SIAM J. OPTIM.
Vol. 29, No. 3, pp. 1772–1798

BACKTRACKING STRATEGIES FOR ACCELERATED DESCENT METHODS WITH SMOOTH COMPOSITE OBJECTIVES[*]

LUCA CALATRONI[†] AND ANTONIN CHAMBOLLE[†]

Abstract. We present and analyze a backtracking strategy for a general fast iterative shrinkage/thresholding algorithm proposed by Chambolle and Pock [*Acta Numer.*, 25 (2016), pp. 161–319] for strongly convex composite objective functions. Unlike classical Armijo-type line searching, our backtracking rule allows for local increasing and decreasing of the descent step size (i.e., proximal parameter) along the iterations. We prove accelerated convergence rates and show numerical results for some exemplar problems.

Key words. composite optimization, forward-backward splitting, acceleration, backtracking, image denoising, elastic net

AMS subject classifications. 90C25, 65F22

DOI. 10.1137/17M1149390

1. Introduction.

The concept of *acceleration* of first-order optimization methods dates back to the seminal work of Nesterov [20]. For a proper, convex, l.s.c. function $F : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ defined on a Hilbert space $\mathcal{X}$ with Lipschitz gradient with constant $L > 0$, solving the abstract optimization problem

$$\min_{x\in\mathcal{X}} \; F(x) \tag{1}$$

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 $O(1/k^2)$ that holds for any function such as F. In the smoother case, i.e., when F is a strongly convex function with parameter $\mu > 0$, Nesterov showed in [21, Theorem 2.1.13] that a lower bound for first-order optimization methods of order $O((\frac{\sqrt{q}-1}{\sqrt{q}+1})^{2k})$ can be shown, with $q := L/\mu \geq 1$ being the *condition number* of F. In this case, improved linear convergence rates of order $O((\frac{\sqrt{q}-1}{\sqrt{q}})^k)$ are proved. Similar results for implicit gradient descent have been studied by Güler [17]. We also refer the reader to [26], where a general framework for inexact accelerated methods is presented.

If the objective function in (1) can be further decomposed as the sum of a convex function f with Lipschitz gradient ∇f and a convex, l.s.c., and nonsmooth function g, i.e., if problem (1) can be rewritten as

$$\min_{x\in\mathcal{X}} \; \{F(x) = f(x) + g(x)\}, \tag{2}$$

different descent methods taking into account the nondifferentiability of F need to be considered. Such approaches go under the name of *composite optimization* methods,

[*] Received by the editors September 27, 2017; accepted for publication (in revised form) April 30, 2019; published electronically July 2, 2019.
https://doi.org/10.1137/17M1149390
Funding: The authors acknowledge the joint ANR/FWF Project "Efficient Algorithms for Nonsmooth Optimization in Imaging" (EANOI), ANR-12-IS01-0003/FWF n. I1148, and the support of the Fondation Mathématique Jacques Hadamard (FMJH).

[†] Centre de Mathématiques Appliquées (CMAP), École Polytechnique CNRS, 91128 Palaiseau CEDEX, France (luca.calatroni@polytechnique.edu, antonin.chambolle@cmap.polytechnique.fr).

after the work of Nesterov [23]. A typical optimization strategy for solving composite optimization problems consists in alternating along the iterations a "forward" (i.e., explicit) gradient descent step taken in correspondence with the differentiable component f and a "backward" (implicit) gradient descent step in correspondence with the nonsmooth part g. Due to this alternation, such an optimization technique is known as *forward-backward* (FB) splitting. The literature on FB splitting methods is extremely vast. Such a strategy was first used in [16] for projected gradient descent, and subsequently popularized within the imaging community with the work of Combettes and Wajs [12]. Acceleration methods for FB splitting were first considered by Nesterov in [21] for projected gradient descent, and later extended by Beck and Teboulle [4] to more general "simple" nonsmooth functions g under the name of fast iterative shrinkage/thresholding algorithm (FISTA). Several variants of FISTA have been considered in works such as [22, 31, 23, 10, 6, 5], and properties such as convergence of the iterates under specific assumptions [8] and monotone variants (M-FISTA) [3, 30] 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 [29, 32, 2] for these studies.

In its original formulation, FISTA requires an estimate on the Lipschitz constant $L_f > 0$ of ∇f. Whenever such an estimate is not easily computable, an Armijo-type backtracking rule [1] can alternatively be used [4, section 4]. By construction, this backtracking strategy requires such an estimate to be nondecreasing along the iterations. From a practical point of view, this condition 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, the convergence speed may suffer if an inaccurate estimate of L_f is computed. To avoid this drawback, in [28] Scheinberg, Goldfarb, and Bai proposed a backtracking strategy for FISTA where an 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 a suitable average of the $k-1$ local estimates of the L_f computed in the previous iterations. The proposed strategy is shown to guarantee acceleration and to outperform the standard Armijo-type backtracking in several numerical examples. Compared to the similar full backtracking strategy proposed by Nesterov in [23], the criterion used in [28] renders more cheaply since it does not require the extra calculation of the ∇f term in correspondence with the proximal step at each iteration.

In the case of strongly convex objective functionals, improved linear convergence rates are expected. Recalling the composite problem (2), the case of a strongly convex component f was first considered for projected gradient descent in [21] and more recently extended by Chambolle and Pock [11] to the case of strongly convex f and g. In this work, we will denote this general FISTA-type algorithm by GFISTA. For GFISTA, linear convergence rates have rigorously been shown, encompassing the quadratic ones of plain FISTA in the nonstrongly convex case. For its practical application, GFISTA requires an estimate of the Lipschitz constant L_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.

Contribution. In this work we analyze a full backtracking strategy for the strongly convex version of FISTA (GFISTA) proposed in [11]. Unlike the standard backtracking rule proposed in the original paper by Beck and Teboulle [4] and based on Armijo line-searching [1], the strategy considered here allows for both increasing and decreasing of the Lipschitz constant estimate, i.e., for both decreasing and increasing of the gradient descent step size. Compared to the full backtracking strategy presented by Nesterov in [23], the one we consider here does not require the evaluation of the gradient of the smooth component in correspondence with the proximal step at each iteration, and thus it renders more cheaply. A similar backtracking strategy was considered by Scheinberg, Goldfarb, and Bai for plain FISTA in [28], but its generalization to the strongly convex case is not straightforward. We address this in this work, presenting a unified framework where the standard FISTA (with and without backtracking) can be derived as a particular case. In the case of strongly convex objectives, we prove linear convergence results by studying in detail the decay speed of the corresponding convergence factors. We validate our theoretical results on some exemplar problems with strongly convex objective functions that can be encountered in imaging or in data analysis. Finally, to relax the dependence on the strong convexity parameters appearing in the algorithm, we combine the backtracking strategy to classical restarting methods [24], which show empirical convergence properties.

Organization of the paper. In section 2 we recall some definitions and standard assumptions used in the modeling of composite optimization problems. In section 3 we present GFISTA, the strongly convex variant of FISTA studied in [11]. Next, in section 4 we analyze an adaptive backtracking strategy for GFISTA and prove the accelerated convergence results by means of technical tools inspired by [21]. Numerical examples confirming our theoretical results are reported in section 5. In the final section (section 6) we summarize the main results of this work and point towards some challenging questions to be addressed in future work.

Remark 1.1. In their recent preprint [14], Florea and Vorobyov propose an algorithm similar to the one described in this work as an extension of their previous work [15]. The convergence result [14, section 3.1, Theorem 2] obtained by the authors is similar to the one presented in our work (see Theorem 4.6) but less accurate since it is based on a worst-case analysis while ours depends on average quantities estimated along the iterations. Furthermore, the arguments used in [14] are completely different from the ones used here. To show the main convergence result, Florea and Vorobyov considered *generalized estimate sequences*, a notion which, starting from the original paper by Nesterov [20], has indeed become very popular in the field of optimization (see, e.g., [17, 18, 26]) due to its easy geometrical interpretation. However, the use of this technique leaves the technical difficulties related to the precise study of the decay speed of the convergence factors somewhat hidden. Inspired by [21] and [4], here we follow a different path, defining appropriate decay factors and extrapolation rules along the iterations that, eventually, will result in an accelerated (linear) convergence rate.

2. Preliminaries and notation.

We are interested in the solution of the composite minimization problem

$$\min_{x\in\mathcal{X}} \ \{F(x) = f(x) + g(x)\}, \tag{3}$$

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 minimize. We denote by $x^* \in \mathcal{X}$ a minimizer of F. We let $f : \mathcal{X} \to \mathbb{R}$ be a differentiable convex function with Lipschitz gradient and $g : \mathcal{X} \to \mathbb{R}\cup\{+\infty\}$ be nonsmooth, convex, and l.s.c. We further denote by L_f the Lipschitz constant of ∇f, so that

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

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

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

Similarly, we will denote by $\mu_g \geq 0$ the strong convexity parameter of g. The strong convexity parameter of the composite functional F in (3) will then be 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 $\mu > 0$.

Remark 2.1. Note that the nonstrongly convex case when $\mu = 0$ reduces (3) to the classical FISTA-type optimization problem. In the case of projected gradient descent, i.e., when solving

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

the case when $\mu_f > 0$ was studied by Nesterov in [21]. The problem can formulated in the form (3), with g being the indicator function of the subset $\mathcal{B}$ (with $\mu_g = 0$), as

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

Note, however, that the proof in [21] actually works for any function g; see [11] for more details.

In order to write the FB optimization step, 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 $T_\tau : \mathcal{X} \to \mathcal{X}$, which is defined as follows:

$$\bar{x} \mapsto \hat{x} = T_\tau \bar{x} := \text{prox}_{\tau g}(\bar{x} - \tau \nabla f(\bar{x})),$$

where $\text{prox}_{\tau g}$ denotes the proximal mapping operator defined by

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

Note that, in order to exploit some properties of the proximal mapping operator above, for $\eta > 0$ we will also make use of the following notation:

$$\text{prox}_g^\eta(z) = \arg\min_{y \in \mathcal{X}} \left(g(y) + \frac{1}{2}\|z-y\|_{\eta^{-1}}^2 \right), \quad z \in \mathcal{X}, \tag{4}$$

where the weighted norm is defined by $\|w\|_{\eta^{-1}}^2 = \langle \eta^{-1} w, w \rangle$.

