Backtracking strategies for accelerated descent methods with smooth composite objectives

Luca Calatroni, Antonin Chambolle

Backtracking strategies for accelerated descent methods with smooth composite objectives

Luca Calatroni[†], Antonin Chambolle [†]

Abstract. We present and analyse a backtracking strategy for a general Fast Iterative Shrinkage/Thresholding Algorithm which has been recently proposed in [11] for strongly convex objective functions. Differently from classical Armijo-type line searching, our backtracking rule allows for local increase and decrease of the Lipschitz constant estimate along the iterations. For such strategy accelerated convergence rates are proved and numerical results are shown for some exemplar imaging problems.

1. Introduction

The concept of *acceleration* of first-order optimisation methods dates back to the seminal work of Nesterov [19]. For a proper, convex, l.s.c. function $F : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ defined on a Hilbert space $\mathcal{X}$ and having Lipschitz gradient with constant $L > 0$, solving the abstract optimisation problem

$$\min_{x \in \mathcal{X}} \; F(x) \tag{1.1}$$

by means of an accelerated iterative method means improving the convergence rate $O(1/k)$ achieved after $k \geq 1$ iterations of standard gradient descent methods in order to (almost) match the universal lower bound of $O(1/k^2)$ holding for any function such as F. In the smoother case, i.e. when F is a strongly convex with parameter $\mu > 0$, Nesterov showed in [20, Theorem 2.1.13] that a lower bound for first-order optimisation methods of the order $O((\frac{\sqrt{q}-1}{\sqrt{q}+1})^{2k})$ can be shown, with $q := L/\mu \geq 1$ being the *condition number* of the function F. In this case, improved improved linear convergence rates of the order $O((\frac{\sqrt{q}-1}{\sqrt{q}})^k)$ are proved. Similar results for implicit gradient descent have similarly been studied by Güler [17]. We also refer the reader to [24], where a general framework for accelerated methods is presented.

If the objective function in (1.1) can be further decomposed into the sum of a convex function f with Lipschitz gradient and a convex, l.s.c. and non-smooth component g as

$$\min_{x \in \mathcal{X}} \; \{F(x) = f(x) + g(x)\}\,, \tag{1.2}$$

alternative descent methods taking into account the non-differentiability of F need to be considered. Typically, they go under the name of *composite optimisation* methods, after the work of Nesterov [22]. A typical optimisation strategy for solving composite optimisation problems consists in alternating a 'forward' (i.e. explicit) gradient descent step taken in the differentiable component f with a 'backward' (implicit) implicit gradient descent step in the non-smooth part g along the iterations. Due to this feature, such optimisation technique is traditionally known as *forward-backward* (FB) splitting. The literature on FB splitting methods is extremely vast. Historically, such strategy has firstly been used in [16] for projected gradient descent, and subsequently attracted the interest of the imaging community after the work of Combettes and Wajs [12]. Acceleration methods for FB splitting has been considered by Nesterov in [20] for

[†] Centre de Mathématiques Appliquées (CMAP), École Polytechnique CNRS, 91128, Palaiseau Cedex, France.

e-mail: luca.calatroni@polytechnique.edu, antonin.chambolle@cmap.polytechnique.fr

projected gradient descent, and popularised later on by Beck and Teboulle in [4] for more general 'simple' non-smooth functions g under the name of Fast Iterative Shrinkage/Thresholding Algorithm (FISTA). Several variants of FISTA have been considered in a number of work such as [21, 29, 22, 10, 6, 5] just to mention a few, and further properties such as convergence of the iterates under specific assumptions ([8]) and monotone variants (M-FISTA) [3, 28] have also been studied. In the case when only an approximate evaluation of the FB operators up to some error can be provided, accelerated convergence rates can also be shown. We refer the reader to [27, 30, 2] for this study.

In its basic formulation, FISTA requires an estimate on the Lipschitz constant $L_f > 0$ of ∇f. Whenever such estimate is not easily computable, an Armijo-type backtracking rule [1] can alternatively be used. From a computational point of view, this backtracking strategy is very limiting since it requires such estimate to be non-decreasing along the iterations. From a practical point of view, this conditions implies that if a large value of this constant is computed in the early iterations, a corresponding small (or even smaller!) gradient step size will be used in the later iterations. As a consequence, convergence speed may suffer if an inaccurate estimate of L_f is computed. To avoid this drawback, Scheinberg, Goldfarb and Bai have proposed in [26] a backtracking strategy for FISTA where and adaptive increasing and decreasing of the estimated Lipschitz constant along the iterations is allowed. In particular, a Lipschitz constant estimate is computed locally at each iterate $k \geq 1$ in terms of an average of the $k-1$ local estimates of the Lipschitz constant estimated in the previous iterations. The proposed strategy is shown to guarantee acceleration and to outperform the standard Armijo-type backtracking in several numerical examples.

In the case of strongly convex objective functionals, improved linear convergence rates are expected. Recalling the composite problem (1.2), the case of strongly convex components f has firstly been considered for projected gradient descent in [20] and recently generalised in the monograph by Chambolle and Pock [11] where a general variant of FISTA (which we will denote in this work by GFISTA) allowing both f and g to be strongly convex has been studied. For that, linear convergence rates have rigorously been shown, encompassing the ones holding for standard FISTA in the non-strongly convex case. For its practical application, GFISTA requires an estimate of the Lipschitz constant of ∇f, which paves the way for the design of robust and fast backtracking strategies similar to the ones described above. We address this problem in this work.

Remark. Florea and Vorobyov studied n their very recent preprint[15] similar problems to the ones described in this work, as an extension of their previous work [14]. Although the convergence result obtained by the authors (compare [15, Theorem 2, Section 3.1]) is exactly the same as the one presented in our work (see Theorem 4.10), their analysis is completely different from the one we use here. In particular, to show the main convergence result, the authors introduce the idea of *generalised estimate sequences*. Starting from the original paper by Nesterov [19], the use of estimate sequences has indeed become very popular in the field of optimisation (see, e.g., [17, 18, 24], just to mention a few) due to its easy geometrical interpretation. However, the use of this technique leaves the technical difficulties related to the study of the decay speed of the convergence factors somehow hidden. Inspired by the classical results studied by Nesterov in [20] and by Beck and Teboulle in [4], we follow here a different path, defining appropriate decay factors and extrapolation rules along the iterations which, eventually, will result in an accelerated (linear) convergence rate.

Contribution. In this work we propose a full backtracking strategy for the general strongly convex version of FISTA (GFISTA) proposed in [11]. Differently from the standard backtracking rule proposed in the original paper by Beck and Teboulle [4] and based essentially on an Armijo line searching [1], the proposed strategy allows for both increasing and decreasing of the Lipschitz constant estimate, i.e. for both decreasing and increasing of the gradient descent step size, respectively. A similar backtracking strategy has been proposed by Scheinberg, Godfarb and Bai in [26] for the non-strongly convex case, but its generalisation to the strongly convex case is not straightforward. We address this gap in this work, presenting a unified framework where the standard FISTA algorithm (with and without backtracking) can be derived as a particular case. In the case of strongly convex objectives, we proved linear convergence studying in detail the decay speed of the convergence factors. We validate our theoretical results on some imaging denoising problems with strongly convex objective functions.

Organisation of the paper. In Section 2 we recall some definitions and standard assumptions for composite optimisation. In Section 3 we present the GFISTA strongly convex variant of FISTA proposed in [11] and recall the main convergence result proved therein. Next, in Section 4 we propose a backtracking strategy for GFISTA and prove the accelerate convergence results by means of several technical tools à la Nesterov (see, e.g., [20]). Numerical examples confirming our theoretical results are reported in Section 5. In the final Section 6 we summarise the main results of this work and picture some challenging question to be addressed in future work.

2. Preliminaries and notation

We are interested in the solution of the composite minimisation problem

$$\min_{x\in\mathcal{X}} \ \{F(x) = f(x) + g(x)\}\,, \tag{2.1}$$

where $\mathcal{X}$ is a (possibly infinite-dimensional) Hilbert space endowed with norm $\|\cdot\| = \langle\cdot,\cdot\rangle^{1/2}$ and $F : \mathcal{X} \to \mathbb{R}\cup\{+\infty\}$ is a convex, l.s.c. and proper functional to minimise. We denote by $x^* \in \mathcal{X}$ a minimiser of F. We assume that F can be decomposed into the sum of a differentiable convex function $f : \mathcal{X} \to \mathbb{R}$ with Lipschitz gradient and a non-smooth, convex and l.s.c. function $g : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$. We will denote the Lipschitz constant of ∇f by $L_f > 0$ so that

$$\|\nabla f(y) - \nabla f(x)\| \leq L_f\|y - x\|, \qquad \text{for any } x, y \in \mathcal{X}.$$

The strong convexity parameters of f will be denoted by $\mu_f \geq 0$ so that for any $t \in [0, 1]$, by definition, there holds

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) - \frac{\mu_f}{2}t(1-t)\|x-y\|^2, \quad \text{for any } x, y \in \mathcal{X}.$$

Similarly, by $\mu_g \geq 0$ we will denote the strong convexity parameter of g. The strong convexity parameter of the functional F will be then the sum $\mu = \mu_f + \mu_g$.

In this work we are particularly interested in the case when at least one of the two parameters μ_f and μ_g is strictly positive, so that $\mu > 0$.

The case $\mu = 0$ reduces (2.1) to the classical FISTA-type optimisation problem. In the case of projected gradient descent, i.e.

$$\min_{x\in\mathcal{B}\subset\mathcal{X}} \ f(x),$$

the case $\mu_f > 0$ has been studied by Nesterov in [20] and can be formulated as a composite problem of the form (2.1) with g being the indicator function of the subset $\mathcal{B}$ (for which $\mu_g = 0$)

as:

$$\min_{x\in\mathcal{X}} f(x)+\delta_{\mathcal{B}}(x), \quad \text{with} \quad \delta_{\mathcal{B}}=\begin{cases}0, & \text{if } x\in\mathcal{B}\\ +\infty, & \text{if } x\notin\mathcal{B}.\end{cases}$$

Our analysis covers both these scenarios as special cases, but it allows also the more general scenario of strongly convex components g.

