Inexact and accelerated proximal point algorithms

Saverio Salzo

DISI, Università di Genova

Via Dodecaneso 35 - 16146, Genova, Italy

email: salzo@disi.unige.it

Silvia Villa

DIMA & DISI, Università di Genova

Via Dodecaneso 35 - 16146, Genova, Italy

email: villa@dima.unige.it

We present inexact accelerated proximal point algorithms for minimizing a proper lower semicontinuous and convex function. We carry on a convergence analysis under different types of errors in the evaluation of the proximity operator, and we provide corresponding convergence rates for the objective function values. The proof relies on a generalization of the strategy proposed in [14] for generating estimate sequences according to the definition of Nesterov, and is based on the concept of ε-subdifferential. We show that the convergence rate of the exact accelerated algorithm $1/k^2$ can be recovered by constraining the errors to be of a certain type.

Keywords: accelerated proximal point algorithms, global convergence rates, approximation criteria

Mathematical Subject Classification: 90C25 (49D37, 65K10)

1 Introduction

Given a proper, convex and lower semicontinuous function $F : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ defined on a Hilbert space $\mathcal{H}$, we consider the optimization problem

$$\min_{x \in \mathcal{H}} F(x). \tag{P}$$

The proximal point algorithm, as introduced by Martinet first [17] and later generalized by Rockafellar [25] is designed to cope with problem (P) and generates for any starting point x_0 a minimizing sequence for F by the rule

$$x_{k+1} = \operatorname{prox}_{\lambda_k F}(x_k),$$

where $\text{prox}_{\lambda_k F}(x_k) := \text{argmin}_{y\in\mathcal{H}} \left\{ F(y) + \frac{1}{2\lambda_k}\|y - x_k\|^2 \right\}$, with $\lambda_k > 0$. Denoting by F^* the optimal value of (P), the sequence of the objective function values $F(x_k)$ converges to F^* under minimal assumptions on the λ_k's. Furthermore, the convergence rate $F(x_k) - F^* = O(1/k)$ has been shown if the minimum F^* is attained [13].

Motivated by several applications in the context of image processing, inverse problems and machine learning, recently there has been an active interest in accelerations and modifications of the classical proximal point algorithm (see e.g. [11, 4, 3]).

Resorting to the ideas contained in the seminal work of Nesterov [23, 22], Güler, in [14], devises an elegant way to accelerate the proximal point algorithm achieving the convergence rate $F(x_k) - F^* = O(1/k^2)$ — which is optimal for a first order method in the sense defined in [21]. Note that, until now accelerated methods provide only convergence of the objective function values, in general without any guarantee of convergence on the sequence $(x_k)_{k\in\mathbb{N}}$.

The computational effort of accelerated methods is comparable with that of the standard proximal point algorithm and mainly lies in the minimization subproblem required to calculate the proximal point at each iteration. In fact, very often in the applications, a formula for the proximity operator is not available in closed form. This happens for instance when applying proximal methods to image deblurring with total variation [9, 5, 2], or to structured sparsity regularization problems in machine learning and inverse problems [16, 12, 20]. In those cases the proximity operator is usually computed using ad hoc algorithms, and therefore inexactly. For this reason, it is indeed critical to study the convergence of the algorithms under possible perturbations of proximal points. This program has been pursued in the pioneering paper by Rockafellar [25] for what concerns the basic proximal point algorithm, and under different notions of admissible approximations of proximal points. Since then, there has been a growing interest in inexact implementations of proximal methods and many works appeared, treating the problem under different perspectives, see e.g. [26, 27, 28, 29, 8, 32].

As regards the accelerated schemes, the paper [14] by Güler is the first one dealing with computational errors. The analysis of convergence is performed first in the exact case and is then carried on, in Theorem 3.1, to handle certain types of computational errors, preserving the $1/k^2$ rate of convergence. Unfortunately, we discovered such theorem contains a subtle error and therefore the proof of convergence for the inexact case remains an open issue (see Remark 2).

In this paper we analyze the convergence of accelerated and inexact procedures for the proximal point algorithm. The derivation of the algorithm relies on the machinery of estimate sequences, as first proposed by Nesterov, see [22]. Inspired by the results in [14], we present a more flexible method to build estimate sequences, that can be adapted easily to different notions of approximation for the proximal point. We show two facts: first, that leveraging on a different concept of admissible errors (named of *type 2*), it is still possible to get quadratic convergence of inexact and accelerated schemes, and secondly, that using even a generalization of the type of errors considered in [14] (named of *type 1*), convergence of the inexact and accelerated proximal point algorithm is guaranteed, but only with the rate $1/k$. In both cases conditions on the asymptotic behavior of the errors' magnitude are needed. It is worth noting, in handling errors of type 2, that the convergence of the inexact algorithm holds even if the sequence of errors is not summable,

which contrasts the common requirement of summability of errors in the related literature (see e.g. [25]).

The paper is organized as follows: in Section 2 we introduce the definition of different types of errors for the evaluation of the proximity operator and give several examples for the simple but representative case of the projection operator onto a convex set. In Section 3 we quickly review the theory of estimate sequences and we propose a new method to construct them. Convergence analysis of the algorithms is performed in Section 4, where the general scheme is applied according to two different notions of error. Finally, in Section 5 we state the given algorithms in equivalent forms, allowing for simpler formulations, and comparisons with other well-known methods.

2 General setting

2.1 Assumptions

Let $\mathcal{H}$ be a Hilbert space and consider a proper, convex and lower semicontinuous function $F : \mathcal{H} \to (-\infty, +\infty]$. We focus on the optimization problem

$$\inf_{x \in \mathcal{H}} F(x).$$

We denote by F^* the infimum of F and we do not require the infimum to be attained, neither to be finite.

2.2 Inexact computations of the proximal point

The algorithms analyzed in this paper are based on the computation of the proximity operator of the function F, introduced by Moreau [19], and then made popular in the optimization literature by Martinet [17] and Rockafellar [25, 24]. For $\lambda > 0$ and $y \in \mathcal{H}$, *the proximal point of y with respect to λF* is defined by setting

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

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

$$z = \operatorname{prox}_{\lambda F}(y) \iff 0 \in \partial\Phi_\lambda(z) \iff \frac{y - z}{\lambda} \in \partial F(z), \tag{1}$$

where ∂ denotes the subdifferential operator. The last equivalence shows also that $\operatorname{prox}_{\lambda F}(y) = (I + \lambda\partial F)^{-1}(y)$.

From a practical point of view the computation of the proximity operator can be as difficult as the computation of a solution of the initial problem (even though the strong convexity of Φ_λ can be a key advantage), so it is essential to replace the proximal point with an approximate version of it. We introduce here three concepts of approximation of the proximal point. The first two are based on the notion of ε-subdifferential. The second has already been considered in [1], while the third one has been first proposed by

Rockafellar in [25] and further investigated in [14]. For other notions of approximations used in the context of proximal point algorithms see also [7, 26, 27, 28, 29, 8, 32].

We recall that the *ε-subdifferential of F at the point $z \in \mathrm{dom}F$* is the set

$$\partial_\varepsilon F(z) = \{\xi \in \mathcal{H} \,:\, F(x) \geq F(z) + \langle x - z, \xi\rangle - \varepsilon,\ \forall x \in \mathcal{H}\}\,. \tag{2}$$

For our purposes, it is worth noting that in general it holds

$$0 \in \partial_\varepsilon F(z) \iff F(z) \leq \inf F + \varepsilon.$$

All the notions of approximation we are going to introduce are of absolute type and are based on the relaxation of conditions characterizing the proximal point — see in particular those given in equation (1). Since in the applications the proximal point is often sought by applying an iterative algorithm for minimizing Φ_λ, the following notion of approximate proximal point is very natural.

Definition 1. We say that $z \in \mathcal{H}$ *is a type 1 approximation of* $\mathrm{prox}_{\lambda F}(y)$ *with ε-precision* and we write $z \approx_1 \mathrm{prox}_{\lambda F}(y)$ if and only if

$$0 \in \partial_{\frac{\varepsilon^2}{2\lambda}} \Phi_\lambda(z). \tag{AT1}$$

It is important to note (see [25, 14]) that if $z \approx_1 \mathrm{prox}_{\lambda F}(y)$ with ε-precision, then

$$z \in \mathrm{dom}F \ \text{ and } \ \|z - \mathrm{prox}_{\lambda F}(y)\| \leq \varepsilon.$$

Indeed, being Φ_λ strongly convex with modulus $1/\lambda$ and $0 \in \partial\Phi_\lambda(\mathrm{prox}_{\lambda F}(y))$, we have

$$\Phi_\lambda(z) - \Phi_\lambda(\mathrm{prox}_{\lambda F}(y)) \geq \frac{1}{2\lambda}\|z - \mathrm{prox}_{\lambda F}(y)\|^2$$

The statement follows from the fact that

$$\frac{\varepsilon^2}{2\lambda} \geq \Phi_\lambda(z) - \Phi_\lambda^*$$

being $0 \in \partial_{\varepsilon^2/(2\lambda)}\Phi_\lambda(z)$.

Another notion of approximation is obtained by relaxing the last equation in (1) in the way specified in the next definition.