3. A general fast iterative shrinkage/thresholding algorithm.

The fast iterative shrinkage/thresholding algorithm, proposed in [4], is a very popular optimization strategy to minimize composite functionals F as in (3) with convergence guarantees of order $O(1/k^2)$. Such an algorithm extends Nesterov's approach in [21] for the case of smooth constrained minimization to more general nonsmooth functions g. In the strongly convex case when $\mu > 0$, linear convergence rates have been shown in [11] by means of a careful study of the decay of the composite functional toward its optimal value. In the following, we will refer to this extension to the strongly convex case as general FISTA (GFISTA).

For conciseness, in Algorithm 1 we unify FISTA and GFISTA, 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 finding a useful recursion starting from the following descent rule for F, which holds for every $x \in \mathcal{X}$ and for $\hat{x} = T_\tau \bar{x}$ with $\bar{x} \in \mathcal{X}$:

$$F(\hat{x}) + (1+\tau\mu_g)\frac{\|x-\hat{x}\|^2}{2\tau} \leq F(x) + (1-\tau\mu_f)\frac{\|x-\bar{x}\|^2}{2\tau}, \quad \tau > 0. \tag{5}$$

Inequality (5) is classically used as a starting point to study convergence rates. Its proof is a trivial consequence of a general property holding for strongly convex functions. We report it in Lemma A.2.

Starting from (5), the general technique to perform a convergence analysis consists in taking as element $x \in \mathcal{X}$ the convex combination of the kth iterate x_k of the algorithm considered and a generic point (such as x^*), and by means of (strong) convexity assumptions, 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 a factor then needs to be made by means of the technical properties of the iterates of the algorithm and of its extrapolation parameters. We refer the reader to the work of Nesterov [21] for a review of these techniques applied to standard cases and to [11] for a survey of their applications in the context of imaging.

The result in Theorem 3.2 generalizes the ones proved for FISTA in [21, 4]. In particular, the standard FISTA convergence rate of $O(1/k^2)$ proved in [4, Theorem 4.4] in the nonstrongly 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 and used as an input to find admissible gradient parameters $\tau > 0$. We refer the reader to [22, 23, 31] for similar results proved for variants of FISTA.

Algorithm 1 FISTA and GFISTA (no backtracking).

```
Input: $0 < \tau \leq 1/L_f$, $\mu \geq 0$, $x^0 = x^{-1} \in \mathcal{X}$, $q := \tau\mu/(1+\tau\mu_g) \in [0,1)$, and $t_0 \in \mathbb{R}$ s.t. $0 \leq t_0 \leq 1/\sqrt{q}$.
for $k \geq 0$ do
    $y^k = x^k + \beta_k(x^k - x^{k-1}),$
(6) $x^{k+1} = T_\tau y^k = \mathrm{prox}_{\tau g}(y^k - \tau\nabla f(y^k)),$
    where
(7) $t_{k+1} = \frac{1 - qt_k^2 + \sqrt{(1-qt_k^2)^2 + 4t_k^2}}{2},$
    $\beta_k = \frac{t_k - 1}{t_{k+1}} \frac{1 + \tau\mu_g - t_{k+1}\tau\mu}{1 - \tau\mu_f}$
end for
```

Remark 3.1 (FISTA updates). Note that in the case when $\mu = 0$ the update rules for t_{k+1} and β_k in (7) simplify to

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \qquad \beta_k = \frac{t_k - 1}{t_{k+1}}, \tag{8}$$

which are the standard FISTA updates considered by Beck and Teboulle in [4].

THEOREM 3.2 (see [21] and [11, Theorem B.1]). *Let $\tau > 0$ with $\tau \leq 1/L_f$, $q := \mu\tau/(1+\tau\mu_g)$, and x^* be a minimizer of F. If $\sqrt{q}t_0 \leq 1$ with $t_0 \geq 0$, then the sequence (x^k) produced by* (6) *in Algorithm* 1 *satisfies*

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

and $r_k(q)$ is defined by

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

Backtracking. Whenever an estimate of L_f is not available, backtracking techniques can be used. For FISTA, an Armijo-type backtracking rule was proposed in the original paper by Beck and Teboulle [4]. In this case, similar convergence rates to those above can be proved. Furthermore, in order to improve the speed of the algorithm and by allowing an increase in the step size τ in the neighborhoods of "flat" points of the function f (i.e., where L_f is small), a full backtracking strategy for FISTA was considered by Scheinberg, Goldfarb, and Bai in [28].

The typical inequality to check in the design of any backtracking strategy can be derived from (5) (see Lemma A.2) and reads

$$F(\hat{x})+(1+\tau\mu_g)\frac{\|x-\hat{x}\|^2}{2\tau}+\left(\frac{\|\hat{x}-\bar{x}\|^2}{2\tau} - D_f(\hat{x},\bar{x})\right) \leq F(x)+(1-\tau\mu_f)\frac{\|x-\bar{x}\|^2}{2\tau}, \tag{9}$$

where $D_f(\hat{x},\bar{x}) := f(\hat{x}) - f(\bar{x}) - \langle\nabla f(\bar{x}), \hat{x}-\bar{x}\rangle \leq \frac{L_f}{2}\|\hat{x}-\bar{x}\|^2$ is the Bregman distance of f between $\hat{x}$ and $\bar{x}$. Note that, in the case when no backtracking is performed, condition (9) 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 $0 < \tau \leq 1/L_f$ with L_f known. However, by letting τ vary, one can alternatively check condition (9) along the iterations of the algorithm and redefine τ_k at each iteration $k \geq 1$ so as 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 is classically done) or increase depending on the validity of the following inequality:

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

where the constant $\rho \in (0,1)$ is chosen in advance and where the choice of $\bar{x}$ and $\hat{x}$ will be made precise in the following section. Note that this condition entails that the inequality

$$\tau_k \geq \frac{\rho}{L_f} \tag{10}$$

holds at any iteration. Heuristically, condition (CB2) favors the step size τ_k to 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 analysis performed in [11, section 4 and Appendix B], we prove that the backtracking strategy described above and applied to GFISTA (Algorithm 1) enjoys accelerated convergence rates, which turn out to be linear in the case when $\mu > 0$.

For an arbitrary $t \geq 1$, $k \geq 0$, and $\tau > 0$ we start from inequality (9) and choose the point x to be the convex combination $x = ((t-1)x^k + x^*)/t$, where x^k is an iterate of the algorithm we are going to define and x^* is a minimizer of F. For the other points, we set $\bar{x} = y^{k+1}$ and $\hat{x} = x^{k+1} = T_\tau y^{k+1}$. The formula for y^{k+1} will be specified next.

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

$$\begin{aligned} & t^2\left(F(x^{k+1}) - F(x^*)\right) + \frac{1+\tau\mu_g}{2\tau}\|x^* - x^{k+1} - (t-1)(x^{k+1}-x^k)\|^2 \\ & + t^2(t-1)\frac{\mu(1-\tau\mu_f)}{1+\tau\mu_g - t\tau\mu}\frac{\|x^k - y^{k+1}\|^2}{2} \\ & \qquad \leq t(t-1)\left(F(x^k) - F(x^*)\right) \\ & \qquad\quad + \frac{1+\tau\mu_g - t\tau\mu}{2\tau}\|x^* - x^k - t\frac{1-\tau\mu_f}{1+\tau\mu_g - t\tau\mu}(y^{k+1}-x^k)\|^2. \end{aligned} \tag{11}$$

We now set $t = t_{k+1}$, let $\tau = \tau_{k+1}$, and define the following quantities:

$$\tau'_{k+1} := \frac{\tau_{k+1}}{1+\tau_{k+1}\mu_g} > 0, \tag{12}$$

$$q_{k+1} := \mu\tau'_{k+1} = 1 - \frac{1-\tau_{k+1}\mu_f}{1+\tau_{k+1}\mu_g} \in [0,1), \tag{13}$$

$$\omega_{k+1} := \frac{1+\tau_{k+1}\mu_g - t_{k+1}\tau_{k+1}\mu}{1+\tau_{k+1}\mu_g} = 1 - t_{k+1}q_{k+1} \in (0,1], \tag{14}$$

$$\beta_{k+1} := \frac{t_k - 1}{t_{k+1}}\frac{1+\tau_{k+1}\mu_g - t_{k+1}\tau_{k+1}\mu}{1-\tau_{k+1}\mu_f} = \omega_{k+1}\frac{t_k-1}{t_{k+1}}\frac{1+\tau_{k+1}\mu_g}{1-\tau_{k+1}\mu_f}, \tag{15}$$

where we can assume $\mu_f < L_f$, so that $\tau < 1/L_f$.