For the application of the FB optimisation, a standard descent step in the differentiable component f is combined with an implicit gradient descent step for g. For any $\tau>0$ and for $\bar{x}\in\mathcal{X}$ we then introduce the corresponding FB operator $T_\tau:\mathcal{X}\to\mathcal{X}$:

$$\bar{x}\mapsto\hat{x}=T_\tau\bar{x}:=\operatorname{prox}_{\tau g}\left(\bar{x}-\tau\nabla f(\bar{x})\right), \tag{2.2}$$

where $\operatorname{prox}_{\tau g}$ is the proximal mapping operator defined by:

$$\operatorname{prox}_{\tau g}(z):=\arg\min_{y\in\mathcal{X}}\left(g(y)+\frac{1}{2\tau}\|z-y\|^2\right), \quad z\in\mathcal{X}.$$

3. A General Fast Iterative Shrinkage/Thresholding Algorithm

The original FISTA algorithm proposed in [4] is a very popular optimisation strategy to minimise a composite functionals F like (2.1) with rate of convergence $O(1/k^2)$. Originally proposed by Nesterov in [20] in the case of smooth constrained minimisation, FISTA extends Nesterov's approach for more general non-smooth functions g. A more general extension of the FISTA algorithm encoding also the strongly-convex case $\mu>0$ has been studied in [11]. Will refer to this extension in the following as GFISTA.

For the sake of conciseness, we report in Algorithm 1 both the FISTA and GFISTA algorithms followed by the convergence result [11, Theorem B.10]. Its proof is rather technical and can be found in [11, Appendix B]: the key idea consists in defining precise decay factors in the classical descent rule for F holding for every $x\in\mathcal{X}$ and for $\hat{x}=T_\tau\bar{x}$, with $\bar{x}\in\mathcal{X}$:

$$F(\hat{x})+(1+\tau\mu_g)\frac{\|x-\hat{x}\|^2}{2\tau}\leq F(x)+(1-\tau\mu_f)\frac{\|x-\bar{x}\|^2}{2\tau}, \quad \tau>0. \tag{3.1}$$

Inequality (3.1) is classically used as a starting point to study precisely convergence rates. The proof of such inequality is a trivial consequence of a general property holding for strongly convex functions. We report its proof in Lemma 6.2 in the Appendix.

The general technique to perform a convergence analysis consists in taking as element $x\in\mathcal{X}$ the convex combination of the k-th iterate of the algorithm x_k and a generic point (such as x^*) and, by means of (strong) convexity assumptions, in defining an appropriate decay factor by which a recurrence relation for the algorithm starting from the initial guess x_0 can be derived. To show acceleration, a detailed study of such factor needs to be done by means of technical properties of the iterates of the algorithm and of its extrapolation parameters. We refer to the work of Nesterov [20] for a review of these techniques applied to more standard cases and to [11] to an exhaustive survey of their applications in the context of Imaging.

The result reported in Theorem 3.1 generalises the ones holding for FISTA and studied in [20, 4]. In particular, the standard FISTA convergence rate of $O(1/k^2)$ proved in [4, Theorem 4.4] in the non-strongly convex case ($\mu=q=0$ and $t_0=0$) turns out to be a particular case, while improved linear convergence is shown whenever the composite functional F is μ-strongly convex ($\mu>0$) and an estimate on the Lipschitz constant L_f is available. We refer the reader to [21, 22, 29] for similar results proved for variants of FISTA.

Algorithm 1 FISTA and GFISTA (no backtracking)

Input: $0 < \tau \leq 1/L_f, x^0 = x^{-1} \in \mathcal{X}, q = \tau\mu/(1+\tau\mu_g) \in [0,1)$ and $t_0 \in \mathbb{R}$ s.t. $0 \leq t_0 \leq 1/\sqrt{q}$.

for $k \geq 0$ **do**

$$y^k = x^k + \beta_k(x^k - x^{k-1}) \tag{3.2}$$

$$x^{k+1} = T_\tau y^k = \mathrm{prox}_{\tau g}(y^k - \tau\nabla f(y^k)) \tag{3.3}$$

where:

if $\mu = 0$ **then**

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2} \tag{3.4}$$

$$\beta_k = \frac{t_k - 1}{t_{k+1}}$$

else if $\mu > 0$ **then**

$$t_{k+1} = \frac{1 - qt_k^2 + \sqrt{(1-qt_k^2)^2 + 4t_k^2}}{2} \tag{3.5}$$

$$\beta_k = \frac{t_k - 1}{t_{k+1}} \frac{1+\tau\mu_g - t_{k+1}\tau\mu}{1-\tau\mu_f}$$

end if

end for

Theorem 3.1 (Theorem B.1 [11]). *Let $\tau > 0$ with $\tau \leq 1/L_f$ and let $q := \frac{\mu\tau}{1+\tau\mu_g}$. If $\sqrt{q}t_0 \leq 1$ with $t_0 \geq 0$, then the sequence (x^k) produced by the Algorithm 1 in* (3.3) *in both cases* (3.4) *and* (3.5) *satisfies*

$$F(x^k) - F(x^*) \leq r_k(q)\left(t_0^2(F(x^0) - F(x^*)) + \frac{1+\tau\mu_g}{2}\|x - x^*\|^2\right), \tag{3.6}$$

where x^* is a minimiser of F and:

$$r_k(q) = \min\left\{\frac{4}{(k+1)^2}, (1+\sqrt{q})(1-\sqrt{q})^k, \frac{(1-\sqrt{q})^k}{t_0^2}\right\}. \tag{3.7}$$

Backtracking. Whenever an estimate of L_f is not available, backtracking techniques can be used. For FISTA, an Armijo-type backtracking rule has been proposed in the original paper of Beck and Teboulle [4]. For that, similar convergence rates as above can be proved. Furthermore, in order to improve the speed of the algorithm allowing also the increasing of the step size τ in the neighbourhoods of 'flat' points of the functional F (i.e. where L_f is small), a full backtracking strategy for FISTA has been considered by Scheinberg, Goldfarb and Bai in [26].

Typically, the key inequality to check when designing any backtracking strategy can be derived similarly as (3.1) (see Lemma 6.2 in the Appendix) and reads:

$$F(\hat{x}) + (1+\tau\mu_g)\frac{\|x-\hat{x}\|^2}{2\tau} \leq F(x) + (1-\tau\mu_f)\frac{\|x-\bar{x}\|^2}{2\tau} + \left(\frac{\|\hat{x}-\bar{x}\|^2}{2} - D_f(\hat{x},\bar{x})\right), \tag{3.8}$$

where $D_f(\hat{x}, \bar{x}) := f(\hat{x}) - f(\bar{x}) - \langle \nabla f(\bar{x}), \bar{x} - \hat{x} \rangle$ is the Bregman distance of f between $\hat{x}$ and $\bar{x}$. Note that in the case when no backtracking is performed, condition (3.8) is satisfied as long as:

$$D_f(\hat{x}, \bar{x}) \leq \frac{\|\hat{x} - \bar{x}\|^2}{2\tau}, \tag{CB}$$

which is clearly true for constant τ whenever $\tau \leq 1/L_f$ and L_f known. However, by letting τ vary, one can alternatively check condition (3.8) along the iterations of the algorithm and redefine τ_k at each iteration $k \geq 1$ in order to compute a local Lipschitz constant estimate.

In the following, we will indeed use this rule for the design of a backtracking strategy for Algorithm 1 with $\mu > 0$. In order to allow robust backtracking, we will allow the step size τ_k to either decrease (as it is classically done) or increase depending on the validity of the following inequality:

$$\frac{2D_f(\hat{x}, \bar{x})}{\|\hat{x} - \bar{x}\|^2} > C_{bt} \left(\frac{1}{\tau_k} \right), \tag{CB2}$$

where the constant $C_{bt} \in (0, 1)$ has been chosen in advance. Heuristically, in our backtracking the step size τ_k will be decreased at iteration $k \geq 1$ whenever the estimate of the Lipschitz constant given by the left hand side in the inequality above is 'too close' to $1/\tau_k$, i.e. whenever (CB2) is verified, and increased otherwise.

4. A backtracking strategy for GFISTA algorithm 1

Following the approach proposed in [11, Section 4, Appendix B], we prove that a backtracking strategy applied to the GFISTA algorithm 1 enjoys accelerated convergence rates.

For an arbitrary $t \geq 1$, $k \geq 0$ and $\tau > 0$ we start from inequality (3.8) and set the point x to be the convex combination of an iterate x^k of the algorithm we are going to define and x^*, that is $x = ((t-1)x^k + x^*)/t$, while for the other points we set $\bar{x} = y^k$ and $\hat{x} = x^{k+1} = T_\tau y^k$. The formula for y^k will be specified in the following.

After multiplication by t^2 and using the convexity of F we get:

$$\begin{aligned} t^2 \left(F(x^{k+1}) - F(x^*) \right) + \frac{1 + \tau\mu_g}{2\tau} \|x^* - x^{k+1} - (t-1)(x^{k+1} - x^k)\|^2 \\ + \; t^2(t-1) \frac{\tau\mu(1 - \tau\mu_f)}{1 + \tau\mu_g - t\mu\tau} \frac{\|x^k - y^k\|^2}{2\tau} \leq t(t-1) \left(F(x^k) - F(x^*) \right) \\ + \; \frac{1 + \tau\mu_g - t\mu\tau}{2\tau} \|x^* - x^k - t \frac{1 - \tau\mu_f}{1 + \tau\mu_g - t\mu\tau} (y^k - x^k)\|^2. \end{aligned} \tag{4.1}$$

We now set $t = t_{k+1}$ in the inequality above and define the following quantities:

$$q_{k+1} := \frac{\tau\mu}{1 + \tau\mu_g} = 1 - \frac{1 - \tau\mu_f}{1 + \tau\mu_g} \in [0, 1), \tag{4.2}$$

$$\omega_{k+1} := \frac{1 - q_{k+1} t_{k+1}}{1 - q_{k+1}} = \frac{1 + \tau\mu_g - t_{k+1}\tau\mu}{1 - \tau\mu_f}, \tag{4.3}$$

$$\beta_k := \omega_{k+1} \frac{t_k - 1}{t_{k+1}}, \tag{4.4}$$

where the dependence on $k+1$ of the term in (4.2) will be clarified in the following and where we can assume $\mu_f < L_f$, so that $\tau < 1/L_f$. If this is not satisfied, then the function f is quadratic and the optimisation problem becomes trivial.