Definition 2. We say that $z \in \mathcal{H}$ is a *type 2 approximation of* $\mathrm{prox}_{\lambda F}(y)$ *with ε-precision* and we write $z \approx_2 \mathrm{prox}_{\lambda F}(y)$ if and only if

$$\frac{y - z}{\lambda} \in \partial_{\frac{\varepsilon^2}{2\lambda}} F(z). \tag{AT2}$$

The condition in equation (AT2) can be written equivalently as

$$y \in z + \lambda\partial_{\frac{\varepsilon^2}{2\lambda}} F(z) \iff z \in (I + \lambda\partial_{\frac{\varepsilon^2}{2\lambda}} F)^{-1}(y).$$

Recalling that the proximity operator of λF is defined as $(I + \lambda\partial F)^{-1}$, the admissible approximations of type 2 can be interpreted as a kind of an ε-enlargement of the proximity operator. A similar concept of error has been proposed for non accelerated proximal algorithms in [1] and very recently in the preprint [18]. Finally we recall the concept introduced in [25] and treated by Güler in [14].

Definition 3. We say that $z \in \mathcal{H}$ is a *type 3 approximation of* $\operatorname{prox}_{\lambda F}(y)$ *with* ε*-precision* and we write $z \approx_3 \operatorname{prox}_{\lambda F}(y)$ if and only if

$$d(0, \partial\Phi_\lambda(z)) \leq \frac{\varepsilon}{\lambda}. \tag{AT3}$$

Condition (AT3) can be written in another way. In fact

$$\begin{aligned} d(0, \partial\Phi_\lambda(z)) \leq \varepsilon/\lambda &\iff \exists\, e \in \mathcal{H},\ \|e\| \leq \varepsilon/\lambda,\ e \in \partial\Phi_\lambda(z) \\ &\iff \exists\, e \in \mathcal{H},\ \|e\| \leq \varepsilon,\ (y - z + e)/\lambda \in \partial F(z) \\ &\iff \exists\, e \in \mathcal{H},\ \|e\| \leq \varepsilon,\ z = (I + \lambda\partial F)^{-1}(y + e). \end{aligned}$$

Therefore, we find out that

$$z \approx_3 \operatorname{prox}_{\lambda F}(y) \text{ with } \varepsilon\text{-precision} \iff \exists\, e \in \mathcal{H},\ \|e\| \leq \varepsilon,\ z = \operatorname{prox}_{\lambda F}(y + e).$$

This equivalence means that an approximation of type 3 of the proximal point of y is nothing but the exact proximal point of an ε-perturbation of y.

The concepts just introduced are not independent one of each other. In order to clarify the relationships among them, we start with some equivalent formulations of the first concept of approximation introduced above. Theorem 2.8.7 in [31] and formula (1.2.5) in Chap. XI of [15] ensures that the $\varepsilon^2/(2\lambda)$-subdifferential of Φ_λ can be written as

$$\begin{aligned} \partial_{\frac{\varepsilon^2}{2\lambda}}\Phi_\lambda(z) &= \bigcup_{0 \leq \varepsilon_1 + \varepsilon_2 \leq \frac{\varepsilon^2}{2\lambda}} \partial_{\varepsilon_1} F(z) + \partial_{\varepsilon_2} \frac{1}{2\lambda}\|\cdot - y\|^2(z) \\ &= \bigcup_{0 \leq \varepsilon_1 + \varepsilon_2 \leq \frac{\varepsilon^2}{2\lambda}} \partial_{\varepsilon_1} F(z) + \left\{ \frac{z - y + e}{\lambda} \,:\, \frac{\|e\|^2}{2\lambda} \leq \varepsilon_2 \right\}. \end{aligned}$$

From the last formula we shall derive some equivalent descriptions of the first type of approximation, which will be proved to be useful in studying convergence of the algorithms and that we summarize in the following lemma.

Lemma 1. *The following statements are equivalent:*

i) $z \approx_1 \operatorname{prox}_{\lambda F}(y)$ *with* ε*-precision;*

ii) $\exists\,(\varepsilon_1, \varepsilon_2) \in \mathbb{R}^2_+,\ 0 \leq \varepsilon_1 + \varepsilon_2 \leq \varepsilon^2/(2\lambda), \exists\,\xi \in \partial_{\varepsilon_1} F(z), \exists\, e \in \mathcal{H},\ \|e\|^2/(2\lambda) \leq \varepsilon_2$ *such that* $\lambda\xi + (z - y + e) = 0$*;*

iii) $\exists\,(\varepsilon_1, \varepsilon_2) \in \mathbb{R}^2_+,\ 0 \leq \varepsilon_1^2 + \varepsilon_2^2 \leq \varepsilon^2, \exists\, e \in \mathcal{H},\ \|e\| \leq \varepsilon_2$ *such that* $(y - z + e)/\lambda \in \partial_{\varepsilon_1^2/(2\lambda)} F(z)$.

This lemma sheds light also on the link among the various definitions of error given so far. In particular, one can see that, in some sense, approximations of type 2 and 3 are extreme cases of approximations of type 1, corresponding to the choices $\varepsilon_2 = 0$ and $\varepsilon_1 = 0$, respectively. The next proposition states this fact more formally.

Proposition 1. *The following implications hold true*

1. $z \approx_2 \operatorname{prox}_{\lambda F}(y)$ *with* ε*-precision* $\implies$ $z \approx_1 \operatorname{prox}_{\lambda F}(y)$ *with* ε*-precision;*

2. $z \approx_3 \operatorname{prox}_{\lambda F}(y)$ *with* ε*-precision* $\implies$ $z \approx_1 \operatorname{prox}_{\lambda F}(y)$ *with* ε*-precision.*

If in addition F *is strongly convex of modulus* $\mu > 0$

3. $z \approx_1 \operatorname{prox}_{\lambda F}(y)$ *with* ε*-precision* $\implies$ $z \approx_2 \operatorname{prox}_{\lambda F}(y)$ *with* $\varepsilon\sqrt{(\lambda\mu+1)/\lambda\mu}$*-precision; thus* $z \approx_2 \operatorname{prox}_{\lambda F}(y)$ *with* $\sqrt{2}\varepsilon$*-precision, if* $\lambda \geq 1/\mu$.

Proof. 1. If we take $(y-z)/\lambda \in \partial_{\varepsilon^2/(2\lambda)} F(z)$, then choosing $\varepsilon_1 = \varepsilon$, $\varepsilon_2 = 0$ (and $e = 0$), condition *iii)* in Lemma 1 is satisfied and therefore we get $z \approx_1 \operatorname{prox}_{\lambda F}(y)$ with ε-precision.
2. Since $z \approx_3 \operatorname{prox}_{\lambda F}(y)$, there exists $e \in \mathcal{H}$ with $\|e\| \leq \varepsilon$ such that $(y - z + e)/\lambda \in \partial F(z)$, then choosing $\varepsilon_1 = 0$ and $\varepsilon_2 = \varepsilon$ makes condition *iii)* again fulfilled.
3. From the definition

$$0 \in \partial_{\frac{\varepsilon^2}{2\lambda}} \Phi_\lambda(z) \iff F(x) + \frac{1}{2\lambda}\|x-y\|^2 \geq F(z) + \frac{1}{2\lambda}\|z-y\|^2 - \frac{\varepsilon^2}{2\lambda} \quad \forall\, x \in \mathcal{H},$$

and on the other hand

$$\|z-y\|^2 = -\|z-x\|^2 + \|x-y\|^2 + 2\langle x-z, y-z\rangle.$$

Thus, we have

$$0 \in \partial_{\frac{\varepsilon^2}{2\lambda}} \Phi_\lambda(z) \iff F(x) \geq F(z) + \frac{1}{\lambda}\langle x-z, y-z\rangle - \frac{1}{2\lambda}\|x-z\|^2 - \frac{\varepsilon^2}{2\lambda}, \quad \forall\, x \in \mathcal{H}.$$

Since F is strongly convex of modulus $\mu > 0$, for every $\theta \in (0,1]$, writing the previous inequality replacing x with $\theta x + (1-\theta)z$, we obtain

$$\begin{aligned}
\theta F(x) + (1-\theta)F(z) &\geq F(\theta x + (1-\theta)z) + \frac{\mu}{2}\theta(1-\theta)\|x-z\|^2 \\
&\geq F(z) + \frac{1}{\lambda}\langle \theta(x-z), y-z\rangle - \frac{1}{2\lambda}\|\theta(x-z)\|^2 - \frac{\varepsilon^2}{2\lambda} \\
&\qquad + \frac{\mu}{2}\theta(1-\theta)\|x-z\|^2.
\end{aligned}$$