We now define the following update for y^{k+1}:

$$y^{k+1} = x^k + \beta_{k+1}(x^k - x^{k-1}) \tag{16}$$

for any $k \geq 0$. After further multiplying (11) by τ'_{k+1}, we thus deduce

$$\begin{aligned} & \tau'_{k+1}t_{k+1}^2\left(F(x^{k+1}) - F(x^*)\right) + \frac{1}{2}\|x^* - x^{k+1} - (t_{k+1}-1)(x^{k+1}-x^k)\|^2 \\ & \qquad \leq \tau'_{k+1}t_{k+1}(t_{k+1}-1)\left(F(x^k) - F(x^*)\right) \\ & \qquad\qquad + \frac{\omega_{k+1}}{2}\|x^* - x^k - (t_k - 1)(x^k - x^{k-1})\|^2. \end{aligned} \tag{17}$$

Let us now assume that, for every $k \geq 1$, the inequality

$$\tau'_{k+1} t_{k+1}(t_{k+1}-1) \leq \omega_{k+1} \tau'_k t_k^2 \tag{18}$$

holds, and that the same holds for the iteration $k=0$ by defining $T_0^2 := \tau'_0 t_0^2$ implicitly by

$$T_0^2 = \frac{\tau'_1 t_1(t_1-1)}{\omega_1} = \frac{\tau_1 t_1(t_1-1)}{1+\tau_1\mu_g - t_1\tau_1\mu}, \tag{19}$$

which is positive whenever

$$1 \leq t_1 < \frac{1+\tau_1\mu_g}{\tau_1\mu} = \frac{1}{q_1}. \tag{20}$$

Then, we get from (17) that, for any $k \geq 0$,

$$\begin{aligned} &\tau'_{k+1} t_{k+1}^2 \left(F(x^{k+1}) - F(x^*)\right) + \frac{1}{2}\|x^* - x^{k+1} - (t_{k+1}-1)(x^{k+1}-x^k)\|^2 \\ &\quad \leq \omega_{k+1}\left(\tau'_k t_k^2 \left(F(x^k) - F(x^*)\right) + \frac{1}{2}\|x^* - x^k - (t_k-1)(x^k - x^{k-1})\|^2\right). \end{aligned} \tag{21}$$

By now applying (21) recursively and letting $x^0 = x^{-1} \in \mathcal{X}$, we find the following convergence inequality:

$$F(x^k) - F(x^*) \leq \theta_k \left(T_0^2\left(F(x^0) - F(x^*)\right) + \frac{1}{2}\|x^* - x^0\|^2\right), \tag{22}$$

where the decay rate of the factor

$$\theta_k := \frac{\prod_{i=1}^k \omega_i}{\tau'_k t_k^2} \tag{23}$$

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.

Assuming that (18) holds with an equality sign, i.e.,

$$\tau'_{k+1} t_{k+1}(t_{k+1}-1) = \omega_{k+1} \tau'_k t_k^2, \tag{24}$$

and after recalling the definition of ω_{k+1} in (14), we find the following update rule for the elements of sequence (t_k), $k \geq 1$:

$$\begin{aligned} t_{k+1} &= \frac{1 - q_{k+1}\frac{\tau'_k}{\tau'_{k+1}} t_k^2 + \sqrt{\left(1 - q_{k+1}\frac{\tau'_k}{\tau'_{k+1}} t_k^2\right)^2 + 4\frac{\tau'_k}{\tau'_{k+1}} t_k^2}}{2} \\ &= \frac{1 - q_k t_k^2 + \sqrt{\left(1-q_k t_k^2\right)^2 + 4\frac{q_k}{q_{k+1}} t_k^2}}{2} \geq 0, \end{aligned} \tag{25}$$

by (13) and (12).

We can now present a new version of GFISTA with backtracking (Algorithm 2).

Algorithm 2 GFISTA with backtracking.

```
Parameters: $\mu_f$, $\mu_g$, $\tau_1 > 0$, $\rho \in (0,1)$.
Initialization: $q_1 := \mu\tau_1/(1+\tau_1\mu_g)$, $x^0 = x^{-1} \in \mathcal{X}$, $x_1 = T_{\tau_1}(x_0)$, and $t_1 \in \mathbb{R}$ s.t. $1 \le t_1 \le 1/\sqrt{q_1}$.
for $k = 1, 2, \dots$ do
    (26)    $\tau_{k+1}^0 = \frac{\tau_k}{\rho}$;
    for $i = 0, 1, \dots$ repeat
        $\tau_{k+1} = \rho^i \ \tau_{k+1}^0$,
        $q_{k+1} = \frac{\mu\tau_{k+1}}{1+\tau_{k+1}\mu_g}$,
        $t_{k+1} = \frac{1 - q_k t_k^2 + \sqrt{(1-q_k t_k^2)^2 + 4\frac{q_k}{q_{k+1}} t_k^2}}{2}$,
        $\beta_{k+1} = \frac{t_k - 1}{t_{k+1}} \frac{1+\tau_{k+1}\mu_g - t_{k+1}\tau_{k+1}\mu}{1-\tau_{k+1}\mu_f}$,
        (27)    $y^{k+1} = x^k + \beta_{k+1}(x^k - x^{k-1})$,
        (28)    $x^{k+1} = T_{\tau_{k+1}} \ y^{k+1} = \mathrm{prox}_{\tau_{k+1}g}(y^{k+1} - \tau_{k+1}\nabla f(y^{k+1}))$,
    until $D_f(x^{k+1}, y^{k+1}) \le \|x^{k+1} - y^{k+1}\|^2/2\tau_{k+1}$.
end for
```

We remark that compared to the algorithm studied in [23, section 4], Algorithm 2 has a lower per-iteration cost. The reason for that is that the backtracking criterion considered in [23] requires, at any iteration k, the computation of the quantity $\nabla f(T_{\tau_{k+1}} y^k)$, whereas our backtracking condition (CB) is based on the calculation of D_f for which only the computation of $\nabla f(y^k)$ is required, thus avoiding the calculation of ∇f in the proximal step. In many applications (e.g., compressed sensing), this difference can be quite crucial: the extra evaluation of ∇f in one point in fact requires two matrix-vector multiplications compared to a single one required for functional evaluation. Similar considerations have already been made for FISTA with full backtracking in [28] since the stopping criterion for the backtracking procedure considered therein similar to the one used in our Algorithm 2.

Remark 4.1. A risk with Algorithm 2 is that when the step τ_k is correctly estimated, by performing (26) we systematically overestimate its value and have to perform a backtracking step to correct it. A simple strategy to avoid this issue consists in calculating (26) in practice only when we observe that, in the previous step,

$$D_f(x^k, y^k) \le \rho \frac{\|x^k - y^k\|^2}{2\tau_k}.$$

If not, we keep $\tau_{k+1}^0 = \tau_k$. This does not modify the convergence analysis.

Remark 4.2 (no backtracking). When no backtracking is performed along the iterations, $\tau_k = \tau_{k+1}$ for any k and the ratio q_k/q_{k+1} in (25) is constantly equal to 1. In this case, the update rule (25) is the same as that used in (7) for GFISTA without backtracking; cf. [11, Appendix B].

Remark 4.3 (FISTA with backtracking). In the nonstrongly convex case ($\mu_f = \mu_g = q_k = 0$ for every k), (25) 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 Scheinberg, Goldfarb, and Bai in [28] for adaptive backtracking of plain FISTA.

We now prove a fundamental property of the sequence (t_k) defined by (25).

LEMMA 4.4. *Let the sequence* (t_k) *be defined by the update rule* (25). *Then*

$$t_k \geq 1 \quad \textit{for any } k \geq 1.$$

Proof. We simply observe that since $q_k \leq 1$ for every k it holds that

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

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

LEMMA 4.5. *Let* $\sqrt{q_1} t_1 \leq 1$. *Then, it holds that*

$$\sqrt{q_k} t_k \leq 1. \tag{29}$$

Proof. We proceed by induction. By assumption, the initial step $k = 1$ holds. Let us assume that (29) holds for some $k \geq 1$. By (24), we get

$$q_{k+1} t_{k+1}^2 = q_{k+1} t_{k+1} + \omega_{k+1} q_k t_k^2 = 1 + \omega_{k+1}(q_k t_k^2 - 1) \leq 1$$

by simply applying the induction assumption. □

Note that the condition $t_1 \leq 1/\sqrt{q_1}$ combined with $t_1 \geq 1$ results in the following bound:

$$1 \leq t_1 \leq \sqrt{1 + \frac{1 - \tau_1 \mu_f}{\tau_1 \mu}}. \tag{30}$$

Furthermore, since $1/\sqrt{q_1} < 1/q_1$, such a condition also guarantees (20). In particular, $t_1 = 1$ is an admissible choice.

4.2. Convergence rates. In this section, we follow [21, 11] to derive a precise estimate of the factor θ_k in (23).

The following convergence result shows that the backtracking strategy applied to GFISTA guarantees accelerated linear convergence rates given in terms of *averaging* quantities defined in terms of the Lipschitz constant estimates along the iterations. Remarks on our result in comparison to the those studied in analogous works [28, 14] are given below.

THEOREM 4.6 (convergence rates). *Let T_0 be defined as in* (19). *If $1 \leq t_1 \leq 1/\sqrt{q_1}$, then the sequence (x^k) produced by Algorithm* 2 *with* (25), (14), (15), *and* (16) *satisfies*

$$F(x^k) - F(x^*) \leq r_k \left(T_0^2 \left(F(x^0) - F(x^*) \right) + \frac{1}{2} \|x^0 - x^*\|^2 \right), \tag{31}$$

where r_k is defined by

$$r_k := \min \left\{ \frac{4\bar{L}_k}{k^2}, (L_1 - \mu_f)(1 - \sqrt{\bar{q}_k})^{k-1} \right\}, \tag{32}$$

and the average quantities $\bar{L}_k$ and $\sqrt{\bar{q}_k}$ are defined by

$$\sqrt{\bar{L}_k} := \frac{1}{\frac{1}{k} \sum_{i=1}^k \frac{1}{\sqrt{L_i - \mu_f}}}, \qquad \sqrt{\bar{q}_k} := \frac{1}{k-1} \sum_{i=2}^k \sqrt{\frac{\mu}{L_i + \mu_g}}, \tag{33}$$

with $L_i := 1/\tau_i$.

Proof. We recall the definition of θ_k given in (23) and start by computing the $O(1/k^2)$ factor in (32) following [21, 11].

