ACCELERATED AND INEXACT FORWARD-BACKWARD ALGORITHMS

SILVIA VILLA *, SAVERIO SALZO †, LUCA BALDASSARRE ‡, AND ALESSANDRO VERRI §

Abstract. We propose a convergence analysis of accelerated forward-backward splitting methods for composite function minimization, when the proximity operator is not available in closed form, and can only be computed up to a certain precision. We prove that the $1/k^2$ convergence rate for the function values can be achieved if the admissible errors are of a certain type and satisfy a sufficiently fast decay condition. Our analysis is based on the machinery of estimate sequences first introduced by Nesterov for the study of accelerated gradient descent algorithms. Furthermore, we give a global complexity analysis, taking into account the cost of computing admissible approximations of the proximal point. An experimental analysis is also presented.

Key words. convex optimization, accelerated forward-backward splitting, inexact proximity operator, estimate sequences, total variation

AMS subject classifications. 90C25, 49M07, 65K10, 94A08

1. Introduction.

Let $\mathcal{H}$ be a Hilbert space and consider the optimization problem

$$\inf_{x\in\mathcal{H}} f(x) + g(x) =: F(x), \tag{P}$$

where

H1) $g : \mathcal{H} \to \overline{\mathbb{R}}$ is proper, lower semicontinuous (l.s.c.) and convex,

H2) $f : \mathcal{H} \to \mathbb{R}$ is convex differentiable and ∇f is L-Lipschitz continuous on $\mathcal{H}$ with $L > 0$, namely

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{H}.$$

We denote by F_* the infimum of F. We do not require in general the infimum to be attained, neither to be finite. It is well-known that problem (P) covers a wide range of signal recovery problems (see [18] and references therein), including constrained and regularized least-squares problems [27, 25, 51, 21], (sparse) regularization problems in image processing, such as total variation denoising and deblurring (see e.g. [50, 13, 12]), as well as machine learning tasks involving nondifferentiable penalties (see e.g. [4, 23, 42]).

The variety of applications to real life problems stimulated the search of simple first-order methods to solve (P), which can be applied to large scale problems. In this area, a significant amount of research has been devoted to *forward–backward splitting methods*, that allow to decouple the contributions of the functions f and g in a gradient descent step determined by f and in a backward implicit step induced by g [17, 18, 35]. These schemes are also known under the name of *proximal gradient*

*Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy (Silvia.Villa@iit.it)

†DIBRIS, University of Genova, Via Dodecaneso 35, 16145, Genova, Italy (Saverio.Salzo@unige.it).

‡University College London, Dept. of Computer Science, Gower Street, London WC1E 6BT, United Kingdom l.baldassarre@cs.ucl.ac.uk

§DIBRIS, University of Genova, Via Dodecaneso 35, 16145, Genova, Italy (Alessandro.Verri@unige.it).

methods [61], since the implicit step relies on the computation of the so called proximity operator, introduced by Moreau in [39]. Though appealing for their simplicity, gradient-based methods often exhibit a slow speed of convergence. For this reason, resorting to the ideas contained in the work of Nesterov [44], there has recently been an active interest in accelerations and modifications of the classical forward-backward splitting algorithm [61, 45, 7]. We will study the following general accelerated scheme

$$\left| \begin{array}{l} x_{k+1} = \operatorname{prox}_{\lambda_k g}(y_k - \lambda_k \nabla f(y_k)), \\ y_{k+1} = c_{1,k} x_{k+1} + c_{2,k} x_k + c_{3,k} y_k, \end{array} \right. \tag{1.1}$$

for suitably chosen constants $c_{i,k}$, $(i = 1, 2, 3, k \in \mathbb{N})$ and parameters $\lambda_k > 0$ — where $\operatorname{prox}_{\lambda_k g} : \mathcal{H} \to \mathcal{H}$ denotes the *proximity operator* associated to $\lambda_k g$. In particular, choosing $c_{3,k} = 0$, procedure (1.1) encompasses the popular *Fast Iterative Shrinkage Thresholding Algorithm* (FISTA), whose optimal (in the sense of [43]) $1/k^2$ convergence rate for the objective values $F(x_k) - F_*$ has been proved in [7]. Furthermore, the effectiveness of such accelerations has been tested empirically on several relevant problems (see e.g. [6, 8]).

Unfortunately, the proximity operator is in general not available in exact form or its computation may be very demanding. Just to mention some examples, this happens when applying proximal methods to image deblurring with total variation [12, 6, 26], or to structured sparsity regularization problems in machine learning and inverse problems [67, 28, 33, 42, 49, 2]. In those cases, the proximity operator is usually computed using ad hoc algorithms, and therefore inexactly. See [17] for a list of possible approaches. In the end, the entire procedure for solving problem (P) is constituted by two nested loops: an external one of type (1.1) and an internal one which serves to approximately compute the proximity operator occurring in the first row of (1.1). Hence, the problem of studying the convergence of accelerated forward-backward algorithms under possible perturbations of proximal points arises. In [6], FISTA is applied to the TV image deblurring problem and empirically it is shown to possibly generate divergent sequences when the prox subproblem is solved inexactly. However, no theoretical analysis is carried out for the role of inexactness in the convergence and acceleration properties of the algorithm.

1.1. Main contributions. From a theoretical point of view, the contribution of this paper is threefold: first, we show that by considering a suitable notion of admissible approximation of the proximal point, it is possible to get quadratic convergence of the inexact version of the accelerated forward-backward scheme (1.1). In particular, we prove that the proposed algorithm shares the $1/k^2$ convergence rate in the objective values if the computation of the proximity operator at the k-th step is performed up to a precision ε_k, with $\varepsilon_k = O(1/k^q)$ and $q > 3/2$. This assumption clearly implies summability of the errors, which is a common requirement in similar contexts (see e.g. [48, 18]). We underline however that, for slower convergence rates, summability can be avoided and the requirement $\varepsilon_k = O(1/k^q)$ with $q > 1/2$, is sufficient. The second main contribution of the paper is the study of the global iteration complexity of (1.1), which takes also into account the cost of computing admissible approximations of the proximity operator. Furthermore, we show that the proposed inexactness criterion has an equivalent formulation in terms of duality gap, that can be easily checked in practice. This allows to handle most significant penalty terms and different algorithms to compute the proximal point, as for instance those in [12, 19, 14]. This resolves the issue of convergence and applicability of the two-loops algorithm for many real-life

problems, in the same spirit of [15].

The third contribution concerns the techniques we employ to obtain the result. The algorithm derivation relies on the machinery of the estimate sequences. Leveraging on the ideas developed in [52], we propose a flexible method to build estimate sequences, that can be easily adapted to deal with inexactness in accelerated forward-backward algorithms. It is worth to mention that this framework includes the well-known FISTA [7].

Finally, we performed numerical experiments investigating the impact of errors on the acceleration property. We also illustrate the effectiveness of the proposed notion of inexactness on two real-life problems, making performance comparisons with the non accelerated version, and a benchmark primal-dual algorithm.

1.2. Related Work. Forward-backward algorithms belong to the wider class of proximal splitting methods [17]. All these methods require the computation of the proximity operator, consequently approximations of proximal points have been studied in a number of papers, and the following list does not claim to be exhaustive. For non accelerated schemes, convergence in the presence of errors has been addressed in various contexts ranging from proximal point algorithms [3, 48, 29, 34, 20, 19, 1, 59], hybrid extragradient-proximal point algorithms [55, 56, 57, 63], generalized proximal algorithms using Bregman distances [24, 58, 11] and forward-backward splitting [18].

On the other hand, only very recently, accelerated proximal methods under inexact evaluation of the proximity operator have been studied. In [31, 52] the classical *proximal point algorithm* is treated ($f = 0$ in (1.1)). Paper [38] considers inexact *accelerated hybrid extragradient-proximal* methods, but actually the framework is shown to include only the case of the *exact* accelerated forward-backward algorithm. In [22], convergence rates for an *accelerated projected-subgradient* method is proved. The case of an exact projection step is considered, and the authors assume the availability of an oracle that yields global lower and upper bounds on the function. Although interesting, it leads to a slower convergence rates than proximal-gradient methods. Summarizing, none of the studies above covers the case of accelerated inexact forward-backward algorithms.

Finally, we mention the subsequent, but independent, work [54], where an analysis of an accelerated proximal-gradient method with inexact proximity operator is given too, and the same convergence rates are proved. While the accelerated scheme is very similar (though not exactly equal[1]), the employed techniques are completely different. In particular, the estimate sequences framework which motivates the updating rules for the parameters and auxiliary sequences are not used in [54]. The inexactness notion is different as well: our choice is more demanding, but leads to a better (weaker) dependence on the errors decay. For instance, in [54] the authors obtain convergence of the algorithm for $\varepsilon_k = O(1/k^{1+\delta})$, while we only need $\varepsilon_k = O(1/k^{1/2+\delta})$, and the optimal convergence rate of the algorithm for $\varepsilon_k = O(1/k^{2+\delta})$, while Theorem 4.4 requires only $\varepsilon_k = O(1/k^{3/2+\delta})$. For a comparison between the two errors see Section 2. For completeness, in Appendix A we show that the framework of estimate sequences can handle the type of errors considered in [54] as well, but only $1/k$ convergence rate can be obtained.

Note also that none of the above mentioned papers study the rate of convergence of the nested algorithm, as we do in Section 6.

[1] There, the sequence y_k in (1.1), is updated by setting $c_{3,k} = 0$, and the choice of the parameters $c_{1,k}, c_{2,k}$ is different too.

1.3. Outline of the paper. In Section 2, we give a notion of admissible approximation of proximal points and discuss its applicability. Section 3 reviews the framework of Nesterov's estimate sequences and gives a general updating rule for recursively constructing estimate sequences for convex problems. In Section 4, we present a new general accelerated scheme for forward-backward splitting algorithms and a convergence theorem under admissible approximations of proximal points. In Section 5, we rewrite the algorithm in equivalent forms, which encompass other popular algorithms. In Section 6, we discuss the subproblem of computing inexact proximal points and the complexity of the resulting global nested algorithm. Finally, Section 7 contains a numerical evaluation of the effect of errors in the computation of the proximal points on the forward-backward algorithm (1.1).

2. Inexact proximal points. The algorithms analyzed in this paper are based on the computation of the proximity operator of a convex function, introduced by Moreau [39, 40, 41], and then made popular in the optimization literature by Martinet [36] and Rockafellar [48, 47].

Let $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ be the extended real line. For a proper, convex and l.s.c. function $g : \mathcal{H} \to \overline{\mathbb{R}}$, $\lambda > 0$ and $y \in \mathcal{H}$, *the proximal point of y with respect to λg* is defined by setting

$$\operatorname{prox}_{\lambda g}(y) := \operatorname*{argmin}_{x \in \mathcal{H}} \left\{ g(x) + \frac{1}{2\lambda}\|x - y\|^2 \right\} \tag{2.1}$$

and the mapping $\operatorname{prox}_{\lambda g} : \mathcal{H} \to \mathcal{H}$ is called the *proximity operator of λg*. If we let $\Phi_\lambda(x) = g(x) + \frac{1}{2\lambda}\|x-y\|^2$, the first order optimality condition for a convex minimum problem yields

$$z = \operatorname{prox}_{\lambda g}(y) \iff 0 \in \partial\Phi_\lambda(z) \iff \frac{y - z}{\lambda} \in \partial g(z), \tag{2.2}$$

where ∂ denotes the subdifferential operator.

We already noted that, from a practical point of view, it is essential to replace the proximal point with an approximate version of it.

2.1. The proposed notion. We employ here a concept of approximation of the proximal point based on ε-subdifferential, which is indeed a relaxation of condition (2.2). We recall that, for $\varepsilon \geq 0$, the *ε-subdifferential of g at the point $z \in \operatorname{dom} g$* is the set $\partial_\varepsilon g(z) = \{\xi \in \mathcal{H} : g(x) \geq g(z) + \langle x - z, \xi\rangle - \varepsilon, \, \forall x \in \mathcal{H}\}$.

Definition 2.1. *Let $\varepsilon \geq 0$. We say that $z \in \mathcal{H}$ is an* approximation of $\operatorname{prox}_{\lambda g}(y)$ with ε-precision *and we write $z \cong_\varepsilon \operatorname{prox}_{\lambda g}(y)$ if and only if*

$$\frac{y - z}{\lambda} \in \partial_{\frac{\varepsilon^2}{2\lambda}} g(z). \tag{2.3}$$

Note that if $z \cong_\varepsilon \operatorname{prox}_{\lambda g}(y)$, then necessarily $z \in \operatorname{dom} g$; therefore, the allowed approximations are always feasible. This notion has first been proposed, in the context of the proximal point algorithm, in [34] and used successfully in e.g. [1, 19, 52]. A relative version of criterion (2.3) has recently been proposed for non accelerated proximal methods in the preprint [37], which allows to interpret the (exact) forward-backward splitting algorithm as an instance of an inexact proximal point algorithm.

Example 1. *We describe the case where g is the indicator function of a closed and convex set C, and the proximity operator is consequently the projection onto C,*

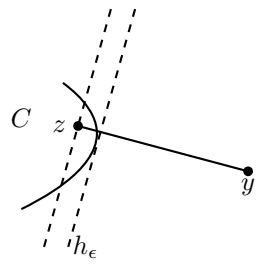

FIG. 2.1. *Admissible approximation of $P_C(y)$*

denoted by P_C. Given $y \in \mathcal{H}$, it holds

$$z \cong_\varepsilon P_C(y) \iff z \in C \text{ and } \langle x - z, y - z\rangle \leq \frac{\varepsilon^2}{2} \quad \forall x \in C. \tag{2.4}$$

Recalling that the projection $P_C(y)$ of a point y is the unique point $z \in C$ which satisfies $\langle x - z, y - z\rangle \leq 0$ for all $x \in C$, approximations of this type are therefore the points enjoying a relaxed formulation of this property. From a geometric point of view, the characterization of projection ensures that the convex set C is entirely contained in the half-space determined by the tangent hyperplane at the point $P_C(y)$, namely $C \subseteq \{x \in X : \langle x - P_C(y), y - P_C(y)\rangle \leq 0\}$. Figure 2.1 depicts an admissible approximation of $P_C(y)$. To check that z satisfies condition (2.4)*, it is enough to verify that C is entirely contained in the negative half-space determined by the (affine) hyperplane of equation*

$$h_\varepsilon : \quad \langle x - z, \frac{y - z}{\|y - z\|}\rangle = \frac{\varepsilon^2}{2\|y - z\|}.$$

which is normal to $y - z$ and at distance $\varepsilon^2/(2\|y - z\|)$ from z.