Hence, simplifying, dividing by θ and then requiring $\lambda\mu(1-\theta) \geq \theta$, we finally get

$$\begin{aligned}
F(x) &\geq F(z) + \frac{1}{\lambda}\langle x-z, y-z\rangle + \frac{1}{2\lambda}\Big(\lambda\mu(1-\theta) - \theta\Big)\|x-z\|^2 - \frac{\varepsilon^2}{2\lambda\theta} \\
&\geq F(z) + \langle x-z, \frac{y-z}{\lambda}\rangle - \frac{\varepsilon^2}{2\lambda\theta},
\end{aligned}$$

which implies $(y-z)/\lambda \in \partial_{\varepsilon^2/(2\lambda\theta)} F(z)$. If we choose $\theta = \lambda\mu/(1+\lambda\mu)$, then $(y-z)/\lambda \in \partial_{\varepsilon^2(1+\lambda\mu)/(2\lambda^2\mu)} F(z)$, meaning $z \approx_2 \operatorname{prox}_{\lambda F}(z)$ with $\varepsilon\sqrt{(1+\lambda\mu)/(\lambda\mu)}$-precision. Moreover, if $\lambda \geq 1/\mu$, it is possible to choose $\theta = 1/2$ obtaining $(y-z)/\lambda \in \partial_{\sqrt{2}\varepsilon} F(z)$. □

Example 1. To clarify what kind of approximations are allowed applying the various error criteria, we describe the case where F is the indicator function of a closed and convex set C, and the proximal operator is consequently the projection onto C, denoted by P_C. Given $y \in \mathcal{H}$, it holds

$$z \approx_1 P_C(y) \text{ with } \varepsilon\text{-precision} \iff z \in C \text{ and } \|z - y\|^2 \leq d(y, C)^2 + \varepsilon^2.$$

As noted above this is the less restrictive concept of error. If $y \notin C$, approximations of type 1 do not necessarily belong to the boundary of C, but lie in the portion of C belonging to a ball centered in y with a radius greater then $d(y, C)$. An example is shown in Figure 1.

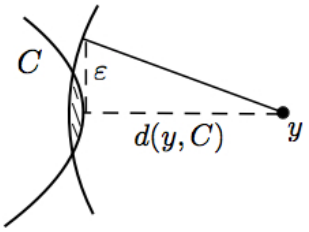

Figure 1: Type 1 approximations

The second notion of approximation, when we deal with projections, can be formulated in the following way:

$$z \approx_2 P_C(y) \text{ with } \varepsilon\text{-precision} \iff z \in C \text{ and } \langle x - z, y - z\rangle \leq \frac{\varepsilon^2}{2} \quad \forall x \in C. \tag{3}$$

Recalling that the projection $P_C(y)$ of a point y is the unique point $z \in C$ which satisfies $\langle x - z, y - z\rangle \leq 0$, approximations of type 2 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 $C \subseteq \{x \in X : \langle x - P_C(y), y - P_C(y)\rangle \leq 0\}$. In Figure 2 an admissible approximation of $P_C(y)$ is depicted. To check that z satisfies inequality (3) for all $x \in C$, 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.

Approximations of type 3 are the points belonging to C that can be written as the projection of points belonging to a ball of radius ε centered at y:

$$z \approx_3 P_C(y) \text{ with } \varepsilon\text{-precision} \iff z = P_C(y + e), \text{ with } \|e\| \leq \varepsilon.$$

Therefore, if $y \notin C$, those approximations belong to the boundary of C (see Figure 3).

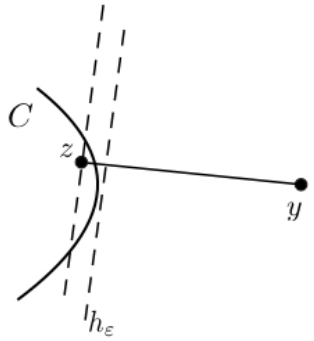

Figure 2: Type 2 approximations

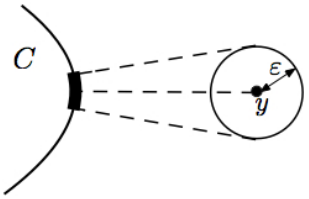

Figure 3: Type 3 approximations

We remark that if C is bounded, an approximation of type 3 can be regarded as a type 2 approximation. More precisely:

$$z \approx_3 P_C(y) \text{ with } \varepsilon\text{-precision} \implies z \approx_2 P_C(y) \text{ with } \sqrt{2\text{diam}(C)\varepsilon}\text{ -precision.}$$

To this purpose, suppose $z \approx_3 P_C(y)$ with ε-precision. Then, there exists $e \in \mathcal{H}$, $\|e\| \leq \varepsilon$ such that $z = P_C(y+e)$ and for all $x \in C$ it holds

$$\begin{aligned}\langle x - z, y - z\rangle &= \langle x - P_C(y+e), y + e - P_C(y+e)\rangle - \langle x - P_C(y+e), e\rangle \\ &\leq \|x - P_C(y+e)\|\|e\| \\ &\leq \text{diam}(C)\varepsilon.\end{aligned}$$

3 Nesterov's estimate sequences

In [22], Nesterov illustrates a flexible mechanism to produce minimizing sequences for an optimization problem. The idea is to generate recursively a sequence of simple functions that approximate F in the sense introduced below. In this section we briefly describe this method and provide new general results for constructing quadratic estimate sequences when F is convex.

3.1 General framework

Definition 4. 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 $F : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ iff

$$\forall x \in \mathcal{H}, \forall k \in \mathbb{N} : \ \varphi_k(x) - F(x) \leq \beta_k(\varphi_0(x) - F(x)) \quad \text{and} \quad \beta_k \to 0. \tag{4}$$

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 1. *Let $((\varphi_k)_{k\in\mathbb{N}}, (\beta_k)_{k\in\mathbb{N}})$ be an estimate sequence of F and denote by φ_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*

$$F(x_k) \leq \varphi_k^* + \delta_k, \tag{5}$$

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

$$F(x_k) \leq \beta_k(\varphi_0(x) - F(x)) + \delta_k + F(x). \tag{6}$$

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

$$F(x_k) - F^* \leq \beta_k(\varphi_0(x^*) - F^*) + \delta_k$$

Proof. Suppose that x_k and δ_k satisfy (5). Then for $x \in \mathcal{H}$, we have

$$F(x_k) \leq \varphi_k^* + \delta_k \leq \varphi_k(x) + \delta_k.$$

Using that φ_k is an estimate sequence of F we get

$$F(x_k) \leq \beta_k(\varphi_0(x) - F(x)) + F(x) + \delta_k, \qquad \forall\, x \in \mathrm{dom}F.$$

Furthermore, if $\beta_k \to 0$ and $\delta_k \to 0$ we obtain

$$\limsup_{k\to+\infty} F(x_k) \leq \limsup_{k\to+\infty} \beta_k(\varphi_0(x) - F(x)) + F(x) + \delta_k = F(x).$$

The previous inequality, holding for every $x \in \mathrm{dom}F$, yields $\limsup_{k\to+\infty} F(x_k) \leq F^*$. Thus, the following chain of inequalities holds

$$F^* \leq \liminf_{k\to+\infty} F(x_k) \leq \limsup_{k\to+\infty} F(x_k) \leq F^*$$

proving that (x_k) is a minimizing sequence for F. If there exists x^* such that $F(x^*) = F^*$, inequality (6) can be specialized for $x = x^*$ giving

$$F(x_k) - F^* \leq \beta_k(\varphi_0(x^*) - F^*) + \delta_k$$

□

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. This result has been also stressed in the already cited paper [14] in Theorem 4.1. However, the hypothesis of attainability of the infimum is required if an estimate of the rate of convergence is needed.

Remark 1. An estimate sequence is usually generated by checking a recursive inequality. Indeed, if $(\varphi_k)_{k\in\mathbb{N}}$ satisfies

$$\varphi_{k+1}(x) - F(x) \leq (1-\alpha_k)(\varphi_k(x) - F(x)), \tag{7}$$

with $0 \leq \alpha_k < 1$, then $((\varphi_k)_{k\in\mathbb{N}}, (\beta_k)_{k\in\mathbb{N}})$ is an estimate sequence of F with

$$\beta_k = \prod_{i=0}^{k-1}(1-\alpha_i) \tag{8}$$

provided that $\sum_{i=0}^{+\infty} \alpha_i = +\infty$.

In fact, iterating the inequality (7) k times, we obtain the basic inequality of estimate sequences given in (4) with β_k as in (8). Furthermore $\beta_k \in (0,1]$, and the following equivalence holds

$$\beta_k \to 0 \Longleftrightarrow \sum_{i=0}^{\infty} \alpha_i = +\infty. \tag{9}$$

For proving (9) first note that by definition

$$\beta_k \to 0 \iff \prod_{i=0}^{k-1}(1-\alpha_i) \to 0 \iff \sum_{i=0}^{+\infty} \log(1-\alpha_i) = -\infty.$$

Now suppose that $\sum_{i=0}^{\infty} \alpha_i = +\infty$. Then, using the inequality $\log(1-x) \leq -x$, which holds for every $x \in [0,1)$, we get

$$\sum_{i=0}^{\infty} \log(1-\alpha_i) \leq \sum_{i=0}^{\infty} -\alpha_i = -\infty,$$

implying that $\beta_k \to 0$.

For the converse, assume that $\beta_k \to 0$. There are two cases: either there exists a subsequence $(\alpha_{i_l})_{l\in\mathbb{N}}$ satisfying $\alpha_{i_l} \geq 1/2$, and the proof in this case is finished, or α_i definitely satisfies $\alpha_i \leq 1/2$. If the latter condition holds, then using the inequality $\log(1-x) \geq -2x$, which holds for $x \in [0, 1/2)$, we get

$$-\infty = \sum_{i=0}^{\infty} \log(1-\alpha_i) \geq -2\sum_{i=0}^{\infty} \alpha_i,$$

and therefore the proof is concluded.