We first notice that from (24) we can deduce

$$1 - \frac{1}{t_{k+1}} = \omega_{k+1} \frac{\tau'_k t_k^2}{\tau'_{k+1} t_{k+1}^2} = \frac{\theta_{k+1}}{\theta_k} \leq 1, \tag{34}$$

which also shows that θ_k is nonincreasing. Thus, we have

$$\frac{1}{\sqrt{\theta_{k+1}}} - \frac{1}{\sqrt{\theta_k}} = \frac{\theta_k - \theta_{k+1}}{\sqrt{\theta_k \theta_{k+1}}(\sqrt{\theta_k} + \sqrt{\theta_{k+1}})} \geq \frac{\theta_k - \theta_{k+1}}{2\theta_k \sqrt{\theta_{k+1}}}. \tag{35}$$

Now, by applying (34), we get

$$\frac{1}{\sqrt{\theta_{k+1}}} - \frac{1}{\sqrt{\theta_k}} \geq \frac{1}{2t_{k+1}\sqrt{\theta_{k+1}}}.$$

We now recall definitions (14) and (23), and use Lemma 4.4 to find

$$\begin{aligned} t_{k+1}\sqrt{\theta_{k+1}} &= \frac{1}{\sqrt{\tau'_{k+1}}} \prod_{i=1}^{k+1} \sqrt{\omega_i} \leq \sqrt{\frac{\omega_{k+1}}{\tau'_{k+1}}} = \sqrt{\frac{1}{\tau'_{k+1}} - \mu t_{k+1}} \\ &\leq \sqrt{\frac{1}{\tau'_{k+1}} - \mu} = \sqrt{\frac{1}{\tau_{k+1}} - \mu_f}, \end{aligned}$$

whence we obtain

$$\frac{1}{\sqrt{\theta_{k+1}}} - \frac{1}{\sqrt{\theta_k}} \geq \frac{1}{2\sqrt{\frac{1}{\tau_{k+1}} - \mu_f}}.$$

Applying this recursively, we get that, for any $k \geq 1$,

$$\frac{1}{\sqrt{\theta_k}} \geq \frac{1}{2} \sum_{i=1}^k \frac{1}{\sqrt{\frac{1}{\tau_i} - \mu_f}}. \tag{36}$$

Note that for $i = 1$ we indeed have

$$\theta_1 = \frac{1-\mu t_1 \tau_1'}{\tau_1' t_1^2} = \frac{1-\mu_g(t_1-1)\tau_1 - \mu_f t_1 \tau_1}{\tau_1 t_1^2} \leq \frac{1}{\tau_1} - \mu_f, \tag{37}$$

since $t_1 \geq 1$ by (30). We then deduce

$$\frac{1}{\sqrt{\theta_1}} \geq \frac{1}{2\sqrt{\frac{1}{\tau_1} - \mu_f}}.$$

After setting $L_i = 1/\tau_i$ in (36), we get

$$\sqrt{\theta_k} \leq \frac{2}{k}\sqrt{\bar{L}_k}, \tag{38}$$

where $\sqrt{\bar{L}_k}$ is defined in (33).

To get the linear rates, we notice that, by Lemma 4.5, relation (34), and definition (13), we have

$$\theta_k = \theta_1 \prod_{i=2}^{k}\left(1 - \frac{1}{t_i}\right) \leq \theta_1 \prod_{i=2}^{k}(1-\sqrt{q_i}) \tag{39}$$

$$\leq \theta_1 \prod_{i=2}^{k}\left(1 - \sqrt{\frac{\mu}{L_i + \mu_g}}\right) \leq \theta_1 (1-\sqrt{\bar{q}})^{k-1}, \tag{40}$$

which follows by the concavity of the function logarithm and the definition (33) of $\sqrt{\bar{q}_k}$. We then get from (39) that

$$\theta_k \leq \theta_1(1-\sqrt{\bar{q}_k})^{k-1} \leq (L_1 - \mu_f)(1-\sqrt{\bar{q}_k})^{k-1}$$

by (37). Combining this with (38) we get the final rate (32). □

Note that the averaging term $\bar{L}_k$ appearing above is always smaller than the actual average of the terms $(L_i - \mu_f)$, since

$$\sqrt{\bar{L}_k} \leq \frac{1}{k}\sum_{i=1}^{k}\sqrt{L_i - \mu_f} \leq \sqrt{\frac{1}{k}\sum_{i=1}^{k}(L_i - \mu_f)}. \tag{41}$$

Furthermore, whenever L_f is known, recalling (10), we can deduce the following bounds for the terms defined in (33):

$$\sqrt{\bar{L}_k} \leq \sqrt{\frac{L_f - \rho\mu_f}{\rho}}, \qquad \sqrt{\bar{q}_k} \geq \sqrt{\frac{\rho\mu}{L_f + \rho\mu_g}}.$$

Hence, the convergence rate r_k in (32) can be estimated as

$$r_k \leq \frac{1}{\rho}\min\left\{\frac{4(L_f - \rho\mu_f)}{k^2}, (L_f - \rho\mu_f)\left(1 - \sqrt{\frac{\rho\mu}{L_f + \rho\mu_g}}\right)^{k-1}\right\}.$$

Finally, as far as the choice of T_0 is concerned, note that, by (19), when $t_1 = 1$, we have $T_0 = 0$.

Remark 4.7 (FISTA with backtracking). Note that in the nonstrongly convex case ($\mu = q_k = 0$ for all k) the global convergence rates (31) and (32) are analogous to [28, Theorem 3.3], that is

$$F(x^k) - F(x^*) \leq \frac{2\tilde{L}_k \|x^0 - x^*\|^2}{\rho k^2}, \tag{42}$$

where the term $\tilde{L}_k$ is defined by

$$\tilde{L}_k := \frac{(\sum_{i=1}^k \sqrt{L_i})^2}{k^2}.$$

Note in fact that whenever $\mu_f = 0$ our definition (33) relates to the one above via (41).

Remark 4.8. The worst-case convergence result [14, Theorem 2] is obtained via the analysis of generalized estimate sequences. In [14, section 4] some comments on the extrapolated form of their algorithm and its relation with the strongly convex variant of 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 (14) and (25), respectively, the arguments used by Florea and Vorobyov are different from the ones used here. More importantly, compared to a worst-case analysis, the convergence result in Theorem 4.6 is more precise, since it provides quantitative convergence estimates in terms of the average quantities $\sqrt{\bar{L}_k}$ and $\sqrt{\bar{q}_k}$ estimated along the iterations.

4.3. 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 nonincreasing. A straightforward modification of the proposed algorithm (Algorithm 2) enforcing such a property and used in several papers [3, 31] consists in taking as x^k any point such that $F(x^k) \leq F(T_{\tau_k} y^k)$. Recalling the definition of ω_{k+1} in (14), the update rule (27) for extrapolation can then be changed to

$$\begin{aligned} y^{k+1} &= x^k + \beta_{k+1}\left(x^k - x^{k-1}\right) + \omega_{k+1}\frac{t_k}{t_{k+1}}\frac{1+\tau_{k+1}\mu_g}{1-\tau_{k+1}\mu_f}\left(T_{\tau_k} y^k - x^k\right) \\ &= x^k + \beta_{k+1}\left(\left(x^k - x^{k-1}\right) + \frac{t_k}{t_k - 1}\left(T_{\tau_k} y^k - x^k\right)\right). \end{aligned} \tag{C2$_m$}$$

One can easily check that starting from (5) and replacing in (11) x^{k+1} by $T_\tau y^{k+1}$ with the update rule above the same computations of the previous sections carry on and the same convergence rates are obtained. Condition (C2$_m$) also suggests a natural choice for x^k. In fact, one can simply set

$$x^k = \begin{cases} T_{\tau_k}(y^k) & \text{if } F(T_{\tau_k} y^k) \leq F(x^{k-1}), \\ x^{k-1} & \text{otherwise,} \end{cases} \tag{43}$$

so that in either case one of the two terms in (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 recently been considered in [30], where, despite the further computational costs required to compute the value x^k, an empirical linear convergence rate is also observed for the standard FISTA applied to strongly convex objectives. The rigorous proof of such a convergence property is an interesting question for future research.

5. Numerical examples.

In this section we report some numerical experiments to confirm numerically the convergence result of Algorithm 2 in Theorem 4.6. We also discuss some heuristic restarting strategies [24] in the case when the strong convexity parameters are unknown.

5.1. TV-Huber ROF denoising.

We start by considering a strongly convex variant of the well-known Rudin–Osher–Fatemi (ROF) image denoising model [25] based on the use of total variation (TV) regularization. In its discretized form and for a given noisy image $u^0 \in \mathbb{R}^{m\times n}$ corrupted by Gaussian noise with zero mean and variance σ^2, the original ROF model reads

$$\min_u \ \lambda\|Du\|_{p,1} + \frac{1}{2}\|u-u^0\|_2^2. \tag{44}$$

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

$$\|Du\|_{p,1} = \sum_{i=1}^{m}\sum_{j=1}^{n} |(Du)_{i,j}|_p = \sum_{i=1}^{m}\sum_{j=1}^{n} \left((Du)_{i,j,1}^p + (Du)_{i,j,2}^p\right)^{1/p}, \tag{45}$$

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 regularization parameter $\lambda > 0$ in (44) weights the action of TV-regularization against the fitting with the Gaussian data given by the ℓ^2 term.

Taking $p=2$ in (45), we now follow [11, Examples 4.7 and 4.14] and consider a similar denoising model where a strongly convex variant of TV is employed. This can be obtained, for instance, using the C^1-Huber smoothing function $h_\varepsilon : \mathbb{R}\to\mathbb{R}$ defined for a parameter $\varepsilon > 0$ 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 (45) removes the singularity in a neighborhood of zero by means of a quadratic term and leaves the TV term almost unchanged otherwise. The resulting Huber–ROF image denoising model then reads

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

with

$$H_\varepsilon(u) := \sum_{i=1}^{m}\sum_{j=1}^{n} h_\varepsilon\left(\sqrt{(Du)_{i,j,1}^2 + (Du)_{i,j,2}^2}\right). \tag{46}$$

The dual problem of (46) reads

$$\min_{\boldsymbol{p}} \ \frac{1}{2}\|D^*\boldsymbol{p} - u^0\|_2^2 + \frac{\varepsilon}{2\lambda}\|\boldsymbol{p}\|_2^2 + \delta_{\{\|\cdot\|_{2,\infty}\le\lambda\}}(\boldsymbol{p}), \tag{47}$$

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

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

Note that (47) is the sum of a function f with Lipschitz gradient and a nonsmooth function g, which are respectively given by

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

The gradient of the differentiable component f reads

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

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

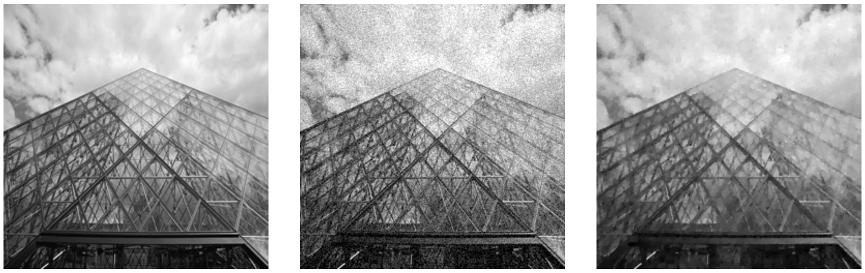
The function g is strongly convex with parameter $\mu_g = \mu = \varepsilon/\lambda$ and its proximal map $\hat{\boldsymbol{p}} = \operatorname{prox}_{\tau g}(\tilde{\boldsymbol{p}})$ can be easily computed pixelwise as

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

since, due to the general properties of proximal maps with added squared ℓ^2 terms (see Lemma A.3), it holds that

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

Note that the same example has also been considered for similar verifications in [15, section 4.2]: our results are in fact in good agreement with the ones reported therein.

