1} - z^\star \right\|_\Gamma \leq \left\| z^k - z^\star \right\|_\Gamma. \tag{IV.19}$$

On the other hand for all $k \in I_{\mathrm{aa}}$, by the triangle inequality and for all $z^\star \in \operatorname{zer}(\bar{\mathscr{R}})$, we have

$$\left\| z^{k+1} - z^\star \right\|_\Gamma \leq \left\| z^k - z^\star \right\|_\Gamma + \left\| z^{k+1} - z^k \right\|_\Gamma.$$

Using the safeguard condition (IV.9) and that $z^+ = z^{k+1}$ for all $k \in I_{\mathrm{aa}}$, we obtain

$$\left\| z^{k+1} - z^\star \right\|_\Gamma \leq \left\| z^k - z^\star \right\|_\Gamma + DV(z^k). \tag{IV.20}$$

Using the fact that $\mathbb{E}_k \left\| z^{k+1} - z^\star \right\|_\Gamma = \left\| z^{k+1} - z^\star \right\|_\Gamma$ holds for all $k \in I_{\mathrm{aa}}$ since the acceleration method is deterministic, by combining (IV.19) and (IV.20), we conclude that

$$\mathbb{E}_k \left\| z^{k+1} - z^\star \right\|_\Gamma \leq \left\| z^k - z^\star \right\|_\Gamma + \sigma_k \tag{IV.21}$$

holds for all $k \in \mathbb{N}$, where

$$\sigma_k = \begin{cases} 0 & k \in I_{\mathrm{bm}} \\ DV(z^k) & k \in I_{\mathrm{aa}} \end{cases}.$$

Due to (IV.8), $(\sigma_k)_{k\in\mathbb{N}}$ is summable and $(\|z^k - z^\star\|)_{k\in\mathbb{N}}$ converges a.s. [Combettes and Pesquet, 2015, Lemma 2.2] and is therefore a.s. bounded. Next, by squaring both sides of (IV.20), for all $k \in I_{\text{aa}}$, we get

$$\left\|z^{k+1} - z^\star\right\|_\Gamma^2 \leq \left\|z^k - z^\star\right\|_\Gamma^2 + 2\left\|z^k - z^\star\right\|_\Gamma DV(z^k) + (DV(z^k))^2.$$

Defining $\beta_k := 2\|z^k - z^\star\|_\Gamma DV(z^k) + (DV(z^k))^2$ and using $\mathbb{E}_k \|z^{k+1} - z^\star\|_\Gamma = \|z^{k+1} - z^\star\|_\Gamma$ for all $k \in I_{\text{aa}}$, we get for all $k \in I_{\text{aa}}$ that

$$\mathbb{E}_k \left\|z^{k+1} - z^\star\right\|_\Gamma^2 \leq \left\|z^k - z^\star\right\|_\Gamma^2 + \beta_k. \tag{IV.22}$$

Since we have concluded that $(\|z^k - z^\star\|_\Gamma)_{k\in I_{\text{aa}}}$ is bounded a.s. and $(V(z^k))_{k\in I_{\text{aa}}}$ is absolutely summable, $(\beta_k)_{k\in I_{\text{aa}}}$ is a.s. absolutely summable as well. Combining (IV.18) and (IV.22) implies that

$$\mathbb{E}_k \left\|z^{k+1} - z^\star\right\|_\Gamma^2 + \nu_k \leq \left\|z^k - z^\star\right\|_\Gamma^2 + \eta_k, \tag{IV.23}$$

where

$$\eta_k = \begin{cases} 0 & k \in I_{\text{bm}} \\ \beta_k & k \in I_{\text{aa}} \end{cases}, \qquad \text{and} \qquad \nu_k = \begin{cases} \zeta_k & k \in I_{\text{bm}} \\ 0 & k \in I_{\text{aa}} \end{cases}.$$

Therefore, by [Combettes and Pesquet, 2015, Proposition 2.3.i], $(\nu_k)_{k\in\mathbb{N}}$ is summable. Now, in [Combettes and Pesquet, 2015, Proposition 2.3] setting $\phi : z \mapsto z^2$, [Combettes and Pesquet, 2015, Proposition 2.3.iii] implies that $(\|z^k - z^\star\|_\Gamma^2)_{k\in\mathbb{N}}$ and evidently $(\|z^k - z^\star\|)_{k\in\mathbb{N}}$ are convergent.