3.2 Construction of an estimate sequence

We present in this section a general procedure for generating an estimate sequence of a proper, lower semicontinuous and convex function $F : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$. Remember an estimate sequence of F is a pair of sequences, a sequence of (simple) functions φ_k and a numerical sequence β_k. Besides this, in order to make things work, we also need a third sequence of points x_k.

First of all, let us deal with the generation of the sequence of functions. Denote by $\mathcal{F}(\mathcal{H},\mathbb{R})$ the space of functions from $\mathcal{H}$ to $\mathbb{R}$. For a given F, we define an updating rule for functions $\varphi \in \mathcal{F}(\mathcal{H},\mathbb{R})$, depending on the choice of four parameters $(z,\varepsilon,\xi,\alpha) \in \mathrm{dom}F \times \mathbb{R}_+ \times \mathcal{H} \times [0,1)$, as

$$U(z,\varepsilon,\xi,\alpha) : \mathcal{F}(\mathcal{H},\mathbb{R}) \to \mathcal{F}(\mathcal{H},\mathbb{R})$$

$$U(z,\varepsilon,\xi,\alpha)(\varphi)(x) = (1-\alpha)\varphi(x) + \alpha(F(z) + \langle x - z, \xi\rangle - \varepsilon).$$

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

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

hiding the dependence on the parameters. The same hat notation will be used also for other quantities: in all cases it will stand for an update of the corresponding variable. The iteration of the operator $U(z,\varepsilon,\xi,\alpha)$ will allow us to generate sequences $(\varphi_k)_{k\in\mathbb{N}}$ which turn out to be estimate sequences for F. Indeed, it is easy to see that if $\xi \in \partial_\varepsilon F(z)$, then the following inequality holds

$$\hat{\varphi}(x) - F(x) \le (1-\alpha)(\varphi(x) - F(x))$$

which in fact resembles the recursive inequality of estimate sequences (7). Indeed, recalling the definition of ε-subdifferential (2), note that

$$\begin{aligned} \hat{\varphi}(x) - F(x) &= (1-\alpha)(\varphi(x) - F(x)) + \alpha(F(z) + \langle x - z, \xi\rangle - \varepsilon - F(x)) \\ &\le (1-\alpha)(\varphi(x) - F(x)). \end{aligned}$$

Summarizing, given $((z_k,\varepsilon_k,\xi_k,\alpha_k))_{k\in\mathbb{N}}$, $(z_k,\varepsilon_k,\xi_k,\alpha_k) \in \mathrm{dom}F \times \mathbb{R}_+ \times \mathcal{H} \times [0,1)$ with $\xi_{k+1} \in \partial_{\varepsilon_k} F(z_{k+1})$, $\sum_{k\in\mathbb{N}} \alpha_k = +\infty$, and an arbitrary function $\varphi : \mathcal{H} \to \mathbb{R}$, the sequence defined by setting

$$\begin{cases} \varphi_0 = \varphi \\ \varphi_{k+1} = U(z_{k+1},\varepsilon_k,\xi_{k+1},\alpha_k)\varphi_k, \end{cases} \tag{10}$$

satisfies the inequality (7) and $\beta_k = \prod_{i=0}^{k-1}(1-\alpha_k) \to 0$ thanks to Remark 1, being therefore an estimate sequence of F.

We now describe in detail the update when the starting φ is a quadratic function written in canonical form, namely

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

where clearly $\varphi^* = \inf\varphi$. 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

$$\begin{cases} \hat{\varphi}^* = (1-\alpha)\varphi^* + \alpha F(z) + \alpha\langle \nu - z, \xi\rangle - \dfrac{\alpha^2}{2(1-\alpha)A}\|\xi^2\| - \alpha\varepsilon \\ \hat{A} = (1-\alpha)A \\ \hat{\nu} = \nu - \dfrac{\alpha}{(1-\alpha)A}\xi. \end{cases} \tag{11}$$

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

Next, the problem of generating a sequence $(x_k)_{k\in\mathbb{N}}$ satisfying inequality (5) shall be treated. We state a generalization of Theorem 2.1 and Lemma 3.2 in [14] and of Lemma 2.1 in [6], that will be crucial in the whole subsequent analysis. Though the proof follows closely the cited results, the use of the approximate subdifferential brings about much more flexibility with respect to the original version, as will be clear later.

Lemma 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 $z, \xi \in \mathcal{H}$, $\varepsilon \geq 0$ are given with $\xi \in \partial_\varepsilon F(z)$, defining $\hat{\varphi} = U(z, \varepsilon, \xi, \alpha)(\varphi)$, with $\alpha \in [0, 1)$ and setting $y = (1-\alpha)x + \alpha\nu$, we get*

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

Proof. Consider $\varphi = \varphi^* + A/2\|\cdot - \nu\|^2$ and $(z, \xi) \in \mathcal{H}^2$, $\varepsilon \geq 0$ with $\xi \in \partial_\varepsilon F(z)$ and $\alpha \in [0, 1)$. Define $\hat{\varphi} = U(z, \varepsilon, \xi, \alpha)\varphi$ and suppose $x \in \mathcal{H}$ to satisfy $\delta + \varphi^* \geq F(x)$ for some $\delta \geq 0$. From the expression of $\hat{\varphi}^*$ in (11) we get

$$\hat{\varphi}^* = (1-\alpha)\varphi^* + \alpha F(z) + \alpha\langle \nu - z, \xi\rangle - \frac{\alpha^2}{2(1-\alpha)A}\|\xi\|^2 - \alpha\varepsilon,$$

and, being $\xi \in \partial_\varepsilon F(z)$, we obtain

$$\delta + \varphi^* \geq F(x) \geq F(z) + \langle x - z, \xi\rangle - \varepsilon$$

therefore

$$\begin{aligned}(1-\alpha)\delta + \hat{\varphi}^* &\geq F(z) + \langle (1-\alpha)x + \alpha\nu - z, \xi\rangle - \frac{\alpha^2}{2(1-\alpha)A}\|\xi\|^2 - \varepsilon \\ &= F(z) + \langle (1-\alpha)x + \alpha\nu - (\lambda\xi + z), \xi\rangle \\ &\qquad + \left(\lambda - \frac{\alpha^2}{2(1-\alpha)A}\right)\|\xi\|^2 - \varepsilon.\end{aligned}$$

Defining $y = (1-\alpha)x + \alpha\nu$ and substituting it in the previous equation we get the statement. □

Given $x \in \mathcal{H}$ satisfying $F(x) \leq \varphi^* + \delta$ for some $\delta \geq 0$, the inequality stated in the previous lemma suggests different possibilities to choose an update of x, denoted $\hat{x}$, which satisfies the analogous condition $F(\hat{x}) \leq \hat{\varphi}^* + \hat{\delta}$ for a suitable choice of $\hat{\delta}$. This will be clarified in the following sections.

The last ingredient for completing the framework of estimate sequences is the numerical sequence $(\beta_k)_{k\in\mathbb{N}}$ which should be constructed in such a way that $\beta_k \to 0$. To this purpose and motivated also by the previous lemma we give a further result that is a slight generalization of Lemma 2.2 in [14]. The proof is omitted because it follows the same line of the original one.

Lemma 3. *Given the numerical sequence* $(\lambda_k)_{k\in\mathbb{N}}$, $\lambda_k > 0$ *and* $A > 0$, $a, b > 0$, $a \leq b$, *define* $(A_k)_{k\in\mathbb{N}}$ *and* $(\alpha_k)_{k\in\mathbb{N}}$ *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 $\beta_k := \prod_{i=0}^{k-1}(1-\alpha_i)$ *satisfies*

$$\frac{1}{(1+\sqrt{bA}\sum_{j=0}^{k-1}\sqrt{\lambda_j})^2} \leq \beta_k \leq \frac{1}{(1+(\sqrt{aA}/2)\sum_{j=0}^{k-1}\sqrt{\lambda_j})^2} \tag{12}$$

In particular, $\beta_k \sim 1/(\sum_{j=0}^{k-1}\sqrt{\lambda_j})^2$ *and* $\beta_k \to 0$ *if and only if* $\sum_{k=0}^{\infty}\sqrt{\lambda_k} = +\infty$. *Moreover, if* $\lambda_k \geq \lambda > 0$ *for every* $k \in \mathbb{N}$, *then* $\beta_k = O(1/k^2)$.

4 Convergence analysis of the algorithms

In this section we show how to employ the general framework of estimate sequences introduced in Section 3 for constructing inexact proximal point algorithms of various types according to the different definitions of error given above. Convergence of the algorithms shall be analyzed and the rate of convergence shall be provided as well. In case of errors of type 1, the main result states that the generated sequence is minimizing for F and if a minimizer exists, it shares the convergence rate

$$F(x_k) - F^* = O(1/k).$$

This result corrects the one given in [14]. Indeed, as we shall explain in the following Remark 2, a subtle error is present in Güler's proof that makes vain the conclusion about the quadratic convergence of the algorithm under inexact computation of the proximal points. We were able to fix the problem and recover the convergence of the algorithm even under slightly more general errors, but we could obtain only linear rate of convergence, no matter how fast the error goes to zero. This suggests that the use of an accelerated algorithm under the presence of errors of type 1 could have no effect in practice because potentially equivalent to the non accelerated version. We refer the reader to [10, 2] for some considerations on this fact.

