



HAL
open science

On the Proximal Gradient Algorithm with Alternated Inertia

Franck Iutzeler, Jérôme Malick

► **To cite this version:**

Franck Iutzeler, Jérôme Malick. On the Proximal Gradient Algorithm with Alternated Inertia. *Journal of Optimization Theory and Applications*, 2018, 176 (3), pp.688-710. 10.1007/s10957-018-1226-4 . hal-01685859

HAL Id: hal-01685859

<https://hal.science/hal-01685859>

Submitted on 16 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Proximal Gradient Algorithm with Alternated Inertia

Franck Iutzeler · Jérôme Malick

Received: date / Accepted: date

Abstract In this paper, we investigate the attractive properties of the proximal gradient algorithm with inertia. Notably, we show that using *alternated inertia* yields monotonically decreasing functional values, which contrasts with usual accelerated proximal gradient methods. We also provide convergence rates for the algorithm with alternated inertia based on local geometric properties of the objective function. The results are put into perspective by discussions on several extensions and illustrations on common regularized problems.

Keywords Proximal gradient algorithm · Accelerated methods · Kurdyka-Łojasiewicz inequality

Mathematics Subject Classification (2000) 65K10 · 90C30

1 Introduction

In this paper, we consider the composite convex optimization problem

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + g(x) \quad (1)$$

where the functions f and g are convex, and f is furthermore smooth. This problem can be solved by the proximal gradient algorithm, a special case of Forward-Backward splitting, the iterations of which read

$$x_{k+1} = \underset{\gamma_k g}{\mathbf{prox}}(x_k - \gamma_k \nabla f(x_k)). \quad (2)$$

Recall that the proximal operator $\mathbf{prox}_h : \mathbb{R}^n \rightarrow \mathbb{R}^n$ of a proper lower semi-continuous convex function $h : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined as

$$\mathbf{prox}_h(x) := \underset{w \in \mathbb{R}^n}{\operatorname{argmin}} \left(h(w) + \frac{1}{2} \|w - x\|^2 \right) \quad \text{for all } x \in \mathbb{R}^n.$$

The computation of this operator is simple (closed-form or equivalent to a smaller problem) for several interesting functions such as the ℓ_1 norm, the group norm $\ell_{1,2}$, or the nuclear norm, which are commonly used as regularizers in machine learning and image processing optimization problems; see, e.g., [1–3].

To improve the convergence of the proximal gradient algorithm, *inertial versions* have been extremely popular, both practically and in terms of theoretical rate [4–8]. Adding inertia consists in constructing the next iterate x_{k+1} by combining the outputs y_{k+1} and y_k of the last two proximal steps. Specifically, given

Franck Iutzeler, Corresponding author
Univ. Grenoble Alpes
franck.iutzeler@univ-grenoble-alpes.fr

Jérôme Malick
CNRS, LJK
jerome.malick@univ-grenoble-alpes.fr

a sequence of real non-negative numbers (α_k) , an iteration of the inertial proximal gradient algorithm can be written as

$$y_{k+1} = \underset{\gamma_k g}{\mathbf{prox}}(x_k - \gamma_k \nabla f(x_k)) , \quad x_{k+1} = y_{k+1} + \alpha_{k+1}(y_{k+1} - y_k). \quad (3)$$

The design of the inertial sequence (α_k) affects greatly the performance of the resulting algorithm, and different options are investigated in the literature. Popular choices include Nesterov's optimal sequence [9,10] which leads to the FISTA algorithm [11] or a variant used for proving the iterates convergence [6,8]. These sequences have the form

$$\alpha_{k+1} = \frac{t_k - 1}{t_{k+1}} \quad \text{with } t_0 = 0 \quad \text{and} \quad \begin{aligned} t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\ \text{or } t_{k+1} &= \frac{k + a}{a} \quad \text{with } a > 2. \end{aligned} \quad (4)$$

Both choices lead to increasing sequences going to 1 at rate $1/k$ and are proven to accelerate the worst-case convergence rate of the algorithm from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$. Another variant with $t_{k+1} = ((k+a)/a)^d$ where $d \in]0, 1]$ and $a > \max\{1, (2d)^{\frac{1}{d}}\}$ was shown to be efficient, when the proximal operator is computed approximately [7]. Finally, if f is in addition μ -strongly convex, then the optimal linear rate is attained for fixed $\alpha = (1 - \sqrt{\mu/L})/(1 + \sqrt{\mu/L})$ [12].

Quest for Monotonicity. An attractive feature of the vanilla proximal gradient algorithm is its monotonicity; more precisely, iterates generated by (2) satisfy both

- monotonicity of the iterates in the sense of Fejér: $\|x_{k+1} - x\| \leq \|x_k - x\|$ for any optimal x
- monotonicity of the functional values: $F(x_{k+1}) \leq F(x_k)$.

These properties are well-known and obtained directly from contraction and descent results recalled in Section 2. On the contrary, acceleration by inertia breaks down monotonicity as the iterates generated by inertial variants can circle or oscillate around the set of minimizers [5,13,14]. These kinds of behaviors make accelerated methods sometimes slower than their unaccelerated counterparts (see for instance the non-negative least squares problem in [15]).

Monotonicity (and in particular functional monotonicity) is a highly desirable feature in optimization, in both theory and practice; see for instance the quests for descent in [16–18]. For composite optimization, several algorithms based on descent tests have been proposed to fix the non-monotonicity of accelerated proximal gradient methods, notably: MTwist [17], MFISTA [14], or Monotonous APG [19]. However, all these methods rely on additional function evaluations, or even additional computations, compared to the initial proximal gradient algorithm.

Another strategy is to use inertia every other step of the method: the corresponding algorithm, called proximal gradient with *alternated* inertia, can be formulated as (3) with $\alpha_k = 0$ for k even, and $\alpha_k > 0$ for k odd. Alternated inertia was introduced in [20] as a variation of the proximal point algorithm, and exhibited attractive performances in practice [21]; and it was shown to have potential to achieve good rates (better than both inertia and over-relaxation in some cases) while recovering iterates monotonicity for some range of inertial coefficients. Unfortunately, in the case of the proximal gradient, the range guaranteeing iterates monotonicity is typically $[0, 0.5]$, which prevents from considering usual inertial sequences. Moreover, functional monotonicity has not been investigated yet.

Contributions and Outline. In this paper, we show that the alternated inertial proximal gradient algorithm enjoys a monotonic functional descent for a range of inertial coefficients encompassing the popular sequences of the literature. Furthermore, building on this monotonic decrease, we provide a complexity analysis based on the geometry of the function around optimal points. In addition, we discuss several extensions of the alternated inertial proximal algorithm for i) being resilient to undetected strong convexity, ii) handling non-convex g , and iii) recovering a $1/k^2$ worst-case rate. Finally, we provide numerical examples illustrating the attractive properties of alternated inertia compared to existing (accelerated) proximal gradient algorithms.

The outline of this paper is as follows. Section 2 sets the framework with the assumptions and useful known results. Section 3 is the core of the paper and presents the study of the alternated inertial proximal

gradient algorithm: its monotonicity and the resulting complexity analysis. This analysis is based on the standard Kurdyka-Lojasiewicz inequality but in an original way: a by-product of our developments is indeed a refined analysis of functional convergence rates for general weakly decreasing sequences. Finally, Section 4 discusses extensions and Section 5 presents illustrative numerical experiments.

2 Preliminaries: Assumptions and Recalls on the Proximal Gradient algorithm

In this paper, the working space is \mathbb{R}^n with the standard inner product $\langle \cdot, \cdot \rangle$ and associated Euclidean norm $\| \cdot \|$. Although all forthcoming results generalize to Hilbert spaces, we restrict ourselves to finite-dimensional spaces for simplicity and for being closer to considered applications. We use in our developments standard terminology, notation, and tools from convex analysis and operator theory; we refer to the textbooks [22] and [23]. In particular, for a convex function h , we denote by $\partial h(x)$ the subdifferential of h at x .

We make the following standard assumptions on the functions involved in the composite problem (1).

Assumption 1 (i) $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex L -smooth function (for $L > 0$) i.e. it is differentiable and

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in \mathbb{R}^n;$$

(ii) $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is a convex and proper lower semi-continuous function:

(iii) the set of optimal points $X^* := \operatorname{argmin}_x F(x) \neq \emptyset$ is closed and $F^* := F(x^*) > -\infty$ for any $x^* \in X^*$.

These three points will be assumed to hold for all the results stated in this paper (except for (ii) in Section 4.2 dealing with the non-convex case). For simplicity, we will also assume that the step-sizes of the algorithms are fixed over time: $\gamma_k = \gamma > 0$ for all k ; all the results presented in this paper extend easily to the case when step-sizes are bounded consistently with the Lipschitz constant L . Restricting to fixed step-sizes will slightly enlighten the statements and allow us to focus on the inertia parameters (α_k) which are at the heart of our analysis.