For the last part of the proof, fix $z^\star \in \operatorname{zer}(\bar{\mathscr{R}})$ and denote the set of sequential cluster points of $(z^k)_{k\in\mathbb{N}}$ by $\mathscr{C}$. Since $(\|z^k - z^\star\|)_{k\in\mathbb{N}}$ is convergent, the sequence $(z^k)_{k\in\mathbb{N}}$ is bounded, and therefore, it has at least one convergent subsequence by the Bolzano–Weierstrass theorem. Denote this subsequence by $(z^{n_k})_{k\in\mathbb{N}}$ and its associated sequential cluster point by $z_c^\star = (x_c^\star, y_1^\star, \ldots, y_N^\star)$. As the problem is finite-dimensional, using Lipschitz continuity of the operator $\bar{\mathscr{R}}$ (Proposition 2) we have

$$\|z^{n_k} - z_c^\star\| \to 0 \qquad \Longrightarrow \qquad \left\|\bar{\mathscr{R}} z^{n_k} - \bar{\mathscr{R}} z_c^\star\right\| \to 0$$

which means that $\bar{\mathscr{R}} z^{n_k} \to \bar{\mathscr{R}} z_c^\star$. Note that $(z^{n_k})_{k\in\mathbb{N}}$ is constructed by the points that are generated by either the basic method or the acceleration algorithm. For the subsequence of points in $(z^{n_k})_{k\in\mathbb{N}}$ that are obtained from the basic method, that is $(z^{n_k+1})_{k\in I_{\text{bm}}}$, since $(\nu_{n_k})_{k\in\mathbb{N}}$ is summable, so is $(\zeta_{n_k})_{k\in\mathbb{N}}$. Then, using the same approach as in the last part of the proof of Theroem 1, we can show that $(\bar{\mathscr{R}} z^{n_k+1})_{k\in I_{\text{bm}}}$ converges to zero. For the subsequence of the points in $(z^{n_k})_{k\in\mathbb{N}}$ which are generated by the acceleration algorithm, that is $(z^{n_k+1})_{k\in I_{\text{aa}}}$, it is evident from the definition of the merit function in (IV.10), that convergence of $(V(z^{n_k+1}))_{k\in I_{\text{aa}}}$ to zero—which

is dictated by the safeguard condition—enforces convergence of $(\bar{\mathscr{R}}z^{n_k+1})_{k\in I_{\text{aa}}}$ to zero as well. Therefore, for $(z^{n_k})_{k\in\mathbb{N}}$ as a whole, we have $z^{n_k}\to 0$ as $k\to\infty$. Then, it follows from $\bar{\mathscr{R}}z^{n_k}\to\bar{\mathscr{R}}z_c^\star$ that $\bar{\mathscr{R}}z_c^\star=0$. Thus, $z_c^\star$ belongs to $\operatorname{zer}(\bar{\mathscr{R}})$. The same implication can be made for all other sequential cluster points of $(z^k)_{k\in\mathbb{N}}$ which means that all sequential cluster points of $(z^k)_{k\in\mathbb{N}}$ belong to $\operatorname{zer}(\bar{\mathscr{R}})$, that is $\mathscr{C}\subset\operatorname{zer}(\bar{\mathscr{R}})$. Finally, by [Combettes and Pesquet, 2015, Proposition 2.3.iv], the sequence $(z^k)_{k\in\mathbb{N}}$ converges a.s. to a point $\bar{z}\in\operatorname{zer}(\bar{\mathscr{R}})$. Now, by Proposition 3, $x^k\to\bar{x}$ and for all i, $y_i^k\to\nabla f_i(\bar{x})$ a.s., where $\bar{x}$ is the solution of problem (IV.2). By this, the proof is complete.