Then, we study the algorithm under errors of type 2 (more demanding than 1) and here the situation changes. Indeed, we were able to get again quadratic convergence

$$F(x_k) - F^* = O(1/k^2)$$

if the errors go to zero fast enough. One further result is that, if only errors of type 2 are allowed, the algorithm converges even if errors go to zero more slowly then usually required, with the sum of total errors that can be possibly infinite. This is a remarkable fact that does not occur in the classical (non accelerated) proximal point algorithm where summability of the errors is required, see [25].

4.1 The algorithm with errors of type 1

The first result of this section is an application of Lemma 2. It is essentially the core of Theorem 3.1 given in Güler's paper [14].

Theorem 2. *Fix $\lambda > 0$ and $\varepsilon \geq 0$. Let $x, \nu \in X$, $A > 0$ and $\varphi = \varphi^* + A/2\|\cdot - \nu\|^2$ such that $F(x) \leq \varphi^* + \delta$ for some $\delta \geq 0$. If $\alpha^2/((1-\alpha)A\lambda) = 1$, choosing*

$$t = (1-\alpha)x + \alpha\nu$$

$$\hat{x} \approx_1 \operatorname{prox}_{\lambda F}(t) \qquad \textit{with } \varepsilon\textit{-precision}$$

$$\hat{\varphi} = U(z, 0, \xi, \alpha)\varphi \qquad \textit{with } z = \operatorname{prox}_{\lambda F}(t),\ \xi = \frac{t-z}{\lambda} \in \partial F(\hat{x}).$$

$$\hat{A} = (1-\alpha)A$$

$$\hat{\nu} = \nu - \frac{\lambda}{\alpha}\xi$$

$$\hat{\delta} = (1-\alpha)\delta + \varepsilon^2/2\lambda$$

we have $\hat{\delta} + \hat{\varphi}^* \geq F(\hat{x}) + \dfrac{1}{2\lambda}\|t - \hat{x}\|^2$.

Proof. Let Φ_λ denote here the function $F + 1/(2\lambda)\|\cdot - t\|^2$. Applying Lemma 2 with $\varepsilon = 0$, $y = t$ and $\xi = (t-z)/\lambda$ we get

$$(1-\alpha)\delta + \hat{\varphi}^* \geq F(z) + \frac{1}{2\lambda}\|t - z\|^2 = \Phi_\lambda(z) \geq \Phi_\lambda^* \geq \Phi_\lambda(\hat{x}) - \frac{\varepsilon^2}{2\lambda}$$

the last inequality following from the definition of $\hat{x}$ itself. □

Remark 2. We are now going to discuss a fundamental issue. It is quite natural to think to employ the previous theorem to generate a sequence $(x_k)_{k\in\mathbb{N}}$ satisfying $\delta_k + \varphi_k \geq F(x_k)$ and then conclude relying on Theorem 1 — and actually this is what Güler did in his paper [14] in Theorem 3.1. But, unfortunately this is not a correct reasoning. Indeed, if we look carefully at the statement of the previous theorem, we recognize that the full iteration process should rely on updating the four quantities x, ν, A, α. The point is that the second one is updated by the rule $\hat{\nu} = \nu - (t-z)/\alpha$ which does depend on the exact proximal point we are assuming not to know. Thus, the inductive construction of the sequences cannot be set up by no means. This is exactly where Güler's proof fails, leading to a wrong result. Rather, we would like to update the variable ν using the rule $\hat{\nu} = \nu - (t-\hat{x})/\alpha$ (because $\hat{x}$ is what we actually know) and indeed the author suggests to use it in the statement of Theorem 3.1, even though the proof addresses the other rule, missing the link between them.

To overcome the difficulty discussed in the previous remark, it is necessary to study the case in which ν is known only up to a certain precision. We are therefore going to describe what the effects of a perturbation of ν on the previous results are. Using the notations introduced in Theorem 2, define $u \in \mathcal{H}$ by setting

$$u = \nu + \Delta, \text{ with } \|\Delta\| \leq \eta.$$

If we set

$$\begin{aligned} y &= (1-\alpha)x + \alpha u \\ \hat{x} &\approx_1 \operatorname{prox}_{\lambda F}(y) \quad \text{with } \varepsilon\text{-precision} \end{aligned}$$

it follows that $y = (1-\alpha)x + \alpha\nu + \alpha\Delta = t + \alpha\Delta$. Moreover, from Lemma 1 there exist $\varepsilon_1, \varepsilon_2 \geq 0$ such that $0 \leq \varepsilon_1^2 + \varepsilon_2^2 \leq \varepsilon^2$ and

$$\frac{y - \hat{x} + e}{\lambda} \in \partial_{\frac{\varepsilon_1^2}{2\lambda}} F(\hat{x}), \text{ with } \|e\| \leq \varepsilon_2.$$

Recalling the definition of y, we can rewrite the last equation as

$$\frac{t - \hat{x} + e + \alpha\Delta}{\lambda} \in \partial_{\frac{\varepsilon_1^2}{2\lambda}} F(\hat{x}), \text{ with } \|e + \alpha\Delta\| \leq \varepsilon_2 + \alpha\eta, \tag{13}$$

implying that

$$\hat{x} \approx_1 \operatorname{prox}_{\lambda F}(t) \quad \text{with } (\varepsilon + \alpha\eta)\text{-precision.} \tag{14}$$

This allows for stating a new version of Theorem 2 that can be finally used to build an iterative algorithm.

Theorem 3. *Fix $\lambda > 0$, $\varepsilon > 0$, $x, u \in \mathcal{H}$, $A > 0$, and $\delta, \eta \geq 0$. Suppose that there exist $\varphi^* \in \mathbb{R}$ and $\nu \in \mathcal{H}$ with $\|\nu - u\| \leq \eta$ such that if we set $\varphi = \varphi^* + A/2\|\cdot - \nu\|^2$, we get $\delta + \varphi^* \geq F(x)$. Let $\alpha \in [0, 1[$ be such that $\alpha^2 = (1-\alpha)A\lambda$ and define*

$$\begin{aligned} y &= (1-\alpha)x + \alpha u \\ \hat{x} &\approx_1 \operatorname{prox}_{\lambda F}(y) \text{ with } \varepsilon\text{-precision} \\ \hat{u} &= u - (1/\alpha)(y - \hat{x}) \\ \hat{A} &= (1-\alpha)A \\ \hat{\eta} &= \eta + \varepsilon/\alpha \\ \hat{\delta} &= (1-\alpha)\delta + \frac{(\alpha\hat{\eta})^2}{2\lambda} \end{aligned}$$

Then there exists $\hat{\nu} \in \mathcal{H}$, $\|\hat{\nu} - \hat{u}\| \leq \hat{\eta}$ and $\hat{\varphi}^* \in \mathbb{R}$ such that, if $\hat{\varphi} = \hat{\varphi}^* + \hat{A}/2\|\cdot - \hat{\nu}\|^2$, it holds

$$\begin{aligned} \hat{\varphi} - F &\leq (1-\alpha)(\varphi - F) \\ \hat{\delta} + \hat{\varphi}^* &\geq F(\hat{x}). \end{aligned}$$

More precisely, the function $\hat{\varphi}$ is obtained by $\hat{\varphi} = U(t, 0, \xi, \alpha)\varphi$ with $t = (1-\alpha)x + \alpha\nu$ and $\xi = (t - \operatorname{prox}_{\lambda F}(t))/\lambda$.

Proof. Recalling equation (14) it is possible to apply Theorem 2 with $\varepsilon = \varepsilon + \alpha\eta$. In particular, if $t, \xi, \hat{\nu}, \hat{\varphi}$ are as in Theorem 2, we have

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

Now, being $\hat{x}$ an approximation of type 1 of $z = \text{prox}_{\lambda F}(t)$ with $(\varepsilon + \alpha\eta)$ -precision, we have $\|\hat{x} - z\| \leq \varepsilon + \alpha\eta$ and moreover

$$\begin{aligned}\hat{u} &= u - \frac{1}{\alpha}(y - \hat{x}) = \nu + \Delta - \frac{1}{\alpha}(y - z) - \frac{1}{\alpha}(z - \hat{x}) \\ &= \nu + \Delta - \frac{1}{\alpha}(t + \alpha\Delta - z) - \frac{1}{\alpha}(z - \hat{x}) \\ &= \nu - \frac{1}{\alpha}(t - z) - \frac{1}{\alpha}(z - \hat{x}) \\ &= \hat{\nu} - \frac{1}{\alpha}(z - \hat{x})\end{aligned}$$

Thus $\|\hat{u} - \hat{\nu}\| \leq \alpha^{-1}\|z - \hat{x}\| \leq \eta + \varepsilon/\alpha$. □