In the following we provide an analysis of the notion of inexactness given in Definition 2.1, which will clarify the nature of these approximations and the scope of applicability. To this purpose, we will make use of the duality technique, an approach that is quite common in signal recovery and image processing applications [18, 12, 16]. The starting point is the *Moreau decomposition formula* [41, 18]

$$\mathrm{prox}_{\lambda g}(y) = y - \lambda \mathrm{prox}_{g^*/\lambda}(y/\lambda), \tag{2.5}$$

where $g^* : \mathcal{H} \to \overline{\mathbb{R}}$ is the *conjugate functional* of g defined as $g^*(y) = \sup_{x\in\mathcal{H}}(\langle x, y\rangle - g(x))$. In the cases where $\mathrm{prox}_{g^*/\lambda}$ is easy to compute, formula (2.5) provides a convenient method to find the proximity operator of λg.

A remarkable property of inexact proximal points based on the criterion (2.3) is that, in a sense, the Moreau decomposition still holds. If $y, z \in \mathcal{H}$ and $\varepsilon, \lambda > 0$, then, letting $\eta = \varepsilon/\lambda$, it is

$$z \cong_\eta \mathrm{prox}_{g^*/\lambda}(y/\lambda) \iff y - \lambda z \cong_\varepsilon \mathrm{prox}_{\lambda g}(y)\,. \tag{2.6}$$

This arises immediately from Definition 2.1 and the following equivalence (see Theorem 2.4.4, item (iv), in [65]):

$$y - \lambda z \in \partial_{\frac{\eta^2\lambda}{2}} g^*(z) \iff z \in \partial_{\frac{\varepsilon^2}{2\lambda}} g(y - \lambda z).$$

Next, we prove that the proposed inexactness criterion can be formulated in terms of duality gap. This leads to a very natural and simple test for assessing admissible approximations. Without loss of generality, we consider the case where g has the following structure

$$g(x) = \omega(Bx), \tag{2.7}$$

with $B : \mathcal{H} \to \mathcal{G}$ a bounded linear operator between Hilbert spaces, and $\omega : \mathcal{G} \to \overline{\mathbb{R}}$ a proper, l.s.c. convex function. The structure (2.7) often arises in regularization for ill-posed inverse problems [13, 10, 28, 53, 67, 16]. By definition, finding $\operatorname{prox}_{\lambda g}(y)$ requires the solution of the minimization problem

$$\min_{x\in\mathcal{H}} \omega(Bx) + \frac{1}{2\lambda}\|x-y\|^2 = \min_{x\in\mathcal{H}} \Phi_\lambda(x). \tag{2.8}$$

Then, *Fenchel-Moreau-Rockafellar duality formula* (see Corollary 2.8.5 in [65]) states that, if ω is continuous in Bx_0 for some $x_0 \in \mathcal{H}$, it holds

$$\min_{x\in\mathcal{H}} \Phi_\lambda(x) = -\min_{v\in\mathcal{G}} \Psi_\lambda(v) =: m\,, \tag{2.9}$$

where

$$\Psi_\lambda(v) = \frac{1}{2\lambda}\|\lambda B^* v - y\|^2 + \omega^*(v) - \frac{1}{2\lambda}\|y\|^2\,, \tag{2.10}$$

or equivalently the minimum of the *duality gap* is zero

$$0 = \min_{(x,v)\in\mathcal{H}\times\mathcal{G}} \Phi_\lambda(x) + \Psi_\lambda(v) =: G(x,v)\,. \tag{2.11}$$

Moreover, if $\bar v$ is a solution of the dual problem $\min_v \Psi_\lambda(v)$, then $\bar z = y - \lambda B^*\bar v$ solves the primal problem (2.8). This also means that $\min_v G(y-\lambda B^* v, v) = 0$.

PROPOSITION 2.2. *Let $\eta = \varepsilon/\lambda$, $v \in \mathcal{G}$ and consider the following statements:*

a) $G(y-\lambda B^* v, v) \le \varepsilon^2/(2\lambda)$*;*

b) $B^* v \cong_\eta \operatorname{prox}_{g^*/\lambda}(y/\lambda)$*;*

c) $y - \lambda B^* v \cong_\varepsilon \operatorname{prox}_{\lambda g}(y)$*.*

Then, it holds a) $\Rightarrow$ b) and b) $\Leftrightarrow$ c). Furthermore they are all equivalent in case $\omega^*(v) = g^*(B^*v)$.

Proof. Let us show that *a)* $\Rightarrow$ *b)*. From the definition of G it follows

$$\begin{aligned}
&G(y-\lambda B^* v, v)\\
&\quad= \frac{1}{2\lambda}\left[\|\lambda B^* v\|^2 - 2\langle \lambda B^* v, y\rangle\right] + \frac{1}{2\lambda}\|\lambda B^* v\|^2 + \sup_{w\in\mathcal{H}} \langle w, y-\lambda B^* v\rangle - g^*(w) + \omega^*(v)\\
&\quad= \langle B^* v, \lambda B^* v - y\rangle + \sup_{w\in\mathcal{H}} \langle w, y-\lambda B^* v\rangle - g^*(w) + \omega^*(v)\\
&\quad\ge \sup_{w\in\mathcal{H}} \langle w - B^* v, y-\lambda B^* v\rangle - g^*(w) + g^*(B^* v)\\
&\quad= \sup_{w\in\mathcal{H}} -[g^*(w) - g^*(B^* v) - \langle w - B^* v, y - \lambda B^* v\rangle],
\end{aligned}$$

since $\omega^*(v) \geq g^*(B^*v)$ — with all equalities in case $\omega^*(v) = g^*(B^*v)$. Therefore if $G(y - \lambda B^*v, v) \leq \varepsilon^2/(2\lambda)$, setting $\eta = \varepsilon/\lambda$, it is

$$\forall\, w \in \mathcal{H} \quad g^*(w) - g^*(B^*v) \geq \langle w - B^*v, y - \lambda B^*v\rangle - \frac{\eta^2\lambda}{2}$$

which is equivalent to $y - \lambda B^*v \in \partial_{\eta^2\lambda/2} g^*(B^*v)$ and hence to $B^*v \cong_\eta \operatorname{prox}_{g^*/\lambda}(y/\lambda)$. The equivalence of $b)$ and $c)$ comes directly from the inexact Moreau decomposition formula (2.6). □

REMARK 1. *In Proposition 2.2, the equivalence among statements $a), b), c)$ occurs in the following cases:*

- *ω is positively homogeneous. Since in that case $\omega^* = \delta_S$ with $S = \partial\omega(0)$ and also $g^* = \delta_K$ with $K = \partial g(0) = B^*(S)$. Thus, if $v \in S$, it is $\delta_S(v) = \delta_K(B^*v)$ and*

$$G(y - \lambda B^*v, v) \leq \frac{\varepsilon^2}{2\lambda} \iff \lambda B^*v \cong_\varepsilon P_{\lambda K}(y) \iff y - \lambda B^*v \cong_\varepsilon \operatorname{prox}_{\lambda g}(y)\,.$$

- *B surjective. Since in that case $g^*(B^*v) = \sup_{x\in\mathcal{H}}(\langle Bx, v\rangle - \omega(Bx)) = \omega^*(v)$. For instance, for $B = id$, it holds*

$$G(y - \lambda v, v) \leq \frac{\varepsilon^2}{2\lambda} \iff v \cong_\eta \operatorname{prox}_{g^*/\lambda}(y/\lambda)\,.$$

Summarizing, we have shown that admissible approximations in the sense of Definition 2.1 can be computed by minimizing the duality gap $G(y - \lambda B^*v, v)$. In Section 6, we will provide a simple algorithm for doing this.

2.2. Comparison with other kinds of approximation. Other notions of inexactness for the proximity operator have been considered in the literature. One of the first is

$$d(0, \partial\Phi_\lambda(z)) \leq \frac{\varepsilon}{\lambda}\,, \tag{2.12}$$

which was proposed in [48], and treated also in [30].

Another notion, that we shall treat in the appendix, simply replaces the exact minimization in (2.1) by searching for $\varepsilon^2/(2\lambda)$-minima, that is

$$\Phi_\lambda(z) \leq \inf \Phi_\lambda + \frac{\varepsilon^2}{2\lambda}\,. \tag{2.13}$$

Condition (2.13) is equivalent to $0 \in \partial_{\varepsilon^2/(2\lambda)}\Phi_\lambda(z)$ and implies[2] $\|z - \operatorname{prox}_{\lambda g}(y)\| \leq \varepsilon$. This type of error has been first considered in [3] and then employed for instance in [19, 52, 66]. Paper [52] (Lemma 1) shows that criterion (2.13) is more general than both (2.3), (2.12) and, actually, it is the combination of those error criteria. We also note that (again from Lemma 1 in [52]) the error criterion proposed in [38, 55] for the approximate hybrid extragradient-proximal point algorithm corresponds to a relative version of (2.13).

Here, to help positioning the proposed criterion, we give a proposition and two corollaries that directly link approximations in the sense of (2.3) with those in the sense of (2.13), valid for a sub-class of functions g.

[2] See [52]. This accounts also for ε^2 in (2.13).

PROPOSITION 2.3. *Let $g : \mathcal{H} \to \overline{\mathbb{R}}$ be a proper, convex and l.s.c. function and $y, z \in \mathcal{H}$. Suppose* $\operatorname{dom} g$ *be bounded (in norm). Then, if $0 < \eta \leq \operatorname{diam}(\operatorname{dom} g)$ and $\eta \operatorname{diam}(\operatorname{dom} g) \leq \varepsilon^2/2$, it holds*

$$z \approxeq_\eta \operatorname{prox}_{\lambda g}(y) \text{ in the sense of (2.13)} \implies z \approxeq_\varepsilon \operatorname{prox}_{\lambda g}(y) \text{ in the sense of (2.3)}.$$

Proof. Thanks to Lemma 1 in [52], if (2.13) holds with ε replaced by η, then there exist $\eta_1, \eta_2 \geq 0$ with $\eta_1^2 + \eta_2^2 \leq \eta^2$ and $e \in \mathcal{H}$, $\|e\| \leq \eta_2$, such that $(y + e - z)/\lambda \in \partial_{\eta_1^2/(2\lambda)} g(z)$. Therefore, for every $x \in \operatorname{dom} g$

$$\lambda g(x) - \lambda g(z) \geq \langle x - z, y - z \rangle - \operatorname{diam}(\operatorname{dom} g)\eta_2 - \frac{\eta_1^2}{2}.$$

Now it is easy to show that, if $0 < \eta \leq \operatorname{diam}(\operatorname{dom} g)$, then

$$\sup_{\eta_1^2 + \eta_2^2 \leq \eta^2} \left(\operatorname{diam}(\operatorname{dom} g)\eta_2 + \frac{\eta_1^2}{2} \right) = \operatorname{diam}(\operatorname{dom} g)\eta.$$

Thus, if $\operatorname{diam}(\operatorname{dom} g)\eta \leq \varepsilon^2/2$, it holds $\lambda g(x) - \lambda g(z) \geq \langle x - z, y - z \rangle - \varepsilon^2/2$ for every $x \in \operatorname{dom} g$, which proves that $(y - z)/\lambda \in \partial_\varepsilon g(z)$. □

Proposition 2.3 tells us that for each $\varepsilon > 0$ we can get approximations of proximal points in the sense of Definition 2.1 from approximations in the sense of (2.13) as soon as η is chosen small enough. Combining Proposition 2.3 with (2.6), we also have

COROLLARY 2.4. *Let $g : \mathcal{H} \to \overline{\mathbb{R}}$ be proper, convex and l.s.c. and $y \in \mathcal{H}$. Suppose* $\operatorname{dom} g^*$ *be bounded (in norm). For any $\varepsilon > 0$, if $0 < \sigma \leq \operatorname{diam}(\operatorname{dom} g^*)$ and $\sigma\lambda^2 \operatorname{diam}(\operatorname{dom} g^*) \leq \varepsilon^2/2$, then*

$$z \approxeq_\sigma \operatorname{prox}_{g^*/\lambda}(y/\lambda) \text{ in the sense of (2.13)} \implies y - \lambda z \approxeq_\varepsilon \operatorname{prox}_{\lambda g}(y)$$

for every $z \in \operatorname{dom} g^*$.

Proof. The condition on σ, given in the statement, is equivalent to

$$\sigma \operatorname{diam}(\operatorname{dom} g^*) \leq \eta^2/2$$

with $\eta = \varepsilon/\lambda$. Therefore we can apply Proposition 2.3 to the function g^*, obtaining that for $z \in \operatorname{dom} g^*$, it holds

$$z \approxeq_\sigma \operatorname{prox}_{g^*/\lambda}(y/\lambda) \text{ in the sense of (2.13)} \implies z \approxeq_\eta \operatorname{prox}_{g^*/\lambda}(y/\lambda).$$

Then, the inexact Moreau decomposition formula (2.6) gives $y - \lambda z \approxeq_\varepsilon \operatorname{prox}_{\lambda g}(y)$ □

REMARK 2. *The hypothesis* $\operatorname{dom} g^*$ *be bounded in Corollary 2.4 is satisfied (in finite dimension) for many significant regularization terms, like total variation, nuclear norm and structured sparsity regularization and it has been consider in similar contexts, for instance, in [14, 9]. It implies that g is Lipschitz continuous on* $\operatorname{dom} g$. *Indeed, for a proper, convex function $g : \mathcal{H} \to \overline{\mathbb{R}}$, we have*

$$\operatorname{dom} g^* = \bigcup_{x \in \mathcal{H}} \partial_\varepsilon g(x)$$

for every $\varepsilon > 0$. Therefore, if $\operatorname{dom} g^*$ *is bounded in norm, say by $M \geq 0$, it is easy to see that $|g(x_1) - g(x_2)| \leq M\|x_1 - x_2\| + \varepsilon$ for every $x_1, x_2 \in \operatorname{dom} g$ and $\varepsilon > 0$. Since ε is arbitrary, this shows that g is in fact M-Lipschitz on the entire* $\operatorname{dom} g$.