We now define the proximal gradient operator for $F = f + g$ (for the step $\gamma > 0$) by

$$\mathsf{T}_\gamma(x) := \underset{\gamma g}{\mathbf{prox}}(x - \gamma \nabla f(x)).$$

An iteration of the proximal gradient algorithm (2) can thus be written compactly $x_{k+1} = \mathsf{T}_\gamma(x_k)$ and the set X^* of the optimal solutions of (1) coincide with the fixed points set of T_γ .

In our analysis, we will rely on some basic results about this operator that we recall in the three next lemmas. We refer respectively to [23, Th. 25.8/Cor. 27.9], [6, Lemma 1], and [24, Prop. 13] for these results but out of convenience for curious readers, proofs of the last two lemmas are given in the appendix.

Lemma 2.1 (Contraction) Take $\gamma \in]0, \frac{2}{L}[$. Then, the proximal gradient operator T_γ is ν -averaged:

$$\|\mathsf{T}_\gamma(x) - \mathsf{T}_\gamma(y)\|^2 + \frac{1-\nu}{\nu} \|(x - \mathsf{T}_\gamma(x)) - (y - \mathsf{T}_\gamma(y))\|^2 \leq \|x - y\|^2 \quad \text{for all } x, y \in \mathbb{R}^n$$

with $\nu = 2/(1 + 2 \min\{1, 1/(\gamma L)\}) \in [2/3, 1[$.

Lemma 2.2 (Functional descent) Take $\gamma > 0$. Then,

$$F(\mathsf{T}_\gamma(x)) + \frac{(1-\gamma L)}{2\gamma} \|\mathsf{T}_\gamma(x) - x\|^2 + \frac{1}{2\gamma} \|\mathsf{T}_\gamma(x) - y\|^2 \leq F(y) + \frac{1}{2\gamma} \|x - y\|^2 \quad \text{for all } x, y \in \mathbb{R}^n.$$

Lemma 2.3 (Bound on distance to subdifferential) Take $\gamma > 0$. Then,

$$\operatorname{dist}(0, \partial F(\mathsf{T}_\gamma(x))) \leq \frac{L\gamma + 1}{\gamma} \|x - \mathsf{T}_\gamma(x)\| \quad \text{for all } x \in \mathbb{R}^n.$$

As a direct consequence of these lemmas, we retrieve the well-known monotonicity results for the proximal gradient algorithm. For $\gamma \in]0, 2/L[$ and any $x^* \in X^*$, we have indeed that the sequence generated by (2) satisfies

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \frac{1-\nu}{\nu} \|x_{k+1} - x_k\|^2 \\ F(x_{k+1}) &\leq F(x_k) - \frac{2-\gamma L}{2\gamma} \|x_{k+1} - x_k\|^2 \\ &\leq F(x_k) - \frac{(2-\gamma L)\gamma}{(1+\gamma L)^2} \text{dist}(0, \partial F(x_{k+1}))^2. \end{aligned}$$

Thus, (x_k) converges to a point in X^* monotonically in both *Fejér* and *functional* sense. As already mentioned in the introduction, inertial versions of the algorithm lose monotonicity, which is, roughly speaking, due to local changes in the shape of the function (we refer to the references given in the introduction and numerical experiments). We will see in the next section that alternating inertia allows to keep some monotonicity properties of the initial proximal gradient method.

3 Proximal Gradient Algorithm with Alternated Inertia

In this section, we study the proximal gradient algorithm with alternated inertia, introduced in [20] and [21]. With our notations, an iteration of this algorithm reads, for k even, as

$$\begin{cases} y_{k+1} = \mathsf{T}_\gamma(x_k) & y_{k+2} = \mathsf{T}_\gamma(x_{k+1}) \\ x_{k+1} = y_{k+1} + \alpha_k(y_{k+1} - y_k) & x_{k+2} = y_{k+2} \end{cases}. \quad (5)$$

In Section 3.1, we briefly discuss known results about the iterates monotonicity for this method. We then derive in Section 3.2 a functional monotonicity result for a wide range of inertial parameters. Finally, in Section 3.3, we leverage this monotonicity to provide convergence rates using geometrical properties of the objective function. This study is also a chance to establish a new generic convergence analysis using Kurdyka-Łojasiewicz gradient inequality (see forthcoming Theorem 3.3).

3.1 Recalls about Iterates Monotonicity

The iterates monotonicity of the proximal gradient with alternated inertia is established in [21, Lemma 6].

Theorem 3.1 (Iterates Monotonicity) *Take $\gamma \in]0, \frac{2}{L}[$. If the inertial sequence (α_k) verifies*

$$0 \leq \alpha_k \leq \min \left\{ 1, \frac{1}{\gamma L} \right\} - \frac{1}{2} \quad \text{for all } k > 0,$$

then the sequence (y_{2k}) produced by (5) converges Fejér monotonically to a point in X^ .*

For instance, with the typical choice $\gamma = 1/L$, this result states that taking a fixed inertia $\alpha_k = \alpha \in [0, 0.5]$ leads to a monotonic convergence. This is an interesting property which is verified for the vanilla proximal gradient algorithm but not by other accelerated proximal gradient algorithms; see e.g. [13]. However, this result suffers from the fact that monotonicity cannot be guaranteed when inertia goes to 1, as for the popular Nesterov sequence (4).

3.2 Functional Monotonicity

As recalled in Section 2, the proximal gradient algorithm with $\gamma \in]0, 2/L[$ generates function values $(F(x_k))$ that decrease monotonically to F^* . Interestingly, a monotonic behavior can be retrieved when considering alternated inertia.

Theorem 3.2 (Descent result) *Take $\gamma > 0$, $\alpha_k \geq 0$. Then, for an iteration of (5) with $k > 0$ even,*

$$\begin{aligned} F(y_{k+2}) &\leq F(y_k) - \frac{(2 - \alpha_k - \gamma L)}{2\gamma} \left[\|y_{k+2} - x_{k+1}\|^2 + \|y_{k+1} - x_k\|^2 \right] \\ &\leq F(y_k) - \frac{(2 - \alpha_k - \gamma L)\gamma}{2(1 + \gamma L)^2} \text{dist}(0, \partial F(y_{k+2}))^2. \end{aligned} \quad (6)$$

Proof Applying Lemma 2.2 with $x = y = x_k = y_k$, we get

$$F(y_{k+1}) + \frac{(2 - \gamma L)}{2\gamma} \|y_{k+1} - x_k\|^2 \leq F(y_k) \quad (7)$$

and by applying Lemma 2.2 with $x = x_{k+1}$ and $y = y_{k+1}$, we get

$$F(y_{k+2}) + \frac{(1 - \gamma L)}{2\gamma} \|y_{k+2} - x_{k+1}\|^2 + \frac{1}{2\gamma} \|y_{k+2} - y_{k+1}\|^2 \leq F(y_{k+1}) + \frac{1}{2\gamma} \|x_{k+1} - y_{k+1}\|^2. \quad (8)$$

Summing Eqs. (7) and (8), we obtain

$$\begin{aligned} F(y_{k+2}) &\leq F(y_k) + \frac{1}{2\gamma} \|x_{k+1} - y_{k+1}\|^2 - \frac{1}{2\gamma} \|y_{k+2} - y_{k+1}\|^2 \\ &\quad - \frac{(2 - \gamma L)}{2\gamma} \|y_{k+1} - x_k\|^2 - \frac{(1 - \gamma L)}{2\gamma} \|y_{k+2} - x_{k+1}\|^2 \\ &= F(y_k) - \frac{1}{2\gamma} \|y_{k+2} - x_{k+1}\|^2 - \frac{1}{\gamma} \langle y_{k+2} - x_{k+1}; x_{k+1} - y_{k+1} \rangle \\ &\quad - \frac{(2 - \gamma L)}{2\gamma} \|y_{k+1} - x_k\|^2 - \frac{(1 - \gamma L)}{2\gamma} \|y_{k+2} - x_{k+1}\|^2 \\ &= F(y_k) - \frac{\alpha_k}{\gamma} \langle y_{k+2} - x_{k+1}; y_{k+1} - y_k \rangle \\ &\quad - \frac{(2 - \gamma L)}{2\gamma} \|y_{k+1} - x_k\|^2 - \frac{(2 - \gamma L)}{2\gamma} \|y_{k+2} - x_{k+1}\|^2 \\ &\leq F(y_k) - \frac{(2 - \alpha_k - \gamma L)}{2\gamma} \left[\|y_{k+2} - x_{k+1}\|^2 + \|y_{k+1} - x_k\|^2 \right] \end{aligned}$$

where the final inequality comes from Cauchy-Schwarz and Young's inequalities. Lemma 2.3 then directly gives the second part of the result. \square

Instantiating this result with the standard step-size $\gamma = 1/L$, we have that the choice $\alpha_k = \alpha = 1 - \varepsilon$ for any $\varepsilon \in]0, 1]$ leads to a monotonic functional convergence. This is in contrast with Section 3.1 where the condition for iterates monotonic convergence was $\alpha \leq 0.5$. Interestingly, if (α_k) is chosen as in (4) or any sequence valued in $[0, 1]$, the functional descent is preserved (although maybe not the convergence). Note also that, unlike MFISTA, MAPG, or MTwist, no functional evaluation is needed to guarantee this monotonicity. Thus, this monotonicity brought by alternating inertia has a practical interest; we will see in the next section that it also has a theoretical interest as it opens the door for a complexity analysis.