In contrast to what happens in Theorem 2, all the updating rules in Theorem 3 no longer depend on unknown quantities, and therefore allow for the definition of an iterative process. More precisely, given two sequences of positive numbers $(\lambda_k)_{k\in\mathbb{N}}, (\varepsilon_k)_{k\in\mathbb{N}}$, $A > 0$ and $x_0 \in \text{dom}(F), u_0 = x_0, \eta_0 = 0$ we can iteratively build sequences as follows

$$\left\lfloor \begin{array}{l} \alpha_k = \dfrac{\sqrt{(A_k\lambda_k)^2 + 4A_k\lambda_k} - A_k\lambda_k}{2} \\ y_k = (1 - \alpha_k)x_k + \alpha_k u_k \\ x_{k+1} \approx_1 \text{prox}_{\lambda_k F}(y_k) \text{ with } \varepsilon_k\text{-precision} \\ A_{k+1} = (1 - \alpha_k)A_k \\ u_{k+1} = u_k - \dfrac{1}{\alpha_k}(y_k - x_{k+1}) \\ \eta_{k+1} = \eta_k + \dfrac{\varepsilon_k}{\alpha_k} \\ \delta_{k+1} = (1 - \alpha_k)\delta_k + \dfrac{(\alpha_k\eta_{k+1})^2}{2\lambda_k}. \end{array} \right. \tag{IAPPA1}$$

We note that the α_k in algorithm (IAPPA1) is obtained as the only positive solution of the equation $\alpha_k^2 = (1 - \alpha_k)A_k\lambda_k$.

Remark 3. Theorem 3 ensures the existence of a parallel sequence of points $(\nu_k)_{k\in\mathbb{N}}$, $\nu_k \in \mathcal{H}$ such that $\|u_k - \nu_k\| \leq \eta_k$ and of a corresponding estimate sequence $(\varphi_k)_{k\in\mathbb{N}}$, $\varphi_k = \varphi_k^* + A_k\|\cdot - \nu_k\|^2$ satisfying (7) and such that $F(x_k) \leq \varphi_k^* + \delta_k$. The same theorem also shows that the sequence of the φ_k's can be defined recursively by means of $\varphi_{k+1} = U(t_k, 0, \xi_{k+1}, \alpha_k)(\varphi_k)$ with $t_k = (1 - \alpha_k)x_k + \alpha_k\nu_k$, $\xi_{k+1} = (t_k - \text{prox}_{\lambda_k F}(t_k))/\lambda_k$. Of course all the sequences of parameters ν_k and ξ_k are unknown and consequently the estimate sequence itself is unknown. However, as the algorithm above shows, the explicit expression of the φ_k's is not essential in constructing the full iterative process. In conclusion, the estimate sequence $(\varphi_k)_{k\in\mathbb{N}}$, even though unknown, is indeed underlying the iterative process and make things work.

While we are not interested in the expression of φ_k, we need an explicit formula for the asymptotic behavior of β_k. From Lemma 3 in the previous section with $a = b = 1$,

we know the asymptotic behavior of the sequence of the β_k's and sufficient conditions to make it convergent and even an $O(1/k^2)$.

The aim of the rest of this section is to explicitly compute the cumulative errors δ_k and η_k, for which we have only a recursive definition, and to determine a rate of convergence to zero. From the definition $\eta_k = \eta_{k-1} + \varepsilon_{k-1}/\alpha_{k-1}$, we get

$$\eta_k = \sum_{i=0}^{k-1} \frac{\varepsilon_i}{\alpha_i}$$

taking into account that $\eta_0 = 0$. As concerns δ_k, mimicking the reasoning followed in Lemma 3.3 in [14], being $\delta_0/\beta_0 = 0$, one can get

$$\frac{\delta_k}{\beta_k} = \frac{1}{2} \sum_{i=0}^{k-1} \frac{(\alpha_i \eta_{i+1})^2}{\lambda_i \beta_{i+1}}.$$

Exploiting the condition $\alpha_i^2/(A_{i+1}\lambda_i) = 1$ and formula $A_{i+1} = \beta_{i+1}A$, we get $\alpha_i^2/(\lambda_i\beta_{i+1}) = A$ and hence

$$\delta_k = \frac{A\beta_k}{2} \sum_{i=0}^{k-1} \eta_{i+1}^2, \qquad \eta_{i+1} = \sum_{j=0}^{i} \frac{\varepsilon_j}{\alpha_j} \tag{15}$$

From the latter formula, it is possible to derive a rate of convergence for algorithm (IAPPA1).

Theorem 4. *Consider the proximal point algorithm described in (IAPPA1) for a sequence $\lambda_k > 0$ satisfying*

$$\lambda_j \leq M\lambda_i \text{ whenever } j \leq i \text{ for some } M > 0. \tag{16}$$

Then, if $\varepsilon_k = O(1/k^q)$ with $q > 3/2$, the sequence $(x_k)_{k\in\mathbb{N}}$ is minimizing for F and if in addition F has a minimizer the following rate of convergence holds

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

Proof. Convergence, as well as relative rate declared in the statement, will be deduced from corresponding convergence properties and rate of the sequences δ_k and β_k, using the key results about estimate sequences stated in Theorem 1.

Concerning the sequence $(1/\alpha_j)_{j\in\mathbb{N}}$, from the equation $\alpha_j^2 = \beta_{j+1}A\lambda_j$ and Lemma 3 (with $a = b = 1$), we have

$$\frac{\sqrt{A\lambda_j}}{1 + \sqrt{A}\sum_{k=0}^{j}\sqrt{\lambda_k}} \leq \alpha_j \leq \frac{\sqrt{A\lambda_j}}{1 + \sqrt{A}/2\sum_{k=0}^{j}\sqrt{\lambda_k}}$$

Taking into account condition (16) we obtain

$$\frac{1}{\alpha_j} \leq \frac{1}{\sqrt{A}} + \sum_{k=0}^{j} \sqrt{\lambda_k/\lambda_j} \leq \frac{1}{\sqrt{A}} + \sqrt{M}(j+1),$$

and hence $1/\alpha_j = O(j)$. Thanks to the assumption on ε_j we thus have

$$\frac{\varepsilon_j}{\alpha_j} \leq \frac{c}{(j+1)^p} \qquad \text{with } p = q - 1 > 1/2, \text{ for some } c > 0.$$

This implies that

$$\begin{aligned}
\eta_{i+1} = \sum_{j=0}^{i} \frac{\varepsilon_j}{\alpha_j} &\leq c \sum_{j=1}^{i+1} \frac{1}{j^p} = c\left(1 + \sum_{j=2}^{i+1} \frac{1}{j^p}\right) \\
&\leq c\left(1 + \int_1^{i+1} t^{-p} \mathrm{d}t\right) \\
&= \begin{cases} \dfrac{c}{(1-p)}\left((i+1)^{1-p} - p\right) & \text{if } 1/2 < p < 1 \\ c\,(1 + \log(i+1)) & \text{if } p = 1 \\ \dfrac{c}{(p-1)}\left(p - \dfrac{1}{(i+1)^{p-1}}\right) & \text{if } p > 1 \end{cases}
\end{aligned}$$

Let us suppose $p \neq 1$. Taking into account that the function $t \mapsto (t^{1-p} - p)^2$ is increasing for $t \geq 1$, we get

$$\begin{aligned}
\sum_{i=0}^{k-1} \eta_{i+1}^2 &\leq \frac{c^2}{(1-p)^2} \sum_{i=1}^{k} (i^{(1-p)} - p)^2 \\
&\leq \frac{c^2}{(1-p)^2} \int_1^{k+1} (t^{(1-p)} - p)^2 \mathrm{d}t \\
&= \frac{c^2}{(1-p)^2} \left(\frac{(k+1)^{3-2p}}{3-2p} - 2p \frac{(k+1)^{2-p}}{2-p} + p^2 k - \frac{4p^2 - 7p + 2}{(3-2p)(2-p)} \right) \qquad (17)
\end{aligned}$$

On the other hand, if $p = 1$ we have

$$\begin{aligned}
\sum_{i=0}^{k-1} \eta_{i+1}^2 &\leq c^2 \sum_{i=1}^{k} (1 + \log i)^2 \\
&\leq c^2 \Big(1 + \int_2^{k+1} (1 + \log t)^2 \mathrm{d}t\Big) \\
&= c^2 \big(k + (k+1)\log^2(k+1) - 2\log^2 2\big) \qquad (18)
\end{aligned}$$

Putting together (17) and (18) we get

$$\sum_{i=0}^{k-1} \eta_{i+1}^2 = \begin{cases} O(k^{3-2p}) & \text{if } 1/2 < p < 1 \\ O(k \log^2 k) & \text{if } p = 1 \\ O(k) & \text{if } p > 1. \end{cases}$$

Combining this result with the convergence rate $O(1/k^2)$ of β_k mentioned in Lemma 3, and recalling that $p = q - 1$, we finally get

$$\delta_k = \begin{cases} O(k^{3-2q}) & \text{if } 3/2 < q < 2 \\ O(\log^2 k / k) & \text{if } q = 1 \\ O(1/k) & \text{if } q > 2. \end{cases}$$

□

4.2 The algorithm with errors of type 2

The aim of this section is to prove that considering a different type of approximations, namely approximations of type 2, the rate of convergence of the exact version of the accelerated proximal point algorithm can be recovered. While the structure of the section is essentially the same of the previous one, what makes the difference is the recursive update of the quantity δ_k.