In case g is positively homogeneous, namely $g(\alpha x) = \alpha g(x)$ for $\alpha \geq 0$, λg is positively homogeneous too and $(\lambda g)^* = \delta_{\partial(\lambda g)(0)} = \delta_{\lambda K}$ with $K := \partial g(0)$. The inexact Moreau decomposition (2.6), applied to λg, gives

$$z \approxeq_\varepsilon P_{\lambda K}(y) \iff y - z \approxeq_\varepsilon \operatorname{prox}_{\lambda g}(y) \tag{2.14}$$

meaning that we can approximate the proximity operator of λg by means of an approximation of the projection onto the closed and convex set λK. Thus, we can further specialize Corollary 2.4, obtaining that

$$z \approxeq_\sigma P_{\lambda K}(y) \text{ in the sense of (2.13)} \implies y - z \approxeq_\varepsilon \operatorname{prox}_{\lambda g}(y)$$

if $\sigma\lambda \mathrm{diam} K \leq \varepsilon^2/2$.

3. Nesterov's estimate sequences. In [44], Nesterov illustrates a flexible mechanism to produce minimizing sequences for an optimization problem. The idea is to recursively generate a sequence of simple functions that approximate F. In this section, we briefly describe this method and review the general results obtained in [52] for constructing quadratic estimate sequences when F is convex. We do not provide proofs, referring to the mentioned works for details.

3.1. General framework. We start by providing the definition and motivation of estimate sequences.

DEFINITION 3.1. *A pair of sequences* $(\varphi_k)_{k\in\mathbb{N}}$, $\varphi_k : \mathcal{H} \to \mathbb{R}$ *and* $(\beta_k)_{k\in\mathbb{N}}$, $\beta_k \geq 0$ *is called an* estimate sequence *of a proper function* $F : \mathcal{H} \to \overline{\mathbb{R}}$ *iff*

$$\forall x \in \mathcal{H}, \forall k \in \mathbb{N} : \ \varphi_k(x) - F(x) \leq \beta_k(\varphi_0(x) - F(x)) \quad \text{and} \quad \beta_k \to 0. \tag{3.1}$$

The next statement represents the main result about estimate sequences and explains how to use them to build minimizing sequences and get corresponding convergence rates.

THEOREM 3.2. *Let* $((\varphi_k)_{k\in\mathbb{N}}, (\beta_k)_{k\in\mathbb{N}})$ *be an estimate sequence of* F *and denote by* $(\varphi_k)_*$ *the infimum of* φ_k. *If, for some sequences* $(x_k)_{k\in\mathbb{N}}$, $x_k \in \mathcal{H}$ *and* $(\delta_k)_{k\in\mathbb{N}}, \delta_k \geq 0$, *we have*

$$F(x_k) \leq (\varphi_k)_* + \delta_k, \tag{3.2}$$

then for any $x \in \mathrm{dom} F$

$$F(x_k) \leq \beta_k(\varphi_0(x) - F(x)) + \delta_k + F(x). \tag{3.3}$$

Thus, if $\delta_k \to 0$ *(being also* $\beta_k \to 0$*),* $(x_k)_{k\in\mathbb{N}}$ *is a minimizing sequence for* F, *that is* $\lim_{k\to\infty} F(x_k) = F_*$. *If in addition the infimum* F_* *is attained at some point* $x_* \in \mathcal{H}$, *then the following rate of convergence holds true*

$$F(x_k) - F_* \leq \beta_k(\varphi_0(x_*) - F_*) + \delta_k \,.$$

We point out that the previous theorem provides convergence of the sequence $(F(x_k))_{k\in\mathbb{N}}$ to the infimum of F without assuming any existence of a minimizer for F, neither the boundedness from below. However, the hypothesis of attainability of the infimum is required if an estimate of the convergence rate is needed.

3.2. Construction of an estimate sequence. In this section, we review a general procedure, introduced in [52], for generating an estimate sequence of a proper, l.s.c. and convex function $F : \mathcal{H} \to \overline{\mathbb{R}}$. First of all, we deal with the generation of the sequence of functions $(\varphi_k)_{k\in\mathbb{N}}$.

Denote by $\mathcal{F}(\mathcal{H},\mathbb{R})$ the space of functions from $\mathcal{H}$ to $\mathbb{R}$. Given F, we define an updating rule for functions $\varphi \in \mathcal{F}(\mathcal{H},\mathbb{R})$, depending on the choice of four parameters $(z,\eta,\xi,\alpha) \in \mathrm{dom}F \times \mathbb{R}_+ \times \mathcal{H} \times [0,1)$, as

$$U(z,\eta,\xi,\alpha) : \mathcal{F}(\mathcal{H},\mathbb{R}) \to \mathcal{F}(\mathcal{H},\mathbb{R})$$
$$U(z,\eta,\xi,\alpha)(\varphi)(x) = (1-\alpha)\varphi(x) + \alpha(F(z) + \langle x - z, \xi\rangle - \eta)\,.$$

Hereafter, for convenience, we will often denote the update of φ simply by $\hat{\varphi}$, that is

$$\hat{\varphi} := U(z,\eta,\xi,\alpha)(\varphi)$$

hiding the dependence on the parameters. The same hat notation will also be used for other quantities: in all cases it will stand for an update of the corresponding variable.

One can see that, given $((z_k,\eta_k,\xi_k,\alpha_k))_{k\in\mathbb{N}}$, $(z_k,\eta_k,\xi_k,\alpha_k) \in \mathrm{dom}F \times \mathbb{R}_+ \times \mathcal{H} \times [0,1)$ with $\xi_{k+1} \in \partial_{\eta_k} F(z_{k+1})$ and an arbitrary function $\varphi : \mathcal{H} \to \mathbb{R}$, the sequence defined by setting $\varphi_0 = \varphi$ and $\varphi_{k+1} = U(z_{k+1},\eta_k,\xi_{k+1},\alpha_k)\varphi_k$ satisfies

$$\varphi_{k+1}(x) - F(x) \le (1-\alpha_k)(\varphi_k(x) - F(x))\,, \tag{3.4}$$

and the pair $((\varphi_k)_{k\in\mathbb{N}}, (\beta_k)_{k\in\mathbb{N}})$, with $\beta_k = \prod_{i=0}^{k-1}(1-\alpha_i)$, is an estimate sequence of F provided that $\sum_{k\in\mathbb{N}} \alpha_k = +\infty$.

If the starting φ is a quadratic function written in canonical form, namely

$$\varphi(x) = \varphi_* + \frac{A}{2}\|x-\nu\|^2, \quad \text{with } \varphi_* \in \mathbb{R},\ A > 0,\ \nu \in \mathcal{H},$$

then, for an arbitrary choice of the parameters, the update $\hat{\varphi}$ of φ introduced above is still a quadratic function, that can be written in canonical form as $\hat{\varphi}(x) = \hat{\varphi}_* + \frac{\hat{A}}{2}\|x-\hat{\nu}\|^2$, with

$$\begin{cases} \hat{\varphi}_* = (1-\alpha)\varphi_* + \alpha F(z) + \alpha\langle \nu - z, \xi\rangle - \dfrac{\alpha^2}{2(1-\alpha)A}\|\xi\|^2 - \alpha\eta \\ \hat{A} = (1-\alpha)A \\ \hat{\nu} = \nu - \dfrac{\alpha}{(1-\alpha)A}\xi. \end{cases} \tag{3.5}$$

This means that the subset of quadratic functions is closed with respect to the action of the operator $U(z,\eta,\xi,\alpha)$, which therefore induces a transformation on the relevant parameters defining their canonical form, depending of course on (z,η,ξ,α).

Next, we will need to generate a sequence $(x_k)_{k\in\mathbb{N}}$ satisfying inequality (3.2) and to study the asymptotic behavior of β_k. To this aim we recall two lemmas, whose proofs are provided in [52], that will be crucial in the whole subsequent analysis.

LEMMA 3.3. *Let $x,\nu \in \mathcal{H}$, $A > 0$ and $\varphi = \varphi_* + A/2\|\cdot - \nu\|^2$ be such that $F(x) \le \varphi_* + \delta$ for some $\delta \ge 0$. If $z,\xi \in \mathcal{H}$, $\eta \ge 0$ are given with $\xi \in \partial_\eta F(z)$, defining $\hat{\varphi} = U(z,\eta,\xi,\alpha)(\varphi)$, with $\alpha \in [0,1)$ and setting $y = (1-\alpha)x + \alpha\nu$, we get*

$$(1-\alpha)\delta + \eta + \hat{\varphi}^* \ge F(z) + \frac{\lambda}{2}\left(2 - \frac{\alpha^2}{(1-\alpha)A\lambda}\right)\|\xi\|^2 + \langle y - (\lambda\xi + z), \xi\rangle$$

for every $\lambda > 0$.

LEMMA 3.4. *Given the sequence* $(\lambda_k)_{k\in\mathbb{N}}$, $\lambda_k \geq \lambda > 0$ *and* $A > 0$, $a, b > 0$, $a \leq b$, *define* $(A_k)_{k\in\mathbb{N}}$ *and* $(\alpha_k)_{k\in\mathbb{N}}$ *recursively, such that* $A_0 = A$ *and for* $k \in \mathbb{N}$

$$\begin{aligned} &\alpha_k \in [0,1), \text{ with } a \leq \frac{\alpha_k^2}{(1-\alpha_k)A_k\lambda_k} \leq b \\ &A_{k+1} = (1-\alpha_k)A_k. \end{aligned}$$

Then, the sequence defined by setting $\beta_k := \prod_{i=0}^{k-1}(1-\alpha_i)$ *satisfies* $\beta_k = O(1/k^2)$. *Moreover, if* $(\lambda_k)_{k\in\mathbb{N}}$ *is also bounded from above,* $\beta_k \sim 1/k^2$.

4. Derivation of the general algorithm. In this section, we show how the mechanism of estimate sequences can be used to generate an inexact version of accelerated forward-backward algorithms. A general theorem of convergence will also be provided.

We shall assume both the hypotheses H1) and H2), given in the introduction, be satisfied. The following lemma generalizes a well-known result [7] and will enable us to build an appropriate estimate sequence.

LEMMA 4.1. *For any* $x, y \in \mathcal{H}$, $z \in \mathrm{dom} g$, $\varepsilon \geq 0$ *and* $\zeta \in \partial_\varepsilon g(z)$ *it holds*

$$F(x) \geq F(z) + \langle x - z, \nabla f(y) + \zeta\rangle - \frac{L}{2}\|z-y\|^2 - \varepsilon. \tag{4.1}$$

In other words, $\nabla f(y) + \zeta \in \partial_\eta F(z)$, *with* $\eta = L/2\|z-y\|^2 + \varepsilon$.

Proof. Fix $y, z \in \mathcal{H}$, since ∇f is L-Lipschitz continuous we get

$$f(y) \geq f(z) - \langle z - y, \nabla f(y)\rangle - \frac{L}{2}\|z-y\|^2 . \tag{4.2}$$

On the other hand, being f convex, we have $f(x) \geq f(y) + \langle x - y, \nabla f(y)\rangle$, which combined with (4.2) gives

$$f(x) \geq f(z) + \langle x - z, \nabla f(y)\rangle - \frac{L}{2}\|z-y\|^2 . \tag{4.3}$$

Since g is convex and $\zeta \in \partial_\varepsilon g(z)$, we have $g(x) \geq g(z) + \langle x - z, \zeta\rangle - \varepsilon$, that summed with (4.3), gives the statement. □

Combining the previous Lemma 4.1 with Lemma 3.3, we derive the following result.

LEMMA 4.2. *Let* $x, \nu \in \mathcal{H}$, $A > 0$ *and* $\varphi = \varphi_* + A/2\|\cdot - \nu\|^2$ *be such that* $F(x) \leq \varphi_* + \delta$ *for some* $\delta \geq 0$. *If* $z, \zeta \in \mathcal{H}, \varepsilon \geq 0, \lambda > 0$ *are given with* $\zeta \in \partial_{\varepsilon^2/(2\lambda)} g(z)$, *setting* $y = (1-\alpha)x + \alpha\nu$ *and* $\eta = L/2\|y-z\|^2 + \varepsilon^2/(2\lambda)$ *and defining* $\hat{\varphi} = U(z, \eta, \nabla f(y) + \zeta, \alpha)\varphi$, *with* $\alpha \in [0,1)$, *we get*

$$\begin{aligned} (1-\alpha)\delta + \frac{\varepsilon^2}{2\lambda} + \hat{\varphi}^* \geq F(z) + \frac{\lambda}{2}\left(2 - \frac{\alpha^2}{(1-\alpha)A\lambda}\right)\|\nabla f(y) + \zeta\|^2 \\ + \langle y - (\lambda(\nabla f(y) + \zeta) + z), \nabla f(y) + \zeta\rangle - \frac{L}{2}\|y-z\|^2 \end{aligned}$$

The next result shows how to choose ζ in order to derive an iterative procedure.