If the two conditions:

$$0 < \omega_{k+1} \leq 1, \tag{C1}$$

$$y^k = x^k + \beta_k(x^k - x^{k-1}), \tag{C2}$$

are satisfied for any $k \geq 0$, using them in (4.1) we note that after neglecting some positive terms on the left hand side, the following inequality can be derived:

$$\begin{aligned} t_{k+1}^2 \left(F(x^{k+1}) - F(x^*)\right) + \frac{1-\tau\mu_f}{2\tau}\|x^* - x^{k+1} - (t_{k+1}-1)(x^{k+1}-x^k)\|^2 \\ \leq t_{k+1}(t_{k+1}-1)\left(F(x^k) - F(x^*)\right) + \frac{\omega_{k+1}(1-\tau\mu_f)}{2\tau}\|x^* - x^k - (t_k-1)(x^k - x^{k-1})\|^2. \end{aligned} \tag{4.5}$$

Defining now the following quantity:

$$\tau' := \frac{\tau}{1-\tau\mu_f} > 0, \tag{4.6}$$

and by multiplying the inequality (4.1) by τ', we get:

$$\begin{aligned} \tau' t_{k+1}^2 \left(F(x^{k+1}) - F(x^*)\right) + \frac{1}{2}\|x^* - x^{k+1} - (t_{k+1}-1)(x^{k+1}-x^k)\|^2 \\ \leq \tau' t_{k+1}(t_{k+1}-1)\left(F(x^k) - F(x^*)\right) \\ + \frac{\omega_{k+1}}{2}\|x^* - x^k - (t_k-1)(x^k - x^{k-1})\|^2. \end{aligned} \tag{4.7}$$

We now want to perform backtracking and let the step size τ vary along the iterations; we then set $\tau = \tau_{k+1}$ in (4.6), so that

$$\tau' = \tau'_{k+1} = \frac{\tau_{k+1}}{1-\tau_{k+1}\mu_f}. \tag{4.8}$$

Note that the dependence on $k+1$ in definition (4.2) is now clear since the sequence (q_k) is also let vary along the iterations. The elements of such sequence will play the role of local inverse condition numbers in the following.

If for every $k \geq 0$ the following inequality holds:

$$\tau'_{k+1} t_{k+1}(t_{k+1}-1) \leq \omega_{k+1}\tau'_k t_k^2, \tag{C3}$$

then we can easily get from (4.7):

$$\begin{aligned} \tau'_{k+1} t_{k+1}^2 \left(F(x^{k+1}) - F(x^*)\right) + \frac{1}{2}\|x^* - x^{k+1} - (t_{k+1}-1)(x^{k+1}-x^k)\|^2 \\ \leq \omega_{k+1}\left(\tau'_k t_k^2\left(F(x^k) - F(x^*)\right) + \frac{1}{2}\|x^* - x^k - (t_k-1)(x^k - x^{k-1})\|^2\right). \end{aligned} \tag{4.9}$$

Applying (4.9) recursively, we finally find the following convergence inequality

$$F(x^k) - F(x^*) \leq \theta_k \left(\tau'_0 t_0^2\left(F(x^0) - F(x)\right) + \frac{1}{2}\|x^* - x^0\|^2\right)., \tag{4.10}$$

where the factor

$$\theta_k := \frac{\prod_{i=1}^{k} \omega_i}{\tau'_k t_k^2} \tag{4.11}$$

needs to be studied to determine the speed of convergence of $F(x^k)$ to the optimal value $F(x^*)$. We will do this in the following sections using some technical properties of the sequences defined above.

4.1. **Update rule.** From (C3), we impose the following update rule for the elements of sequence (t_k):

$$\tau'_{k+1} t_{k+1}(t_{k+1} - 1) = \omega_{k+1} \tau'_k t_k^2, \tag{4.12}$$

which provides:

$$t_{k+1} = \frac{1 - \frac{q_{k+1}}{1-q_{k+1}} \frac{\tau'_k}{\tau'_{k+1}} t_k^2 + \sqrt{\left(\frac{q_{k+1}}{1-q_{k+1}} \frac{\tau'_k}{\tau'_{k+1}} t_k^2 - 1\right)^2 + 4 \frac{\tau'_k}{\tau'_{k+1}} \frac{t_k^2}{1-q_{k+1}}}}{2} \geq 0. \tag{4.13}$$

We can now present the GFISTA algorithm with backtracking.

Algorithm 2 GFISTA with backtracking

Input: μ_f, μ_g, $\tau_0 > 0$, $q_0 := \mu\tau_0/(1 + \tau_0\mu_g)$, $\rho \in (0, 1)$, $x^0 = x^{-1} \in \mathcal{X}$ and $t_0 \in \mathbb{R}$ s.t. $0 \leq t_0 \leq 1/\sqrt{q_0}$.

for $k \geq 0$ **do**

$$y^k = x^k + \beta_k(x^k - x^{k-1}).$$

Set $i_{bt} = 0$;

if (CB2) is **then**

while (CB) is **not** verified **and** $i_{bt} \leq i_{max}$ **do**

reduce step-size: $\tau_{k+1} = \rho^{i_{bt}} \tau_k$;

Compute

$$x^{k+1} = T_{\tau_{k+1}} y^k = \text{prox}_{\tau_{k+1} g}(y^k - \tau_{k+1} \nabla f(y^k)) \tag{4.14}$$

$i_{bt} = i_{bt} + 1$;

end while

else if not (CB2) **then**

increase step-size: $\tau_{k+1} = \frac{\tau_k}{\rho}$;

Compute x_{k+1} using (4.14);

end if

Set

$$\tau'_{k+1} = \frac{\tau_{k+1}}{1 - \tau_{k+1}\mu_f}, \qquad q_{k+1} = \frac{\mu\tau_{k+1}}{1 + \tau_{k+1}\mu_g}.$$

Compute t_{k+1} using the update rule (4.13).

Set

$$\beta_{k+1} = \frac{1 - q_{k+1}t_{k+1}}{1 - q_{k+1}} \frac{t_k - 1}{t_{k+1}}.$$

end for

Remark 4.1 (No backtracking)**.** *When no backtracking is performed along the iterations $\tau_k = \tau_{k+1}$ for any $k \geq 0$ and the ratio τ'_k/τ'_{k+1} in* (4.13) *is constantly equal to one. In this case, the update rule* (4.13) *resembles the one used in* (3.5) *for GFISTA without backtracking, although it slightly differs because of the choice of the sequence ω_k having elements given by* (4.3)*. A different choice for such elements consistent with* [11, Appendix B] *from which* (3.5) *could be derived as a particular case would be*

$$\omega_{k+1} = 1 - q_{k+1}t_{k+1}. \tag{4.15}$$

However, under this choice only suboptimal convergence rate results can be proved if backtracking is performed, as preliminary calculations showed, see Remark 4.11.

Note that in the non-strongly convex case ($q_k = 0$ for every k) with no backtracking, the update rule (4.13) *is exactly the same* (3.4) *for the original FISTA algorithm* [20, 4]*.*

Remark 4.2 (FISTA with backtracking)**.** *In the non-trongly convex case,* (4.13) *reduces to*

$$t_{k+1} = \frac{1 + \sqrt{1 + 4\frac{\tau_k}{\tau_{k+1}}t_k^2}}{2},$$

which is exactly the same update rule considered by Goldfarb et al. in [26] *for fast backtracking of the classical FISTA algorithm.*

While condition (C2) can be guaranteed just by imposing the update of the sequence (y^k) depending on the parameter β_k in (4.4) (see Section 4.4 for monotone updates), in order to prove the strict decay in (C3), we need to make sure that (C1) is guaranteed for any $k \geq 0$. We do this using the following lemma on the sequence (t_k).

Lemma 4.3. *Let the sequence (t_k) be defined by the update rule* (4.13)*. Then:*

$$t_k \geq 1 \qquad \text{for any } k \geq 1.$$

Proof. By definition (4.2) $2 - q_k > 1 > q_k$. We can then estimate t_k from below as:

$$\begin{aligned} t_k =& \frac{1 - \frac{q_k}{1-q_k}\frac{\tau'_{k-1}}{\tau'_k}t_{k-1}^2 + \sqrt{\left(\frac{q_k}{1-q_k}\frac{\tau'_{k-1}}{\tau'_k}t_{k-1}^2 - 1\right)^2 + 4\frac{t_{k-1}^2}{1-q_k}\frac{\tau'_{k-1}}{\tau'_k}}}{2} \\ =& \frac{1 - \frac{q_k}{1-q_k}\frac{\tau'_{k-1}}{\tau'_k}t_{k-1}^2 + \sqrt{1 + 2(2-q_k)\frac{t_{k-1}^2}{1-q_k}\frac{\tau'_{k-1}}{\tau'_k} + \left(\frac{q_k}{1-q_k}\frac{\tau'_{k-1}}{\tau'_k}\right)^2 t_{k-1}^4}}{2} \\ \geq& \frac{1 - \frac{q_k}{1-q_k}\frac{\tau'_{k-1}}{\tau'_k}t_{k-1}^2 + \sqrt{\left(1 + \frac{q_k}{1-q_k}\frac{\tau'_{k-1}}{\tau'_k}t_{k-1}^2\right)^2}}{2} = 1. \end{aligned}$$

□

We can now verify the boundness condition (C1) of the sequence (ω_k).

Proposition 4.4. *For any $k \geq 1$, $\omega_k \in (0, 1]$, i.e. condition* (C1) *is verified.*

Proof. Let $k \geq 1$. We first check the condition

$$\omega_k = \frac{1 - q_k t_k}{1 - q_k} \leq 1$$

which clearly holds iff $q_k t_k \geq q_k$. In the case $q_k = 0$, we note that the equality is trivially verified, otherwise we can simply divide by q_k and and apply Lemma 4.3 to conclude.

We now need to check whether $\omega_k > 0$. Since $1 - q_k > 0$ by definition (4.2), we need to check only whether $q_k t_k < 1$. We proceed by contradiction and assume that $q_k t_k \geq 1$. By multiplying equality (4.12) by q_k, we have:

$$q_k \tau_k' t_k^2 = q_k \tau_k' t_k + (1 - q_k t_k) \frac{q_k}{1 - q_k} \tau_{k-1}' t_{k-1}^2.$$

Since the second term on the right hand side is nonpositive by assumption, we infer

$$q_k \tau_k' t_k^2 \leq q_k \tau_k' t_k,$$

which implies

$$q_k t_k \leq q_k < 1,$$

by definition of q_k (4.2), which is a contradiction. Therefore, $q_k t_k < 1$ and $\omega_k > 0$. □

For the following convergence proofs, the following technical lemma will be crucial.