Theorem 5. *Fix $\lambda > 0$ and $\varepsilon \geq 0$. Let $x, \nu \in X$, $A > 0$ and $\varphi = \varphi^* + A/2\|\cdot - \nu\|^2$ be such that $F(x) \leq \varphi^* + \delta$ for some $\delta \geq 0$. If $\alpha^2/((1-\alpha)A\lambda) \leq 2$, choosing*

$$
\begin{aligned}
&y = (1-\alpha)x + \alpha\nu \\
&\hat{x} \approx_2 \operatorname{prox}_{\lambda F}(y), \quad \textit{with } \varepsilon\textit{-precision} \\
&\hat{\varphi} = U(\hat{x}, \varepsilon, \xi, \alpha)\varphi, \quad \textit{with } \xi = \frac{y - \hat{x}}{\lambda} \in \partial_{\frac{\varepsilon^2}{2\lambda}} F(\hat{x}) \\
&\hat{A} = (1-\alpha)A \\
&\hat{\nu} = \nu - \frac{\alpha}{\hat{A}}\xi \\
&\hat{\delta} = (1-\alpha)\delta + \frac{\varepsilon^2}{2\lambda}
\end{aligned}
$$

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

Proof. By Lemma 2, if we take $\hat{x} \in \mathcal{H}$ such that $(y - \hat{x})/\lambda \in \partial_{\frac{\varepsilon^2}{2\lambda}} F(\hat{x})$ and define $\xi = (y - \hat{x})/\lambda$, we get $\xi \in \partial_{\frac{\varepsilon^2}{2\lambda}} F(\hat{x})$ and $y - (\lambda\xi + \hat{x}) = 0$, hence

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

which gives the required result, after substituting ξ with $(y - \hat{x})/\lambda$. □

Theorem 5 allows for defining an iterative procedure as follows. For fixed sequences $(\lambda_k)_{k\in\mathbb{N}}$, $(\varepsilon_k)_{k\in\mathbb{N}}$ with $\lambda_k > 0$, $\varepsilon_k \geq 0$ and $A > 0$, $a \in (0, 2]$, we set $A_0 = A$, $\delta_0 = 0$ and

$x_0 = \nu_0 \in \mathrm{dom}(F)$ and for each $k \in \mathbb{N}$ we define

$$\left[\begin{array}{l} \alpha_k \in [0,1) \text{ such that } a \leq \dfrac{\alpha_k^2}{(1-\alpha_k)A_k\lambda_k} \leq 2 \\ y_k = (1-\alpha_k)x_k + \alpha_k\nu_k \\ x_{k+1} \approx_2 \mathrm{prox}_{\lambda_k F}(y_k) \text{ with } \varepsilon_k\text{-precision} \\ A_{k+1} = (1-\alpha_k)A_k \\ \nu_{k+1} = \nu_k - \dfrac{\alpha_k}{(1-\alpha_k)A_k\lambda_k}(y_k - x_{k+1}) \\ \delta_{k+1} = (1-\alpha_k)\delta_k + \dfrac{\varepsilon_k^2}{2\lambda_k} \end{array}\right. \tag{IAPPA2}$$

Then by setting $\xi_{k+1} = (y_k - x_{k+1})/\lambda_k$, we get two sequences $(x_k)_{k\in\mathbb{N}}$ e $(\xi_k)_{k\in\mathbb{N}}$ such that $\xi_{k+1} \in \partial_{\varepsilon_k^2/(2\lambda_k)}F(x_{k+1})$. With this hypothesis, we have already noted that the construction of a sequence $(\varphi_k)_{k\in\mathbb{N}}$ as in (10) gives an estimate sequence provided that $\beta_k = \prod_{i=0}^{k-1}(1-\alpha_i) \to 0$, and the last condition is true due to Lemma 3 with $b = 2$ if $\lambda_k \geq \lambda > 0$ — actually in this case $(\beta_k)_{k\in\mathbb{N}} = O(1/k^2)$.

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 5, also $\delta_k + \varphi_k^* \geq F(x_k)$. If $\delta_k \to 0$, the sequence $(x_k)_{k\in\mathbb{N}}$ is a minimizing sequence for F. The previous bounds obtained on β_k allow us to impose explicit conditions on the error sequence ε_k in order to get a convergent proximal point algorithm.

Remark 4. To be more precise, Theorem 5 tell us more, that is

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

From this inequality, as done in the proof of Theorem 1, it follows that

$$\frac{c_{k-1}}{2\lambda_{k-1}}\|y_{k-1} - x_k\|^2 + F(x_k) \leq \beta_k(\varphi_0(x) - F(x)) + F(x) + \delta_k$$

for any $x \in \mathrm{dom}F$. Again, if x^* is a minimizer of F

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

The last result first shows that, if $c_k \geq 2 - a > 0$ and λ_k is kept bounded from above, then $\|y_{k-1} - x_k\| \to 0$. Secondly, it suggests that choosing α_k such that c_k is as great as possible could improve the practical speed of convergence.

Concerning the structure of the error term δ_k, it is easy to prove (see again Lemma 3.3 in [14]) that the solution of the last difference equation in (IAPPA2) is given by

$$\delta_k = \frac{\beta_k}{2}\sum_{i=0}^{k-1}\frac{\varepsilon_i^2}{\lambda_i\beta_{i+1}}. \tag{19}$$

Combining equation (19) with the bounds on β_k in Lemma 3, we get

$$\delta_k \leq \frac{\beta_k}{2} \sum_{i=0}^{k-1} \varepsilon_i^2 \frac{(1+\sqrt{2A}\sum_{j=0}^{i}\sqrt{\lambda_j})^2}{\lambda_i}. \tag{20}$$

This implies that if the series $\sum_{i=0}^{k-1} \varepsilon_i^2 (1+\sqrt{2A}\sum_{j=0}^{i}\sqrt{\lambda_j})^2/\lambda_i$ is convergent, then δ_k converges to zero at the same rate of β_k. This holds if we impose the error ε_k to satisfy at each step

$$\varepsilon_k \leq \frac{c\sqrt{\lambda_k}}{(k+1)^p(1+\sqrt{2A}\sum_{j=0}^{k}\sqrt{\lambda_j})}, \tag{21}$$

for some positive constant c and for $p > 1/2$. If we do not ask for the same convergence rate of β_k, but only for convergence, obviously it is enough to impose less stringent conditions on ε_k. The theorem below, specializing the results outlined above to the case when λ_k satisfies condition (22), is the analogous of Theorem 3.3 in [14] and gives the convergence rate estimates for the proximal point algorithm where errors of type 2 in the computation of the proximity operator are admitted.

Theorem 6. *Consider the proximal point algorithm described in (IAPPA2) for a sequence λ_k satisfying*

$$\lambda_j \leq M\lambda_i \qquad \text{whenever } j \leq i \text{ for some } M > 0. \tag{22}$$

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 in addition F has a minimizer the following rate of convergence holds

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

Proof. If (22) is true, then Lemma 3 implies $\beta_k = O(1/k^2)$. Indeed, by (22) it follows $k\sqrt{\lambda_0/M} \leq \sum_{i=0}^{k-1}\sqrt{\lambda_i}$, and thus $\beta_k \leq 1/(1+(\sqrt{aA}/2)k\sqrt{\lambda_0}/M)^2$.

On the other hand, again from Lemma 3 with $b = 2$, we get

$$\begin{aligned} \frac{1}{\lambda_i\beta_{i+1}} &\leq \left(\frac{1}{\sqrt{\lambda_i}} + 2\sqrt{A}\sum_{j=0}^{i}\sqrt{\lambda_j/\lambda_i}\right)^2 \\ &\leq \left(\sqrt{M/\lambda_0} + 2(i+1)\sqrt{AM}\right)^2 \\ &\leq c(i+1)^2 \end{aligned}$$

for a properly chosen constant $c > 0$, thus the error δ_k can be majorized as follows

$$\begin{aligned} \delta_k &= \frac{\beta_k}{2}\sum_{i=0}^{k-1}\frac{\varepsilon_i^2}{\lambda_i\beta_{i+1}}. \\ &\leq \frac{c}{2(k+1)^2}\sum_{i=0}^{k-1}\varepsilon_i^2(i+1)^2. \end{aligned}$$

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

5 Equivalent forms of the algorithms

We finish by rewriting the two algorithms (IAPPA1/2) given in the previous section in equivalent but simpler forms. We discuss the second algorithm because the scheme is more general, providing for the first algorithm just the final result.