THEOREM 4.3. *Fix $\lambda > 0$, $\varepsilon > 0$. Let $x, \nu \in \mathcal{H}$, $A > 0$ and $\varphi = \varphi_* + A/2\|\cdot - \nu\|^2$ be such that $F(x) \le \varphi_* + \delta$ for some $\delta \ge 0$. If $\alpha^2/((1-\alpha)A\lambda) \le 2 - \lambda L$, choosing*

$$
\begin{aligned}
&y = (1-\alpha)x + \alpha\nu \\
&\hat{x} \approx_\varepsilon \operatorname{prox}_{\lambda g}(y - \lambda\nabla f(y)), \quad \zeta = \frac{y-\hat{x}}{\lambda} - \nabla f(y) \quad (\in \partial_{\frac{\varepsilon^2}{2\lambda}} g(\hat{x})) \\
&\hat{\varphi} = U(\hat{x}, \eta, \nabla f(y) + \zeta, \alpha)\varphi, \quad \textit{with } \eta = \frac{L}{2}\|y-\hat{x}\|^2 + \frac{\varepsilon^2}{2\lambda} \\
&\hat{A} = (1-\alpha)A \\
&\hat{\nu} = \nu - \frac{\alpha}{\hat{A}}(\nabla f(y) + \zeta) \\
&\hat{\delta} = (1-\alpha)\delta + \frac{\varepsilon^2}{2\lambda},
\end{aligned}
$$

we have $\hat{\delta} + \hat{\varphi}^* \ge F(\hat{x}) + \dfrac{c}{2\lambda}\|y-\hat{x}\|^2 \ge F(\hat{x})$ with $c = 2 - \lambda L - \alpha^2/(\hat{A}\lambda) \ge 0$.

Proof. Applying Lemma 4.2, with $z = \hat{x}$ and ζ defined above, taking into account that $y - (\lambda(\nabla f(y) + \zeta) + \hat{x}) = 0$, we get

$$
(1-\alpha)\delta + \frac{\varepsilon^2}{2\lambda} + \hat{\varphi}^* \ge F(\hat{x}) + \frac{1}{2\lambda}\left(2 - \lambda L - \frac{\alpha^2}{(1-\alpha)A\lambda}\right)\|y-\hat{x}\|^2.
$$

If we choose λ and α such that $\alpha^2/(1-\alpha)A\lambda \le 2 - \lambda L$ we immediately obtain the statement of the theorem. ∎

We are now ready to define a general *accelerated and inexact forward-backward splitting* (AIFB) algorithm and to prove its convergence rate. For fixed numbers $A > 0, a \in\ ]0, 2]$, a sequence of parameters $(\lambda_k)_{k\in\mathbb{N}}$, $\lambda_k \in\ ]0, (2-a)/L]$ and a sequence of errors $(\varepsilon_k)_{k\in\mathbb{N}}$ with $\varepsilon_k \ge 0$, we set $A_0 = A, \delta_0 = 0$ and $x_0 = \nu_0 \in \operatorname{dom} F$ and for every $k \in \mathbb{N}$, we recursively define

$$
\left|\begin{aligned}
&\alpha_k \in [0,1) \quad \text{such that} \quad a \le \frac{\alpha_k^2}{(1-\alpha_k)A_k\lambda_k} \le 2 - \lambda_k L \\
&y_k = (1-\alpha_k)x_k + \alpha_k\nu_k \\
&x_{k+1} \approx_{\varepsilon_k} \operatorname{prox}_{\lambda_k g}(y_k - \lambda_k\nabla f(y_k)) \\
&A_{k+1} = (1-\alpha_k)A_k \\
&\nu_{k+1} = \nu_k - \frac{\alpha_k}{(1-\alpha_k)A_k\lambda_k}(y_k - x_{k+1}) \\
&\delta_{k+1} = (1-\alpha_k)\delta_k + \frac{\varepsilon_k^2}{2\lambda_k}.
\end{aligned}\right. \tag{AIFB}
$$

Then, by setting $\xi_{k+1} = (y_k - x_{k+1})/\lambda_k$, we get two sequences $(x_k)_{k\in\mathbb{N}}$ and $(\xi_k)_{k\in\mathbb{N}}$ such that $\xi_{k+1} \in \partial_{\eta_k} F(x_{k+1})$, where $\eta_k = \frac{L}{2}\|y_k - x_{k+1}\|^2 + \varepsilon_k^2/(2\lambda_k)$. Therefore, the sequence of functions $(\varphi_k)_{k\in\mathbb{N}}$ defined as $\varphi_{k+1} = U(x_{k+1}, \eta_k, \xi_{k+1}, \alpha_k)\varphi_k$ is an estimate sequence of F provided that $\beta_k = \prod_{i=0}^{k-1}(1-\alpha_i) \to 0$. The last condition holds true due to Lemma 3.4.

Moreover, starting from $\varphi_0 = F(x_0) + A_0/2\|\cdot - \nu_0\|^2$, we have $\delta_0 + \varphi_0^* \geq F(x_0)$ and, by induction, applying Theorem 4.3, also $\delta_k + (\varphi_k)_* \geq F(x_k)$ for every $k \geq 1$. If $\delta_k \to 0$, the sequence $(x_k)_{k\in\mathbb{N}}$ is a minimizing sequence for F.

REMARK 3. *To be more precise, Theorem 4.3 implies*

$$\delta_{k+1} + \varphi_{k+1}^* \geq F(x_{k+1}) + \frac{c_k}{2\lambda_k}\|y_k - x_{k+1}\|^2 \quad \textit{with } c_k = 2 - \lambda_k L - \frac{\alpha_k^2}{(1-\alpha_k)A_k\lambda_k}\,.$$

From this inequality, along the lines of the proof of Theorem 1 in [52], if x_* is a minimizer of F, one has

$$\frac{c_{k-1}}{2\lambda_{k-1}}\|y_{k-1} - x_k\|^2 + (F(x_k) - F_*) \leq \beta_k(\varphi_0(x_*) - F_*) + \delta_k\,.$$

The last result shows that, if $c_k \geq c > 0$ (e.g. if $2 - \lambda_k L - a \geq c > 0$), being λ_k bounded from above, then $\|y_{k-1} - x_k\| \to 0$.

Concerning the structure of the error term δ_k, it is easy to prove (see Lemma 3.3 in [30]) that the solution of the last difference equation in AIFB is given by

$$\delta_k = \frac{\beta_k}{2}\sum_{i=0}^{k-1}\frac{\varepsilon_i^2}{\lambda_i\beta_{i+1}}\,. \tag{4.4}$$

The behavior of β_k established in Lemma 3.4 allows us to impose explicit conditions on the error sequence ε_k in order to get a convergent algorithm. The following theorem is the main result of the paper.

THEOREM 4.4. *Consider the AIFB algorithm for a bounded sequence $\lambda_k \in [\lambda, (2-a)/L]$, and fixed $\lambda \in\ ]0, 2/L[$ and $a \in\ ]0, 2 - \lambda L[$.*
Then, if $\varepsilon_k = O(1/k^q)$ with $q > 1/2$, the sequence $(x_k)_{k\in\mathbb{N}}$ is minimizing for F and if the infimum of F is attained the following bounds on the rate of convergence hold true

$$F(x_k) - F_* = \begin{cases} O\left(1/k^2\right) & \textit{if } q > 3/2 \\ O\left(1/k^2\right) + O\left(\log k/k^2\right) & \textit{if } q = 3/2 \\ O\left(1/k^2\right) + O\left(1/k^{2q-1}\right) & \textit{if } q < 3/2. \end{cases}$$

Proof. Since $\lambda_k \in [\lambda, (2-a)/L]$, by Lemma 3.4, $\beta_k \sim 1/k^2$. Hence the error δ_k can be majorized as follows

$$\delta_k = \frac{\beta_k}{2}\sum_{i=0}^{k-1}\frac{\varepsilon_i^2}{\lambda_i\beta_{i+1}} \leq \frac{c}{2\lambda(k+1)^2}\sum_{i=0}^{k-1}\varepsilon_i^2(i+1)^2\,,$$

for a properly chosen constant $c > 0$. If $\varepsilon_k = O(1/(k+1)^q)$, the last inequality implies

$$\delta_k \leq \frac{\tilde{c}}{(k+1)^2}\sum_{i=0}^{k-1}\frac{1}{(i+1)^{2(q-1)}}.$$

The sequence $\sum_{i=0}^{k-1} 1/(i+1)^{2(q-1)}$ is convergent if $q > 3/2$, it is an $O(\log k)$ if $q = 3/2$ and an $O((k+1)^{3-2q})$ if $q < 3/2$. ∎

The rates of convergence given in Theorem 4.4 hold for the function values and not for the iterates, as usual for accelerated schemes [7, 61]. In particular, we proved

that the proposed algorithm shares the convergence rate of the exact one, if the errors ε_k in the computation of the proximity operator in (1.1) decay as $1/k^q$ and $q > 3/2$. We underline that summability of the errors is not required to get convergence, which is guaranteed for $q > 1/2$. If the infimum is not achieved, it is not possible to get a convergence rate for $F(x_k) - F_*$, but inequality (3.3) ensures that a solution within accuracy σ requires $O(1/\sqrt{\sigma})$ iterations if $q > 3/2$ and $O(1/\sigma^{1/(2q-1)})$ if $1/2 < q < 3/2$. We finally point out that the results given in Theorem 4.4 provide lower bounds for the convergence rates of AIFB algorithm, meaning that faster empirical rates might be observed for particular instances of problem (P).

Backtracking stepsize rule. As other forward-backward splitting schemes, the above procedure requires the explicit knowledge of the Lipschitz constant of ∇f. Often in practice, especially for large scale problems, computing L might be too demanding. For this reason, procedures allowing the use of a forward-backward splitting algorithm while avoiding the computation of L have been proposed [45, 7]. In this section, we describe how the so called *backtracking procedure* can be applied in our context as well, when L is not known. The key idea is the fact that the statement of Lemma 4.1 still holds if $y \in \mathcal{H}$, $z \in \operatorname{dom} g$ and $M > 0$ satisfy the inequality

$$f(y) \geq f(z) - \langle z - y, \nabla f(y) \rangle - \frac{M}{2}\|z - y\|^2. \tag{4.5}$$

Then, a straightforward generalization of Theorem 4.3 yields the key inequality $\hat{\delta} + \hat{\varphi}^* \geq F(\hat{x})$, for $y, \hat{x}$ satisfying (4.5). These two facts allow us to add a subroutine to AIFB, denoted BT, without affecting its convergence rate. More precisely, the direct choice of λ_k and α_k and the computation of y_k and x_{k+1} in AIFB is substituted at each step by means of the following function:

$$(M_k, \lambda_k, \alpha_k, y_k, x_{k+1}) = BT(M_{k-1}, \varepsilon_k, x_k, \nu_k),$$

where M_{k-1} is the current guess for L. Let $\gamma > 1$, for arbitrary M, ε, x, ν, we define $BT(M, \varepsilon, x, \nu)$ by iteratively constructing the finite sequence $((\tilde{M}_i, \tilde{\lambda}_i, \tilde{\alpha}_i, \tilde{y}_i, \tilde{x}_{i+1}))_{i=0}^m$, for $i \geq 0$ as

$$\begin{cases} \tilde{M}_i = \gamma^i M, \qquad \tilde{\lambda}_i \in]0, (2-a)/\tilde{M}_i] \\ \tilde{\alpha}_i \in [0,1) \quad \text{such that} \quad a \leq \dfrac{\tilde{\alpha}_i^2}{(1-\tilde{\alpha}_i)A\tilde{\lambda}_i} \leq 2 - \tilde{\lambda}_i \tilde{M}_i \\ \tilde{y}_i = (1-\tilde{\alpha}_i)x + \tilde{\alpha}_i \nu \\ \tilde{x}_{i+1} \cong_\varepsilon \operatorname{prox}_{\tilde{\lambda}_i g}(\tilde{y}_i - \tilde{\lambda}_i \nabla f(\tilde{y}_i)) \end{cases}$$

We then let $BT(M, \varepsilon, x, \nu) = (\tilde{M}_m, \tilde{\lambda}_m, \tilde{\alpha}_m, \tilde{y}_m, \tilde{x}_{m+1})$, where m is defined as

$$m := \min\{i \in \mathbb{N} : f(\tilde{y}_i) \geq f(\tilde{x}_{i+1}) - \langle \tilde{x}_{i+1} - \tilde{y}_i, \nabla f(\tilde{y}_i) \rangle - \tilde{M}_i \|\tilde{x}_{i+1} - \tilde{y}_i\|^2/2\}.$$

Note that m is finite, since $\lim_i \tilde{M}_i = +\infty$ (being $\gamma > 1$) and the condition (4.5) is satisfied by any point when $M \geq L$.

5. Recovering FISTA. We show that the proposed general algorithm can be rewritten in equivalent forms, which include the well-known FISTA [7].

We prove that the sequence ν_k in AIFB can be eliminated, achieving a first alternative form of the algorithm. To this purpose, let us define

$$a_k := \frac{\alpha_k^2}{(1-\alpha_k)A_k\lambda_k} \in [a, 2[\tag{5.1}$$

Then, the updating rule for ν can be written as

$$\nu_{k+1} = \nu_k - \frac{a_k}{\alpha_k}(y_k - x_{k+1}) \tag{5.2}$$

From the definition of y_k in AIFB, we get $\nu_k = \alpha_k^{-1}(y_k - (1-\alpha_k)x_k)$ and substituting into (5.2), we have

$$\nu_{k+1} = \Big(\frac{1}{\alpha_k} - 1\Big)(x_{k+1} - x_k) + \frac{1}{\alpha_k}(1 - a_k)(y_k - x_{k+1}) + x_{k+1}\,.$$

Thus, substituting that expression of ν_{k+1} in $y_{k+1} = (1-\alpha_{k+1})x_{k+1} + \alpha_{k+1}\nu_{k+1}$ and computing the solution $\alpha_k \in [0, 1[$ of equation (5.1), AIFB can be equivalently written as

$$\left|\begin{aligned} \alpha_k &= \frac{\sqrt{(A_k a_k \lambda_k)^2 + 4A_k a_k \lambda_k} - A_k a_k \lambda_k}{2} \\ x_{k+1} &\approxeq_{\varepsilon_k} \operatorname{prox}_{\lambda_k g}(y_k - \lambda_k \nabla f(y_k)) \\ y_{k+1} &= x_{k+1} + \alpha_{k+1}\Big(\frac{1}{\alpha_k} - 1\Big)(x_{k+1} - x_k) + (1 - a_k)\frac{\alpha_{k+1}}{\alpha_k}(y_k - x_{k+1}) \\ A_{k+1} &= (1-\alpha_k)A_k\ . \end{aligned}\right. \tag{5.3}$$

This form depends on an extra arbitrary numerical sequence $(a_k)_{k\in\mathbb{N}}$ with $0 < a \leq a_k < 2$ and resembles the one given in equations (34) - (36) in [61].