Lemma 4.5. *Let $\sqrt{q_0} t_0 \leq 1$. Then, there holds:*

$$\sqrt{q_k} t_k \leq 1. \tag{4.16}$$

Proof. We start noticing that by definitions (4.2) and (4.8), we have that for any $k \geq 1$

$$\mu \tau_k' = q_k \frac{1 + \tau_k \mu_g}{1 - \tau_k \mu_f},$$

whence:

$$\frac{q_k}{1 - q_k} t_k^2 = \frac{q_{k+1}}{1 - q_{k+1}} \frac{\tau_k'}{\tau_{k+1}'} t_k^2. \tag{4.17}$$

We proceed by induction. By assumption, the initial step $k = 0$ holds. Let us assume that (4.16) holds for some $k \geq 1$. We multiply (4.12) by q_{k+1} and get:

$$q_{k+1} t_{k+1}^2 = q_{k+1} t_{k+1} + (1 - q_{k+1} t_{k+1}) \frac{q_{k+1}}{1 - q_{k+1}} \frac{\tau_k'}{\tau_{k+1}'} t_k^2.$$

Proceeding by contradiction as in Lemma 4.4, we can show that $0 \leq q_{k+1} t_{k+1} < 1$.

Now, using (4.17) and induction we have:

$$\begin{aligned}
q_{k+1} t_{k+1} &= q_{k+1} t_{k+1} + (1 - q_{k+1} t_{k+1}) \frac{q_{k+1}}{1 - q_{k+1}} \frac{\tau_k'}{\tau_{k+1}'} t_k^2 \\
&= q_{k+1} t_{k+1} + (1 - q_{k+1} t_{k+1}) \frac{1}{1 - q_k} q_k t_k^2 \\
&\leq \left(1 - \frac{1}{1 - q_k}\right) q_{k+1} t_{k+1} + \frac{1}{1 - q_k} \\
&< 1 - \frac{1}{1 - q_k} + \frac{1}{1 - q_k} = 1.
\end{aligned}$$

□

4.2. **Upper bounds.** In order to study the convergence rate in (4.9), we need to study the factor θ_k defined in (4.11). In the spirit of Nesterov's [20, Lemma 2.2.4], we do so using an induction argument to provide estimates bounding such term by a $O(1/k^2)$ factor.

Before doing so, we first notice that from the equality (4.12), for every $k \geq 0$ we can infer:

$$\left(\sqrt{\tau'_{k+1}}t_{k+1} - \frac{\sqrt{\tau'_{k+1}}}{2}\right)^2 = \omega_{k+1}\tau'_k t_k^2 + \frac{\tau'_{k+1}}{4} \;\geq\; \omega_{k+1}\tau'_k t_k^2,$$

whence:

$$\sqrt{\tau'_{k+1}}t_{k+1} \geq \frac{\sqrt{\tau'_{k+1}}}{2} + \sqrt{\omega_{k+1}\tau'_k}t_k. \tag{4.18}$$

Starting from this inequality we can show the following lemma.

Lemma 4.6. *Let (τ'_k, t_k) a sequence satisfying (4.12). Then, for every $k \geq 1$, there holds:*

$$\tau'_k t_k^2 \geq \left(\sum_{i=1}^{k-1} a_i^{(k-1)} \frac{\sqrt{\tau'_i}}{2} + \frac{\sqrt{\tau'_k}}{2}\right)^2, \tag{4.19}$$

where, for every $i = 1, \ldots, k-1$:

$$a_i^{(k-1)} := \prod_{j=i}^{k-1} \sqrt{\omega_{j+1}}. \tag{4.20}$$

Remark 4.7. *Note that for $i = k-1$*

$$a_{k-1}^{(k-1)} = \sqrt{\omega_k},$$

and the sum in (4.19) is empty for $k = 1$.

Proof. We proceed by induction. The initial case $k = 1$ is trivial since, by Remark 4.7 and by the fact that $t_1 \geq 1$ by Lemma 4.3 we have:

$$\tau'_1 t_1^2 \geq \tau'_1 \geq \frac{\tau'_1}{4}.$$

Let us now assume that (4.19)- (4.20) hold for some $k > 1$. We have:

$$\begin{aligned}
\sqrt{\tau'_{k+1}}t_{k+1} &\geq \frac{\sqrt{\tau'_{k+1}}}{2} + \sqrt{\omega_{k+1}}\sqrt{\tau'_k}t_k \geq \frac{\sqrt{\tau'_{k+1}}}{2} + \sqrt{\omega_{k+1}}\left(\sum_{i=1}^{k-1} a_i^{(k-1)} \frac{\sqrt{\tau'_i}}{2} + \frac{\sqrt{\tau'_k}}{2}\right) \\
&= \frac{\sqrt{\tau'_{k+1}}}{2} + \left(\sum_{i=1}^{k-1} \sqrt{\omega_{k+1}} a_i^{(k-1)} \frac{\sqrt{\tau'_i}}{2} + \sqrt{\omega_{k+1}}\frac{\sqrt{\tau'_k}}{2}\right) = \\
&= \frac{\sqrt{\tau'_{k+1}}}{2} + \sum_{i=1}^{k} a_i^{(k)} \frac{\sqrt{\tau'_i}}{2},
\end{aligned}$$

by (4.18) and since for any $i = 1, \ldots, k$:

$$\sqrt{\omega_{k+1}} a_i^{(k-1)} = \prod_{j=i}^{k} \sqrt{\omega_{j+1}} = a_i^{(k)}.$$

□

We now provide a lower bound on the sum of coefficients appearing in (4.19).

Lemma 4.8. *In the condition of Lemma 4.6, for every $k \geq 1$ there holds:*

$$\sum_{i=1}^{k} a_i^{(k)} \geq \left(\prod_{i=1}^{k+1} \sqrt{\omega_i} \right) k. \tag{4.21}$$

Proof. Since by Lemma 4.4 $0 < \omega_i \leq 1$ for any $i = 1, \ldots, k+1$, we have that $0 < \sqrt{\omega_i} \leq 1$. Therefore, we have:

$$\begin{aligned}
\sum_{i=1}^{k} a_i^{(k)} &= \sum_{i=1}^{k} \prod_{j=i}^{k} \sqrt{\omega_{j+1}} = \sqrt{\omega_2 \ldots \omega_{k+1}} + \sqrt{\omega_3 \ldots \omega_{k+1}} + \ldots + \sqrt{\omega_{k+1}} \\
&= \sqrt{\omega_2 \ldots \omega_{k+1}} \left(1 + \frac{1}{\sqrt{\omega_2}} + \ldots + \frac{1}{\sqrt{\omega_2 \ldots \omega_{k+1}}} \right) \\
&\geq \sqrt{\omega_2 \ldots \omega_{k+1}} \underbrace{(1 + 1 + \ldots + 1)}_{k \text{ times}} \geq a_1^{(k)}\ k \\
&\geq (\sqrt{\omega_1 \ldots \omega_{k+1}})\, k = \left(\prod_{i=1}^{k+1} \sqrt{\omega_i} \right) k.
\end{aligned}$$

□

We conclude this section with another Lemma which will be used to get the $O(1/k^2)$ factor.

Lemma 4.9. *For every $k \geq 1$ the following inequality holds:*

$$\frac{\prod_{i=1}^{k+1} \omega_i}{\left(1 + k \prod_{i=1}^{k+1} \sqrt{\omega_i} \right)^2} \ \leq\ \frac{1}{(1 + (k+1))^2}. \tag{4.22}$$

Proof. Let us fix $k \geq 1$. We define $\gamma_{k+1} := \prod_{i=1}^{k+1} \sqrt{\omega_i}$ so that $\gamma_{k+1}^2 = \prod_{i=1}^{k+1} \omega_i$. We want to show:

$$\frac{\gamma_{k+1}^2}{(1 + k\gamma_{k+1})^2} \ \leq\ \frac{1}{(1 + (k+1))^2}, \tag{4.23}$$

which is true if and only if:

$$4(k+1)\gamma_{k+1}^2 - 2k\gamma_{k+1} - 1 \leq 0$$

which is verified whenever γ_{k+1} is in the range:

$$-\frac{1}{k+1} \leq \gamma_{k+1} \leq 1. \tag{4.24}$$

Since by Lemma 4.4 $\gamma_{k+1} = \prod_{i=1}^{k+1} \sqrt{\omega_i} \in (0, 1]$ and $k \geq 1$ is arbitrary, we conclude that (4.24) is true, whence (4.22) holds. □

4.3. **Convergence rates.** In this section, we combine the technical results proved above to derive a precise estimate of the factor θ_k in (4.11).

We perform a *worst-case convergence* analysis. Denoting by $L_0 = \frac{1}{\tau_0}$ the initial Lipschitz constant estimate, we define the following two qualities:

$$L_w := \max\left\{\frac{L_f}{\rho}, \rho L_0\right\}, \qquad q_w := \frac{\mu}{L_w + \mu_g}, \tag{4.25}$$

where q_w has to be understood then as the worst-case inverse condition number along the iterates. Note that by definition :

$$q_k \geq q_w \qquad \text{for all } k \geq 1. \tag{4.26}$$

The following convergence result shows that the proposed backtracking strategy applied to the GFISTA algorithm guarantees accelerated convergence rates, which in the case of strong convexity objectives are in fact linear. Comments on our result in comparison to the ones studied in analogous cases are given in the following remarks.