3.3 Complexity Analysis

The functional monotonicity of an algorithm can be combined with some geometric properties of the objective function in order to derive convergence rates. Two types of geometric profiles are often used: error bounds or Kurdyka-Łojasiewicz gradient inequalities (see the extended survey in [25] or the associated article [24]). For a nonsmooth¹ function $\Phi: \mathbb{R}^n \rightarrow \mathbb{R}$ achieving its minimum Φ^* (so that $\text{argmin } \Phi \neq \emptyset$), these local properties write for $x \notin \text{argmin } \Phi$ as

– <i>error bounds</i>	$\varphi(\Phi(x) - \Phi^*) \geq \text{dist}(x, \text{argmin } \Phi)$
– <i>Kurdyka-Łojasiewicz</i> (KL)	$\varphi'(\Phi(x) - \Phi^*) \text{dist}(0, \partial\Phi(x)) \geq 1$

¹ For a nonsmooth (possibly nonconvex) function $\Phi: \mathbb{R}^n \rightarrow \mathbb{R}$, we denote by $\partial\Phi(x)$ the limiting (Fréchet) subdifferential at x [26]. If Φ is convex, this subdifferential coincides with the standard convex subdifferential.

where φ is a smooth increasing concave function, called *desingularizing function* (see [27] or [24] for details). Typical desingularizing functions are of the form $\varphi(t) = Ct^\theta/\theta$ for $\theta \in]0, 1]$ and $C > 0$. Interestingly, with this class of desingularizing functions, the two properties are equivalent for convex functions with the same desingularizing function [24, Th. 5]. Also, in this case, we can reformulate the KL property in a simpler way: Φ has the KL property (with $\theta \in]0, 1]$ and $C > 0$) if

$$\text{dist}(0, \partial\Phi(x)) \geq \frac{1}{C}(\Phi(x) - \Phi^*)^{1-\theta} \quad \text{for all } x \notin \text{argmin } \Phi. \quad (9)$$

The KL property may look strong at a first sight but it turns out to be widely satisfied. In particular, any convex semi-algebraic function verifies the KL property [27]. In some special cases, we know explicitly the parameters θ and C of the desingularizing function; see the examples in [24, Sec. 3]. For instance, we know that ℓ_1 -regularized least-squares functions

$$\Phi(x) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda_1 \|x\|_1$$

have the KL property (9) on ℓ_1 -balls with $\varphi(s) = C\sqrt{s}$ with an explicit constant C from A, b, λ_1 and the size of the ball [24, Lem. 10].

The KL property is particularly well-suited for complexity analysis of descent algorithms: see in particular [28, 29] for so-called subgradient descent algorithms, or [19] for algorithms showing a descent property of the form

$$\Phi(x_{k+1}) \leq \Phi(x_k) - a \text{ dist}^2(0, \partial\Phi(x_{k+1})).$$

However, Theorem 3.2 does not offer such properties for typical $\gamma = 1/L$ and Nesterov-like inertial sequences: the decreasing term in (6) may go to 0 but not faster than $1/k$ which we denote² $a_k = \Omega(1/k)$. We thus provide an extension of existing results to cover our situation: the following theorem states functional convergence rates for general algorithms showing the weak descent property (10), as the one of alternated inertia proximal gradient algorithm.

Theorem 3.3 (General convergence with weak descent) *Let Φ verify the KL property (9). Suppose that there are a non-negative sequence (a_k) and an \mathbb{R}^n -valued sequence (x_k) verifying*

$$\Phi(x_{k+1}) \leq \Phi(x_k) - a_k \text{ dist}^2(0, \partial\Phi(x_{k+1})) \quad \text{and} \quad \sum_{k=1}^{\infty} a_k = +\infty \quad (10)$$

then $\Phi(x_k)$ converges to Φ^* and

$$\Phi(x_{k+1}) - \Phi^* = \begin{cases} \mathcal{O}\left(\frac{1}{(\sum_{\ell=0}^k \min\{a_\ell, C_l\})^{\frac{1}{1-\theta}}}\right) & \text{for } \theta \in]0, 0.5[\\ \mathcal{O}\left(\exp\left(-\frac{1}{2C^2} \sum_{\ell=0}^k \min\{a_\ell, C_e\}\right)\right) & \text{for } \theta \in [0.5, 1[\\ 0 \quad \forall k \geq K := \inf\left\{k : \sum_{\ell=0}^k a_\ell > C^2(\Phi(x_0) - \Phi^*)\right\} & \text{for } \theta = 1 \end{cases}$$

for positive constants C_l, C_e . Then, depending on the behavior of (a_k) divided in three regimes:

- (a) $a_k \geq a > 0$
- (b) $a_k \geq 0$ with $a_k = \Omega\left(\frac{1}{k^d}\right)$, $d \in]0, 1[$
- (c) $a_k \geq 0$ with $a_k = \Omega\left(\frac{1}{k}\right)$

we get the functional convergence rates displayed in Table 1.

Proof The monotonicity of $(\Phi(x_k))$ implies that $\Phi(x_k) \rightarrow \bar{\Phi}$. Since $\sum_{k=1}^{\infty} a_k = +\infty$, we have moreover that a subsequence of $(\text{dist}(0, \partial\Phi(x_k)))$ vanishes. By using (9), this yields that the corresponding subsequence of $(\Phi(x_k))$ converges to Φ^* , hence $\bar{\Phi} = \Phi^*$.

² $a_k = \Omega(b_k)$ if $\exists a, K$ such that $\forall k \geq K$ we have $a_k \geq a.b_k$

	$\theta \in]0, 0.5[$	$\theta \in [0.5, 1[$	$\theta = 1$
(a)	$\mathcal{O}\left(\frac{1}{k^{1+\frac{2\theta}{1-2\theta}}}\right)$	$\mathcal{O}\left(\left[\frac{C^2}{C^2+a}\right]^k\right)$	finite
(b)	$\mathcal{O}\left(\frac{1}{k^{1+\frac{2\theta-d}{1-2\theta}}}\right)$	$\mathcal{O}\left(\exp\left(-\frac{C'}{2C^2}k^{1-d}\right)\right)$	finite
(c)	$\mathcal{O}\left(\frac{1}{\log(k)^{\frac{1}{1-2\theta}}}\right)$	$\mathcal{O}\left(\frac{1}{k^{\frac{C^m}{2C^2}}}\right)$	finite

Table 1 Functional convergence rates with $C' = \liminf_k s_k/k^{1-d}$, $C^m = \liminf_k s_k/\log(k)$ and $s_k = \sum_{\ell \leq k} \min(a_\ell, 2C^2)$.

Define $r_k := \Phi(x_k) - \Phi^*$, then $r_k \rightarrow 0$ monotonically. Thus, one can again use the KL property (9) and the descent on $\Phi(x_k)$ to get

$$r_{k+1} \leq r_k - \frac{a_k}{C^2} r_{k+1}^{2-2\theta} \Leftrightarrow \frac{a_k}{C^2} \leq (r_k - r_{k+1}) r_{k+1}^{2\theta-2}. \quad (11)$$

which will be our core equation for rates derivations, which we separate in three cases depending on θ .

Case $\theta = 1$. Eq. (11) becomes $r_{k+1} \leq r_k - \frac{a_k}{C^2}$, so by summing this inequality we get

$$r_{k+1} \leq r_0 - \frac{1}{C^2} \sum_{\ell=0}^k a_\ell$$

and as (a_k) is a non-negative sequence verifying $\sum_{k=1}^\infty a_k = +\infty$, there is $K < \infty$, such that $\forall k \geq K$, $\frac{1}{C^2} \sum_{\ell=0}^k a_k > r_0$ leading to $r_{k+1} = \Phi(x_{k+1}) - \Phi^* < 0$ which contradicts Φ^* being the minimum of Φ . Thus, we must have $\Phi(x_{k+1}) = \Phi^*$ for all $k \geq K$, i.e. finite convergence.

