Adaptive Fista

Peter Ochs and Thomas Pock

November 15, 2017

Abstract

In this paper we propose an adaptively extrapolated proximal gradient method, which is based on the accelerated proximal gradient method (also known as FISTA), however we locally optimize the extrapolation parameter by carrying out an exact (or inexact) line search. It turns out that in some situations, the proposed algorithm is equivalent to a class of SR1 (identity minus rank 1) proximal quasi-Newton methods. Convergence is proved in a general non-convex setting, and hence, as a byproduct, we also obtain new convergence guarantees for proximal quasi-Newton methods. In case of convex problems, we can devise hybrid algorithms that enjoy the classical $O(1/k^2)$-convergence rate of accelerated proximal gradient methods. The efficiency of the new method is shown on several classical optimization problems.

1 Introduction

The introduction of accelerated gradient methods by Nesterov in [37] has arguably revolutionized the world of large-scale convex and non-smooth optimization. Computationally, they are as simple and efficient as plain gradient descent, but come along with a much faster rate of convergence. Therefore, they found numerous applications in modern signal and image processing, and machine learning. It is well-known that these accelerated gradient methods are "optimal" for the class of smooth and convex (not necessarily strongly-convex) problems with Lipschitz continuous gradient, in the sense that their worst case complexity is proportional to the theoretical lower complexity bound of first-order methods for this class of problems [36, 38]. The mechanism of accelerated gradient methods can be interpreted and explained in many different ways and hence, the magic of acceleration is still subject of intensive research efforts.

Accelerated gradient methods can be seen as a modification of the heavy-ball method of Polyak [45]. While the heavy-ball method already achieves an optimal convergence rate on smooth and strongly convex functions with Lipschitz continuous gradient, accelerated gradient methods are optimal for the complete class of smooth convex problems with Lipschitz continuous gradient [38].

In the original work [38], the acceleration is explained by the concept of so called estimation sequences, which generate a sequence of simple convex functions approximating the original function. Accelerated gradient methods also show strong connections to the Störmer–Verlet method [27] for discretizing second order ordinary differential equations (ODEs). Using this connection, both the heavy-ball method [45] and Nesterov's original method [37] can be seen as particular types of discrete-time approximations to the "heavy-ball with friction" dynamical system. The main principle here is to attach a "mass" to the sequence of points generated which accelerates when moving down the landscape of the objective function. More recent works use this relation to investigate the properties of accelerated gradient methods in the framework of ODEs [49, 2]. Finally, accelerated gradient methods can also be explained by accelerated primal-dual methods that are based on an optimal choice of dynamic step sizes in the primal and dual spaces [12].

For quadratic functions, both the heavy-ball method and accelerated gradient methods show striking similarities to the conjugate gradient method [28]. However, while in the conjuagte gradient method, the step size and extrapolation parameters are chosen locally using an exact line (or plane) search, accelerated gradient methods derive their parameters from global properties of the objective such as the smallest and largest eigenvalues of the corresponding system matrix. On the one hand, the conjugate gradient method is optimal and comes along with a finite convergence property. On the other hand, they are much harder to generalized to non-quadratic functions. At this point, accelerated gradient methods take their advantage. They can be applied not only to quadratic problems but to the whole class of one times differentiable functions with Lipschitz continuous gradient.

Accelerated gradient methods have been generalized to be applicable to a class of convex optimization problems that can be written as the sum of a differentiable function with Lipschitz continuous gradient and a non-smooth function with easy to compute proximal map. The most popular instance is the Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [4] but also other, more general schemes have been proposed. See the work of Tseng [50] for an excellent presentation and unification of a whole family of accelerated proximal gradient methods.

Finally, accelerated gradient methods have also turned out to perform very well on non-convex optimization problems. However, while their empirical performance is often comparable to the convex case, their convergence properties are still hardly understood from a theoretical point of view [25, 33, 21, 42].