We can formulate the algorithm in yet another, simpler form, replacing the two numerical sequences $(\alpha_k)_{k\in\mathbb{N}}$ and $(A_k)_{k\in\mathbb{N}}$ with a new one. By defining $t_k = 1/\alpha_k$ the update t_{k+1} can be computed recursively. Indeed, being $\alpha_k^2 = a_k A_{k+1}\lambda_k$ and taking into account (5.1) for $k+1$, we have

$$t_{k+1}^2 - t_{k+1} - \frac{\lambda_k}{\lambda_{k+1}}\frac{a_k}{a_{k+1}}t_k^2 = 0$$

which can be solved in the unknown t_{k+1}. Therefore, a third form of the algorithm reads as follows

$$\left|\begin{aligned} t_{k+1} &= \frac{1 + \sqrt{1 + 4(a_k\lambda_k)t_k^2/(a_{k+1}\lambda_{k+1})}}{2} \\ x_{k+1} &\approxeq_{\varepsilon_k} \operatorname{prox}_{\lambda_k g}(y_k - \lambda_k \nabla f(y_k)) \\ y_{k+1} &= x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k) + (1 - a_k)\frac{t_k}{t_{k+1}}(y_k - x_{k+1})\ . \end{aligned}\right. \tag{5.4}$$

Regarding the initialization, we highlight that we are free to choose any $t_0 > 1$ as well as $x_0 = y_0 \in \mathcal{H}$. Indeed $\alpha_0 = 1/t_0 \in\ ]0, 1[$ and $\alpha_0^2/((1-\alpha_0)A_0\lambda_0) = a_0$ holds if we choose $A_0 = A = \alpha_0^2/((1-\alpha_0)a_0\lambda_0)$.

REMARK 4. *We are allowed to choose $t_0 = 1$ in the initialization step because, as one can easily check, with this choice we get $t_1 > 1$ and if $a_0 = 1$, $y_1 = x_1$. Therefore the sequences continue as if they started from (t_1, x_1, y_1).*

The last form of the algorithm, together with Remark 4, shows that we can recover FISTA [7, 61] by choosing $a_k = 1$ and $\lambda_k = \lambda \leq 1/L$, starting with $t_0 = 1$. Moreover, for $f = 0$ and $a_k = 2$, we also obtain the proximal point algorithm given in the appendix of [30].

6. Study of the global nested algorithm. In this section we consider the entire two-loops algorithm that results from the composition of AIFB with an inner algorithm which computes the proximity operator.

6.1. Computing admissible approximations. We first cope with the computation of solutions of the subproblem

$$z \approxeq_\varepsilon \operatorname{prox}_{\lambda g}(y) \tag{6.1}$$

required by the proposed algorithm at each iteration. There are various possibilities to solve problem (6.1). In [20, 19] a bundle algorithm returning an element $z \in \mathcal{H}$ satisfying (6.1) is provided, and convergence in a finite number of steps is proved when g is Lipschitz continuous over bounded sets (see Algorithm 6.1 and Proposition 6.1 in [19]). As in Section 2, we consider the case of $g(x) = \omega(Bx)$. Proposition 2.2 allows to state problem (6.1) as the minimization of the duality gap, in fact $z := y - \lambda B^* v$ solves (6.1), if $v \in \mathcal{G}$ is such that

$$G(y - \lambda B^* v, v) \leq \frac{\varepsilon^2}{2\lambda}. \tag{6.2}$$

It is evident that condition (6.2) can be explicitly checked in practice. Following the same notations of Section 2, let $\overline{v}$ be a solution of the dual problem $\min \Psi_\lambda$ and $\overline{z} = y - \lambda B^* \overline{v}$ the solution of the primal problem (2.8). Then one has

$$0 \in B(\lambda B^* \overline{v} - y) + \partial \omega^*(\overline{v}),$$

or, equivalently, the primal solution $\overline{z}$ satisfies

$$B\overline{z} \in \partial \omega^*(\overline{v}). \tag{6.3}$$

In the following, we show that each algorithm that produces a minimizing sequence for the dual function Ψ_λ yields a minimizing sequence for the duality gap as well, if ω is continuous on the entire $\mathcal{G}$.

THEOREM 6.1. *Let* $\operatorname{dom} \omega = \mathcal{G}$, $(v_n)_{n\in\mathbb{N}}$ *be a minimizing sequence for* Ψ_λ *and set* $z_n = y - \lambda B^* v_n$. *Then it holds*

$$z_n \to \overline{z}\,, \quad G(z_n, v_n) \to 0\,.$$

Moreover, if $\Psi_\lambda(v_n) - \Psi_\lambda(\overline{v}) = O(1/n^p)$ *for some* $p > 0$, *we have*

$$\|z_n - \overline{z}\| = O\left(\frac{1}{n^{p/2}}\right), \qquad G(z_n, v_n) = O\left(\frac{1}{n^{p/2}}\right). \tag{6.4}$$

Proof. We claim that

$$\frac{1}{2\lambda}\|z_n - \overline{z}\|^2 \leq \Psi_\lambda(v_n) - \Psi_\lambda(\overline{v}). \tag{6.5}$$

To prove (6.5), first note that

$$\begin{aligned}
&\frac{1}{2\lambda}\|\lambda B^* v_n - y\|^2 - \frac{1}{2\lambda}\|\lambda B^* \overline{v} - y\|^2 + \langle B\overline{z}, v_n - \overline{v}\rangle \qquad (6.6)\\
&\quad = \frac{1}{2\lambda}\langle \lambda B^*(v_n + \overline{v}) - 2y, \lambda B^*(v_n - \overline{v})\rangle + \frac{1}{2\lambda}\langle 2(y - \lambda B^* \overline{v}), \lambda B^*(v_n - \overline{v})\rangle\\
&\quad = \frac{1}{2\lambda}\langle \lambda B^*(v_n + \overline{v}) - 2\lambda B^* \overline{v}, \lambda B^*(v_n - \overline{v})\rangle\\
&\quad = \frac{1}{2\lambda}\|\lambda B^*(v_n - \overline{v})\|^2\,.
\end{aligned}$$

By (6.3) we obtain $\omega^*(v_n) - \omega^*(\overline{v}) - \langle B\overline{z}, v_n - \overline{v}\rangle \geq 0$. Summing the last equation with (6.6), we get

$$\begin{aligned}\Psi_\lambda(v_n) - \Psi_\lambda(\overline{v}) &= \frac{1}{2\lambda}\|\lambda B^* v_n - y\|^2 - \frac{1}{2\lambda}\|\lambda B^*\overline{v} - y\|^2 + \omega^*(v_n) - \omega^*(\overline{v}) \\ &\geq \frac{1}{2\lambda}\|\lambda B^*(v_n - \overline{v})\|^2 \\ &= \frac{1}{2\lambda}\|z_n - \overline{z}\|^2 .\end{aligned}$$

Since $\operatorname{dom}\omega = \mathcal{G}$, ω is continuous on $\mathcal{G}$ and hence Φ_λ is continuous on $\mathcal{H}$. Therefore $\Phi_\lambda(z_n) \to \Phi_\lambda(\overline{z})$. This implies, being $\Phi_\lambda(\overline{z}) = -\Psi_\lambda(\overline{v})$,

$$G(z_n, v_n) = \Phi_\lambda(z_n) + \Psi_\lambda(v_n) \to \Phi_\lambda(\overline{z}) + \Psi_\lambda(\overline{v}) = 0 .$$

Now suppose that $\Psi_\lambda(v_n) - \Psi_\lambda(\overline{v}) = O(1/n^p)$. Then, the first part of statement (6.4) directly follows from (6.5). Regarding the rate on the duality gap, note that the function Φ_λ is Lipschitz continuous on bounded sets, being convex and continuous. Thus there exists $L_1 > 0$ such that

$$\Phi_\lambda(z_n) - \Phi_\lambda(\overline{z}) \leq L_1\|z_n - \overline{z}\| \leq L_1\sqrt{2\lambda}\,(\Psi_\lambda(v_n) - \Psi_\lambda(\overline{v}))^{1/2} .$$

This shows that the convergence rate stated for the duality gap in (6.4) holds. □

In order to compute admissible approximations of the proximal point, we can choose any minimizing algorithm for the dual problem. A simple choice is the forward-backward splitting algorithm (called also ISTA [7]). In this case, as done in [16], we get the following algorithm (for an arbitrary initialization $v_0 \in \mathcal{G}$)

$$v_{n+1} = \operatorname{prox}_{\frac{\gamma_n}{\lambda}\omega^*}\Big(v_n - \frac{\gamma_n}{\lambda}B(\lambda B^* v_n - y)\Big) \qquad 0 < \gamma_n < \frac{2}{\|B\|^2} . \tag{6.7}$$

Since for this choice $\Psi_\lambda(v_n) - \Psi_\lambda(\overline{v}) = O(1/n)$, this gives the rate $G(z_n, v_n) = O(1/\sqrt{n})$ for the duality gap. We remark that the pair of sequences $(y - \lambda B^* v_n, v_n)$ corresponds exactly to the pair $(\overline{x}_n, y_n)$ generated by the primal-dual Algorithm 1 proposed in [14] when applied to the minimization of $\Phi_\lambda(x) = g(x) + \frac{1}{2\lambda}\|x - y\|^2$ ($\tau = \lambda$, $\theta = 1$).

A more efficient choice is FISTA, resulting in the rate $G(z_n, v_n) = O(1/n)$. The latter will be our choice in the numerical section. For the case of ω positively homogeneous (e.g. total variation), it holds $\omega^* = \delta_S$, with $S = \partial\omega(0)$, and the corresponding dual minimization problem $\min \Psi_\lambda$ becomes a constrained smooth optimization problem. Then, FISTA reduces to an accelerated projected gradient descent algorithm

$$\begin{aligned} v_{n+1} &= P_S\Big(u_n - \frac{\gamma_n}{\lambda}B(\lambda B^* u_n - y)\Big) \qquad 0 < \gamma_n \leq \frac{1}{\|B\|^2} \\ u_{n+1} &= v_{n+1} + \frac{t_n - 1}{t_{n+1}}(v_{n+1} - v_n), \end{aligned} \tag{6.8}$$

with the usual choices for t_n (see Remark 4).

REMARK 5. *We highlight that the results in Theorem 6.1 holds for the more general setting of a minimization problem of the form*

$$\min_{x \in X} \omega(Bx) + \varphi(x)$$

where $\operatorname{dom}\varphi = X$ *and* φ *is strongly convex and differentiable with Lipschitz continuous gradient.*[3] *Indeed, in this case one has* $\overline{z} = \nabla\varphi^*(-B^*\overline{v}), z_n = \nabla\varphi^*(-B^*v_n)$ *and the strong convexity of* φ^* *allows to get the analogous bound of* (6.5)

$$\frac{1}{2L}\|z_n - \overline{z}\|^2 \leq \Psi(z_n) - \Psi(\overline{z})\,.$$

6.2. Global iteration complexity of the algorithm. Each iteration of AIFB consists of a gradient descent step, to which we refer to as *external iteration*, and an inner loop, to approximate the proximity operator of g up to a precision ε_k. Theorem 6.1 proves that using FISTA to solve the dual problem guarantees $G(z_n, v_n) \leq D/n$ for a constant $D > 0$. This shows that $\lceil 2\lambda D/\varepsilon^2 \rceil$ iterations suffice to get a solution of problem (6.1). We note that, under the additional hypotheses $\omega^*(v)/\|v\| \to +\infty$ and γ_n constant, the same number of iterations is sufficient to get the same convergence rate for the gap, using the sequences of ergodic means computed via Algorithm 1 proposed in [14]. On the other hand, the algorithm provided in [19] reaches the same goal in $O(1/\varepsilon^4)$ iterations.

In general, given an (internal) algorithm that solves problem (6.1) in at most

$$\frac{D\lambda}{\varepsilon^{2/p}}, \quad p > 0, \tag{6.9}$$

iterations[4], we can bound the total iteration complexity of the AIFB algorithm. From Theorem 4.4, if we let $\varepsilon_k := 1/k^q$, and take $k \geq N_e$, with

$$N_e := \begin{cases} \lceil (C/\varepsilon)^{\frac{1}{2q-1}} \rceil & \text{if } 1/2 < q < 3/2 \\ \lceil (C/\varepsilon)^{\frac{1}{2}} \rceil & \text{if } q > 3/2 \end{cases}$$

we have $F(x_k) - F_* \leq \varepsilon$, where $C > 0$ is the constant masked in the rates given in Theorem 4.4. Now for each $k \leq N_e$, from the hypothesis (6.9) on the complexity of the internal algorithm, one needs at most $D\lambda_k/\varepsilon_k^{2/p} = D\lambda_k k^{2q/p}$ internal iterations to get an approximate proximal point x_{k+1} in AIFB with precision $\varepsilon_k = 1/k^q$. Summing all the internal iterations from 1 to $\overline{k}$, and if $\lambda_k \leq \overline{\lambda}$, we have

$$N_i = \sum_{k=1}^{\overline{k}} D\lambda_k k^{2q/p} \leq D\overline{\lambda}\int_0^{\overline{k}} t^{2q/p}dt = \frac{D\overline{\lambda}}{2q/p+1}\overline{k}^{2q/p+1}$$

and hence

$$N_i = \begin{cases} O\big(1/\varepsilon^{\frac{2q/p+1}{2q-1}}\big) & \text{if } 1/2 < q < 3/2 \\ O\big(1/\varepsilon^{\frac{2q/p+1}{2}}\big) & \text{if } q > 3/2\,. \end{cases}$$

[3] This is equivalent to require φ^* strongly convex and differentiable with Lipschitz continuous gradient. See Theorems 4.2.1 and 4.2.2 in chapter 4 of [32].