Case $\theta \in [0.5, 1[$. Here $0 < 2 - 2\theta \leq 1$. Since $r_k \rightarrow 0$ monotonically, we have $r_k^{2-2\theta} \geq r_k$ for all $k \geq K$,

$$r_{k+1} \leq r_k - \frac{a_k}{C^2} r_{k+1} \Leftrightarrow r_{k+1} \leq \frac{1}{1 + \frac{a_k}{C^2}} r_k$$

which leads to different convergence modes depending of (a_k) . If it is bounded away from zero, linear convergence arises (case (a)). Else, one gets that the above inequality also holds with a_k replaced by $a'_k = \min(a_k, 2C^2)$, and we have $\log(1/(1 + \frac{a'_k}{C^2})) \leq -\frac{a'_k}{2C^2}$. In addition, if $a_k = \Omega(b_k)$ with $b_k \rightarrow 0$, then $a'_k = \Omega(b_k)$ too so

$$\log(r_{k+1}) \leq \sum_{\ell=0}^k \log\left(\frac{1}{1 + \frac{a'_\ell}{C^2}}\right) + \log(r_0) \leq -\frac{1}{2C^2} \sum_{\ell=0}^k a'_\ell + \log(r_0) \leq \begin{cases} -\frac{C'}{2C^2} k^{1-d} + \log(r_0) & \text{case (b)} \\ -\frac{C^m}{2C^2} \log(k) + \log(r_0) & \text{case (c)} \end{cases}$$

leading, for C_b, C_c two positive constants, to

$$r_{k+1} \leq \begin{cases} C_b \exp(-\frac{C'}{2C^2} k^{1-d}) & \text{case (b)} \\ C_c \frac{1}{k^{C^m/(2C^2)}} & \text{case (c)} \end{cases}$$

Case $\theta \in]0, 0.5[$. Here $-2 < 2\theta - 2 < -1$ so as $0 \leq r_{k+1} \leq r_k$, we have $0 \leq r_k^{2\theta-2} \leq r_{k+1}^{2\theta-2}$.

Define $\phi(t) = \frac{C}{1-2\theta} t^{2\theta-1}$ with the same C, θ as in φ . Let us turn our attention to $\phi(r_{k+1}) - \phi(r_k)$ that we want to lower-bound by a constant. We proceed in two subcases:

If $r_{k+1}^{2\theta-2} \leq 2r_k^{2\theta-2}$. Then we have

$$\begin{aligned} \phi(r_{k+1}) - \phi(r_k) &= - \int_{r_{k+1}}^{r_k} \phi'(t) dt = C \int_{r_{k+1}}^{r_k} t^{2\theta-2} dt \geq C(r_k - r_{k+1}) r_k^{2\theta-2} \\ &\geq \frac{C}{2} (r_k - r_{k+1}) r_{k+1}^{2\theta-2} = \frac{a_k}{2C} := d_{a_k} \end{aligned}$$

If $r_{k+1}^{2\theta-2} \geq 2r_k^{2\theta-2}$. Then we have $r_{k+1}^{2\theta-1} \geq 2^{\frac{2\theta-1}{2\theta-2}} r_k^{2\theta-1}$

$$\phi(r_{k+1}) - \phi(r_k) = \frac{C}{1-2\theta} (r_{k+1}^{2\theta-1} - r_k^{2\theta-1}) \geq \frac{C}{1-2\theta} \left(2^{\frac{2\theta-1}{2\theta-2}} - 1\right) r_k^{2\theta-1} \geq \frac{C}{1-2\theta} \left(2^{\frac{2\theta-1}{2\theta-2}} - 1\right) r_0^{2\theta-1} := d_b$$

Thus, we have $\phi(r_{k+1}) - \phi(r_k) \geq a'_k/(2C)$ with $a'_k = \min\{a_k, 2Cd_b\}$, thus

$$\phi(r_k) = \phi(r_0) + \sum_{\ell=0}^{k-1} \phi(r_{\ell+1}) - \phi(r_\ell) \geq \frac{1}{2C} \sum_{\ell=0}^{k-1} a'_\ell.$$

We have to split into two cases:

case (a) $\phi(r_k) \geq k \min\{a/2C; d_b\}$ hence the result holds by inverting ϕ .

cases (b) and (c), as before $a'_k = \Omega(b_k)$ and

$$\phi(r_{k+1}) \geq \frac{1}{2C} \sum_{\ell=0}^k a'_\ell \geq \begin{cases} C_b k^{1-d} & \text{case (b)} \\ C_c \log(k) & \text{case (c)} \end{cases} \quad r_{k+1} \leq \begin{cases} \left(\frac{C}{(1-2\theta)C_b k^{1-d}} \right)^{\frac{1}{1-2\theta}} & \text{case (b)} \\ \left(\frac{C}{(1-2\theta)C_c \log(k)} \right)^{\frac{1}{1-2\theta}} & \text{case (c)} \end{cases}$$

which leads to the claimed result. \square

The general convergence rates for weakly decreasing sequences thus depend on the geometry of the objective function (controlled by C and θ) and the speed of decrease of $\Phi(x_k)$ is controlled by (a_k) . To our knowledge, this is the first result accepting vanishing (a_k) . Note that the proof of our result uses techniques from [28, Th. 2]; however, it deals with the convergence analysis of functional values and not the iterates sequence (x_k) , in contrast with the analysis of subgradient descent algorithms in [24, 28]. The latter analysis can cover the case of sequence with a stronger descent assumption on the algorithm (typically $\Phi(x_{k+1}) \leq \Phi(x_k) - a\|x_{k+1} - x_k\|^2$). Inertial versions of the proximal gradient algorithm, including the one studied here, do not satisfy this assumption.

Remark 3.4 (Global vs local KL assumptions) *In the above theorem and all next corollaries using the KL property (9), we assume it holds for all $x \in \text{dom } \Phi$ for sake of simplicity (since the main topic of the paper is alternated inertia). In fact, the result holds under milder conditions; for instance, the exact same proof can be carried through without the global KL assumption in the following cases:*

- If (9) holds on $\{x : \Phi(x) \leq \Phi(x_0)\}$ only; then, as $\Phi(x_k) \leq \Phi(x_0)$, the property can be applied for the whole sequence (x_k) .
- If Φ is coercive, (a_k) is non-increasing, and (9) holds on $\{x : \Phi(x) \leq \Phi^* + \eta\}$ for some $\eta > 0$; then (x_k) is bounded, and thus there is converging subsequence verifying $(\text{dist}(0, \partial\Phi(x_{k_n}))) \rightarrow 0$ thus $\Phi(x_k) \rightarrow \Phi^*$ monotonically so after some K , $\Phi(x_k) \leq \Phi^* + \eta$ and (11) holds.
- If (9) holds on $\{x : \text{dist}(0, \partial\Phi(x)) \leq \varepsilon\} \cup \{x : \Phi(x) \leq \Phi^* + \eta\}$ for some $\varepsilon, \eta > 0$; then the property can be used after some K to retrieve convergence to Φ^* and (11) with the same arguments.

In addition, Theorem 3.3 also holds when (9) is verified for some local minimizer $\hat{\Phi}$ instead of Φ^* .

Combining the general convergence result with the derived descent of the proximal gradient algorithm with alternated inertia, we get the following convergence rates for our algorithm of interest.

Corollary 3.5 (Convergence and Rates of alternated inertial proximal gradient) *Let F verify the KL property (9). Then, for the alternated inertial proximal gradient algorithm (5), $F(y_{2k})$ converges to F^* with the functional rates presented in Table 1 depending on the inertia parameters (α_k) taken in $[0, 1]$, divided in three regimes:*

- (a) when $\gamma < 1/L$ and any inertia choice. Or when $\gamma = 1/L$ and limited inertia $\alpha_k \leq \alpha < 1$.
- (b) when $\gamma = 1/L$ and inertia parameters (α_k) converging to 1 at rate $1/k^d$ ($d \in]0, 1[$).
- (c) when $\gamma = 1/L$ and inertia parameters (α_k) converging to 1 at rate $1/k$.

Proof The result follows from the descent result of Theorem 3.2, precisely Eq. (6) which gives

$$F(y_{k+2}) \leq F(y_k) - \frac{(2 - \alpha_k - \gamma L)\gamma}{2(1 + \gamma L)^2} \text{dist}(0, \partial F(y_{k+2}))^2$$

combined with Theorem 3.3 applied with $(x_k) \equiv (y_{2k})$, $\Phi \equiv F$, and $a_k = \frac{(2 - \alpha_k - \gamma L)\gamma}{2(1 + \gamma L)^2}$. \square

The convergence rates are established for various inertial parameters (α_k) and strongly depends on the geometric properties: $\theta = 1$ corresponding to finite convergence, $\theta \in [0.5, 1[$ corresponding to linear or sublinear convergence, and $\theta \in]0, 0.5[$ corresponding to sublinear convergence. In particular, for case (b) corresponding to the inertia type used in [7]: in the least favorable case $\theta \in]0, 0.5[$, convergence is at least $\mathcal{O}(1/k)$ when $d \leq 2\theta$ and can be $\mathcal{O}(1/k^2)$ for $\theta \in]0.25, 0.5[$ with $d \leq 4\theta - 1$. The case (c) covers the popular Nesterov sequence (4).