Appendix B

B.1 Anderson acceleration

Anderson acceleration can be exploited to accelerate convergence of the the fixed–point iteration of the form

$$x^{k+1}=\mathscr{T}(x^k)$$

where $\mathscr{T}:\mathbb{R}^n\to\mathbb{R}^n$ is either a contraction or an averaged operator. A variant of Anderson acceleration, which is equipped with Tikhonov regularization on its inner least–squares problem, is given in Algorithm B.1 [Scieur et al., 2016].

Algorithm B.1 Anderson Acceleration

1: **input:** $y^0\in\mathbb{R}^d$, $m\geq 1$.
2: **for** $k=0,1,2,\ldots$ **do**
3: set $m_k=\min\{m,k\}$
4: find the iterate using the fixed-point map $x^k=\mathscr{T}(y^k)$
5: form $\mathcal{R}^k=(r^{k-m_k},\ldots,r^k)$ where $r^j=y^j-x^j$ for $j\in\{k-m_k,\ldots,k\}$
6: determine $\alpha^{(k)}=(\alpha_0^{(k)},\ldots,\alpha_{m_k}^{(k)})$ that solves

$$\begin{aligned}&\underset{\alpha^{(k)}\in\mathbb{R}^{m_k+1}}{\text{minimize}} && \left\|\mathcal{R}_n\alpha^{(k)}\right\|_2^2+\xi_k\left\|\alpha^{(k)}\right\|_2^2\\ &\text{subject to} && \mathbf{1}^T\alpha^{(k)}=1\end{aligned}$$

7: $y^{k+1}=\sum_{i=0}^{m_k}\alpha_i^{(k)}x^{k-m_k+i}$
8: **end for**

B.2 Limited–memory BFGS

If the objective function of a convex optimization problem is twice continuously differentiable, an effective way of solving it, is to use quasi–Newton methods. One of the most well-known quasi–Newton methods is the limited–memory BFGS

(lBFGS) which has been vastly used in many areas. lBFGS is a variant of BFGS method that uses a limited amount of computer's memory and in that sense is cheaper than its parent, BFGS method. Hence, unlike BFGS algorithm, its limited–memory version can be used to solve large–scale problems. The lBFGS method can be stated as in Algorithm B.2 [Nocedal and Wright, 2006].

Algorithm B.2 limited–memory BFGS

Define: $s^k = x^{k+1} - x^k$, $u^k = \nabla f(x^{k+1}) - \nabla f(x^k)$ and $\rho_k = ((u^k)^T s^k)^{-1}$.
input: x^0 and the memory stack size $m \geq 1$.
for $k = 1, 2, \ldots$ **do**
    $H_0^k = \frac{(s^{k-1})^T u^{k-1}}{(u^{k-1})^T u^{k-1}} I$
    $q = \nabla f(x^k)$
    for $i = k-1, \ldots, \min\{k-m, 0\}$ **do**
        $\alpha_i = \rho_i (s^i)^T q$
        $q = q - \alpha_i u^i$
    end for
    $r = H_0^k q$
    for $i = \min\{k-m, 0\}, \ldots, k-1$ **do**
        $\beta = \rho_i (u^i)^T r$
        $r = r + s_i(\alpha_i - \beta)$
    end for
    $p^k = r$
    compute $x^{k+1} = x^k - \lambda_k p^k$, where λ_k is to satisfy a line search condition
    if $k > m$ **then**
        discard s^{k-m} and u^{k-m}
        compute and save s^k and u^k
    end if
end for

References

Anderson, D. G. (1965). "Iterative procedures for nonlinear integral equations". *Journal of the ACM* **12**:4, pp. 547–560. DOI: 10.1145/321296.321305.