In this paper, we study a modification of the accelerated proximal gradient method for minimizing an objective given by the sum of a smooth function f with Lipschitz continuous gradient and a non-smooth function g with simple proximal mapping. We still rely on a step size parameter related to the global Lipschitz parameter L, however similar to the conjugate gradient method, we propose to locally adapt the extrapolation parameter by means of an exact line search. A simplified version of the algorithm's update step is the following:

$$
\begin{aligned}
y_k^{(\beta)} &= x_k + \beta(x_k - x_{k-1}) \\
x_{k+1}^{(\beta)} &= \operatorname*{argmin}_{x\in\mathbb{R}^N} g(x) + \left\langle \nabla f(y_k^{(\beta)}), x - y_k^{(\beta)} \right\rangle + \frac{L}{2}\|x - y_k^{(\beta)}\|^2 ,
\end{aligned}
$$

where $\beta \in \mathbb{R}$ is a free parameter which is optimized in each step in order to provide the locally fastest rate of decrease of the proximal subproblem. It turns out that if the function f is quadratic, the optimal choice of β makes the scheme equivalent to a proximal SR1 (identity minus rank one) quasi-Newton method. This equivalence yields a connection between accelerated schemes and quasi-Newton methods, which holds even when g is a non-convex function. The convergence that we establish for our method in a general non-smooth non-convex setting translates directly to a class of proximal quasi-Newton methods. Unlike the general class of proximal quasi-Newton methods which usually suffer from the problem that simple proximal mappings become hard to solve, efficient solutions to the "rank-1 proximal mappings" are known [5, 30].

The remainder of the paper is organized as follows: In Section 2 we provide a state-of-the-art review and put the contributions of this paper into related work. In Section 3, we detail the proposed algorithm and give different convergence guarantees for both convex and non-convex problems. Numerical results of the proposed algorithm are provided in Section 4. In the last section we give some conclusions and discuss open problems for future research.

2 Related work

Proximal Gradient Method. The basic update step of our method is a so-called proximal gradient step (also known as forward-backward splitting) [34, 19, 18, 38]. Although, using this basic step only yields an efficient algorithm in many cases, it has been observed that the worst case complexity is not optimal [38] for certain classes of convex optimization problems. This observation led to the exploration of so-called accelerated schemes. The basic proximal gradient step can be accelerated by an additional (computationally cheap) extrapolation step [4, 37, 50] (see also [51]), where the iteration-dependent extrapolation parameter obeys a certain rule derived from global properties of the objective function. In contrast to prescribed rules, in this paper, we explicitly optimize the extrapolation parameter. Surprisingly, for a certain class of problems, the optimized scheme has a closed form expression as a proximal quasi-Newton method, which yields new convergence results.

Classical quasi-Newton Methods. Quasi-Newton methods are intensively studied in the classic context of smooth optimization problems. We refer to [40, 20, 9] for an overview and references. The basic idea of quasi-Newton methods is to successively improve a quadratic approximation to the objective function, i.e., the goal is to approximate second order information (Hessian) using combinations of first order information. Maybe the most widely known and used quasi-Newton method is BFGS [22] or its low memory variant L-BFGS [35], and its extension to bound constraints L-BFGS-B [11]. More results that motivate the acceleration governed by such a variable metric approach are [8, 10, 44].

Quasi-Newton Methods for non-smooth problems. Though, originally designed for smooth optimization problems, the BFGS method shows a good performance on non-smooth problems as well. In [52], the method was reinterpreted and analysed in the non-smooth convex setting. Theoretical guarantees of the original BFGS method for one or two dimensional non-smooth problems could be established in [31, 32]. Motivated by the original global convergence proof of Powell [46], Guo and Lewis [26] substantially extended the previous theoretical guarantees to a class of non-smooth convex problems. In general, convergence cannot be expected.