Notice also that the present analysis does not extend easily to other monotonic accelerated proximal gradient algorithms. Indeed, MTwist [17] and MFISTA [14] only rely on function evaluation to get non-strict functional decrease and thus do not enjoy convergence rates from the above results. Concerning Monotone APG [19], the additional *unaccelerated* step at each iteration makes it fit in the strongest case (a) (the proof of which is inspired from [19]) but at the cost of roughly doubling the computation expenses.

4 Extensions

4.1 Resilience to Strong Convexity

Let us consider the case where F is in addition μ -strongly convex (typically when f is μ -strongly convex and g is simply convex) but this strong convexity is either undetected, local, or μ is unknown. Then, the vanilla proximal gradient algorithm is known to have exponential convergence (see [30] for a proof based on error bounds) but accelerated versions such as FISTA only have polynomial convergence ($1/k^2$ in general). Moreover, these convergence rates clearly show in practice; see e.g. [15] or the numerical illustrations of Section 5. For the alternated inertial proximal gradient algorithm, we establish a better-than-polynomial convergence rate, which moreover translates in practice as shown in the numerical experiments.

Proposition 4.1 (Strongly convex case) *Assume that F is in addition μ -strongly convex and take γ and (α_k) as in Corollary 3.5. Then, the alternated inertial proximal gradient algorithm (5) verifies:*

- better-than-polynomial rate: *provided that γ and (α_k) are chosen as in (a) or (b), $F(y_{2k})$ converges to F^* at rate $\mathcal{O}(\exp(-\nu k^{1-d}))$ for $\nu > 0$ and $d = 0$ (case (a)) or $d \in]0, 1[$ (case (b));*
- iterates convergence: *(y_{2k}) converges to the unique minimizer of F at the same rate.*

Proof As F is μ -strongly convex, we have $F(x) - F^* \geq \frac{\mu}{2}\|x - x^*\|^2 = \mu/2 \operatorname{dist}^2(x, X^*)$ which is a (global) error bound with perspective function $\varphi(t) = \sqrt{2/\mu} t^{0.5}$. Thanks to the equivalence between error bounds and KL property for convex functions [24, Th. 5], we have for the alternated inertial proximal gradient the rates of Corollary 3.5 with the perspective function $\tilde{\varphi}(t) = 2\sqrt{2/\mu} t^{0.5}$ i.e. $\theta = 0.5$ (as the KL property actually holds for all x). So, provided that $(2 - \alpha_k - \gamma L)$ does not vanish as quickly as $1/k$, we recover the rates of the top two lines of the column corresponding to $\theta = 0.5$ in Theorem 3.3. Finally, as $\|y_{2k} - x^*\|^2 \leq 2/\mu(F(y_{2k}) - F^*)$, functional convergence (and rate) leads to iterates convergence (and rate). \square

4.2 Case of Non-Convex g

In general, Kurdyka-Lojasiewicz inequality is particularly suited to study non-convex case as a large number of non-convex functions verify KL properties. However, one has to find a functional decrease in algorithms to establish convergence properties. This decrease may be harder to obtain in the non-convex case. While one can get a functional decrease for the vanilla proximal gradient even when f and g are non-convex³, the presence of (alternated) inertia requires to be able to take $x \neq y$ in Lemma 2.2 which prevents from getting rid of the convexity of f .

For f non-convex and g convex, a solution, used for instance in the algorithm iPiano [31], is to slightly modify the inertial proximal gradient to recover a descent property on some Lyapunov function. Then, convergence and rates can be obtained by an adaptation of the existing results of [27].

³ In the case of the proximal gradient, the proof of Lemma 2.2 recalled in the appendix requires (i) convexity of f in order to take $x \neq y$ (see Eq. (16)); (ii) convexity of g to get the term in $\|\mathbb{T}_\gamma(x) - y\|$ by strong convexity of the proximal surrogate (see Eq. (14)).

Here, we rather study the case where the convexity of g is dropped. The definition of the proximal operator directly extends [26, Def. 1.22] and our analysis indeed follows at the expense of a smaller range of stepsizes and inertia parameters, as formalized in the following proposition.

Proposition 4.2 (Non-convex case) *Let Assumption 1 hold without the convexity of g . Let F verify the KL property (9). Then, the alternated inertial proximal gradient algorithm (5) has the functional rates of Theorem 3.3 depending on (α_k) corresponding to the three regimes:*

- (a) when $\gamma < 1/(2L)$ and any $\alpha_k \leq 1/2$. Or when $\gamma = 1/(2L)$ and limited inertia $\alpha_k \leq \alpha < 1/2$.
- (b) when $\gamma = 1/(2L)$ and inertia parameters (α_k) converging to $1/2$ at rate $1/k^d$ ($d \in]0, 1[$).
- (c) when $\gamma = 1/(2L)$ and inertia parameters (α_k) converging to $1/2$ at rate $1/k$.

Proof We use Lemma A.1 which is the non-convex version of Lemma 2.2. The descent term in Theorem 3.2 is now factored with $(1 - \alpha_k - \gamma L)$ instead of $(2 - \alpha_k - \gamma L)$. Then, all the following results (and notably the rates) hold with this modification since the KL property is independent of convexity. \square

4.3 Proximal Gradient Algorithm with Alternated Extrapolation

In this subsection, we extend the alternated inertia to a more general alternated ‘‘extrapolation’’, the iterations of which take the form

$$\begin{cases} y_{k+1} = \mathsf{T}_\gamma(x_k) & y_{k+2} = \mathsf{T}_\gamma(x_{k+1}) \\ x_{k+1} = \mathbf{extrapolation}(\{y_\ell\}_{\ell \leq k+1}) & x_{k+2} = y_{k+2} \end{cases} \quad (12)$$

where $\mathbf{extrapolation}(\{y_\ell\}_{\ell \leq k+1})$ is a linear combination of past iterates (see for instance the recent [32]). We show here that the following extrapolation step, surprisingly close to *heavy balls*,

$$x_{k+1} = y_{k+1} - \frac{1}{t_{k/2+1}}(y_{k+1} - y_k) + \frac{t_{k/2} - 1}{t_{k/2+1}}(y_k - y_{k-1}) \quad \text{with } (t_k) \text{ as in (4)} \quad (13)$$

guarantees a worst case $\mathcal{O}(1/k^2)$ rate. In other words, alternating extrapolation allows attaining the same rates as standard inertia (e.g. FISTA) where alternating inertia cannot (see Corollary 3.5). Furthermore, if the KL property is additionally assumed then this rate may be improved to a small- o convergence instead of big- \mathcal{O} for $\theta = 0.5$ (with different arguments than in [8]) and faster convergence as θ increases.

Proposition 4.3 (alternated extrapolation) *Take $\gamma \in]0, 1/L[$. Then, the algorithm of (12) with the extrapolation defined in (13) verifies for k odd*

$$F(y_k) - F^* \leq \frac{\|x_0 - x^*\|^2}{2\gamma t_{k/2}^2} = \mathcal{O}\left(\frac{1}{k^2}\right) \quad \text{and} \quad \|\mathsf{T}_\gamma(y_k) - y_k\|^2 = o\left(\frac{1}{t_{k/2}^2}\right) = o\left(\frac{1}{k^2}\right).$$

Moreover, if F verifies the KL property (9); then, we have that for k odd

$$F(y_k) - F^* = o\left(\frac{1}{k^{\frac{1}{1-\theta}}}\right).$$

Proof From the descent lemma with extrapolation [14] with $x = x_k$ for k even, and defining $F_k = F(y_k) - F^*$, we get

$$t_{k/2+1}^2 F_{k+2} - t_{k/2}^2 F_{k+1} \leq -\frac{1}{2\gamma} \left[\|t_{k/2+1} y_{k+2} - (t_{k/2+1} - 1)y_{k+1} - y^*\|^2 - \|t_{k/2+1} x_{k+1} - (t_{k/2+1} - 1)y_{k+1} - y^*\|^2 \right].$$