[4] The constant D in general depends on the starting point and the problem solution set, and at the end by y. If $\operatorname{dom}\omega^*$ is bounded, D can be chosen independently on y, since for most algorithms it is majorized by $\operatorname{diam}(\operatorname{dom}\omega^*)$.

Adding the costs of internal and external iterations together, the global complexity $\mathcal{C}_g$ of the two loops algorithm is

$$\mathcal{C}_g = c_i N_i + c_e N_e = \begin{cases} O\big(1/\varepsilon^{\frac{2q/p+1}{2q-1}}\big) + O\big(1/\varepsilon^{\frac{1}{2q-1}}\big) & \text{if } 1/2 < q < 3/2 \\ O\big(1/\varepsilon^{\frac{2q/p+1}{2}}\big) + O\big(1/\varepsilon^{\frac{1}{2}}\big) & \text{if } q > 3/2\,. \end{cases} \tag{6.10}$$

where c_i and c_e denotes the unitary costs of each type of iteration. From the estimates above, one can easily see that, in each case, the lower global complexity is reached for $q \to 3/2$ and it is

$$\mathcal{C}_g = O(1/\varepsilon^{\frac{p+3}{2p}+\delta})$$

for whatever small $\delta > 0$. For $p = 1$, as it is the case of algorithm (6.8), one obtains a complexity of $O(1/\varepsilon^{2+\delta})$. For $p = 1/2$, which corresponds to the rate of the algorithm studied in [19], we have a global complexity of $O(1/\varepsilon^{7/2+\delta})$. We finally note that for $p \to +\infty$ we have a complexity of $O(1/\varepsilon^{1/2+\delta})$: in other words the global rate of convergence tends to $1/N^2$, in the total number N of iterations, and the algorithm behaves once more as an accelerated method.

We remark that the analysis of the global complexity given above is valid only asymptotically, since we did not estimate any of the constants hidden in the O symbols. However, in real situations constants do matter and, in practice, the most effective accuracy rate q is problem dependent and might be different from $3/2$, as we illustrate in the experiments of Subsection 7.3.

7. Numerical Experiments. In this section, we present two types of experiments. The first one is designed to illustrate the influence of the errors on the behavior of AIFB and on its non accelerated counterpart ISTA. The second one is meant to measure the performance of the two loops algorithm AIFB+algorithm (6.8), in comparison with ISTA+algorithm (6.8), and with the primal-dual algorithm proposed in [14].

7.1. Experiments setup. In all the following cases, we consider the regularized least-squares functional

$$F(x) := \frac{1}{2}\|Ax - y\|_{\mathcal{Y}}^2 + g(x)\,, \tag{7.1}$$

where $\mathcal{H}, \mathcal{Y}$ are Euclidean spaces, $x \in \mathcal{H}$, $y \in \mathcal{Y}$, $A : \mathcal{H} \to \mathcal{Y}$ is a linear operator and $g : \mathcal{H} \to \overline{\mathbb{R}}$ is of type (2.7). In all cases ω will be a norm and the projection onto $S = \partial\omega(0)$ will be explicitly computable.

We minimize F using AIFB in the equivalent form (5.4), with $\lambda_k = \lambda = 1/L$, where $L = \|A^*A\|$. We use $a_k = 1$ (corresponding to FISTA), since we empirically observed that the choice of a_k, if independent of k, does not significantly influence the speed of convergence of the algorithm (although preliminary tests revealed a slightly better performance for $a_k = 0.8$). At each iteration, we employ algorithm (6.8) to approximate the proximity operator of g up to a precision ε_k. The stopping rule for the inner algorithm is given by the duality gap, according to Proposition 2.2, item a). Following Theorem 4.4, we consider sequences of errors of type $\varepsilon_k = C/k^q$, with q, hereafter referred as *accuracy rate*, chosen between 0.1 and 1.7. The coefficient C should be comparable to the magnitude of the duality gap. In fact, it determines the practical constraint on the duality gap at the first iterations: the constraint should be

active, but not too demanding to avoid unnecessary precision. We choose C by solving the equation $G(y_0 - \lambda \nabla f(y_0), 0) = C^2/(2\lambda)$ where G is the duality gap corresponding to the first proximal subproblem encountered in AIFB for $k = 0$, evaluated at $v_0 = 0$. We finally consider an "exact" version, obtained by solving the proximal subproblems at the machine precision.

We analyze two well-known problems: deblurring with total variation regularization and learning a linear estimator via regularized empirical risk minimization with the overlapping group lasso penalty. The numerical experiments are divided in two parts. In the first one, we evaluate the impact of the errors on the convergence rate of AIFB and the (non accelerated) forward-backward splitting (here denoted as ISTA). The plot of the relative objective values $(F(x_k) - F_*)/F_*$ against the number of external iterations for different accuracy rates on the error is shown. We underline that this study is independent of the algorithm chosen to produce an admissible approximation of the proximal points.

In the second part, we assess the overall behavior of the two-loops algorithm, as described in Section 6, using algorithm (6.8) to solve the proximal subproblems. We compare it with the non accelerated version (ISTA) and the Primal-Dual (PRIDU) algorithm proposed by [14] for image deconvolution. For all algorithms we provide CPU time, and the number of external and internal iterations for different precisions. Note that the cost of each external iteration relies mainly in the evaluation of the gradient of the quadratic part of the objective function (7.1). The internal iteration has a similar form, but being the matrix B sparse and structured in both experiments, can be implemented in a fast way. All the numerical experiments have been performed in MATLAB environment[5], on a a desktop iMac with Intel Core i5 CPU, 2,5 Ghz, 6MB cache L3, and 6 GB of RAM.

7.1.1. Deblurring with Total Variation. Regularization with the Total Variation [50, 12, 6] is a widely used technique for deblurring and denoising images, that preserves sharp edges.

In this problem, $\mathcal{H} = \mathcal{Y} = \mathbb{R}^{N \times N}$ is the space of (discrete 2D) images on the grid $[1, N]^2$, A is a linear map representing some blurring operator [6] and y is the observed noisy and blurred datum. The (discrete) *total variation* regularizer is defined as

$$g = \omega \circ \nabla \qquad g(x) = \tau \sum_{i,j=1}^{N} \|(\nabla x)_{i,j}\|_2$$

where $\nabla : \mathcal{H} \to \mathcal{H}^2$ is the (discrete) gradient operator (see [12] for the precise definition) and $\omega : \mathcal{H}^2 \to \mathbb{R}$, $\omega(\mathbf{p}) = \tau \sum_{i,j=1}^{N} \|\mathbf{p}_{i,j}\|_2$ with $\tau > 0$ a regularization parameter and $\|\cdot\|_2$ the euclidean norm in $\mathbb{R}^2$. Note that the matrix corresponding to ∇ is highly sparse (it is bidiagonal). This feature has been taken into account to get an efficient implementation.

We followed the same experimental setup as in [6]. We considered the 256×256 Lena test image, blurred by a 9×9 Gaussian blur with standard deviation 4, followed by additive normal noise with zero mean and standard deviation 10^{-3}. The regularization parameter τ was set to 10^{-3}. Since the blurring operator A is a convolution operator, in the implementation it is common to evaluate it by an FFT based method (see e.g. [6, 14]).

[5]The code is available upon request to the authors

7.1.2. Overlapping group lasso. The group lasso penalty is a regularization term for ill-posed inverse problems arising in statistical learning [64, 33], image processing and compressed sensing [46], enforcing structured sparsity in the solutions. Regularization with this penalty consists in solving a problem of the form (7.1), where $\mathcal{H} = \mathbb{R}^p$, $\mathcal{Y} = \mathbb{R}^m$, A is a data or design matrix and y is a vector of outputs or measurements. Following [33], the overlapping group lasso (OGL) penalty is

$$g(x) = \tau \sum_{i=1}^{r} \left(\sum_{j \in J_i} (w_j^i)^2 x_j^2 \right)^{1/2}, \tag{7.2}$$

where $\mathcal{J} = \{J_1, \dots, J_r\}$ is a collection of overlapping groups of indices such that $\bigcup_{i=1}^r J_i = \{1, \dots, p\}$. The weights w_j^i are defined as

$$w_j^i = \left(\frac{1}{2}\right)^{a_j^i}, \quad \text{with } a_j^i = \#\{J \in \mathcal{J} \,:\, j \in J, J \subset J_i, J \neq J_i\}.$$

This penalty can be written as $\omega \circ B$, with $B = (B_1, \dots, B_r) : \mathbb{R}^p \to \mathbb{R}^{J_1} \times \dots \mathbb{R}^{J_r}$,

$$B_i : \mathbb{R}^p \to \mathbb{R}^{J_i}, \quad B_i x = (w_j^i x_j)_{j \in J_i},$$

and $\omega : \mathbb{R}^{J_1} \times \dots \mathbb{R}^{J_r} \to \overline{\mathbb{R}}$, $\omega(v_1, \dots, v_r) = \tau \sum_{i=1}^r \|v_i\|_2$, where $\|\cdot\|_2$ is the euclidean norm in $\mathbb{R}^{J_i}$.

The matrix A and the datum y are generated from the breast cancer dataset provided by [62]. The dataset consists of expression data for $8,141$ genes in 295 breast cancer tumors (78 metastatic and 17 non-metastatic). The groups are defined according to the canonical pathways from MSigDB [60], that contains 639 groups of genes, 637 of which involve genes from the breast cancer dataset. We restrict the analysis to the 3510 genes that are contained in at least one group. Hence, our data matrix A consists of 295 different expression levels of 3510 genes. The output vector y contains the labels (± 1, metastatic or non-metastatic) of each sample. The structure of the overlapping groups gives rise to a matrix B of size 15126×3510. Despite the high dimensionality, one can take advantage of its sparseness. We analyze two choices of the regularization parameter: $\tau = 0.01$ and $\tau = 0.1$.

7.2. Results - Part I. We run AIFB and its non-accelerated counterpart, ISTA, up to 2.000 external iterations. With the aim of maximizing the effect of inexactness, we require algorithm (6.8) to produce solutions with errors close to the upper bounds $\epsilon_k^2/2\lambda$ prescribed by the theory. We achieve this by reducing the internal step-size length γ_n and using *cold restart*, i.e. initializing at each step algorithm (6.8) with $v_0 = 0$.

As a reference optimal value, F_*, we use the value found afters $10,000$ iterations of AIFB with error rate $q = 1.7$.

As shown in Fig. 7.1, the empirical convergence rate of $(F(x_k) - F_*)/F_*$ is indeed affected by the accuracy rate q: to smaller values of q correspond slower convergence rates both for AIFB and the inexact (non-accelerated) forward-backward algorithm. When the errors in the computation of the proximity operator do not decay fast enough, the convergence rates are much deteriorated and the algorithms can even not converge to the infimum. If the errors decay sufficiently fast, AIFB shows a faster convergence w.r.t. ISTA in both experiments. In contrast, this is not true for accuracy rates $q < 1$, where ISTA has practically the same behavior of AIFB.

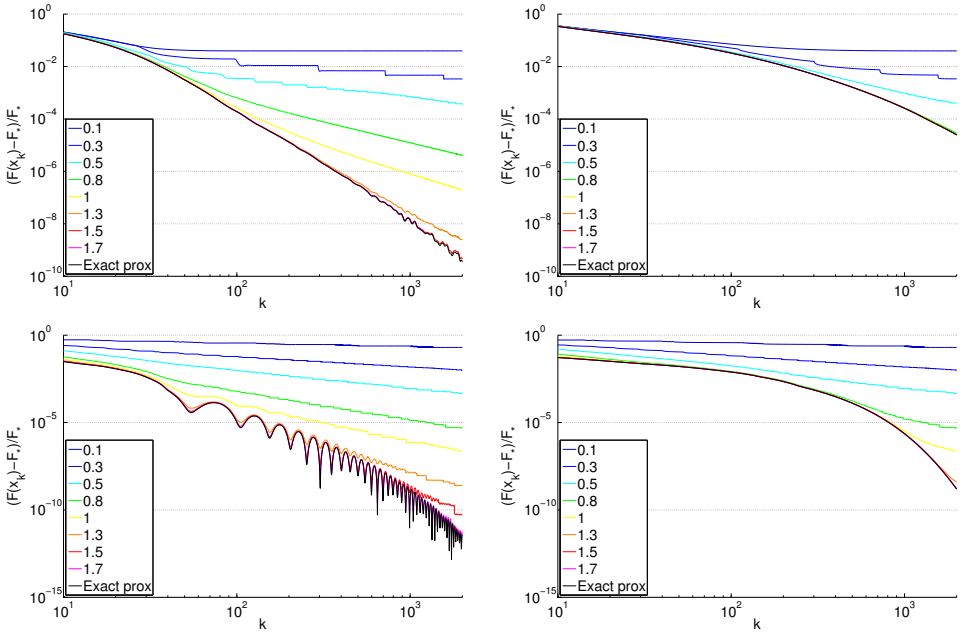

FIG. 7.1. **Impact of the errors on AIFB and ISTA.** *Loglog plots of relative objective value vs external iterations, k, obtained for TV deblurring (upper row) and the OGL problem with regularization parameter $\tau = 10^{-1}$ (bottom row). The AIFB and inexact ISTA for different accuracy rates q in the computation of the proximity operator are shown in the left and right column, respectively. For larger values of the parameter q the curves overlap. It can be seen from visual inspection, that the errors affect the acceleration.*

Moreover, it turns out that AIFB is more sensitive to errors then ISTA. This is more evident in the experiment on TV deblurring. Indeed, for AIFB most curves corresponding to the different accuracy rates are well separated, while for ISTA they are closer to each other, and often completely overlapped. Yet, the overlapping phenomenon in general starts earlier (lower q) for ISTA than AIFB, indicating that no gain is obtained in increasing the accuracy error rates over a certain level, in accordance to the theoretical results.