Theorem 4.10 (Convergence rates)**.** *Let $x_0 \in \mathcal{X}$, $\tau_0 > 0$ and let (x_k) the sequence produced by the GFISTA with backtracking Algorithm 2. If $t_0 \geq 0$ and $\sqrt{q_0}t_0 \leq 1$, we have:*

$$F(x^k) - F(x^*) \leq r_k\left(\rho L_f - \mu_f\right)\left(\frac{\tau_0 t_0^2}{1-\mu_f\tau_0}\left(F(x^0) - F(x^*)\right) + \frac{1}{2}\|x^0 - x^*\|^2\right) \tag{4.27}$$

where the decay rate is defined as:

$$r_k := \min\left\{\frac{4}{(k+1)^2}, (1-\sqrt{q_w})^{k-1}, \frac{(1-\sqrt{q_w})^k}{t_0^2}\right\}. \tag{4.28}$$

Proof. We start observing that for $0 < \rho < 1$, we have that for every $i \geq 0$:

$$\rho L_f \geq \frac{1}{\tau_i} = \frac{1+\tau_i'\mu_f}{\tau_i'} \quad \Longrightarrow \quad \frac{1}{\tau_i'} \leq \rho L_f - \mu_f, \tag{4.29}$$

which we will use to bound every $1/\tau_i'$. Therefore, by applying Proposition 4.6 and Lemma 4.8 we can derive the first $O(1/k^2)$ factor in the definition of r_k given in (4.28) as follows:

$$\begin{aligned}\frac{\prod_{i=1}^k \omega_i}{\tau_k' t_k^2} &\leq 4(\rho L_f - \mu_f)\frac{\prod_{i=1}^k \omega_i}{\left(1 + (k-1)\prod_{i=1}^k \sqrt{\omega_i}\right)^2} \\ &\leq 4(\rho L_f - \mu_f)\,\frac{\prod_{i=1}^k \omega_i}{\left(1 + (k-1)\prod_{i=1}^k \sqrt{\omega_i}\right)^2} \leq \frac{4}{(k+1)^2}\left(\rho L_f - \mu_f\right).\end{aligned}$$

We now follow [11, Appendix B] and note that by the equality (4.12) we get:

$$1 - \frac{1}{t_k} = \omega_k \frac{\tau_{k-1}' t_{k-1}^2}{\tau_k' t_k^2},$$

by which:

$$\tau_0' t_0^2 \theta_k = \frac{\tau_0' t_0^2}{\tau_k' t_k^2} \prod_{i=1}^{k} \omega_i = \prod_{i=1}^{k} \omega_i \frac{\tau_{i-1}' t_{i-1}^2}{\tau_i' t_i^2} = \prod_{i=1}^{k} \left(1 - \frac{1}{t_i}\right) \leq \prod_{i=1}^{k} \left(1 - \sqrt{q_i}\right),$$

thanks to Lemma 4.5. We can now bound the term above using the worst-case inverse number defined in (4.25) and the bound (4.29), thus getting:

$$\theta_k \leq \frac{\left(1 - \sqrt{q_w}\right)^k}{\tau_0' t_0^2} \leq (\rho L_f - \mu_f) \frac{\left(1 - \sqrt{q_w}\right)^k}{t_0^2}, \tag{4.30}$$

whenever $t_0^2 \geq 1$.

Conversely, if $t_0^2 < 1$, proceeding similarly as before we get:

$$\theta_k = \frac{\omega_1}{\tau_k' t_k^2} \prod_{i=2}^{k} \omega_i < \frac{\omega_1}{\tau_1' t_1^2} \prod_{i=2}^{k} \left(1 - \sqrt{q_i}\right) \leq \frac{\omega_1}{\tau_1' t_1^2} \left(1 - \sqrt{q_w}\right)^{k-1}.$$

Since by Lemma 4.3 $t_1 \geq 1$, we can simply bound this term as:

$$\theta_k \leq \frac{\omega_1}{\tau_1' t_1^2} \left(1 - \sqrt{q_w}\right)^{k-1} \leq \frac{1}{\tau_1'} \left(1 - \sqrt{q_w}\right)^{k-1} \leq \left(1 - \sqrt{q_w}\right)^{k-1} (\rho L_f - \mu_f).$$

Combining all together we finally get the convergence decay (4.28) with rate r_k defined in (4.30).

□

Remark 4.11. *As mentioned in Remark 4.1, a different choice of the decaying factor sequence ω_k like* (4.15) *(see for analogy* [11, Appendix B]*) would result in a suboptimal convergence rate depending on the factor $(\rho L_f + \mu_g)$ rather then the one in* (4.27).

Remark 4.12 (FISTA with backtracking)**.** *Note that in the non-strongly convex case ($\mu = q_k = 0$ for all k) and under the choice $t_0 = 0$, the global convergence rate* (4.27)-(4.28) *is actually less accurate than the one derived in* [26, Theorem 3.3]*, which reads:*

$$F(x^k) - F(x^*) \leq \frac{2\rho L_k \|x^0 - x^*\|^2}{(k+1)^2}, \tag{4.31}$$

where the term L_f is replaced by the averaging term L_k defined by

$$\sqrt{L_k} := \frac{\sum_{i=1}^{k} \sqrt{L(f, g, y^i)}}{k},$$

with $L(f, g, y^i)$ being the local Lipschitz constant estimated in each interval $(T_{\tau_i} y^i, y^i)$, for $1 \leq i \leq k$. However, one can in fact improve the estimate (4.31) *using similar techniques as the ones we have described in the previous sections. Namely, since $\tau_k' = \tau_k$ in this case, starting from* (4.10) *and replacing the factor θ_k in* (4.11) *by:*

$$\theta_k = \frac{1}{\tau_k t_k^2},$$

since $\omega_i = 1$ for all $i \geq 1$, one can observe that such term has now the same expression of the analogous factor studied in [11, Appendix B] *up to multiplication of $1/\tau_k$. Hence, using then the basic estimate*

$$\frac{1}{\tau_k} \leq \rho L(f, g, y^k)$$

and via similar techniques to the ones used in this work one can find that the following convergence inequality holds

$$F(x^k) - F(x^*) \leq \frac{2\ \rho L(f, g, y^k)\|x^0 - x^*\|^2}{(k+1)^2},$$

which is more accurate than (4.31) *since it depends only on the Lipschitz constant estimated at iteration k and, as such, it allows for "more local" convergence verifications.*

Remark 4.13. *In* [15, Theorem 2] *the authors derive a convergence rate similar to* (4.27)-(4.30) *using arguments based on generalised estimate sequences. In* [15, Section 4] *some comments on the extrapolated form of their approach and its relation with the strongly-convex variant of the FISTA algorithm 1 are given. Although the expression of the sequence $\{\omega_k\}$ and the update rule for the elements $\{t_k\}$ is similar (but not equal) to our definitions* (4.3) *and* (4.13)*, respectively, the proof of the authors follows a completely different argument based on the used of generalised estimate sequence. Furthermore, note that in our proof the choice of t_0 is let free, whereas in* [15] *is somehow set to a specific value.*

4.4. **Monotone algorithms.** As already noticed for standard FISTA [3, Section V.A] and for GFISTA without backtracking [11, Remark B.3], the convergence of the composite energy F to the optimal value x^* is not guaranteed to be monotone, i.e. non-increasing. A straightforward modification of the GFISTA Algorithm 2 enforcing such property consists in changing the update rule (4.14) by taking as a point x^{k+1} any point such that $F(x^{k+1}) \leq F(T_{\tau_{k+1}} y^k)$. Recalling the definition of ω_{k+1} in (4.3), the update rule (C2) for extrapolation can then be changed as:

$$\begin{aligned} (\mathrm{C2}_m) \qquad y^k &= x^k + \beta_k \left(\left(x^k - x^{k-1}\right) + \omega_{k+1} \frac{t_k}{t_{k+1}} \left(T_{\tau_{k+1}} y^{k-1} - x^k\right) \right) \\ &= x^k + \beta_k \left(\left(x^k - x^{k-1}\right) + \frac{t_k}{t_k - 1} \left(T_{\tau_{k+1}} y^{k-1} - x^k\right) \right). \end{aligned}$$

This modification does not affect the results presented in up to now, since one can easily check that starting from (3.1) and setting $\hat{x} = T_{\tau_{k+1}} y^k$ the same computations of the previous sections can be performed, and the same convergence rates are obtained. Condition ($\mathrm{C2}_m$) suggests also an easy choice of x^{k+1} at each iteration $k \geq 1$. In fact, one can simply set:

$$x^{k+1} = \begin{cases} T_{\tau_{k+1}}(y^k) & \text{if } F(T_{\tau_{k+1}} y^k) \leq F(x^k), \\ x^k & \text{otherwise,} \end{cases} \tag{4.32}$$

so that in either case one of the two terms in ($\mathrm{C2}_m$) vanishes. Whenever the evaluation of the composite functional F is cheap, this choice seems to be the most sensible. Another monotone implementation of FISTA has been recently considered in [28] where despite the further computational costs required to compute the value x^{k+1}, an empirical linear convergence rate is observed also for standard FISTA applied to strongly convex objectives. A rigorous proof of such convergence property is an interesting question of future research.

5. Numerical examples

In this section we report some numerical experiments showing the effectiveness of the backtracking strategy proposed for two exemplar image denoising problems.

5.1. **TV-Huber ROF denoising.** We start considering a strongly convex variant of the well-know Rudin, Osher and Fatemi image denoising model [23] based on the use of Total Variation (TV) regularisation. In its discretised form and for a given noisy image $u^0 \in \mathbb{R}^{m\times n}$ corrupted with Gaussian noise with zero mean and variance σ^2, the original ROF model reads:

$$\min_u \ \lambda\|Du\|_{p,1} + \frac{1}{2}\|u-u^0\|_2^2. \tag{5.1}$$

Here, $Du = ((Du)_1, (Du)_2)$ is the discrete gradient operator discretised using forward finite differences (see, e.g., [7]) and the discrete TV regularisation is defined by:

$$\|Du\|_{p,1} = \sum_{i=1}^{m}\sum_{j=1}^{n} |(Du)_{i,j}|_p = \sum_{i=1}^{m}\sum_{j=1}^{n} \left((Du)_{i,j,1}^p + (Du)_{i,j,2}^p\right)^{1/p}, \tag{5.2}$$

where the value of the parameter p allows for both anisotropic ($p=1$) and isotropic ($p=2$) TV , which is generally preferred to reduce grid bias. The regularisation parameter $\lambda > 0$ in (5.1) weights the action of TV-regularisation against the fitting with the Gaussian data given by the ℓ^2 squared term.