As $y_{k+1} = \mathsf{T}_\gamma(x_k) = \mathsf{T}_\gamma(y_k)$, we have $F(y_{k+1}) \leq F(y_k) - \frac{1}{2\gamma} \|\mathsf{T}_\gamma(y_k) - y_k\|^2$ from Lemma 2.2, thus

$$\begin{aligned} t_{k/2+1}^2 F_{k+2} - t_{k/2}^2 F_k &\leq -\frac{1}{2\gamma} \left[\|t_{k/2+1} y_{k+2} - (t_{k/2+1} - 1)y_{k+1} - y^*\|^2 - \|t_{k/2+1} x_{k+1} - (t_{k/2+1} - 1)y_{k+1} - y^*\|^2 \right] \\ &\quad - \frac{t_{k/2}^2}{2\gamma} \|\mathsf{T}_\gamma(y_k) - y_k\|^2 \end{aligned}$$

and in order to have a right hand side of the form $\frac{1}{2\gamma} [\|u_{k+2}\|^2 - \|u_k\|^2]$, one can choose the extrapolation, that is x_{k+1} so that

$$\|t_{k/2+1} x_{k+1} - (t_{k/2+1} - 1)y_{k+1} - y^*\|^2 = \|t_{k/2} y_k - (t_{k/2} - 1)y_{k-1} - y^*\|^2$$

which naturally leads to taking

$$\begin{aligned} t_{k/2+1}x_{k+1} - (t_{k/2+1} - 1)y_{k+1} &= t_{k/2}y_k - (t_{k/2} - 1)y_{k-1} \\ \Leftrightarrow x_{k+1} &= y_{k+1} - \frac{1}{t_{k/2+1}}(y_{k+1} - y_k) + \frac{t_{k/2} - 1}{t_{k/2+1}}(y_k - y_{k-1}). \end{aligned}$$

This choice of extrapolation leads by summing the above inequality for k even to

$$t_{k/2}^2 F_k + \frac{1}{2\gamma} \sum_{\ell=0}^{k/2-1} t_\ell^2 \|\mathbb{T}_\gamma(y_{2\ell}) - y_{2\ell}\|^2 \leq \frac{1}{2\gamma} \|y_0 - y^*\|^2$$

thus, noticing that all terms above are non-negative:

$$- F_k \leq \frac{1}{2\gamma t_{k/2}^2} \|y_0 - y^*\|^2.$$

$$- \sum_{\ell=0}^{\infty} t_\ell^2 \|\mathbb{T}_\gamma(y_{2\ell}) - y_{2\ell}\|^2 < +\infty \text{ thus } \|\mathbb{T}_\gamma(y_{2\ell}) - y_{2\ell}\|^2 = o(1/t_\ell^2).$$

Interestingly, using Lemma 2.3, we get that for k even, $\text{dist}^2(0, \partial F(y_{k+1})) \leq (L + 1/\gamma)^2 \|\mathbb{T}_\gamma(y_k) - y_k\|^2 = o(\frac{1}{k^2})$. If we also assume that F verifies (9) for some $C > 0$, $\theta \in]0, 1[$; then, we have that for k odd

$$F(y_k) - F^* = o\left(\frac{1}{k^{1-\theta}}\right). \quad \square$$

5 Numerical Experiments

We compare the vanilla proximal gradient algorithm with

- *proposed alternated inertial* version with $\alpha_k = t_{k+1}/(t_k - 1)$, $t_k = ((k + a)/a)^d$, and $d = 0.8$;
- *proposed alternated extrapolation* of Section 4.3;
- *inertial* versions: FISTA [11] and monotonic counterparts MFISTA [14] and MAPG [19] as well as lasso-specific MTwist [17].

All algorithms were tuned as advised in their respective sources, the initial point was drawn randomly and common to all methods. We will compare the speed of convergence of algorithms with respect to their number of iterations. One iteration corresponds to one proximal gradient evaluation; more precisely, the additional functional value computations for MFISTA, MTwist, and MAPG were not taken into account, and an iteration of MAPG is counted double as two proximal gradient steps are computed.

In Section 5.1, we consider the composite problem of ℓ_1 -regularized logistic regression. We show that alternated inertia provides an efficient acceleration in the sense that it uniformly accelerates the proximal gradient even when large stepsizes are taken. Then, in Section 5.2, we address the lasso problem which illustrates the resilience of alternated inertia to strong convexity. Finally, in Section 5.3, we revisit the two previous problems but replace the ℓ_1 -norm with the non-convex $\ell_{0.5}$ -“norm”, this allows us to illustrate the monotonicity and performance of alternated inertia even in a non-convex setting.

5.1 ℓ_1 -Regularized Logistic Regression

We first consider the problem of ℓ_1 -regularized logistic regression on the popular `ionosphere` dataset⁴ containing $m = 351$ binary classified examples $(a_i, y_i) \in \mathbb{R}^n \times \{-1, +1\}$ of feature size $n = 35$. The problem writes as

$$\min_{x \in \mathbb{R}^n} \underbrace{\frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-y_i \langle a_i; x \rangle))}_{f(x)} + \underbrace{\lambda_1 \|x\|_1}_{g(x)}$$

and we fix the regularization parameter λ_1 to 0.1.

In Figure 1, we plot the functional error for all compared algorithms with four choices for stepsize γ : (a) $\gamma = 1/L_u$ where L_u is the usual upper bound for the Lipschitz constant of the gradient of f which is known to be overly pessimistic in most practical cases; and (b,c,d) $\gamma = \gamma_{\max}/\nu$ for $\nu = \{8, 3, 1.5\}$

⁴ <https://archive.ics.uci.edu/ml/datasets/ionosphere>

with γ_{\max} being the greatest stepsize admissible for the proximal gradient algorithm before divergence. This second set of stepsize enables to exhibit the behavior of the compared algorithm with practical and performing stepsizes. We notice that the alternated inertial version exhibits a steady monotonic behavior always outperforming the vanilla proximal gradient. Also, while for smaller stepsizes non-monotonic accelerated versions outperform monotonic ones, the trend shifts for the greater ones. Ultimately, for the greatest stepsize vanilla proximal gradient and alternated inertial counterpart are the most performing. Finally, one can see that the alternated extrapolation performs a bit worse than FISTA with the same characteristic oscillation with a frequency twice as low.

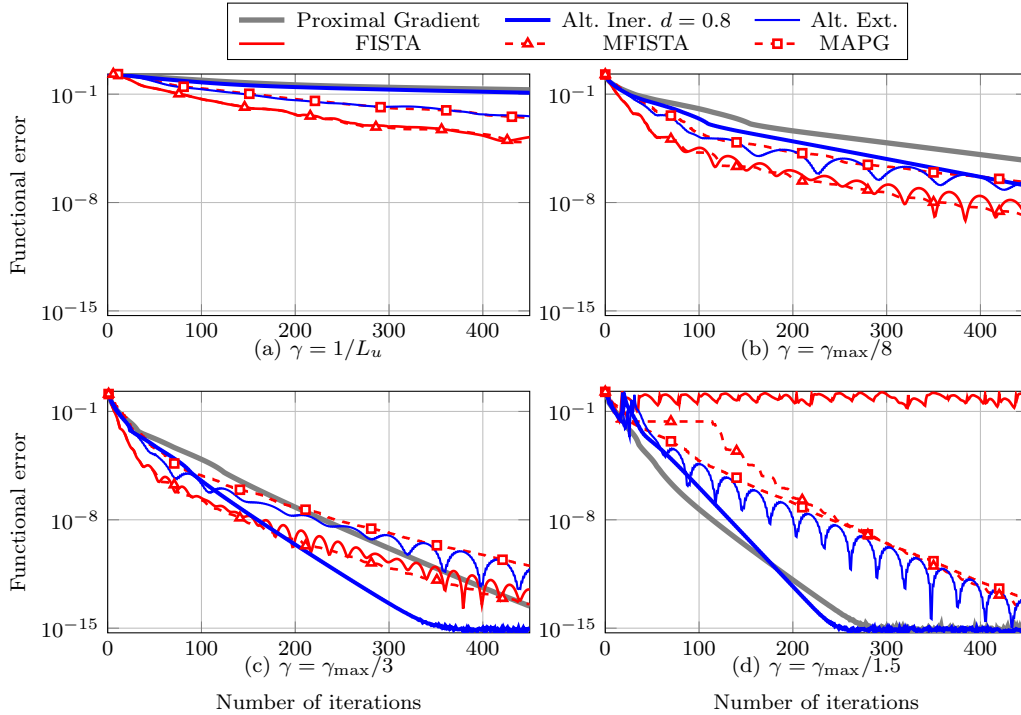


Fig. 1 Comparison of the algorithms on ℓ_1 -regularized logistic regression (functional decrease vs number of iterations). The four cases correspond to four different choices of stepsize γ .

5.2 ℓ_1 -Regularized Least-Squares (Lasso)

We address the lasso problem

$$\min_{x \in \mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{f(x)} + \underbrace{\lambda_1 \|x\|_1}_{g(x)}$$

on synthetic matrix/vector couples $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. A is drawn from the standard normal distribution and $b = Ax_0 + e$ where x_0 is a 10% sparse vector taken from the normal distribution, and e is taken from the normal distribution with standard deviation 0.001. We set λ_1 so that the original sparsity is ultimately recovered.