We prove that the sequence ν_k in the recursive process (IAPPA2) can be replaced with y_k, achieving a first alternative form of the algorithm. To this purpose, let us define

$$a_k = \frac{\alpha_k^2}{(1-\alpha_k)A_k\lambda_k} \in [a, 2] \tag{23}$$

The updating rule for ν can be written as

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

Now from $y_k = (1-\alpha_k)x_k + \alpha_k\nu_k$, we get $\nu_k = \alpha_k^{-1}(y_k - (1-\alpha_k)x_k)$ and we can substitute it into the formula for ν_{k+1} obtaining $\nu_{k+1} = \nu_k - a_k\alpha_k^{-1}(y_k - x_{k+1}) = \alpha_k^{-1}((1-a_k)y_k + a_k x_{k+1} - (1-\alpha_k)x_k)$. If we substitute ν_{k+1} back again into the formula for y_{k+1} we finally obtain

$$\begin{aligned}
y_{k+1} &= (1-\alpha_{k+1})x_{k+1} + \alpha_{k+1}\nu_{k+1} \\
&= (1-\alpha_{k+1})x_{k+1} + \frac{\alpha_{k+1}}{\alpha_k}((1-a_k)y_k + a_k x_{k+1} - (1-\alpha_k)x_k) \\
&= x_{k+1} + \alpha_{k+1}\Big(\frac{a_k}{\alpha_k} - 1\Big)x_{k+1} - \alpha_{k+1}\Big(\frac{1}{\alpha_k} - 1\Big)x_k + (1-a_k)y_k \\
&= x_{k+1} + \alpha_{k+1}\Big(\frac{1}{\alpha_k} - 1\Big)(x_{k+1} - x_k) + (1-a_k)\frac{\alpha_{k+1}}{\alpha_k}(y_k - x_{k+1})
\end{aligned}$$

Thus, the algorithm shows the following final form

$$\left[\begin{array}{l}
\alpha_k = \dfrac{\sqrt{(A_k a_k \lambda_k)^2 + 4A_k a_k \lambda_k} - A_k a_k \lambda_k}{2} \\
x_{k+1} \approx_2 \operatorname{prox}_{\lambda_k F}(y_k) \\
y_{k+1} = x_{k+1} + \alpha_{k+1}\Big(\dfrac{1}{\alpha_k} - 1\Big)(x_{k+1} - x_k) + (1-a_k)\dfrac{\alpha_{k+1}}{\alpha_k}(y_k - x_{k+1}) \\
A_{k+1} = (1-\alpha_k)A_k
\end{array}\right.$$

which depends on an extra arbitrary numerical sequence $(a_k)_{k\in\mathbb{N}}$ with $0 < a \leq a_k \leq 2$.

We can give the algorithm still another form, even simpler, replacing the two numerical sequences $(\alpha_k)_{k\in\mathbb{N}}$ and $(A_k)_{k\in\mathbb{N}}$ with a new one. Just by defining $t_k = 1/\alpha_k$ the update of y_k becomes

$$y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k) + (1 - a_k)\frac{t_k}{t_{k+1}}(y_k - x_{k+1})$$

and t_{k+1} can be computed recursively. Indeed, being $\alpha_k^2 = a_k A_{k+1}\lambda_k$ and taking into account (23) for $k+1$, we have

$$\begin{aligned}\alpha_{k+1}^2 &= a_{k+1}(1 - \alpha_{k+1})A_{k+1}\lambda_{k+1}\\ &= (1 - \alpha_{k+1})\alpha_k^2\frac{a_{k+1}}{a_k}\frac{\lambda_{k+1}}{\lambda_k}.\end{aligned}$$

Making the substitution $t_k = 1/\alpha_k$ in the last equation, we get the equation

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

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

$$\left[\begin{array}{l} t_{k+1} = \dfrac{1 + \sqrt{1 + 4(a_k\lambda_k)t_k^2/(a_{k+1}\lambda_{k+1})}}{2}\\ x_{k+1} \approx_2 \operatorname{prox}_{\lambda_k F}(y_k)\\ y_{k+1} = x_{k+1} + \dfrac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k) + (1 - a_k)\dfrac{t_k}{t_{k+1}}(y_k - x_{k+1})\end{array}\right.$$

This last implementation of the algorithm shows that we can recover the second algorithm given by Güler in the appendix of [14] for the case λ_k be constant, by choosing $a_k = 2$. As a final remark about the choice of the parameters a_k, one can recognize that they are linked with the c_k's introduced in Remark 4, indeed $c_k = 2 - a_k$. Thus, as already noted, the empirical speed of convergence could be improved by making the a_k's smaller then one and close to zero.

The same rearrangement can be done for the first algorithm (IAPPA1). In this case $a_k = 1$ for every $k \in \mathbb{N}$ and the convergent scheme is

$$\left[\begin{array}{l} t_{k+1} = \dfrac{1 + \sqrt{1 + 4\lambda_k t_k^2/\lambda_{k+1}}}{2}\\ x_{k+1} \approx_1 \operatorname{prox}_{\lambda_k F}(y_k)\\ y_{k+1} = x_{k+1} + \dfrac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k)\end{array}\right.$$

This form, if λ_k is constant, resembles popular accelerating schemes designed in general for forward-backward splitting methods, like FISTA [3, 30]. Those algorithms aim at minimizing composite functions of type $F = f + g$, with f convex smooth and g convex possibly nonsmooth. In case $f = 0$ we recover exactly the FISTA procedure.

Acknowledgments. The authors wish to thank Sofia Mosci and Curzio Basso for several useful comments and their willingness to carefully read the manuscript.

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. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring. IEEE Trans. Image Proc., 18(11):2419–2434, 2009.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sciences, 2(1):183–202, 2009.

[4] S. Becker, J. Bobin, and E. Candes. NESTA: A fast and accurate first-order method for sparse recovery. Technical report, California Institute of Technology, 2009.

[5] J. Bect, L. Blanc-Féraud, G. Aubert, and A. Chambolle. A ℓ^1-unified variational framework for image restoration. In T. Pajdla and J. Matas, editors, ECCV 2004, volume 3024 of Lecture Notes in Computer Science, pages 1–13. Springer, Berlin, 2004.

[6] J. R. Birge, L. Qi, and Z. Wei. Convergence analysis of some methods for minimizing a nonsmooth convex function. J. Optim. Theory Appl., 97(2):357–383, 1998.

[7] R. S. Burachik, A. N. Iusem, and B. F. Svaiter. Enlargement of monotone operators with applications to variational inequalities. Set-Valued Anal., 5(2):159–180, 1997.

[8] R. S. Burachik and B. F. Svaiter. A relative error tolerance for a family of generalized proximal point methods. Math. Oper. Res., 26(4):816–831, 2001.

[9] A. Chambolle. An algorithm for total variation minimization and applications. J. Math. Imaging Vis., 20(1-2):89–97, 2004.

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

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

[12] M. Fornasier, editor. Theoretical Foundations and Numerical Methods for Sparse Recovery, volume 9 of Radon Series on Computational and Applied Mathematics. De Gruyter, 2010.

[13] O. Güler. On the convergence of the proximal point algorithm for convex minimization. SIAM J. Control Optim., 29(2):403–419, 1991.

[14] O. Güler. New proximal point algorithms for convex minimization. SIAM J. on Optimization, 2(4):649–664, 1992.

[15] J.-B. Hiriart-Urruty and C. Lemaréchal. Convex analysis and minimization algorithms. II, volume 306 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer-Verlag, Berlin, 1993.

[16] L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning, pages 433–440, New York, NY, USA, 2009. ACM.

[17] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. Rev. Française Informat. Recherche Opérationnelle, 4(Ser. R-3):154–158, 1970.

[18] R.D.C. Monteiro and B.F. Svaiter. Convergence rate of inexact proximal point methods with relative error criteria for onvex optimization, 2010.

[19] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. Bull. Soc. Math. France, 93:273–299, 1965.

[20] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa. Solving structured sparsity regularization with proximal methods. In J. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors, Machine Learning and Knowledge Discovery in Databases, volume 6322 of Lecture Notes in Computer Science, pages 418–433. Springer, 2010.

[21] 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.

[22] Y. Nesterov. Introductory lectures on convex optimization. A basic course, volume 87 of Applied Optimization. Kluwer Academic Publishers, Boston, MA, 2004.

[23] Y. E. Nesterov. An approach to constructing optimal methods for minimization of smooth convex functions. Èkonom. i Mat. Metody, 24(3):509–517, 1988.

[24] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res., 1(2):97–116, 1976.

[25] R. T. Rockafellar. Monotone operators and the proximal point algorithm. SIAM J. Control Optimization, 14(5):877–898, 1976.

[26] 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), volume 36 of Appl. Optim., pages 415–427. Kluwer Acad. Publ., Dordrecht, 2000.

[27] M. V. Solodov and B. F. Svaiter. Error bounds for proximal point subproblems and associated inexact proximal point algorithms. Math. Program., 88(2, Ser. B):371–389, 2000. Error bounds in mathematical programming (Kowloon, 1998).

[28] M. V. Solodov and B. F. Svaiter. An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions. Math. Oper. Res., 25(2):214–230, 2000.

[29] M. V. Solodov and B. F. Svaiter. A unified framework for some inexact proximal point algorithms. Numer. Funct. Anal. Optim., 22(7-8):1013–1035, 2001.

[30] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report, 2008. Submitted to SIAM J. Opt.

[31] C. Zălinescu. Convex analysis in general vector spaces. World Scientific Publishing Co. Inc., River Edge, NJ, 2002.

[32] A. Zaslavski. Convergence of a proximal point method in the presence of computational errors in Hilbert spaces. SIAM J. Optim., 20(5):2413–2421, 2010.