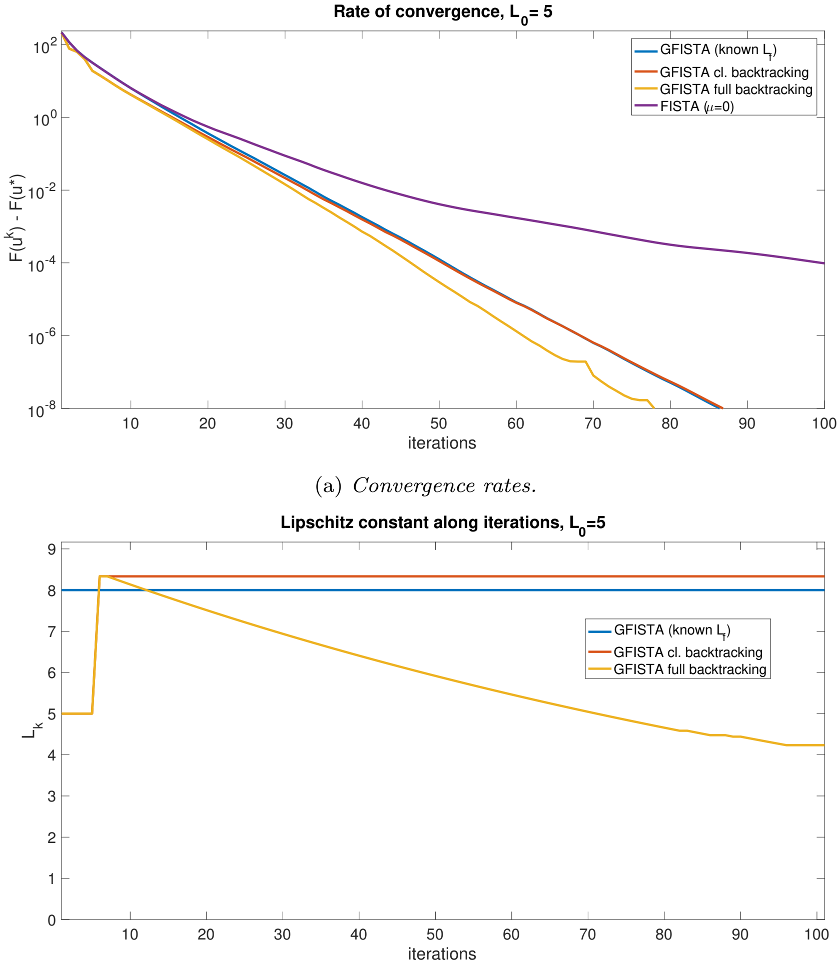


(a) *Original image.* (b) *Noisy version.* (c) *Denoised version.*

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

Parameters. In the following experiments we consider an image $u^0 \in \mathbb{R}^{m\times n}$ with $m = n = 256$ corrupted by Gaussian noise with zero mean and $\sigma^2 = 0.005$; see subfigures (a) and (b) of Figure 1. We set the Huber parameter $\varepsilon = 0.01$ and the regularization parameter $\lambda = 0.1$, so that $\mu_g = \mu = 0.1$. In our comparisons we use 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 4.3, i.e., we use the modified update rules (C2$_m$) and (43). For comparison, we report numerical results where the backtracking strategy is used "classically," i.e., it allows only for increasing of the Lipschitz constant estimate L_k, and "adaptively," i.e., it allows for both increasing and decreasing along the iterations. The backtracking factor ρ is set as $\rho = 0.9$. The initial value is set as $t_1 = 1$. The algorithm is initialized by the gradient of the noisy image u^0, i.e., $\boldsymbol{p}_0 = Du^0$.

To compute an approximation of the optimal solution u^*, we let the plain GFISTA run beforehand for 5000 iterations and store the result for comparison; see Figure 1. We then compute the results by running Algorithms 1 and 2 for `iter` $= 100$ iterations. We report the results computed for two different choices of L_0 that underestimate and overestimate the actual value of L_f, respectively; see Figures 2 and 3.[1] For comparison, we further report the $O(1/k^2)$ convergence rate of the standard FISTA with no strongly convex parameter ($\mu = 0$) encoded.



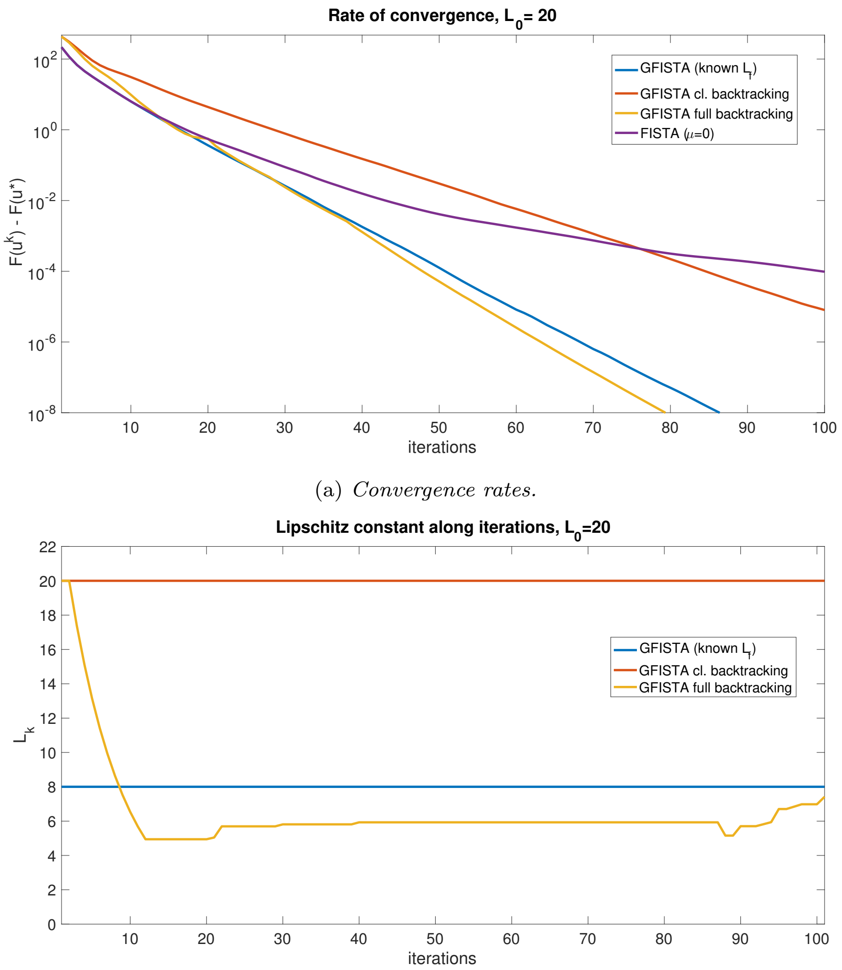
(a) *Convergence rates.*

(b) *Lipschitz constant estimate.*

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

5.2. Strongly convex TV Poisson denoising. In this second example we consider a different denoising model for images corrupted by Poisson noise, which

[1] Color figures are available in the online version of this paper.



(a) *Convergence rates.*

(b) *Lipschitz constant estimate.*

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

is commonly observed in microscopy and astronomy imaging applications. Standard Poisson denoising models using TV regularization are typically combined with a convex, nondifferentiable Kullback–Leibler data-fitting term, which can be consistently derived from the Bayesian formulation of the problem via MAP estimation (see, e.g., [27]). Here, we follow [9] and consider a differentiable version of the Kullback–Leibler data term which, for a given positive noisy image $u^0 \in \mathbb{R}^{m\times n}$ corrupted by Poisson noise, reads

(48)

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

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

$$L_f = \max_{i,j} \frac{u^0_{i,j}}{b^2_{i,j}}, \tag{49}$$

which is well defined, positive, and finite as long as u^0 and b are positive. As a regularization term, we will consider the following ε-strongly convex variant of isotropic TV in (45):

$$g(u) = \lambda \|Du\|_{2,1} + \frac{\varepsilon}{2}\|u\|_2^2, \tag{50}$$

where $\lambda > 0$ again stands for the regularization parameter. Unlike the Huber-TV ROF example, we aim here to apply Algorithm 2 to solve the composite problem

$$\min_u \ \lambda \|Du\|_{2,1} + \frac{\varepsilon}{2}\|u\|_2^2 + \widetilde{KL}(u_0, u) \tag{51}$$

in primal form.