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

Beck, A. and M. Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". *SIAM journal on imaging sciences* **2**:1, pp. 183–202. DOI: 10.1137/080716542.

Chang, C.-C. and C.-J. Lin (2011). "LIBSVM: a library for support vector machines". *ACM Transactions on Intelligent Systems and Technology* **2** (3). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27.

Combettes, P. L. and J.-C. Pesquet (2011). "Proximal splitting methods in signal processing". In: *Fixed-point algorithms for inverse problems in science and engineering*. Springer, pp. 185–212.

Combettes, P. L. and J.-C. Pesquet (2015). "Stochastic quasi-fejér block-coordinate fixed point iterations with random sweeping". *SIAM Journal on Optimization* **25**:2, pp. 1221–1248.

Davis, D. (2016). "Smart: the stochastic monotone aggregated root-finding algorithm". *arXiv preprint arXiv:1601.00698*.

Defazio, A., F. Bach, and S. Lacoste-Julien (2014). "Saga: a fast incremental gradient method with support for non-strongly convex composite objectives". In: *Advances in neural information processing systems*, pp. 1646–1654.

Giselsson, P. and S. Boyd (2015). "Metric selection in fast dual forward–backward splitting". *Automatica* **62**, pp. 1–10.

Giselsson, P., M. Fält, and S. Boyd (2016). "Line search for averaged operator iteration". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 1015–1022. DOI: 10.1109/CDC.2016.7798401.

Johnson, R. and T. Zhang (2013). "Accelerating stochastic gradient descent using predictive variance reduction". In: *Advances in neural information processing systems*, pp. 315–323.

Kovalev, D., S. Horváth, and P. Richtárik (2020). "Don't jump through hoops and remove those loops: svrg and katyusha are better without the outer loop". In: *Algorithmic Learning Theory*. PMLR, pp. 451–467.

Li, X.-L. (2017). "Preconditioned stochastic gradient descent". *IEEE transactions on neural networks and learning systems* **29**:5, pp. 1454–1466.

Liu, D. C. and J. Nocedal (1989). "On the limited memory bfgs method for large scale optimization". *Mathematical programming* **45**:1, pp. 503–528.

Morin, M. and P. Giselsson (2019). "Svag: unified convergence results for sag-saga interpolation with stochastic variance adjusted gradient descent". *arXiv preprint arXiv:1903.09009*.

Morin, M. and P. Giselsson (2020). "Sampling and update frequencies in proximal variance reduced stochastic gradient methods". *arXiv preprint arXiv:2002.05545*.

Nitanda, A. (2014). "Stochastic proximal gradient descent with acceleration techniques". In: *NIPS*.

Nocedal, J. and S. Wright (2006). *Numerical Optimization*. Springer Science & Business Media.

Parikh, N. and S. Boyd (2014). "Proximal algorithms". *Foundations and Trends in optimization* **1**:3, pp. 127–239.

Rodomanov, A. and Y. Nesterov (2021a). "Greedy quasi-newton methods with explicit superlinear convergence". *SIAM J. Optim.* **31**, pp. 785–811.

Rodomanov, A. and Y. Nesterov (2021b). "New results on superlinear convergence of classical quasi-newton methods". *Journal of Optimization Theory and Applications* **188**, pp. 744–769.

Rodomanov, A. and Y. Nesterov (2021c). "Rates of superlinear convergence for classical quasi-newton methods". *Mathematical Programming*, pp. 1–32.

Rosasco, L., S. Villa, and B. C. Vũ (2020). "Convergence of stochastic proximal gradient algorithm". *Applied Mathematics & Optimization* **82**:3, pp. 891–917.

Roux, N. L., M. Schmidt, and F. R. Bach (2012). "A stochastic gradient method with an exponential convergence rate for finite training sets". In: *Advances in neural information processing systems*, pp. 2663–2671.

Schmidt, M., N. Le Roux, and F. Bach (2017). "Minimizing finite sums with the stochastic average gradient". *Mathematical Programming* **162**:1-2, pp. 83–112.