Taking $p=2$ in (5.2), we now follow [11, Examples 4.7 and 4.14] and consider a similar denoising model where a strongly convex variant of the TV regularisation is used. This can be obtained, for instance, using the C^1-Huber smoothing function $h_\varepsilon : \mathbb{R} \to \mathbb{R}$ defined by:

$$h_\varepsilon(t) := \begin{cases} \frac{t^2}{2\varepsilon} & \text{for } |t| \le \varepsilon, \\ |t| - \frac{\varepsilon}{2} & \text{for } |t| > \varepsilon. \end{cases}$$

Applying such smoothing to the TV energy (5.2) removes the singularity in a neighbourhood zero by means of a quadratic term, while leaving the TV term almost unchanged otherwise. The resulting Huber-ROF image denoising model then reads:

$$\min_u \ \lambda H_\varepsilon(u) + \frac{1}{2}\|u-u^0\|_2^2, \tag{5.3}$$

with

$$H_\varepsilon(u) := \sum_{i=1}^{m}\sum_{j=1}^{n} h_\varepsilon\left(\sqrt{(Du)_{i,j,1}^2 + (Du)_{i,j,2}^2}\right). \tag{5.4}$$

The dual problem of (5.4) reads:

$$\min_{\boldsymbol{p}} \frac{1}{2}\|D^*\boldsymbol{p} - u^0\|_2^2 + \frac{\varepsilon}{2\lambda}\|\boldsymbol{p}\|_2^2 + \delta_{\{\|\cdot\|_{2,\infty}\le\lambda\}}(\boldsymbol{p}), \tag{5.5}$$

where $\boldsymbol{p}$ is the dual variable, D^* is the adjoint operator of D (i.e. the discretised negative finite-difference divergence operator) and $\delta_{\{\|\cdot\|_{2,\infty}\le\lambda\}}$ is the indicator function of a ball, i.e. it is defined as:

$$\delta_{\{\|\cdot\|_{2,\infty}\le\lambda\}}(\boldsymbol{p}) = \begin{cases} 0 & \text{if } |\boldsymbol{p}_{i,j}|_2 \le \lambda \text{ for any } i,j, \\ +\infty & \text{otherwise.} \end{cases}$$

Note that (5.5) is the sum of a function f with Lipschitz gradient and a non-smooth function g which are respectively given by:

$$f(\boldsymbol{p}) = \frac{1}{2}\|D^*\boldsymbol{p} - u^0\|_2^2, \qquad g(\boldsymbol{p}) = \frac{\varepsilon}{2\lambda}\|\boldsymbol{p}\|_2^2 + \delta_{\{\|\cdot\|_{2,\infty}\le\lambda\}}(\boldsymbol{p}).$$

Under this notation, the gradient of the differentiable component f reads:

$$\nabla f(\boldsymbol{p}) = D(D^*\boldsymbol{p} - u^0).$$

Furthermore, it is easy to show that its Lipschitz constant L_f can be estimated as $L_f \leq 8$, see, e.g. [7] and that $\mu_f = 0$.

The function g is strongly convex with parameter $\mu_g = \mu = \varepsilon/\lambda$ and its proximal map $\hat{\boldsymbol{p}} = \text{prox}_{\tau g}(\tilde{\boldsymbol{p}})$ can be easily computed pixel-wise as:

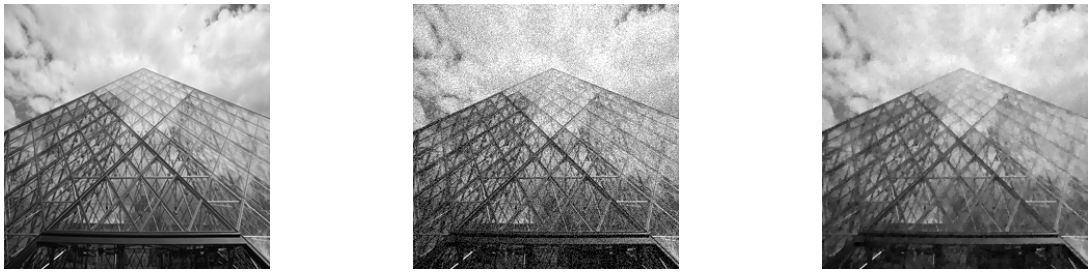
$$\hat{\boldsymbol{p}}_{i,j} = \frac{(1+\tau\mu_g)^{-1}\tilde{\boldsymbol{p}}_{i,j}}{\max\left\{1, (\lambda(1+\tau\mu_g))^{-1}|\tilde{\boldsymbol{p}}_{i,j}|_2\right\}}, \quad \text{for any } i,j,$$

since, due to general property of proximal maps with added squared ℓ^2 terms (see Lemma 6.4 in the Appendix), there holds:

$$\text{prox}_{\tau g}(\tilde{\boldsymbol{p}}) = \text{prox}_{\frac{\tau}{1+\tau\mu_g}\delta_{\left\{\|\cdot\|_{2,\infty}\leq\lambda\right\}}}\left(\frac{\tilde{\boldsymbol{p}}}{1+\tau\mu_g}\right) = \Pi_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}\left(\frac{\tilde{\boldsymbol{p}}}{1+\tau\mu_g}\right).$$

Note that the same example has also been considered for similar verifications in [14, Section 4.2]: our results are in good agreement with the ones reported therein.

Parameters. In the following experiments we consider an image $u^0 \in \mathbb{R}^{m\times n}$ with $m = n = 256$ corrupted with Gaussian noise with zero mean and $\sigma^2 = 0.005$, see Figure 1a-1b. We set the Huber parameter $\varepsilon = 0.01$ and the regularisation parameter $\lambda = 0.1$, so that $\mu_g = \mu = 0.1$. In our comparisons we use the GFISTA algorithms 1 and 2 with and without backtracking using the prior knowledge of L_f given by the estimate $L_f = 8$ and an initial L_0, respectively. To ensure monotone decay we use the modified version described in Section (4.4), i.e. we use the modified update rules (C2$_m$)-(4.32). For comparison, we show results where the backtracking strategy is used 'classically', i.e. it allows only for increasing of the Lipschitz constant estimate and used 'fully', i.e. it allows for both its increasing and decreasing along the iterations. The backtracking factor ρ is set $\rho = 0.9$ and used also as $1/\rho$ for the decreasing of the step size τ_k. The initial value t_0 is set $t_0 = 1$. The algorithm is initialised by the gradient of the noisy image u^0, i.e. $\boldsymbol{p_0} = Du^0$.

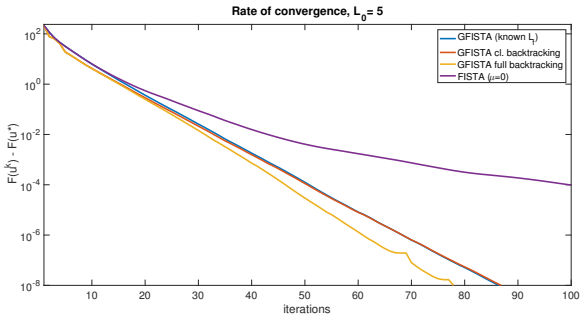


(A) Original image (B) Noisy version (C) Denoised version

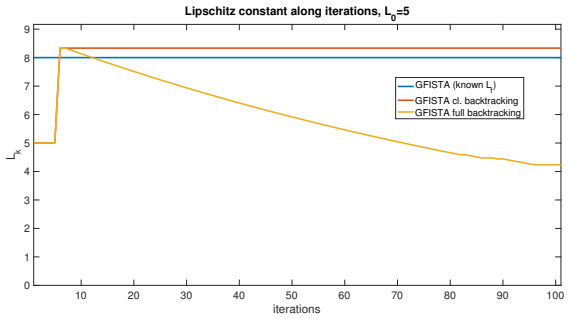
FIGURE 1. Original, noisy and TV-Huber denoised images used. Noise is Gaussian distributed with zero mean and variance $\sigma^2 = 0.005$. The regularisation parameter is $\lambda = 0.1$ and the Huber parameter is $\varepsilon = 0.01$ so that $\mu = 0.1$.

To compute an approximation of the optimal solution u^*, we let the plain GFISTA algorithm run beforehand for 5000 iterations and store the result for comparison, see Figure 1c. We then compute the results running the algorithms for `iter`= 100 iterations. We report the

results computed for two different choices of L_0 which are underestimating and overestimating the actual value of L_f, respectively, see Figure 2 and 3. For comparison, note the $O(1/k^2)$ convergence rate of standard FISTA with no strongly convex parameter ($\mu = 0$) compared to the linear one of the GFISTA variant.

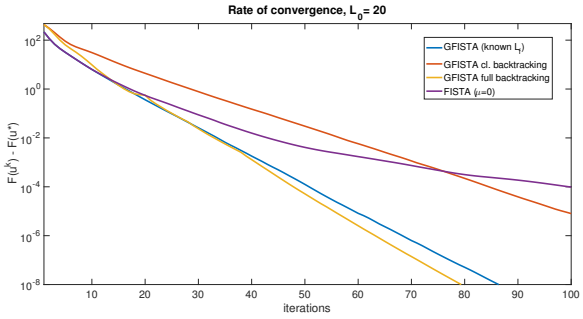


(A) Convergence rates.

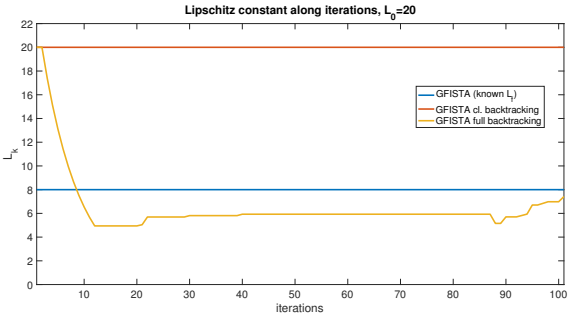


(B) Lipschitz constant estimate.

FIGURE 2. Convergence rates and backtracking of the Lipschitz constant of ∇f starting from the underestimating initial value $L_0 = 5$.



(A) Convergence rates.



(B) Lipschitz constant estimate.

FIGURE 3. Convergence rates and backtracking of the Lipschitz constant of ∇f starting from the overestimating initial value $L_0 = 20$.