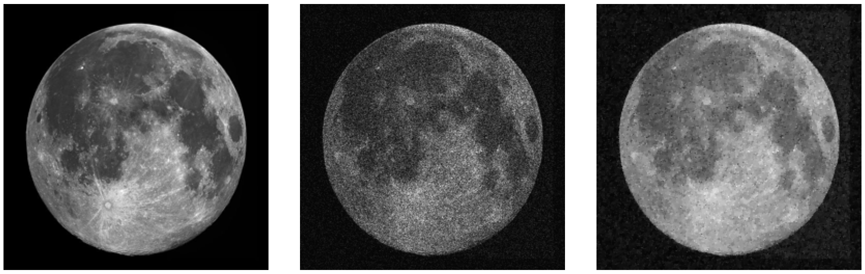
The gradient of the KL term in (48) can be computed easily and the proximal map of g in (50) can be computed using the proximal map of the TV functional due to a general property reported in Lemma A.3, so that, recalling the definition (4), for any z it holds that

$$\text{prox}_{\tau g}(z) = \text{prox}_{\|\cdot\|_{2,1}}^{\frac{\lambda\tau}{1+\varepsilon\tau}} \left(\frac{z}{1+\varepsilon\tau} \right). \tag{52}$$

Thus, for any $\tau > 0$, computing the right-hand side of the equality above corresponds simply to solving the classical ROF problem with regularization parameter $\sigma := \frac{\lambda\tau}{1+\tau\varepsilon}$. We do this using the standard FISTA as an iterative inner solver.

Parameters. We consider an image $u^0 \in \mathbb{R}^{m\times n}$ with $m = n = 256$ corrupted artificially by Poisson noise; see parts (a) and (b) of Figure 4. 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 regularization parameter $\lambda = 0.1$. Clearly, $\mu = \mu_g = \varepsilon$. In order to compute the proximal map (52) we use 10 iterations of the standard FISTA. In the following example the Lipschitz constant of the gradient of the $\widetilde{KL}$ term can be estimated via (49) 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$_m$) and (43)), for which the factor $\rho = 0.8$ is chosen. The initial value is set as $t_1 = 1$. The algorithm is initialized using the given noisy image u^0.

An approximation of the solution u^* is computed beforehand by letting the plain FISTA run for 5000 iterations and then storing it for comparison; see Figure 4(c). Results are then computed by letting the monotone version of GFISTA 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 in GFISTA, linear convergence is achieved, as opposed to the slower convergence of standard FISTA. Furthermore, the local estimate of the Lipschitz constant provided by the full backtracking strategy decreases along the iterations, thus allowing for larger gradient steps and convergence in fewer



(a) *Original.* (b) *Noisy version.* (c) *Denoised version.*

FIG. 4. *Original, noisy, and restored images computed using the strongly convex TV-Poisson denoising model* (51)*. The regularization parameter is $\lambda = 0.2$ and the strong convexity parameter is $\mu = \varepsilon = 0.15$.*

iterations. In Figure 6, we plot the monotone decay of the energy along GFISTA iterates (with and without backtracking) after the monotone modification described in section 4.3.

5.3. Restarting strategies applied to the elastic net. In this final example we test the performance of GFISTA with backtracking (Algorithm 2) in the case when a prior estimate of the strong convexity parameters μ_f and/or μ_g is either misspecified or not available. As a test problem we consider the elastic net regularization model, which, for a given matrix $A \in \mathbb{R}^{m\times m}$, data $y \in \mathbb{R}^m$, and positive parameters λ_1 and λ_2 reads

$$\min_u \left\{ F(u) := \frac{1}{2}\|Au - y\|_2^2 + \lambda_1\|u\|_1 + \frac{\lambda_2}{2}\|u\|_2^2 \right\}. \tag{53}$$

The elastic net is commonly used as a regularized version of the least absolute shrinkage and selection operator (LASSO) estimator by means of a ridge-type quadratic term and it is employed for several parameter identification problems [34] and support vector machine problems [33]. In order to apply Algorithm 2, we split the functional F above as the sum

$$f(u) := \frac{1}{2}\|Au - y\|_2^2 + \frac{\lambda_2}{2}\|u\|_2^2, \qquad g(u) := \lambda_1\|u\|_1. \tag{54}$$

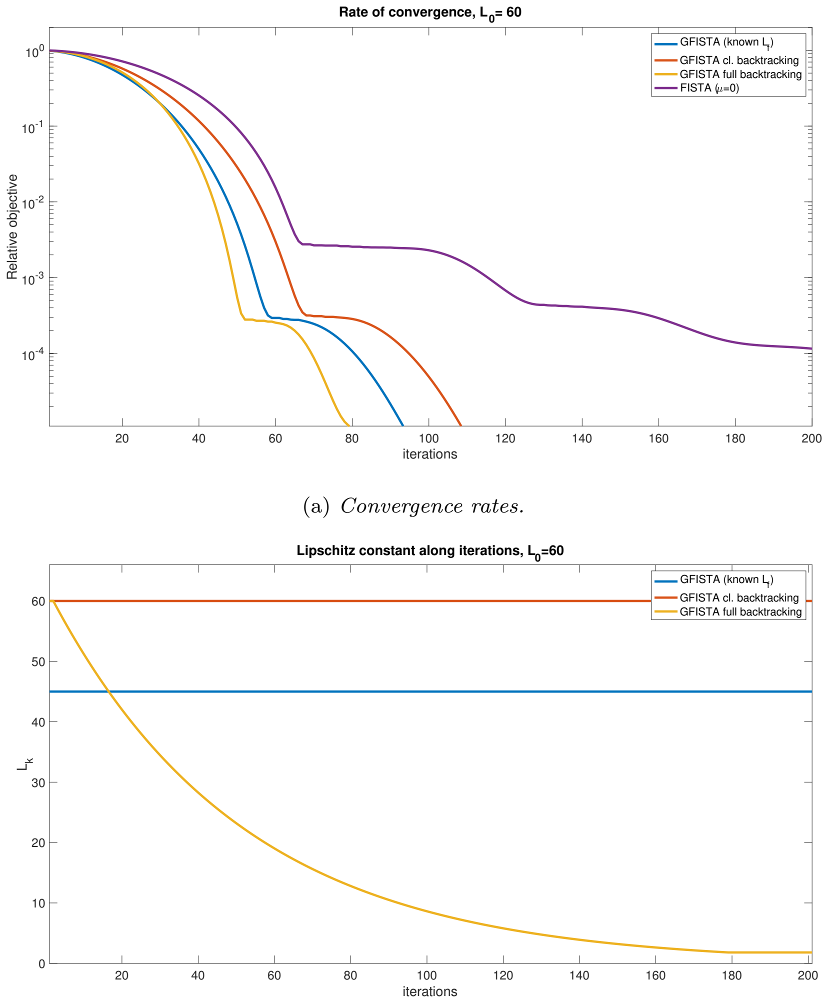
Under this choice, we note that f is differentiable, with Lipschitz-continuous gradient given by

$$\nabla f(u) = A^*(Au - y) + \lambda_2 u.$$

The Lipschitz constant can be calculated as $L_f = \lambda_{\max}(A^*A + \lambda_2\mathbb{I})$, where we denote by $\lambda_{\max}(M)$ the largest eigenvalue of the matrix M. Note that, in the case of large-size problems ($m \gg 1$), such computation of L_f may be prohibitively expensive from a computational point of view. The nonsmooth function g is convex, and for $\tau > 0$ its proximal map can be calculated componentwise by the soft-thresholding operator as

$$(\mathrm{prox}_{\tau g}(z))_i = \mathrm{sign}(z_i)\max\left(|z_i| - \tau\lambda_1, 0\right), \qquad i = 1, \dots, m.$$

Finally, note that f is λ_2-strongly convex, so that $\mu = \mu_f = \lambda_2$.



(a) *Convergence rates.*

(b) *Lipschitz constant estimate.*

FIG. 5. *Convergence rates and backtracking of the Lipschitz constant of* ∇f *in* (48) *starting from the overestimating initial value* $L_0 = 60$. *Rates are shown in terms of the relative objective functional* $\frac{F(u^k)-F(u^*)}{F(u^0)-F(u^*)}$.

Parameters. In the following experiments we solve problem (53) using to a normalized randomly generated operator $A \in \mathbb{R}^{3600\times 3600}$ and for parameters λ_1, λ_2 set as $\lambda_1 = 0.01$ and $\lambda_2 = 10^{-5}$, so that $\mu = \mu_f = \lambda_2$. The Lipschitz constant L_f of ∇f can be estimated in this example as $L_f = 0.0657$. For the backtracking routine, we set the backtracking factor $\rho = 0.95$. Algorithm 2 is initialized with $t_1 = 1$, $L_0 = 1$, and $x_0 = \mathbf{0}$. The plain GFISTA (Algorithm 1) without backtracking is run for 5000 iterations and its solution x^* is stored for comparison. The following results are computed by running the algorithm for `iter` = 100 iterations.

In the first test, we compare once again the performance of GFISTA with backtracking (Algorithm 2) when the prior estimate of L_f is available and when it is not, using both standard Armijo-type backtracking and the adaptive one proposed in this work; see Figure 7. Compared to the examples considered above, note that in this