Proximal quasi-Newton Methods. Driven by the success of proximal splitting methods, the concept of quasi-Newton methods was also applied to optimization problems with more structure than just smoothness, alike the setting of Proximal Gradient Descent. The crucial aspect for the efficiency of such a variable metric Proximal Gradient Method is the evaluation of the proximal mapping, which is often expensive when coordinates are not separated in the objective. There are specific choices of the metric that allow the proximal mapping to be solved efficiently, e.g. when the metric is a diagonal matrix, which preserves the separability. Becker and Fadili [5] derived efficient solutions when the metric is of type "identity plus rank 1". The proximal step was embedded into a zero memory variant of the SR1 quasi-Newton method [40]. Unlike the BFGS-method, which updates the Hessian approximation with a matrix of rank 2, the SR1 method updates the approximation with a rank 1 matrix. Nevertheless, Nocedal and Wright [40] state the observation that the "[...] SR1 method appears to be competitive with the BFGS method". The case of an "identity minus rank 1" metric was studied in [30], which also leads to an efficiently solvable proximal mapping. In [24], interior point methods are used to solve the proximal mapping efficiently for so-called quadratic support functions.

Variable metric versions of the Proximal Gradient Method have been studied without paying special attention to efficiently solving the proximal mapping. The earliest reference is [13, Section 5]. In the broader context of monotone inclusion problems, the general framework for analyzing the convergence of variable metric methods is that of quasi-Fejér sequences [16], which was used in [17] to find a relation to primal–dual schemes. For a variable metric algorithm with mild differentiability assumption (without the usual gradient Lipschitz continuity), we refer to [47]. The convergence of forward–backward splitting with iteration dependent Bregman distances in an infinite dimensional setting, which contains the variable metric proximal step as a special case, was proved in [39].

Non-convex setting. In [14], the metric is constructed to induce a quadratic majorizer in each iteration [29]. Extensions to a block-coordinate descent version are presented in [15] and a combination with an inertial method in [41]. In [6, 7], the choice of the metric enjoys a great flexibility at the cost of an additional line search step in the algorithm. An extension of their framework to a general non-smooth first-order oracle and a flexible choice of Bregman distances was proposed in [43]. A different way to incorporate a variable metric into forward–backward splitting (FBS) was proposed in [48]. They reinterpret FBS as a gradient descent method with a variable metric, which allows them to use some of the machinery from smooth optimization. Besides accelerating optimization methods by a variable metric, accelerated (optimal) gradient methods can be used in the non-convex setting [25, 33, 21] as well, where the goal is to preserve the optimality for convex problems.

2.1 Contribution

Convergence of an optimally extrapolated Proximal Gradient Method (PGM). We study a novel, optimally extrapolated PGM for non-convex optimization problems. If the function is composed of a continuously differentiable function with Lipschitz continuous gradient and a non-smooth function, we prove subsequential convergence to a stationary point and convergence of the objective values. Restricted to convex optimization problems, we propose two variants of our method with an accelerated convergence rate for the functional residual of $O(1/k^2)$ where k is the iteration counter.

Convergence of a non-convex SR1 proximal quasi-Newton method. In another special non-convex setting, if the objective is the sum of a quadratic function and a non-smooth (possibly non-convex) function, we prove our method to be equivalent to a proximal quasi-Newton variant of the SR1 quasi-Newton method [40]. To be more precise, we prove the equivalence between an optimally rank-r extrapolated Proximal Gradient Method and an identity minus rank-r proximal quasi-Newton method. Our convergence results can be applied directly to these methods, leading to the first convergence result of this SR1 proximal quasi-Newton for non-convex optimization problems.

Relation to [5] and [30]. A related proximal quasi-Newton method was considered by Becker and Fadili [5]. However, they adapt a Barzilai–Borwein step length, whereas we relate the step length to the Lipschitz constant. Moreover, the proposed quasi-Newton metric in the evaluation of the proximal mapping is of type "identity plus a rank 1 matrix", whereas ours is of type "identity minus rank 1", and we allow for non-convex functions in the proximal mapping. Another closely related approach is that of Karimi and Vavasis [30], which applies to problems that are the sum of a convex quadratic and a convex non-smooth function. The considered proximal mapping is also of type "identity minus rank 1", however the relation to the SR1 metric [40, Section 8.2] is unclear and the analysis is completely different to ours.