Scieur, D., F. Bach, and A. d'Aspremont (2017). "Nonlinear acceleration of stochastic algorithms". *Advances in Neural Information Processing Systems* **30**. URL: `https://proceedings.neurips.cc/paper/2017/file/fca0789e7891cbc0583298a238316122-Paper.pdf`.

Scieur, D., A. d'Aspremont, and F. Bach (2016). "Regularized nonlinear acceleration". *Advances In Neural Information Processing Systems* **29**. URL: `https://proceedings.neurips.cc/paper/2016/file/bbf94b34eb32268ada57a3be5062fe7d-Paper.pdf`.

Shalev-Shwartz, S. and T. Zhang (2013). "Stochastic dual coordinate ascent methods for regularized loss minimization". *Journal of Machine Learning Research* **14**:Feb, pp. 567–599.

Smith, D. A., W. F. Ford, and A. Sidi (1987). "Extrapolation methods for vector sequences". *SIAM review* **29**:2, pp. 199–233.

Teo, C. H., A. Smola, S. Vishwanathan, and Q. V. Le (2007). "A scalable modular convex solver for regularized risk minimization". In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 727–736.

Themelis, A. and P. Patrinos (2019). "Supermann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators". *IEEE Transactions on Automatic Control* **64**:12, pp. 4875–4890. DOI: 10.1109/TAC.2019.2906393.

Walker, H. F. and P. Ni (2011). "Anderson acceleration for fixed-point iterations". *SIAM Journal on Numerical Analysis* **49**:4, pp. 1715–1735. DOI: 10.1137/10078356X.

Yang, T., R. Jin, S. Zhu, and Q. Lin (2016). "On data preconditioning for regularized loss minimization". *Machine Learning* **103**:1, pp. 57–79.

Zhang, J., B. O'Donoghue, and S. Boyd (2020). "Globally convergent type-i anderson acceleration for nonsmooth fixed-point iterations". *SIAM Journal on Optimization* **30**:4, pp. 3170–3197.

LUNDS UNIVERSITET

Efficient and Flexible First-Order Optimization Algorithms

Hamed Sadeghi

Department of Automatic Control

Popular science summary of the doctoral thesis *Efficient and Flexible First-Order Optimization Algorithms*, December 2022. The thesis can be downloaded from: http://www.control.lth.se/publications

---

There are many applications in science, engineering, and in human daily life in which some sort of *optimization* is being used. For instance, manufacturers aim at designing processes that maximize efficiency of production lines; shipping companies seek to obtain the best routes for delivering parcels to their destination; design engineers try to find the optimal design of a load-carrying structure; and investors seek to create portfolios that maximize the return while avoiding high risks. To find the *best* (or *optimal*) *solution* for such processes, one usually builds a *mathematical model* to describe the underlying optimization problem. The resulting mathematical model includes a quantitative performance measure of the process under study; this measure depends on characteristics or attributes of the process. There is also the possibility of reflecting physical limitations of the underlying process on the mathematical model. In most cases, the mathematical model (of the optimization problem) is complex and includes many parameters and variables; therefore, computer-based *optimization algorithms* are usually used to find its solution. An optimization algorithm is a program that takes a mathematical optimization problem, and after performing a set of mathematical operations on it, finds and returns an approximation to the true solution.

Depending on the mathematical model of the optimization problem, there are a variety of algorithms that can be used to solve it; however, not all of them perform equally fast and efficient. This thesis is an attempt to improve performance of a family of optimization algorithms such that they can solve optimization problems—that they are applicable to—faster and more efficiently. In this thesis, several novel optimization algorithms are proposed. These algorithms are mainly built upon the existing ones, however, they are altered such that, they can exhibit a more favorable behavior than their existing counterparts, i.e., they can find a solution of mathematical optimization problems considerably faster. For instance, for an optimization problem that we considered, using the proposed algorithms, we managed to achieve a 5-10 times speed up—compared to an existing counterpart algorithm—in finding a solution.