The interest of this problem is that we can compute exactly the standard stepsize $\gamma = 1/L$ where $L = 2\|A^T A\|$, which we use for all algorithms. We plot the functional error for all compared algorithms with two different sizes of the matrix A , which correspond to two conditioning levels of the lasso problem: (a) 130×80 and (b) 85×80 .

In Figure 2, we observe that the proximal gradient and the alternated inertial counterpart benefit from a more-than-polynomial rate which enables them to outperform most other methods in the better conditioned case, the alternated inertial version being always significantly better than the vanilla one. In the second case, FISTA, MFISTA, and MTwist perform best although alternated inertia is still competitive.

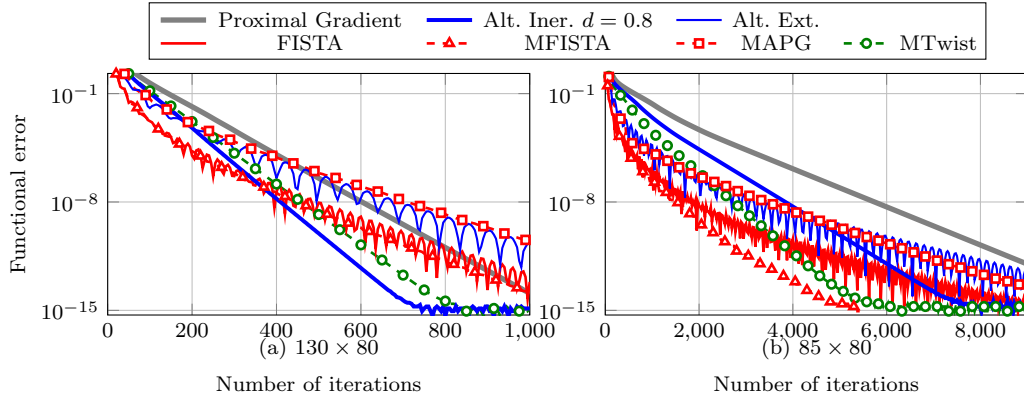


Fig. 2 Comparison of the algorithms on lasso (functional decrease vs number of iterations). The two cases correspond to two problems with two different strong convexity modulus (small and large).

We also checked the wall clock times obtained by IPython’s `timeit` in the situation of Fig. 2 (a) and found that alternated inertia and FISTA roughly take the same the time as the proximal gradient while test-based MFISTA, MAPG, and MTwist are slightly more time consuming due to function evaluations; thus the presented comparisons are more than fair for alternated inertia.

5.3 Non-Convex Problems

In order to investigate the behavior of the compared algorithms in a non-convex setting, we replace the ℓ_1 norm in the two previous problems by the $\ell_{0.5}$ “norm”. This regularization promotes sparsity in a stronger, non-convex, sense than the ℓ_1 norm. This function writes $\|x\|_{0.5}^{0.5} = \sum_{i=1}^n \sqrt{|x_i|}$ and has the attractive feature of having a closed form proximal operator (see [33] and references therein). For these non-convex problems, we do not plot the functional error, as the algorithms may reach different local minimizers depending on the initialization point, but rather: (a) the functional error with respect to the minimal value reached by the algorithm on the run; and (b) the distance of the subgradient to the null vector $\text{dist}^2(0, \partial F(x_k))$. The two figures show that alternated inertia provides a quick and monotonic convergence.

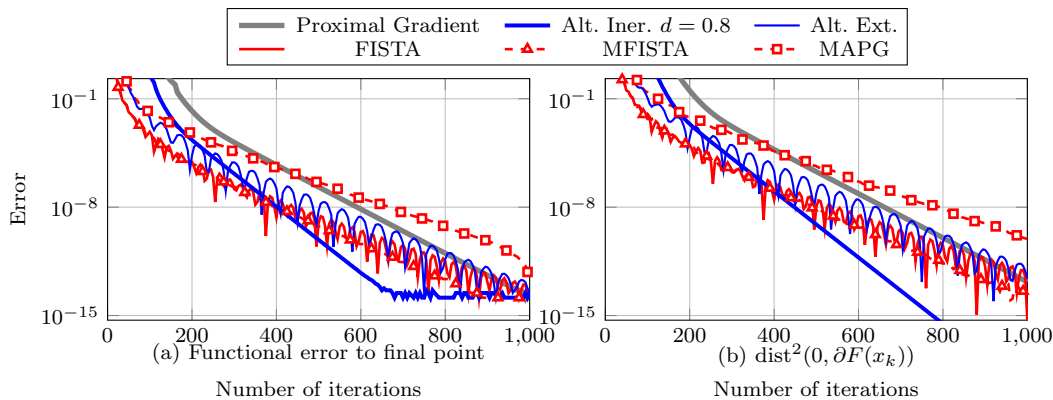


Fig. 3 Comparison of the algorithms on the $\ell_{0.5}$ -regularized logistic regression problem (with $\lambda_{0.5} = 0.002$).

6 Conclusion

Standard algorithms with inertia break down monotonicity; we have proved that adding a proximal gradient step after each inertial step enables to recover monotonicity without extra assumptions on the

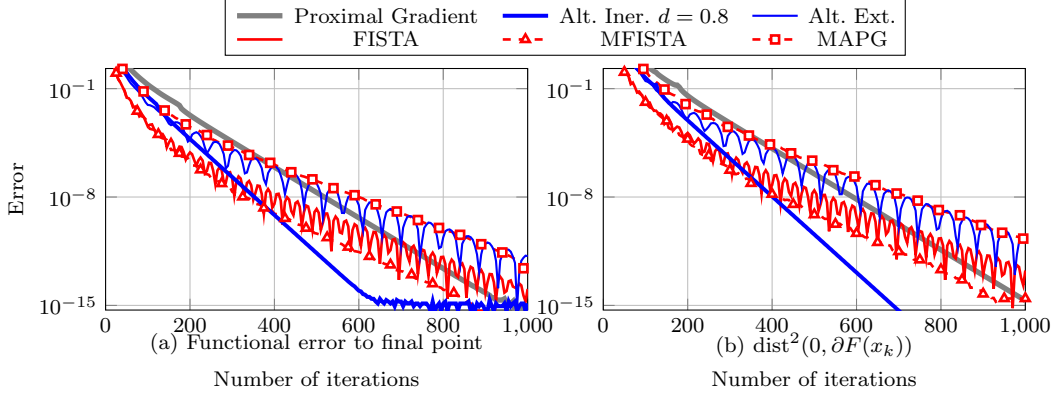


Fig. 4 Comparison of the algorithms on the $\ell_{0.5}$ -regularized linear regression problem (“ $\ell_{0.5}$ -lasso”) with $m = 130$ and $n = 80$. To match the sparsity of x_0 , $\lambda_{0.5}$ was set accordingly to 0.05.

stepsizes or on the inertial sequence. The resulting alternating inertial algorithm enjoys the accelerated behavior of inertial algorithms while keeping the good properties of vanilla proximal gradient algorithms (automatic rate adaptation for strongly convex objectives and generalization for non-convex objectives). The complexity analysis of this algorithm is also the occasion of revealing a general convergence result of weakly decreasing algorithms for minimizing sharp objective functions.

Many extensions of this idea of intermittent inertia are possible; we studied in particular an alternated extrapolation algorithm for which we show that it converges faster as the geometry improves, going beyond the usual worst case rate.

A Appendix with Known Results about the Proximal Gradient

For the sake of completeness, we provide short and direct proofs of known lemmas recalled in Section 2.