3 The optimization problem

The setting of this paper is that of a Euclidean vector space $\mathbb{R}^N$ of dimension N equipped with the standard inner product $\langle\cdot,\cdot\rangle$ and induced norm $\|x\| = \sqrt{\langle x, x\rangle}$ for $x \in \mathbb{R}^N$. Moreover, we use the notation $\|x\|_{\boldsymbol{M}}^2 := \langle x, x\rangle_{\boldsymbol{M}} := \langle x, \boldsymbol{M}x\rangle$ for a matrix $\boldsymbol{M} \in \mathbb{R}^{N\times N}$.

Optimization problem. We consider optimization problems of the following form:

$$\min_{x\in\mathbb{R}^N} f^g(x)\,, \quad f^g(x) := g(x) + f(x)\,, \tag{1}$$

where $f\colon \mathbb{R}^N \to \mathbb{R}$ is a continuously differentiable function with ∇f being L-Lipschitz continuous, $g\colon \mathbb{R}^N \to \overline{\mathbb{R}}$, $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$, is a proper lower semi-continuous (lsc) function, and f^g is bounded from below. We solve the problem by sequential minimization of surrogate functions alike the proximal gradient method (Forward–Backward Splitting).

The update step. Let $\bar{x} \in \mathbb{R}^N$ be the current point of the iterative scheme and $\{d_1, \ldots, d_m\}$ be vectors in $\mathbb{R}^N$ and

$$y^{(\boldsymbol{\beta})} := \bar{x} + \sum_{i=1}^{m} \beta_i d_i \tag{2}$$

for $\boldsymbol{\beta} := (\beta_1, \ldots, \beta_m) \in \mathbb{R}^m$. In matrix–vector notation, we have $y^{(\boldsymbol{\beta})} = \bar{x} + \boldsymbol{D}\boldsymbol{\beta}$ where $\boldsymbol{D} = (d_1, \ldots, d_m) \in \mathbb{R}^{N\times m}$ contains the vectors $d_1, \ldots, d_m$ as columns. The **optimal next iterate** $(\hat{x}, \hat{\boldsymbol{\beta}})$ is computed as follows:

$$\begin{aligned} \hat{x} = \operatorname*{argmin}_{x\in\mathbb{R}^N} \min_{\boldsymbol{\beta}\in\mathbb{R}^m} \; & \ell_f^g(x; y^{(\boldsymbol{\beta})}) + \frac{1}{2}\|x - y^{(\boldsymbol{\beta})}\|_{\boldsymbol{T}}^2, \\ & \ell_f^g(x; y^{(\boldsymbol{\beta})}) := g(x) + f(y^{(\boldsymbol{\beta})}) + \left\langle \nabla f(y^{(\boldsymbol{\beta})}), x - y^{(\boldsymbol{\beta})} \right\rangle \end{aligned} \tag{3}$$

where $\boldsymbol{T}$ is a symmetric positive definite matrix $\mathbb{S}_{++}(N)$ of dimension $N \times N$. However, we also allow for inexact minimizers of (3). We call $(\tilde{x}, \tilde{\boldsymbol{\beta}})$ an **inexact next iterate** if it satisfies the following condition:

$$\ell_f^g(\tilde{x}; y^{(\tilde{\boldsymbol{\beta}})}) + \frac{1}{2}\|\tilde{x} - y^{(\tilde{\boldsymbol{\beta}})}\|_{\boldsymbol{T}}^2 \le f^g(\bar{x}). \tag{4}$$

The right hand side of (4) is the value of the objective in (3) at $\bar{x}$ with $\beta = 0$.

Example 1. Let $\boldsymbol{T} = \alpha^{-1}\boldsymbol{I}$ in (3) where $\boldsymbol{I}$ denotes the identity matrix. The update step is a proximal gradient step at the extrapolated point $y^{(\boldsymbol{\beta})}$. The formula in (3) is equivalent to

$$\hat{x} = \operatorname*{argmin}_{x\in\mathbb{R}^N} g(x) + \frac{1}{2\alpha}\|x - (y^{(\boldsymbol{\beta})} - \alpha\nabla f(y^{(\boldsymbol{\beta})}))\|^2 ,$$

which is exactly the form of an accelerated proximal gradient step (FISTA-update step) [4]. When $\boldsymbol{\beta}$ is chosen iteration-dependent with a behaviour asymptotically like $(1 - 3/k)_{k\in\mathbb{N}}$, where k is the iteration count, the method can be shown to satisfy a convergence rate of $O(1/k^2)$ when f and g are convex functions.