7.3. Results - Part II. This section is the empirical counterpart of Subsection 6.2. Here, we test the global iteration complexity of AIFB and inexact ISTA combined with algorithm (6.8) on the two problems described above. We provide the number of external iterations and the total number of inner iterations. When taking into account the cost of computing the proximity operator, there is a trade-off between the number of external and internal iterations. Since internal and external iterations in general have different computational costs — which depend on the specific problem considered and the machine CPU — the total number of iterations is not a good measure of the algorithm's performance. For instance, on our computer, the ratio between the cost of the external and internal iteration is about 2.15 in the TV deblurring and 2.5 in the OGL problem. Therefore, we also report the CPU time needed to reach a desired accuracy for the relative difference to the optimal value. In this part, we use the *warm-restart* procedure, consisting in initializing algorithm (6.8) with the solution obtained at the previous step. We empirically observed that this initialization strategy drastically reduces the total number of iterations and speeds up the algorithm.

TABLE 7.1

Deblurring with Total Variation regularization, $\tau = 10^{-3}$. *Performance evaluation of AIFB, ISTA and PRIDU, corresponding to different choices of the parameters q and σ, respectively. Concerning AIFB and ISTA, the results are reported only for the q's giving the best results. The entries in the table refer to the CPU time (in seconds) needed to reach a relative difference w.r.t. to the optimal value below the thresholds 10^{-4}, 10^{-6} and 10^{-8}, the number of external iterations (# Ext), and the total number of internal interations (# Int).*

Precision	10^{-4}			10^{-6}			10^{-8}		
Algo	Time	# Ext	# Int	Time	# Ext	# Int	Time	# Ext	# Int
AIFB									
$q = 1$	11.8	137	1062	124.2	905	12313	1750	8776	182006
$q = 1.3$	16.2	118	1600	**63.6**	387	6437	**272.1**	1300	28350
$q = 1.5$	26.0	117	2734	98.7	373	10540	414.5	1085	45297
ISTA									
$q = 0.1$	36.9	1341	1341	147.2	5346	5346	635.4	23031	23031
$q = 0.8$	36.9	1341	1341	147.2	5346	5346	635.4	23031	23031
$q = 1.0$	63.2	1337	4533	189.9	5226	11126	745.1	18224	48333
PRIDU									
$\sigma = 10$	7.4	362	-	165.7	8186	-	4684	231848	-
$\sigma = 12.5$	**6.2**	310	-	132.2	6609	-	3715	185588	-

We compare AIFB and ISTA with PRIDU taken as a benchmark, since it often outperforms state-of-the-art methods, in particular for TV regularization (see the numerical section in [14]).

Algorithm PRIDU depends on two parameters[6] $\sigma, \rho > 0$. In our experiments, we tested two choices, indicated by the authors (in the paper and code as well) for the image deblurring and denoising problem: $\sigma = 10$ and $\rho = 1/(\sigma\|B\|^2)$, and $\rho = 0.01$ (corresponding to $\sigma = 1/(\rho\|B\|^2) = 12.5$ for the TV problem and $\sigma \approxeq 1.07$ for the OGL problem). We implemented the algorithm also for the OGL problem and, as a consequence of preliminary tests, the same choices of parameters turn out to be appropriate too.

On the other hand, AIFB and ISTA, depend on the accuracy rate q. We verified that the best empirical results are obtained choosing q in the range $[1, 1.5]$ for AIFB and $[0.1, 0.5]$ for ISTA. This once more confirms the higher sensitivity to the errors of the accelerated version w.r.t. the basic one. In the tables, we detail the results only for most significant choices of q. We remark that the "exact" version of AIFB (and ISTA), where the prox is computed at machine precision at each step, is not even comparable to the results we reported here.

As concerns the TV problem, AIFB ($q = 1.3$ or $q = 1.5$) outperforms both PRIDU and ISTA, for high precisions. PRIDU exhibits a fast convergence at the beginning, but then explodes in correspondence of higher precisions, for both choices of σ. This is a known drawback of primal-dual algorithms with fixed step-size (see e.g. [9]).

The behavior on the OGL problem is presented for two choices of the regularization parameter, since this heavily influence the results. For $\tau = 0.1$ and precision 10^{-4}, AIFB is the fastest. For the middle precision, all the algorithms' performances are comparable. For the highest precision, PRIDU and ISTA perform better. We notice the very good behavior of ISTA, which is probably due to the warm-restart

[6] Denoted σ and τ in [14]

TABLE 7.2
Breast cancer dataset: Overlapping Group Lasso, $\tau = 10^{-1}$. *Performance evaluation of AIFB, ISTA and PRIDU, corresponding to different choices of the parameters q and σ, respectively. Concerning AIFB and ISTA, the results are reported only for the q's giving the best results. The entries in the table refer to the CPU time (in seconds) needed to reach a relative difference w.r.t. to the optimal value below the thresholds 10^{-4}, 10^{-6} and 10^{-8}, the number of external iterations (# Ext), and the total number of internal interations (# Int).*

Precision	10^{-4}			10^{-6}			10^{-8}		
Algo	Time	# Ext	# Int	Time	# Ext	# Int	Time	# Ext	# Int
AIFB									
$q = 1$	3.9	104	3985	41.5	983	42239	414.1	9748	421769
$q = 1.3$	**2.1**	51	2103	11.2	247	11389	60.4	1179	61915
$q = 1.5$	2.8	50	2857	16.2	199	16945	61.3	548	64518
ISTA									
$q = 0.1$	5.3	1675	1682	10.7	3421	3428	16.0	5124	5131
$q = 0.3$	5.2	1613	1730	10.3	3246	3363	15.9	5065	5182
$q = 0.5$	4.4	1217	1827	**9.5**	2850	3460	**14.9**	4603	5213
$q = 0.8$	7.0	585	6092	15.5	2218	11264	19.8	3599	12645
$q = 1$	12.4	535	12031	26.6	1236	25547	42.1	3606	36508
PRIDU									
$\sigma = 10$	10.5	2901	-	25.4	7040	-	47.4	13141	-
$\sigma = 1.07$	5.8	1602	-	11.0	3026	-	16.1	4452	-

TABLE 7.3
Breast cancer dataset: Overlapping Group Lasso, $\tau = 10^{-2}$. *See caption of Table 7.2.*

Precision	10^{-4}			10^{-6}			10^{-8}		
Algo	Time	# Ext	# Int	Time	# Ext	# Int	Time	# Ext	# Int
AIFB									
$q = 0.8$	11.8	443	11392	74.4	2651	72109	1124	39699	1089732
$q = 1$	12.1	432	11616	44.8	1581	43191	170.9	6004	164849
$q = 1.3$	27.0	431	27311	126.9	1572	129708	502.9	4687	518492
$q = 1.5$	62.0	431	64351	312.5	1572	325868	1303	4686	1362149
ISTA									
$q = 0.1$	34.9	11125	11125	69.4	22111	22111	112.3	35782	35782
$q = 0.3$	34.9	11125	11125	69.4	22111	22111	112.3	35782	35782
$q = 0.5$	35.6	11124	11946	70.1	22109	22931	113.0	35781	36603
$q = 0.8$	133.7	11095	114686	218.3	21883	178405	273.2	35781	203992
$q = 1$	335.7	11093	348408	659.7	21818	643374	882.9	33075	851890
PRIDU									
$\sigma = 10$	21.8	5625	-	44.6	11529	-	**82.5**	21346	-
$\sigma = 1.07$	**4.6**	1178	-	**24.7**	6407	-	827.5	214558	-

strategy combined with the greater stability of ISTA against the errors. Finally, on the OGL with $\tau = 0.01$, AIFB still accelerates ISTA at the lower precisions if q is properly tuned, though at the end ISTA wins. The PRIDU algorithm suffers of the same drawbacks remarked in the TV experiment for $\sigma = 1.07$, but exhibits an overall good performance with $\sigma = 10$.

Summarizing, the performance of algorithm AIFB combined with (6.8) and warm restart is comparable with state-of-the-art algorithms, being sometimes better. To this

purpose, the experiments also give some guidelines for choosing the parameter q. We also show situations where the acceleration is lost, in particular referring to high precision.

Appendix A. Accelerated FB algorithms under error criterion (2.13)**.**

We give here a discussion of the behavior of algorithm AIFB using approximations in the sense of (2.13): this is the error considered in [54]. More precisely, the subsequent analysis shows that if at each step of AIFB x_{k+1} is computed with accuracy ε_k, with $\varepsilon_k = O(1/k^q)$ with $q > 3/2$, relying on our techniques we are able to obtain the following convergence rates on the objective values:

$$F(x_k) - F_* = \begin{cases} O\left(1/k\right) & \text{if } q > 2 \\ O\left(\log^2 k/k\right) & \text{if } q = 2 \\ O\left(1/k^{2q-3}\right) & \text{if } q < 2. \end{cases}$$

These results are weaker that the ones given in Theorem 4.4. This is in line with what was obtained in [52], whereas, as mentioned before, the techniques employed in [54] allow to get the rate of convergence $O(1/k^2)$.

Let us consider Lemma 4.2, where we re-denominates y by t and z by $\hat{x}$. Let $\varphi = \varphi^* + \frac{A}{2}\|\cdot - \nu\|^2$ and $x \in \mathcal{H}$ with $F(x) \leq \varphi^* + \delta$ and $\hat{x}, \zeta \in \mathcal{H}$ with $\zeta \in \partial_{\frac{\varepsilon_1^2}{2\lambda}} g(\hat{x})$. Then, setting $t = (1-\alpha)x + \alpha\nu$ and $\hat{\varphi} = U(\hat{x}, \frac{L}{2}\|t - \hat{x}\|^2 + \frac{\varepsilon_1^2}{2\lambda}, \nabla f(t) + \zeta, \alpha)\varphi$, the conclusion can be equivalently written as

$$(1-\alpha)\delta + \frac{\varepsilon_1^2}{2\lambda} + \hat{\varphi}^* \geq F(\hat{x}) - \frac{1}{2\lambda}\Big\{\frac{\alpha^2}{(1-\alpha)A\lambda}\|\lambda(\nabla f(t) + \zeta)\|^2 \\ - 2\langle t - \hat{x}, \lambda(\nabla f(t) + \zeta)\rangle + L\lambda\|t - \hat{x}\|^2\Big\}.$$

Thus if we assume $\frac{\alpha^2}{(1-\alpha)A\lambda} \leq 1$ and $\lambda L \leq 1$, we have

$$(1-\alpha)\delta + \frac{\varepsilon_1^2}{2\lambda} + \hat{\varphi}^* \geq F(\hat{x}) - \frac{1}{2\lambda}\|t - \hat{x} - \lambda(\nabla f(t) + \zeta)\|^2. \tag{A.1}$$

Now let us take $u \in \mathcal{H}$ with $\|u - \nu\| \leq \eta$, thus $u = \nu - \Delta$ and $\|\Delta\| \leq \eta$ (u is considered as a perturbed center of the quadric φ), and set $t = (1-\alpha)x + \alpha\nu$ and $y = (1-\alpha)x + \alpha u$ (perturbed). Clearly $y = t - \alpha\Delta$, hence $\|y - t\| \leq \alpha\eta$. Let

$$\hat{x} \cong_\varepsilon \operatorname{prox}_{\lambda g}(y - \lambda\nabla f(y)) \text{ in the sense of (2.13)}.$$

Then from Lemma 1 in [52], it is

$$\zeta := \frac{y - \lambda\nabla f(y) - \hat{x} - e}{\lambda} \in \partial_{\frac{\varepsilon_1^2}{2\lambda}} g(\hat{x}), \qquad \|e\| \leq \varepsilon_2, \qquad \varepsilon_1^2 + \varepsilon_2^2 \leq \varepsilon^2 \tag{A.2}$$

and if we set $h = -(\nabla f(t) - \nabla f(y))$, we have

$$\zeta = \frac{y - \hat{x} - e}{\lambda} - \nabla f(y) = \frac{t - \hat{x} - (\alpha\Delta + e)}{\lambda} - \nabla f(t) - h.$$

We have $\|h\| = \|\nabla f(t) - \nabla f(y)\| \leq L\|t - y\| \leq L\alpha\eta$ and

$$\lambda(\nabla f(t) + \zeta) = t - \hat{x} - (\hat{e} + \lambda h), \tag{A.3}$$

where we set $\hat{e} = e + \alpha\Delta$ for brief. We can therefore apply the conclusion (A.1). Thus if $\hat{\varphi} = U(\hat{x}, \frac{L}{2}\|t-\hat{x}\|^2 + \frac{\varepsilon_1^2}{2\lambda}, \nabla f(t) + \zeta, \alpha)\varphi$, we get

$$\begin{aligned}(1-\alpha)\delta + \frac{\varepsilon_1^2}{2\lambda} + \hat{\varphi}^* &\geq F(\hat{x}) - \frac{1}{2\lambda}\|\hat{e} + \lambda h\|^2 \\ &= F(\hat{x}) - \frac{1}{2\lambda}\|e + (t-y) - \lambda(\nabla f(t) - \nabla f(y))\|^2\end{aligned}$$

Now taking into account the *Baillon-Haddad theorem* [5], we have

$$\|e + (t-y) - \lambda(\nabla f(t) - \nabla f(y))\| \leq \|e\| + \|(I - \lambda\nabla f)(t) - (I - \lambda\nabla f)(y)\| \leq \varepsilon_2 + \alpha\eta\,.$$