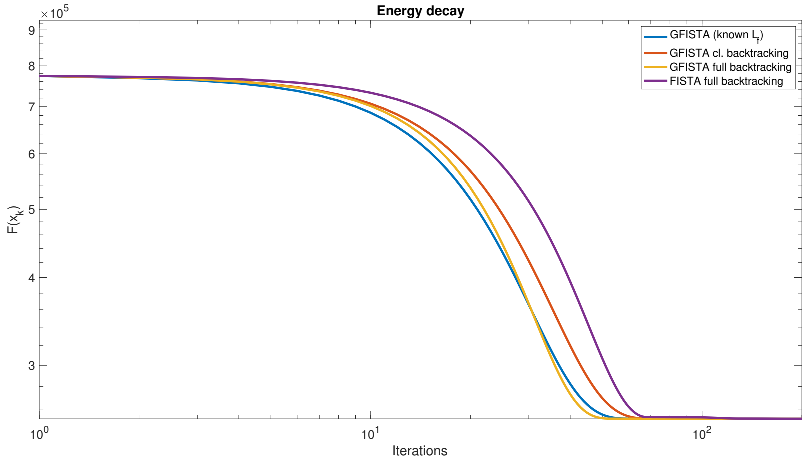


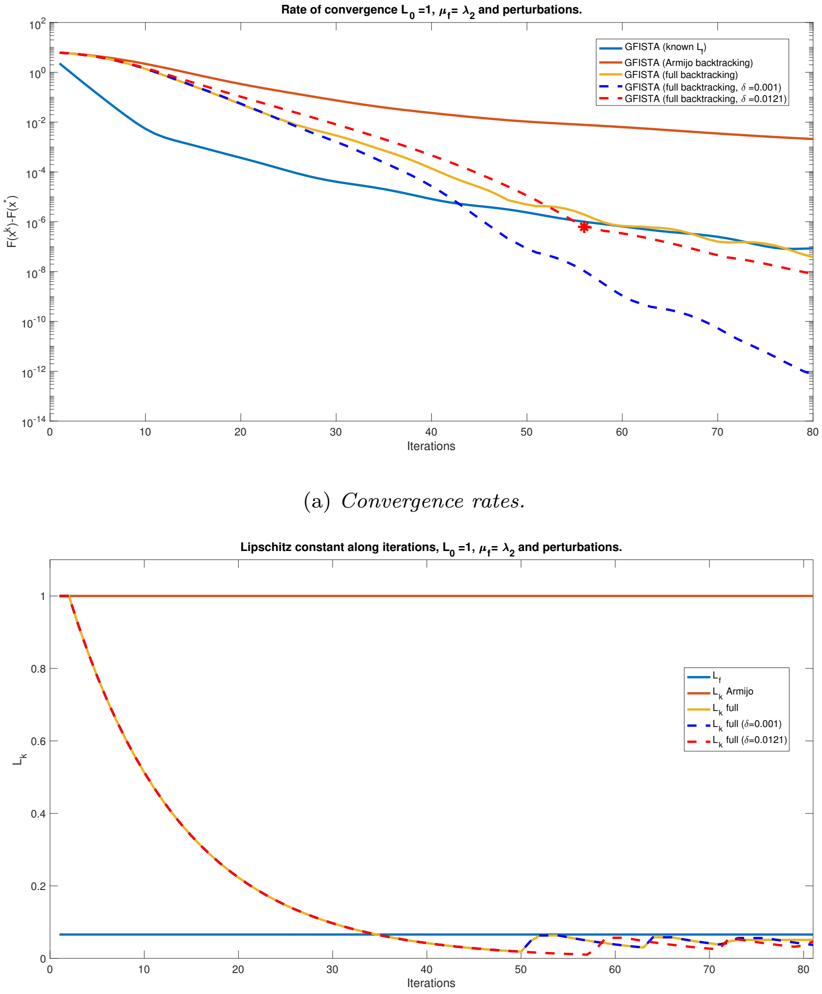
Fig. 6. *Monotone decay along GFISTA iterates (with and without backtracking) after the monotone modifications in* (C2$_m$) *and* (43).

case the strong convexity constant of the problem is encoded in the term f defined in (54), which is accommodated by our strategy. Note, however, that it is typically more efficient encoding strong convexity in the nonsmooth component g, which is treated implicitly, rather than in f, which is treated explicitly. The latter choice would in fact require more restrictive time steps $\tau \leq 1/(L_f + \mu_f)$.

In addition, we also report the results obtained when a "wrong" value of μ_f is used. Given its quadratic behavior, one may in fact suppose that, in addition to the λ_2-strong convexity, some further strong convexity could be hidden in the quadratic data-fitting term. In the following, we report the results obtained by applying Algorithm 2 with full backtracking for a perturbed value of μ_f given by $\mu_f = \lambda_2 + \delta$ for a small perturbation $0 < \delta \ll 1$. Note that under such modification the natural condition $\mu_f < L_k$ may be violated along the iterations, thus preventing the algorithm from converging. Whenever this happens, we decrease the value μ_f by a factor ρ, redefine the term q_k appearing in Algorithm 2 in correspondence to this new value, and carry on with the algorithm. In this way convergence is always guaranteed and also large misspecifications of μ_f can be treated.

Provided such verification is performed along the iterations, these tests suggest that encoding further, hidden, strong convexity information in the model (53) can improve the convergence rates of Algorithm 2.

Motivated by these considerations, we perform in the following a further numerical test where we assume that the values of the strong convexity parameters μ_f and μ_g (and, consequently, μ) are unknown. In several applications, it is actually very hard to provide an explicit estimation of such parameters. Moreover, as we have seen in the examples above, some hidden strong convexity can still not be detected explicitly by only looking at the structure of the functions f and g. An indirect way to estimate strong convexity consists in restarting the algorithm depending on a certain criterion; see, e.g., [23]. In [24] two heuristic restarting procedures based on the evaluation of the composite functional and of a (generalized) gradient are studied. These two restarting approaches have become very popular and, more recently, some others have been proposed, for instance, in [19, 13]. Here, we follow [24] and apply the two function- and



(a) *Convergence rates.*

(b) *Lipschitz constant estimate.*

FIG. 7. *Convergence rates and backtracking of the Lipschitz constant of* ∇f *in* (54) *starting from the overestimating initial value* $L_0 = 1$. *In the convergence plot, solid lines refer to the case when* $\mu_f = \lambda_2$, *while dashed lines refer to "wrong" values of* μ_f, *which is perturbed as* $\mu_f = \lambda_2 + \delta$. *In the red dashed line we also scatter the point corresponding to the iteration violating the condition* $\mu_f < L_k$ *along the iterations, which requires the reduction of* μ_f.

gradient-based restarting procedures to Algorithm 2 with full backtracking to solve the elastic net problem above under the same choice of parameters as above and assuming no prior knowledge on the values of μ_f and μ_g, i.e., setting $\mu = 0$ in the following. As discussed in [24, section 5.2] the two restarting criteria to consider for FISTA-type algorithms are the following.

- *Function adaptive restart*: restart the algorithm whenever

$$F(u^{k+1}) > F(u^k). \tag{55}$$

- *Gradient adaptive restart*: restart the algorithm whenever

$$(y^k - u^{k+1})^T(u^{k+1} - u^k) > 0. \tag{56}$$

Compared to the function-based restarting scheme, the gradient adaptive restart is observed to be more stable around x^*. Furthermore, there is no extra computational cost in applying such restarting to Algorithm 2 since all the quantities appearing in (56) have already been calculated during the backtracking phase. We remark that this second approach goes under the name of "gradient" restart since one can interpret, for each $k \geq 0$, the FB step (28) in Algorithm 2 as a *generalized* gradient step defined by

$$x^{k+1} = \mathrm{prox}_{\tau_{k+1} g}(y^k - \tau_{k+1}\nabla f(y^k)) =: y^k - \tau_{k+1}G(y^k).$$

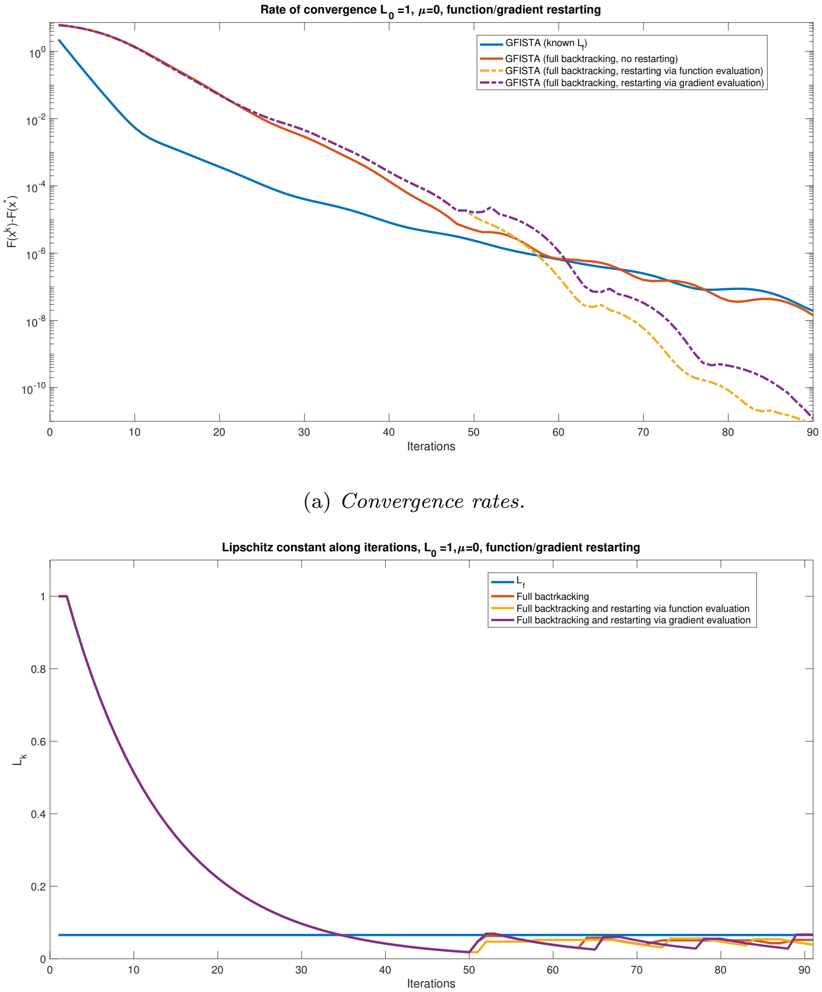
The restarting condition (56) would in this case read $G(y^k)^T(u^{k+1} - u^k) > 0$. In Figure 8, we report the convergence plots and the Lipschitz constant variations for the solution of the elastic net problem (53) via GFISTA (Algorithm 2) with full backtracking combined with the two restarting strategies above. We observe a faster linear convergence compared to the fully backtracked GFISTA, which, heuristically, can therefore be adapted and efficiently employed for a strongly convex problem with no prior estimate on the strong convexity constant μ. A rigorous proof of these convergence results is left for future research.