Proof of Lemma 2.2 Let $x \in \mathbb{R}^n$; by definition, we have

$$\mathsf{T}_\gamma(x) = \operatorname{argmin}_w \left(\gamma g(w) + \frac{1}{2} \|w - (x - \gamma \nabla f(x))\|^2 \right) = \operatorname{argmin}_w \left(\underbrace{f(x) + g(w) + \langle w - x; \nabla f(x) \rangle + \frac{1}{2\gamma} \|w - x\|^2}_{s_x(w)} \right)$$

and, as it is defined as the minimizer of $\frac{1}{\gamma}$ -strongly convex surrogate function s_x , we have for any $y \in \mathbb{R}^n$ that $s_x(\mathsf{T}_\gamma(x)) + \frac{1}{2\gamma} \|\mathsf{T}_\gamma(x) - y\|^2 \leq s_x(y)$ so

$$f(x) + g(\mathsf{T}_\gamma(x)) + \langle \mathsf{T}_\gamma(x) - x; \nabla f(x) \rangle + \frac{\|\mathsf{T}_\gamma(x) - x\|^2}{2\gamma} + \frac{\|\mathsf{T}_\gamma(x) - y\|^2}{2\gamma} \leq f(x) + g(y) + \langle y - x; \nabla f(x) \rangle + \frac{\|y - x\|^2}{2\gamma}. \quad (14)$$

Now we use (i) the descent lemma on L -smooth function f (see [23, Th. 18.15]) to show that

$$f(\mathsf{T}_\gamma(x)) \leq f(x) + \langle \mathsf{T}_\gamma(x) - x; \nabla f(x) \rangle + \frac{L}{2} \|\mathsf{T}_\gamma(x) - x\|^2 \quad (15)$$

and (ii) the convexity of f to have

$$f(x) + \langle y - x; \nabla f(x) \rangle \leq f(y). \quad (16)$$

Using Eq. (15) on the left hand side of (14) and Eq. (16) on the right hand side, we get

$$f(\mathsf{T}_\gamma(x)) + g(\mathsf{T}_\gamma(x)) + \frac{(1 - \gamma L) \|\mathsf{T}_\gamma(x) - x\|^2}{2\gamma} + \frac{\|\mathsf{T}_\gamma(x) - y\|^2}{2\gamma} \leq f(y) + g(y) + \frac{\|y - x\|^2}{2\gamma}. \quad \square$$

Proof of Lemma 2.3 Let $x \in \mathbb{R}^n$, and let $y = \mathsf{T}_\gamma(x) \in \operatorname{argmin}_w \left(\gamma g(w) + \frac{1}{2} \|w - (x - \gamma \nabla f(x))\|^2 \right)$, then

$$0 \in \gamma \partial g(y) + y - x + \gamma \nabla f(x) \quad \Leftrightarrow \quad 0 \in \nabla f(y) + \partial g(y) + \nabla f(x) - \nabla f(y) + \frac{1}{\gamma}(y - x)$$

so $\nabla f(y) - \nabla f(x) + \frac{1}{\gamma}(x - y) \in \partial F(y)$, thus we have $\operatorname{dist}(0, \partial F(y)) \leq \|\nabla f(y) - \nabla f(x) + \frac{1}{\gamma}(x - y)\| \leq \left(L + \frac{1}{\gamma} \right) \|x - y\|$. \square

Lemma A.1 *Let Assumption 1 hold but with g possibly nonconvex, and take $\gamma > 0$. Then, for any $x, y \in \mathbb{R}^n$,*

$$F(\mathbb{T}_\gamma(x)) + \frac{(1-\gamma L)}{2\gamma} \|\mathbb{T}_\gamma(x) - x\|^2 \leq F(y) + \frac{1}{2\gamma} \|x - y\|^2.$$

Proof Let $x \in \mathbb{R}^n$; by definition, we have, as in the proof of Lemma 2.2,

$$\mathbb{T}_\gamma(x) = \operatorname{argmin}_w \left(\gamma g(w) + \frac{1}{2} \|w - (x - \gamma \nabla f(x))\|^2 \right) = \operatorname{argmin}_w \left(\underbrace{f(x) + g(w) + \langle w - x; \nabla f(x) \rangle}_{s_x(w)} + \frac{1}{2\gamma} \|w - x\|^2 \right)$$

and, as it is defined as a minimizer of (non necessarily convex) surrogate function s_x , we have for any $y \in \mathbb{R}^n$ that $s_x(\mathbb{T}_\gamma(x)) \leq s_x(y)$ (which differs from the convex case of Lemma 2.2) thus

$$f(x) + g(\mathbb{T}_\gamma(x)) + \langle \mathbb{T}_\gamma(x) - x; \nabla f(x) \rangle + \frac{\|\mathbb{T}_\gamma(x) - x\|^2}{2\gamma} \leq f(x) + g(y) + \langle y - x; \nabla f(x) \rangle + \frac{\|y - x\|^2}{2\gamma}. \quad (17)$$

The proof then follows the same lines as that of Lemma 2.2. \square

Acknowledgements The work of the authors is partly supported by the PGMO Grant *Advanced Non-smooth Optimization Methods for Stochastic Programming*.

References

1. Chambolle, A., De Vore, R.A., Lee, N.Y., Lucier, B.J.: Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing* **7**(3), 319–335 (1998)
2. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics* **57**(11), 1413–1457 (2004)
3. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation for ℓ_1 -minimization: Methodology and convergence. *SIAM Journal on Optimization* **19**(3), 1107–1130 (2008)
4. Alvarez, F.: Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in hilbert space. *SIAM Journal on Optimization* **14**(3), 773–782 (2004)
5. Lorenz, D., Pock, T.: An inertial forward-backward algorithm for monotone inclusions. *Journal of Mathematical Imaging and Vision* **51**(2), 311–325 (2014)
6. Chambolle, A., Dossal, C.: On the convergence of the iterates of the “fast iterative shrinkage/thresholding algorithm”. *Journal of Optimization Theory and Applications* **166**(3), 968–982 (2015)
7. Aujol, J.F., Dossal, C.: Stability of over-relaxations for the forward-backward algorithm, application to fista. *SIAM Journal on Optimization* **25**(4), 2408–2433 (2015)
8. Attouch, H., Peypouquet, J.: The rate of convergence of Nesterov’s accelerated forward-backward method is actually faster than $1/k^2$. *SIAM Journal on Optimization* **26**(3), 1824–1834 (2016)
9. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady* **27**(2), 372–376 (1983)
10. Güler, O.: New proximal point algorithms for convex minimization. *SIAM Journal on Optimization* **2**(4), 649–664 (1992)
11. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* **2**(1), 183–202 (2009)
12. Nesterov, Y.: Smooth minimization of non-smooth functions. *Mathematical programming* **103**(1), 127–152 (2005)
13. Maingé, P.E.: Convergence theorems for inertial km-type algorithms. *Journal of Computational and Applied Mathematics* **219**(1), 223–236 (2008)
14. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing* **18**(11), 2419–2434 (2009)
15. Malitsky, Y., Pock, T.: A first-order primal-dual algorithm with linesearch. *arXiv preprint arXiv:1608.08883* (2016)
16. Correa, R., Lemaréchal, C.: Convergence of some algorithms for convex minimization. *Mathematical Programming* **62**(2), 261–275 (1993)
17. Bioucas-Dias, J.M., Figueiredo, M.A.: A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing* **16**(12), 2992–3004 (2007)
18. Fuentes, M., Malick, J., Lemaréchal, C.: Descentwise inexact proximal algorithms for smooth optimization. *Computational Optimization and Applications* **53**(3), 755–769 (2012)
19. Li, H., Lin, Z.: Accelerated proximal gradient methods for nonconvex programming. In: *Advances in neural information processing systems*, pp. 379–387 (2015)
20. Mu, Z., Peng, Y.: A note on the inertial proximal point method. *Statistics, Optimization & Information Computing* **3**(3), 241–248 (2015)
21. Iutzeler, F., Hendrickx, J.M.: A generic linear rate acceleration of optimization algorithms via relaxation and inertia. *arXiv preprint arXiv:1603.05398* (2016)
22. Hiriart-Urruty, J.B., Lemaréchal, C.: *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg (1993). Two volumes

23. Bauschke, H.H., Combettes, P.L.: Convex analysis and monotone operator theory in Hilbert spaces. Springer Science & Business Media (2011)
24. Bolte, J., Nguyen, T.P., Peypouquet, J., Suter, B.W.: From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming* pp. 1–37 (2015)
25. NGuyen, T.P.: Kurdyka-lojasiewicz and convexity: algorithms and applications. Ph.D. thesis, Toulouse University (2017)
26. Rockafellar, R.T., Wets, R.J.B.: Variational Analysis. Springer (1998)
27. Bolte, J., Daniilidis, A., Lewis, A.: The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization* **17**(4), 1205–1223 (2007)
28. Attouch, H., Bolte, J.: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming* **116**(1), 5–16 (2009)
29. Frankel, P., Garrigos, G., Peypouquet, J.: Splitting methods with variable metric for kurdyka-lojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications* **165**(3), 874–900 (2015)
30. Karimi, H., Nutini, J., Schmidt, M.: Linear convergence of gradient and proximal-gradient methods under the polyak-lojasiewicz condition. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 795–811 (2016)
31. Ochs, P., Chen, Y., Brox, T., Pock, T.: ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences* **7**(2), 1388–1419 (2014)
32. Liang, J., Fadili, J., Peyré, G.: A multi-step inertial forward-backward splitting method for non-convex optimization. In: Advances in Neural Information Processing Systems, pp. 4035–4043 (2016)
33. Chartrand, R., Yin, W.: Nonconvex sparse regularization and splitting algorithms. In: Splitting Methods in Communication, Imaging, Science, and Engineering, pp. 237–249. Springer (2016)