Therefore reordering the inequality above, it holds

$$(1-\alpha)\delta + \frac{1}{2\lambda}(\varepsilon_1^2 + (\varepsilon_2 + \alpha\eta)^2) + \hat{\varphi}^* \geq F(\hat{x})\,.$$

But, taking into account the third inequality in (A.2), it is

$$(\varepsilon + \alpha\eta)^2 = \varepsilon^2 + (\alpha\eta)^2 + 2\varepsilon\alpha\eta \geq \varepsilon_1^2 + \varepsilon_2^2 + (\alpha\eta)^2 + 2\varepsilon_2\alpha\eta = \varepsilon_1^2 + (\varepsilon_2 + \alpha\eta)^2.$$

Thus we finally obtain

$$(1-\alpha)\delta + \frac{(\varepsilon + \alpha\eta)^2}{2\lambda} + \hat{\varphi}^* \geq F(\hat{x})\,.$$

To conclude, we need only to evaluate the quantity $\|\hat{u} - \hat{\nu}\|$, where $\hat{u}$ and $\hat{\nu}$ are the updating of the centers u, ν, which, taking into account formula (3.5), are defined as follows

$$\begin{cases}\hat{\nu} = \nu - \dfrac{\alpha}{\lambda(1-\alpha)A}\lambda(\nabla f(t) + \zeta) & (t \text{ and } \zeta \text{ are unknowns});\\ \hat{u} = u - \dfrac{\alpha}{\lambda(1-\alpha)A}(y - \hat{x}) & (y \text{ and } \hat{x} \text{ are known quantities}).\end{cases}$$

Evidently, taking into account (A.3) and that $y = t - \alpha\Delta$ and $\hat{e} = e + \alpha\Delta$, it is

$$\begin{aligned}\hat{\nu} &= \nu - \frac{\alpha}{\lambda(1-\alpha)A}[y - \hat{x} - (e + \lambda h)] \\ &= u - \frac{\alpha}{\lambda(1-\alpha)A}(y - \hat{x}) + \Delta + \frac{\alpha}{\lambda(1-\alpha)A}(e + \lambda h) \\ &= \hat{u} + \frac{1}{\alpha}\left[\alpha\Delta + \frac{\alpha^2}{\lambda(1-\alpha)A}(e + \lambda h)\right]\end{aligned}$$

If we set for brief $\gamma := \alpha^2/(\lambda(1-\alpha)A)$, then

$$\hat{\nu} = \hat{u} + \frac{1}{\alpha}(\gamma e + (t-y) - \gamma\lambda(\nabla f(t) - \nabla f(y)))$$

and if we suppose $\gamma \leq 1$, we have

$$\|\gamma e + (t-y) - \gamma\lambda(\nabla f(t) - \nabla f(y))\| \leq \varepsilon_2 + \|(t-y) - \gamma\lambda(\nabla f(t) - \nabla f(y))\| \leq \varepsilon + \alpha\eta,$$

where we took into account again the Baillon-Haddad theorem. Concluding our proof, if we set $\hat{\eta} = \eta + \varepsilon/\alpha$, it holds

$$(1-\alpha)\delta + \frac{(\alpha\hat{\eta})^2}{2\lambda} + \hat{\varphi}^* \geq F(\hat{x}), \qquad \|\hat{v} - \hat{u}\| \leq \hat{\eta}\,.$$

Thus, the errors behaves exactly in the same manner as in Theorem 3 of [52] and hence we can get the same conclusion of the subsequent Theorem 4. We note also that we required only $\alpha^2/(\lambda(1-\alpha)A) \leq 1$ and $\lambda L \leq 1$.

REFERENCES

[1] Y. I. Alber, R. S. Burachik, and A. N. Iusem, *A proximal point method for nonsmooth convex optimization problems in Banach spaces*, Abstr. Appl. Anal., 2 (1997), pp. 97–120.

[2] A. Argyriou, C.A. Micchelli, M. Pontil, L. Shen, and Y. Xu, *Efficient first order methods for linear composite regularizers.* arXiv:1104.1436v1, 2011.

[3] A. Auslender, *Numerical methods for nondifferentiable convex optimization*, Math. Programming Stud., (1987), pp. 102–126. Nonlinear analysis and optimization (Louvain-la-Neuve, 1983).

[4] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, *Optimization with sparsity-inducing penalties*, Foundations and Trends in Machine Learning, 4 (2012), pp. 1–106.

[5] H. H. Bauschke and P. L. Combettes, *The Baillon-Haddad Theorem Revisited*, J. Convex Anal., 17 (2010), pp. 781–787.

[6] A. Beck and M. Teboulle, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring*, IEEE Trans. Image Proc., 18 (2009), pp. 2419–2434.

[7] ———, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sciences, 2 (2009), pp. 183–202.

[8] S. Becker, J. Bobin, and E. Candes, *NESTA: A fast and accurate first-order method for sparse recovery*, SIAM J. on Imaging Sciences, 4 (2011), pp. 1–39.

[9] S. Bonettini and V. Ruggiero, *On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration*, Journal of Mathematical Imaging and Vision, pp. 1–18. DOI: 10.1007/s10851-011-0324-9.

[10] K. Bredies, *A forward-backward splitting algorithm for the minimization of non-smooth convex functionals in Banach space*, Inverse Problems, 25 (2009), pp. 015005, 20.

[11] R. S. Burachik and B. F. Svaiter, *A relative error tolerance for a family of generalized proximal point methods*, Math. Oper. Res., 26 (2001), pp. 816–831.

[12] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vis., 20 (2004), pp. 89–97.

[13] A. Chambolle and P.-L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.

[14] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.

[15] C. Chaux, J.-C. Pesquet, and N. Pustelnik, *Nested iterative algorithms for convex constrained image recovery problems*, SIAM J. Imaging Sci., 2 (2009), pp. 730–762.

[16] P. L. Combettes, Đ. Dũng, and B. C. Vũ, *Dualization of signal recovery problems*, Set-Valued Var. Anal., 18 (2010), pp. 373–404.

[17] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, eds., Springer-Verlag, 2010.

[18] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200 (electronic).

[19] R. Cominetti, *Coupling the proximal point algorithm with approximation methods*, J. Optim. Theory Appl., 95 (1997), pp. 581–600.

[20] R. Correa and C. Lemarechal, *Convergence of some algorithms of convex minimization*, Math. Progr., 62 (1993), pp. 261–275.

[21] I. Daubechies, G. Teschke, and L. Vese, *Iteratively solving linear inverse problems under general convex constraints*, Inverse Problems and Imaging, 1 (2007), pp. 29–46.

[22] O. Devolder, F. Glineur, and Y. Nesterov, *First-order methods of smooth convex optimization with inexact oracle.* http://www.optimization-online.org/DB_FILE/2010/12/2865.pdf, 2011.

[23] J. Duchi and Y. Singer, *Efficient online and batch learning using forward backward splitting*, Journal of Machine Learning Research, 10 (2009), pp. 2899–2934.

[24] J. Eckstein, *Approximate iterations in Bregman-function-based proximal algorithms*, Math. Programming, 83 (1998), pp. 113–123.

[25] B. Eicke, *Iteration methods for convexly constrained ill-posed problems in Hilbert space*, Numer. Funct. Anal. Optim., 13 (1992), pp. 413–429.

[26] E. Esser, X. Zhang, and T.F. Chan, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM J. Imaging Sci., 3 (2010), pp. 1015–1046.
[27] M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, tech. report, IEEE Journal of Selected Topics in Signal Processing, 2007.
[28] M. Fornasier, ed., *Theoretical Foundations and Numerical Methods for Sparse Recovery*, vol. 9 of Radon Series on Computational and Applied Mathematics, De Gruyter, 2010.
[29] O. Güler, *On the convergence of the proximal point algorithm for convex minimization*, SIAM J. Control Optim., 29 (1991), pp. 403–419.
[30] O. Güler, *New proximal point algorithms for convex minimization*, SIAM J. Optim., 2 (1992), pp. 649–664.
[31] B. He and X. Yuan, *An accelerated inexact proximal point algorithm for convex minimization*, J. Optim. Theory Appl., 154 (2012), pp. 536–548.
[32] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms. II*, vol. 306 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, 1993.
[33] R. Jenatton, J.-Y. Audibert, and F. Bach, *Structured variable selection with sparsity-inducing norms*, Journal of Machine Learning Research, 12 (2011), pp. 2777–2824.
[34] B. Lemaire, *About the convergence of the proximal method*, in Advances in optimization (Lambrecht, 1991), vol. 382 of Lecture Notes in Econom. and Math. Systems, Springer, Berlin, 1992, pp. 39–51.
[35] P.-L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.
[36] B. Martinet, *Régularisation d'inéquations variationnelles par approximations successives*, Rev. Française Informat. Recherche Opérationnelle, 4 (1970), pp. 154–158.
[37] R.D.C. Monteiro and B.F. Svaiter, *Convergence rate of inexact proximal point methods with relative error criteria for convex optimization.* http://www.optimization-online.org/DB_HTML/2010/08/2714.html, 2010.
[38] R. Monteiro and B. Svaiter, *An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods*, (2011).
[39] J.-J. Moreau, *Fonctions convexes duales et points proximaux dans un espace hilbertien*, C. R. Acad. Sci. Paris, 255 (1962), pp. 2897–2899.
[40] ———, *Propriétés des applications "prox"*, C. R. Acad. Sci. Paris, 256 (1963), pp. 1069–1071.
[41] ———, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.
[42] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa, *Solving structured sparsity regularization with proximal methods*, in Machine Learning and Knowledge Discovery in Databases, J. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, eds., vol. 6322 of Lecture Notes in Computer Science, Springer, 2010, pp. 418–433.
[43] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*, A Wiley-Interscience Publication, John Wiley & Sons Inc., New York, 1983. Translated from the Russian and with a preface by E. R. Dawson, Wiley-Interscience Series in Discrete Mathematics.
[44] Y. Nesterov, *Introductory lectures on convex optimization. A basic course*, vol. 87 of Applied Optimization, Kluwer Academic Publishers, Boston, MA, 2004.
[45] ———, *Gradient methods for minimizing composite objective function*, tech. report, CORE Discussion Papers from Université Catholique de Louvain, Center for Operations Research and Econometrics No 2007/076, May 2009.
[46] G. Peyré and J. Fadili, *Group sparsity with overlapping partition functions*, in Proc. EUSIPCO 2011, 2011, pp. 303–307.
[47] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.
[48] ———, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optimization, 14 (1976), pp. 877–898.
[49] L. Rosasco, S. Mosci, M. S. Santoro, A. Verri, and S. Villa, *A regularization approach to nonlinear variable selection*, in Proceedings of the 13 International Conference on Artificial Intelligence and Statistics, 2010.
[50] L.I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.
[51] A. Sabharwal and L. C. Potter, *Convexly constrained linear inverse problems: iterative least-squares and regularization*, IEEE Trans. Signal Process., 46 (1998), pp. 2345–2352.

[52] S. Salzo and S. Villa, *Inexact and accelerated proximal point algorithm*, Journal of Convex Analysis, 19 (2012).
[53] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen, *Variational methods in imaging*, vol. 167 of Applied Mathematical Sciences, Springer, New York, 2009.
[54] M. Schmidt, N. Le Roux, and F. Bach, *Convergence rates of inexact proximal-gradient methods for convex optimization.* arXiv:1109.2415v2.
[55] M. V. Solodov and B. F. Svaiter, *A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator*, Set-Valued Anal., 7 (1999), pp. 323–345.
[56] M. V. Solodov and B. F. Svaiter, *A comparison of rates of convergence of two inexact proximal point algorithms*, in Nonlinear optimization and related topics (Erice, 1998), vol. 36 of Appl. Optim., Kluwer Acad. Publ., Dordrecht, 2000, pp. 415–427.
[57] M. V. Solodov and B. F. Svaiter, *Error bounds for proximal point subproblems and associated inexact proximal point algorithms*, Math. Program., 88 (2000), pp. 371–389. Error bounds in mathematical programming (Kowloon, 1998).
[58] ———, *An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions*, Math. Oper. Res., 25 (2000), pp. 214–230.
[59] ———, *A unified framework for some inexact proximal point algorithms*, Numer. Funct. Anal. Optim., 22 (2001), pp. 1013–1035.
[60] A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, T.R. Golub, E.S. Lander, et al., *Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles*, Proceedings of the National Academy of Sciences of the United States of America, 102 (2005), p. 15545.
[61] P. Tseng, *Approximation accuracy, gradient methods, and error bound for structured convex optimization*, Mathematical Programming, 125 (2010), pp. 263–295. 10.1007/s10107-010-0394-2.
[62] M.J. Van De Vijver, Y.D. He, L.J. Van't Veer, H. Dai, A.A.M. Hart, D.W. Voskuil, G.J. Schreiber, J.L. Peterse, C. Roberts, M.J. Marton, et al., *A gene-expression signature as a predictor of survival in breast cancer*, New England Journal of Medicine, 347 (2002), pp. 1999–2009.
[63] Y. Yao and N. Shahzad, *Strong convergence of a proximal point algorithm with general errors*, Optimization Letters, 6 (2012), pp. 621–628.
[64] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, Journal of the Royal Statistical Society, Series B, 68 (2006), pp. 49–67.
[65] C. Zălinescu, *Convex analysis in general vector spaces*, World Scientific Publishing Co. Inc., River Edge, NJ, 2002.
[66] A. Zaslavski, *Convergence of a proximal point method in the presence of computational errors in Hilbert spaces*, SIAM J. Optim., 20 (2010), pp. 2413–2421.
[67] P. Zhao, G. Rocha, and B. Yu, *The composite absolute penalties family for grouped and hierarchical variable selection*, Ann. Statist., 37 (2009), pp. 3468–3497.