6. Conclusions and outlook.

We studied 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 properties of strongly convex functions and upon multiplication by appropriate terms, we derived in section 4 the convergence estimate (22), whose decay factor (23) was 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 [21] and on general properties of the extrapolation sequences defined. Our main result is reported in Theorem 4.6, where accelerated linear convergence rates are proved in terms of average quantities depending on the estimated values along the iterations. Our theoretical results are verified numerically on some exemplar problems in section 5.

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 L_k 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 to provide strict decay in the convergence inequality (22) and defined not only in terms of the step sizes but also in terms of the strong convexity parameters of f and g, resulting in more refined convergence rate estimates. Finally, our approach has a lower per-iteration computational cost than the one studied by Nesterov in [23], since it avoids calculation of the gradient of the smooth component in the proximal step. Accelerated convergence rates were proved and defined in terms of average quantities depending on the estimates performed along the iterations.

Further research should focus on the analysis of the proposed backtracking approach combined with the restarting procedures à la Candès used in section 5.3 for situations when the strong convexity parameters μ_f and μ_g are unknown. In this work we showed heuristically good performance only for the case of function- and gradient-based restarting procedures, but it would be of great interest to also further explore the recently proposed approaches by Fercoq and Qu [13] where the restarting



(a) *Convergence rates.*

(b) *Lipschitz constant estimate.*

FIG. 8. *Convergence rates and backtracking of the Lipschitz constant of* ∇f *in* (54) *with and without restarting based on function* (55) *and gradient* (56) *criteria. Initial (overestimating) value* $L_0 = 1$ *and* $\mu = 0$.

does not require any condition but combines past iterates of the algorithm appropriately. A rigorous analysis of a combined backtracking-restarting procedure would be very interesting for designing an algorithm that is fully adaptive to local convexity and smoothness of its functions.

Finally, it would be interesting to test the robustness and the performance of our algorithm on other strongly convex, possibly large-scale problems coming from the fields of image and data analysis with various condition numbers.

Appendix A. Some useful lemmas. In this appendix we prove some general results which have been used in our work. We start with a general inequality used to derive the descent rule (5). Its proof is a consequence of a trivial property of strongly convex functions.

Lemma A.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 minimizer of h, the following property holds:*

$$h(x) \geq h(\hat{x}) + \frac{\mu_h}{2}\|x - \hat{x}\|^2 \tag{57}$$

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

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

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

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 (57). ∎

An immediate consequence of this general property is the proof of the descent rule (5) used in section 3 as a starting point for our convergence estimates. We follow [11, 31].

Lemma A.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 an 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 $F = f + g$:*

$$F(x) + (1 - \tau\mu_f)\frac{\|x - \bar{x}\|^2}{2\tau} \geq F(\hat{x}) + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau} \quad \textit{for any } x \in \mathcal{X}. \tag{58}$$

Proof. By definition, $\hat{x}$ is the minimizer 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 $x \in \mathcal{X}$,

$$\begin{aligned}
&F(x) + (1 - \tau\mu_f)\frac{\|x - \bar{x}\|^2}{2\tau} \\
&\quad\geq g(x) + f(\bar{x}) + \langle\nabla f(\bar{x}), x - \bar{x}\rangle + \frac{\|x - \bar{x}\|^2}{2\tau} \\
&\quad\geq g(\hat{x}) + f(\bar{x}) + \langle\nabla f(\bar{x}), \hat{x} - \bar{x}\rangle + \frac{\|\hat{x} - \bar{x}\|^2}{2\tau} + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau} \\
&\quad\geq g(\hat{x}) + f(\hat{x}) + \frac{1 - \tau L_f}{2\tau}\|\hat{x} - \bar{x}\|^2 + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau} \\
&\quad= F(\hat{x}) + \frac{1 - \tau L_f}{2\tau}\|\hat{x} - \bar{x}\|^2 + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau}, \qquad (59)
\end{aligned}$$

where the first inequality holds by strong convexity of f, the second one is a simple application of Lemma 57, 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 (59) and get (58). ∎

We finally report a general property of proximal mappings that 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. We recall the notation in (4).

LEMMA A.3. *Let $h : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ be a convex, proper, and l.s.c. function. For $\alpha, \varepsilon > 0$ let*

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

Then, it holds that

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

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

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

Acknowledgment. The authors would like to thank the anonymous referees, whose valuable comments significantly improved the quality of the manuscript.

REFERENCES

[1] L. ARMIJO, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3, http://dx.doi.org/10.2140/pjm.1966.16.1.

[2] J.-F. AUJOL AND C. DOSSAL, *Stability of over-relaxations for the forward-backward algorithm, application to FISTA*, SIAM J. Optim., 25 (2015), pp. 2408–2433.

[3] A. BECK AND M. TEBOULLE, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Trans. Image Process., 18 (2009), pp. 2419–2434.

[4] A. BECK AND M. TEBOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[5] S. BONETTINI, F. PORTA, AND V. RUGGIERO, *A variable metric forward-backward method with extrapolation*, SIAM J. Sci. Comput., 38 (2016), pp. A2558–A2584.

[6] M. BURGER, A. SAWATZKY, AND G. STEIDL, *First-order algorithms in variational image processing*, in Splitting Methods in Communication, Imaging, Science, and Engineering, R. Glowinski, S. Osher, and W. Yin, eds., Sci. Comput., Springer, Cham, 2016, pp. 345–407, https://doi.org/10.1007/978-3-319-41589-5_10.

[7] A. CHAMBOLLE, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97, https://doi.org/10.1023/B:JMIV.0000011325.36760.1e.

[8] A. CHAMBOLLE AND C. DOSSAL, *On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm"*, J. Optim. Theory Appl., 166 (2015), pp. 968–982.

[9] A. CHAMBOLLE, M. EHRHARDT, P. RICHTÁRIK, AND C. SCHÖNLIEB, *Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications*, SIAM J. Optim., 28 (2018), pp. 2783–2808, https://doi.org/10.1137/17M1134834.

[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 J. Comput. Math., 1 (2015), pp. 29–54.

[11] A. Chambolle and T. Pock, *An introduction to continuous optimization for imaging*, Acta Numer., 25 (2016), pp. 161–319, https://doi.org/10.1017/S096249291600009X.

[12] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200, https://doi.org/10.1137/050626090.

[13] Q. Fercoq and Z. Qu, *Restarting Accelerated Gradient Methods with a Rough Strong Convexity Estimate*, preprint, 2016, https://arxiv.org/pdf/1609.07358.

[14] M. I. Florea and S. Vorobyov, *A Generalized Accelerated Composite Gradient Method: Uniting Nesterov's Fast Gradient Method and FISTA*, preprint, 2017, https://arxiv.org/abs/1705.10266.

[15] M. I. Florea and S. A. Vorobyov, *An accelerated composite gradient method for large-scale composite objective problems*, IEEE Trans. Signal Process., 67 (2019), pp. 444–459, https://doi.org/10.1109/TSP.2018.2866409.

[16] A. A. Goldstein, *Convex programming in Hilbert space*, Bull. Amer. Math. Soc., 70 (1964), pp. 709–710, https://doi.org/10.1090/S0002-9904-1964-11178-2.

[17] O. Güler, *New proximal point algorithms for convex minimization*, SIAM J. Optim., 2 (1992), pp. 649–664, https://doi.org/10.1137/0802032.

[18] H. Lin, J. Mairal, and Z. Harchaoui, *A universal catalyst for first-order optimization*, in Adv. Neural Inf. Process. Syst. 28, Curran Associates, Red Hook, NY, 2015, pp. 3384–3392; available at https://papers.nips.cc/paper/5928-a-universal-catalyst-for-first-order-optimization.pdf.

[19] Q. Lin and L. Xiao, *An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization*, Comput. Optim. Appl., 60 (2015), pp. 633–674, https://doi.org/10.1007/s10589-014-9694-4.

[20] Y. Nesterov, *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk, 269 (1983), pp. 543–547 (in Russian); Sov. Math. Dokl., 27 (1983), 372–367 (in English).

[21] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Springer, New York, 2004.

[22] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152, https://doi.org/10.1007/s10107-004-0552-5.

[23] Y. Nesterov, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161, https://doi.org/10.1007/s10107-012-0629-5.

[24] B. O'Donoghue and E. Candès, *Adaptive restart for accelerated gradient schemes*, Found. Comput. Math., 15 (2015), pp. 715–732, https://doi.org/10.1007/s10208-013-9150-3.

[25] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268, https://doi.org/10.1016/0167-2789(92)90242-F.

[26] S. Salzo and S. Villa, *Inexact and accelerated proximal point algorithms*, J. Convex Anal., 19 (2012), pp. 1167–1192.

[27] A. Sawatzky, *(Nonlocal) Total Variation in Medical Imaging*, Ph.D. thesis, University of Münster, Germany, 2011.

[28] K. Scheinberg, D. Goldfarb, and X. Bai, *Fast first-order methods for composite convex optimization with backtracking*, Found. Comput. Math., 14 (2014), pp. 389–417, https://doi.org/10.1007/s10208-014-9189-9.

[29] M. Schmidt, N. Roux, and F. Bach, *Convergence rates of inexact proximal-gradient methods for convex optimization*, in Adv. Neural Inf. Process. Syst. 24, Curran Associates, Red Hook, NY, 2011, pp. 1458–1466; available at http://books.nips.cc/papers/files/nips24/NIPS2011_0839.pdf.

[30] S. Tao, D. Boley, and S. Zhang, *Local linear convergence of ISTA and FISTA on the LASSO problem*, SIAM J. Optim., 26 (2016), pp. 313–336.

[31] P. Tseng, *On Accelerated Proximal Gradient Methods for Convex-Concave Optimization*, Working paper, University of Washington, Seattle, WA, 2008; available at http://www.csie.ntu.edu.tw/~b97058/tseng/papers/apgm.pdf.

[32] S. Villa, S. Salzo, L. Baldassarre, and A. Verri, *Accelerated and inexact forward-backward algorithms*, SIAM J. Optim., 23 (2013), pp. 1607–1633.

[33] L. Wang, J. Zhu, and H. Zou, *The doubly regularized support vector machine*, Statist. Sinica, 16 (2006), pp. 589–615.

[34] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 67 (2005), pp. 301–320.