5.2. **Strongly convex TV Poisson denoising.** In this second example we consider another denoising model for images corrupted with Poisson noise, which is commonly observed in microscopy and astronomy imaging applications. Standard Poisson denoising models using Total Variation regularisation are typically combined with a convex, non-differentiable Kullback-Leibler data fitting term, which can be consistently derived from the Bayesian formulation of the problem via MAP estimation (see, e.g., [25]). In the following, we follow [9] and consider a differentiable version of the Kullback-Leibler fidelity which, for a given positive noisy image $u^0 \in \mathbb{R}^{m\times n}$ corrupted with Poisson noise reads:

(5.6)

$$f(u) = \tilde{KL}(u_0, u) := \sum_{i=1}^{m}\sum_{j=1}^{n} \begin{cases} u_{i,j} + b_{i,j} - u^0_{i,j} + u^0_{i,j}\log\left(\frac{u^0_{i,j}}{u_{i,j}+b_{i,j}}\right) & \text{if } u_{i,j} \geq 0, \\ \frac{u^0_{i,j}}{2b^2_{i,j}}u^2_{i,j} + \left(1 - \frac{u^0_{i,j}}{b_{i,j}}\right)u_{i,j} + b_{i,j} - u^0_{i,j} + u^0_{i,j}\log\left(\frac{u^0_{i,j}}{b_{i,j}}\right) & \text{otherwise,} \end{cases}$$

where $b \in \mathbb{R}^{m\times n}$ stands for the background image which can be typically estimated from the data at hand. It is easy to verify that $\nabla \tilde{KL}(u_0, u)$ has a Lipschitz constant L_f which can be estimated as

$$L_f = \max_{i,j} \frac{u^0_{i,j}}{b^2_{i,j}}, \tag{5.7}$$

which it is well-defined, positive and finite as long as u^0 and b are positive. As a regularisation term, we will consider the following ε-strongly convex variant of isotropic TV in (5.2):

$$g(u) = \lambda \|Du\|_{2,1} + \frac{\varepsilon}{2}\|u\|_2^2, \tag{5.8}$$

where $\lambda > 0$ stands again for the regularisation parameter. Differently from the Huber-TV ROF example, we aim here to apply the GFISTA algorithm to solve the primal formulation of the composite problem:

$$\min_u \ \lambda \|Du\|_{2,1} + \frac{\varepsilon}{2}\|u\|_2^2 + \tilde{KL}(u_0, u). \tag{5.9}$$

The gradient of the KL term (5.6) can be easily computed and the proximal map of g in (5.8) can be computed using the proximal map of the TV functional due to a general property reported in Lemma 6.3 in the appendix, so that for any z:

$$\operatorname{prox}_{\tau g}(z) = \operatorname{prox}^{\frac{\lambda\tau}{1+\varepsilon\tau}}_{\|\cdot\|_{2,1}}\left(\frac{z}{1+\varepsilon\tau}\right). \tag{5.10}$$

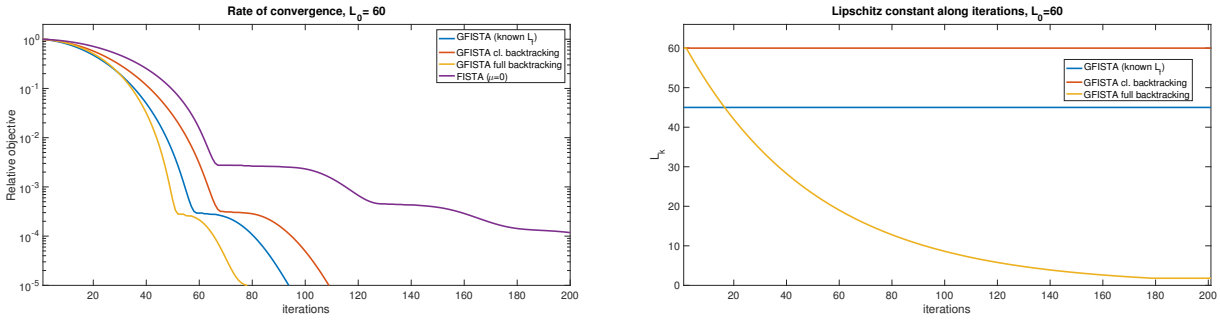
For any $\tau > 0$, computing the right hand side of the equality above corresponds simply to solve the classical ROF problem with regularisation parameter $\sigma := \frac{\lambda\tau}{1+\tau\varepsilon}$. We do that using standard FISTA [4] as an iterative inner solver.

Parameters. We consider an image $u^0 \in \mathbb{R}^{m\times n}$ with $m = n = 256$ corrupted artificially with Poisson noise, see Figure 4a-4b. For simplicity, we consider a constant background with $b_{i,j} = 1$ for all i, j. We set the strong convexity parameter $\varepsilon = 0.15$ and the regularisation parameter $\lambda = 0.1$. Clearly $\mu = \mu_g = \varepsilon$. In order to compute the proximal map (5.10) we use 10 iterations of standard FISTA. In the following example the Lipschitz constant of the gradient of the $\tilde{KL}$ term can be estimated via (5.7) as $L_f = 45$. We report in the following the results computed using the monotone variant of GFISTA algorithm 1 without backtracking and with classical and full backtracking (Algorithm 2 with monotone updates (C2$_m$)-(4.32)), for which the factor $\rho = 0.8$ is chosen. The initial value t_0 is set $t_0 = 1$. The algorithm is initialised using the given noisy image u^0.

An approximation of the solution u^* is computed beforehand by letting the plain GFISTA algorithm run for 5000 iterations and then stored for comparison, see Figure 4c. The results are computed letting the monotone version of the GFISTA algorithms run for `iter` $= 200$ iterations. In Figure 5 we report the results computed for a value of L_0 overestimating the actual one given by L_f and in comparison with standard FISTA with no strongly convex modification. Once again we can observe that by incorporating the strongly convex modification of GFISTA linear convergence is achieved, in comparison with slower convergence of standard FISTA. Furthermore, the local estimate of the Lipschitz constant provided by the full backtracking strategy proposed decreases along the iterations, thus allowing for larger gradient steps and convergence in fewer iterations. In Figure 6, we plot the monotone decay of the energy along the GFISTA iterates (with and without backtracking) after the monotone modification described in Section (4.4).

(A) Original (B) Noisy version (C) Denoised version

FIGURE 4. Original, noisy and restored image computed using the strongly convex TV-Poisson denoising model (5.9). The regularisation parameter is $\lambda = 0.2$ and the strong convexity parameter is $\mu = \varepsilon = 0.15$.



(A) Convergence rates. (B) Lipschitz constant estimate.

FIGURE 5. Convergence rates and backtracking of the Lipschitz constant of ∇f in (5.6) starting from the overestimating initial value $L_0 = 60$. Rates are shown in terms of the relative objective functional: $\frac{F(u^k)-F(u^*)}{F(u^0)-F(u^*)}$.

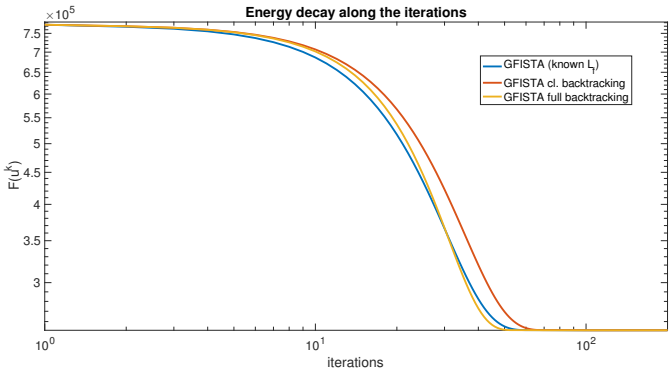


FIGURE 6. Monotone decay along the GFISTA iterates (with and without backtracking) after the monotone modification (C2$_m$)-(4.32).

6. Conclusions and outlook

We proposed a fast backtracking strategy for the strongly convex variant of FISTA proposed in [11] and based on a inequality condition expressed in terms of the Bregman distance, see Section 3. Using standard property of strongly convex functions and upon multiplication of appropriate terms, we have derived in Section 4 the convergence estimate (4.10) whose decay factor (4.11) has been then studied carefully to estimate the convergence speed of Algorithm 2. Our analysis is essentially based on classical technical tools similar to the ones used in Nesterov [20] and on general properties of the extrapolation sequences defined. Our main result is reported in Theorem 4.10 where accelerated $O(1/k^2)$ and worst-case $O((1-\sqrt{q_w})^k)$ linear convergence rates are proved, in agreement with the standard lower complexity bounds for strongly convex functions. Our theoretical results are verified numerically in Section 5 on some image denoising problems.

The backtracking strategy proposed is fast and robust since it allows for adaptive adjustment of the gradient step size (i.e. the proximal map parameter) depending on the local 'flatness' of the gradient of the component f in the objective functional, i.e. on the local estimate of L_f. In other words, in flat regions (small L_f) larger step sizes are promoted, whereas where large variations of ∇f occur (large L_f), small steps are preferred for a more accurate descent. From an algorithmic point of view, extrapolation is performed using suitable parameters providing strict decay in the convergence inequality (4.10) and defined not only in terms of the step sizes, but also in terms of the strong convexity parameters of f and g and resulting in more refined convergence rate estimates. Finally, compared to the standard FISTA, more freedom is left to the initial extrapolation parameter t_0, typically set to either zero or one.

Further research could address the the design of similar algorithms whenever the strong convexity parameters μ_f and μ_g are not known *a-priori*, in order to design a complete backtracking strategy dealing with composite functions with unknown convexity parameter. In this regard, Nesterov proposed in [22] a restarting scheme where an adaptive estimate of the strong convexity parameter of the objective functions is proposed. More recently, Ferocq and Qu have considered in [13] a similar strategy. However, the application of such ideas to GFISTA algorithm 2 is not straightforward and it is an interesting open problem to address in the future research.

Finally, we would like to test the effectiveness of the proposed algorithm on other imaging problems to test its validity, for instance, on image denoising and deblurring problems with general data terms.

Acknowledgements. The authors acknowledge the joint ANR/FWF Project "Efficient Algorithms for Nonsmooth Optimization in Imaging" (EANOI) FWF n. I1148 / ANR-12-IS01-0003.

Appendix

In this appendix we prove some general results which has been used in our work.

We start with a general inequality used to derive the descent rule (3.1). Its proof is a consequence of a trivial property of strongly convex functions.

Lemma 6.1. *If $h : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ is strongly convex with parameter $\mu_h > 0$ and $\hat{x} \in \mathcal{X}$ is a minimiser of h, the following property holds:*

$$h(x) \geq h(\hat{x}) + \frac{\mu_h}{2}\|x - \hat{x}\|^2, \tag{6.1}$$

for any $x \in \mathcal{X}$.

Proof. By definition of μ_h-strong convexity, for any $x, y \in \mathcal{X}$ there holds:

$$h(x) \geq h(y) + \langle p, y - x \rangle + \frac{\mu_h}{2}\|x - y\|^2,$$

where $p \in \partial h(y)$, the subdifferential of h evaluated in y. Taking $y = \hat{x}$, since $0 \in \partial h(\hat{x})$, we get (6.1). □

An immediate consequence of this general property is the proof of the descent rule (3.1) used in Section 3 as a starting point of our convergence estimates. We follow [11, 29].

Lemma 6.2. *Let $f : \mathcal{X} \to \mathbb{R}$ be a μ_f-strongly convex function with Lipschitz gradient with constant L_f and $g : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ be a l.s.c., μ_g-strongly convex function. Then, defining for any $\bar{x} \in \mathcal{X}$ and any $0 < \tau < 1/L_f$ the forward-backward map: $T_\tau : \bar{x} \mapsto prox_{\tau g}(\bar{x} - \tau \nabla f(\bar{x})) =: \hat{x}$, the following inequality holds for the composite functional $F = f + g$:*

$$F(x) + (1 - \tau\mu_f)\frac{\|x - \bar{x}\|^2}{2\tau} \geq F(\hat{x}) + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau}, \qquad \text{for any } x \in \mathcal{X}. \tag{6.2}$$

Proof. By definition, $\hat{x}$ is the minimiser of the function $h : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ defined by:

$$h : x \mapsto g(x) + f(\bar{x}) + \langle f(\bar{x}, x - \bar{x}\rangle + \frac{\|x - \bar{x}\|^2}{2\tau}.$$

The function h is strongly convex with parameter $\mu_h := (\tau\mu_g + 1)/\tau$. Hence, for any $x \in \mathcal{X}$:

$$\begin{aligned}
F(x) + (1 - \tau\mu_f)\frac{\|x - \bar{x}\|^2}{2\tau} &\geq g(x) + f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x}\rangle + \frac{\|x - \bar{x}\|^2}{2\tau} \\
&\geq g(\hat{x}) + f(\bar{x}) + \langle \nabla f(\bar{x}), \hat{x} - \bar{x}\rangle + \frac{\|\hat{x} - \bar{x}\|^2}{2\tau} + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau} \\
&\geq g(\hat{x}) + f(\hat{x}) + \frac{1 - \tau L_f}{2\tau}\|\hat{x} - \bar{x}\|^2 + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau}, \\
&= F(\hat{x}) + \frac{1 - \tau L_f}{2\tau}\|\hat{x} - \bar{x}\|^2 + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau}, \qquad (6.3)
\end{aligned}$$

where the first inequality holds by strong convexity of f, the second one is a simple application of Lemma 6.1 and the last one follows from the Lipschitz continuity of ∇f. Since $\tau L_f < 1$ by assumption, we can neglect the third term in (6.3) and get (6.2). □

We finally report a general properties of proximal mappings which we used in our numerical experiments in Section 5. For a general convex function h it essentially allows a straightforward calculation of the proximal map of the composite ε-strongly convex function $g := \alpha h + \frac{\varepsilon}{2}\|\cdot\|_2^2$ in terms of the proximal map of h itself.

Lemma 6.3. *Let $h : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ a convex, proper and l.s.c. function. For $\alpha, \varepsilon > 0$ let g be defined as:*

$$g(x) := \alpha h(x) + \frac{\varepsilon}{2}\|x\|^2, \qquad x \in \mathcal{X}.$$

Then, there holds:

$$\text{prox}_{\tau g}(z) = \text{prox}_h^{\frac{\alpha\tau}{1+\varepsilon\tau}}\left(\frac{z}{1 + \varepsilon\tau}\right), \quad \text{for any } \tau > 0 \text{ and } z \in \mathcal{X}. \tag{6.4}$$

Proof. Let $\tau > 0$ and $z \in \mathcal{X}$. We have the following chain of equalities:

$$\begin{aligned}
\operatorname{prox}_{\tau g}(z) = \operatorname{prox}_g^{\tau}(z) &= \arg\min_{y\in\mathcal{X}} g(y) + \frac{1}{2\tau}\|y-z\|^2 \\
&= \arg\min_{y\in\mathcal{X}} h(y) + \frac{1+\tau\varepsilon}{2\alpha\tau}\|y\|^2 + \frac{1}{2\alpha\tau}\|z\|^2 \pm \frac{\varepsilon}{2\alpha}\|z\|^2 - \frac{1}{\alpha\tau}\langle y, z\rangle \\
&= \arg\min_{y\in\mathcal{X}} h(y) + \frac{1}{2\frac{\alpha\tau}{1+\tau\varepsilon}}\|y\|^2 + \frac{1}{2\alpha\tau(1+\varepsilon\tau)}\|z\|^2 - \frac{1}{\alpha\tau}\langle y, z\rangle \\
&= \arg\min_{y\in\mathcal{X}} h(y) + \frac{1}{2\frac{\alpha\tau}{1+\tau\varepsilon}}\|y\|^2 + \frac{1}{2\frac{\alpha\tau}{1+\tau\varepsilon}}\|\frac{z}{1+\varepsilon\tau}\|^2 - \frac{1+\varepsilon\tau}{\alpha\tau}\langle y, \frac{z}{1+\varepsilon\tau}\rangle \\
&= \arg\min_{y\in\mathcal{X}} h(y) + \frac{1}{2\frac{\alpha\tau}{1+\tau\varepsilon}}\|y - \frac{z}{1+\varepsilon\tau}\|^2 = \operatorname{prox}_h^{\frac{\alpha\tau}{1+\varepsilon\tau}}(\frac{z}{1+\varepsilon\tau}).
\end{aligned}$$

□

References

1. L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives.*, Pacific Journal of Mathematics **16** (1966), no. 1, 1–3.
2. J.-F. Aujol and C. Dossal, *Stability of over-relaxations for the forward-backward algorithm, application to FISTA*, SIAM Journal on Optimization **25** (2015), no. 4, 2408–2433.
3. A. Beck and M. Teboulle, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Transactions on Image Processing **18** (2009), no. 11, 2419–2434.
4. ———, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences **2** (2009), no. 1, 183–202.
5. S. Bonettini, F. Porta, and V. Ruggiero, *A variable metric forward backward method with extrapolation*, SIAM Journal of Scientific Computing **38** (2016), no. 4, 2558–2584.
6. M. Burger, A. Sawatzky, and G. Steidl, *First-order algorithms in variational image processing*, pp. 345–407, Springer International Publishing, Cham, 2016.
7. A. Chambolle, *An algorithm for total variation minimization and applications*, Journal of Mathematical Imaging and Vision **20** (2004), no. 1, 89–97.
8. A. Chambolle and C. Dossal, *On the convergence of the iterates of the fast iterative shrinkage/thresholding algorithm*, Journal of Optimization theory and Applications **166** (2015), no. 3, 968–982.
9. A. Chambolle, M. J. Ehrhardt, P. Richtarik, and C.-B. Schönlieb, *Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications*, (2017), arXiv preprint: `https://arxiv.org/abs/1706.04957`.
10. A. Chambolle and T. Pock, *A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions*, SMAI-Journal of Computational Mathematics **1** (2015), 29–54.
11. ———, *An introduction to continuous optimization for imaging*, Acta Numerica **25** (2016), 161–319.
12. P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Modeling & Simulation **4** (2005), no. 4, 1168–1200.
13. Q. Ferocq and Q. Zheng, *Restarting accelerated gradient methods with a rough strong convexity estimate*, (2016), arXiv preprint: `https://arxiv.org/pdf/1609.07358`.
14. M. I. Florea and S. Vorobyov, *An accelerated composite gradient method for large-scale composite objective problems*, (2016), arXiv preprint: `https://arxiv.org/pdf/1612.02352`.
15. ———, *A generalized accelerated composite gradient method: uniting Nesterov's fast gradient method and FISTA*, (2017), arXiv preprint: `https://arxiv.org/abs/1705.10266`.
16. A. A. Goldstein, *Convex programming in hilbert space*, Bulletin of the American Mathematical Society **70** (1964), no. 5, 709–710.
17. O. Güler, *New proximal point algorithms for convex minimization*, SIAM Journal on Optimization **2** (1992), no. 4, 649–664.

18. H. Lin, J. Mairal, and Z. Harchaoui, *A universal catalyst for first-order optimization*, Proceedings of the 28th International Conference on Neural Information Processing Systems (Cambridge, MA, USA), NIPS'15, MIT Press, 2015, pp. 3384–3392.
19. Y. Nesterov, *A method for solving the convex programming problem with convergence rate $O(1/k^2)$*, Soviet Mathematics Doklady **269** (1983), no. 3, 543–547.
20. _____, *Introductory lectures on convex optimization*, vol. 87 ed., Boston : Kluwer Academic Publishers, 2004.
21. _____, *Smooth minimization of non-smooth functions*, Mathematical Programming **103** (2005), no. 1, 127–152.
22. _____, *Gradient methods for minimizing composite functions*, Mathematical Programming **140** (2013), no. 1, 125–161.
23. L. Rudin, S. Osher, and E. Fatemi, *Nonlinear Total Variation based noise removal algorithms*, Physica D **60** (1992), 259–268.
24. S. Salzo and S. Villa, *Inexact and accelerated proximal point algorithms*, Journal of Convex Analysis **19** (2012), no. 4, 1167–1192.
25. A. Sawatzky, *(nonlocal) total variation in medical imaging*, Ph.D. thesis, University of Muenster, Germany, 2011.
26. K. Scheinberg, D. Goldfarb, and X. Bai, *Fast first-order methods for composite convex optimization with backtracking*, Foundations of Computational Mathematics **14** (2014), no. 3, 389–417.
27. M. Schmidt, N. Roux, and F. Bach, *Convergence rates of inexact proximal-gradient methods for convex optimization*, Advances in Neural Information Processing Systems 24 (J. Shawe-taylor, R.s. Zemel, P. Bartlett, F.c.n. Pereira, and K.q. Weinberger, eds.), 2011, pp. 1458–1466.
28. S. Tao, D. Boley, and S. Zhang, *Local linear convergence of ISTA and FISTA on the LASSO problems*, SIAM Journal on Optimization **26** (2016), no. 1, 313–336.
29. P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, (2008), `http://www.csie.ntu.edu.tw/~b97058/tseng/papers/apgm.pdf`.
30. S. Villa, S. Salzo, L. Baldassarre, and A. Verri, *Accelerated and inexact forward-backward algorithms*, SIAM Journal on Optimization **23** (2013), no. 3, 1607–1633.