As the proposed update step in (3) takes the same form, but optimizes the update with respect to $\boldsymbol{\beta}$, each step guarantees a better objective value than an accelerated proximal gradient step. However, this is not enough to guarantee the same rate of convergence, which requires a global picture of the objective.

If the minimum w.r.t. β in (3) is attained at $\beta = 0$, the update step reduces to a standard proximal gradient step

$$\hat{x} = \operatorname*{argmin}_{x\in\mathbb{R}^N} g(x) + \frac{1}{2\alpha}\|x - (\bar{x} - \alpha\nabla f(\bar{x}))\|^2 .$$

3.1 Relation to Proximal Quasi-Newton Methods

In this section, we show that for a quadratic function f, the optimal parameter $\boldsymbol{\beta}$ can be computed analytically. Using the optimal $\boldsymbol{\beta}$ in the update step yields an interpretation of the update step (3) as a proximal gradient step in a modified metric without extrapolation, also known as proximal quasi-Newton method. The modified metric is of type identity minus rank r, where r is the number of linearly independent columns of the matrix $\boldsymbol{D}$. Proximal quasi-Newton methods have recently drawn attention, as for $r = 1$, the associated proximal mapping can be evaluated efficiently [5, 30].

Theorem 2. Consider the problem in (3). Suppose that $f(x) = \frac{1}{2}\langle x, \boldsymbol{H}x\rangle + \langle b, x\rangle + c$ is quadratic with Hessian $\boldsymbol{H}$, $b \in \mathbb{R}^N$, $c \in \mathbb{R}$, the matrix $\boldsymbol{T}$ is chosen such that $\boldsymbol{M} := \boldsymbol{T} - \boldsymbol{H} \in \mathbb{S}_{++}(N)$, and the columns of $\boldsymbol{D}$ are linearly independent. Then, the inner optimization problem w.r.t. $\boldsymbol{\beta}$ is solved by

$$\boldsymbol{\beta}^* = (\boldsymbol{D}^\top \boldsymbol{M}\boldsymbol{D})^{-1}\boldsymbol{D}^\top \boldsymbol{M}(x - \bar{x})\,.$$

The optimization problem in (3) is equivalent to the following:

$$\hat{x} = \operatorname*{argmin}_{x\in\mathbb{R}^N}\, g(x) + \frac{1}{2}\|x - \bar{x} + \boldsymbol{Q}^{-1}\nabla f(\bar{x})\|_{\boldsymbol{Q}}^2\,,$$

where

$$\boldsymbol{Q} := \boldsymbol{T} - \boldsymbol{U}^\top \boldsymbol{U} \qquad \text{with} \qquad \boldsymbol{U} := (\boldsymbol{D}^\top \boldsymbol{M}\boldsymbol{D})^{-\frac{1}{2}}\boldsymbol{D}^\top \boldsymbol{M}$$

and $\boldsymbol{U}^\top\boldsymbol{U}$ is of rank m. The inverse metric is

$$\boldsymbol{Q}^{-1} = \boldsymbol{T}^{-1} + \boldsymbol{T}^{-1}\boldsymbol{U}^\top(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{T}^{-1}\boldsymbol{U}^\top)^{-1}\boldsymbol{U}\boldsymbol{T}^{-1}\,.$$

Proof. Since f is a quadratic function, we have

$$f(y^{(\boldsymbol{\beta})}) = f(\bar{x}) + \left\langle \nabla f(\bar{x}), y^{(\boldsymbol{\beta})} - \bar{x}\right\rangle + \frac{1}{2}\left\langle y^{(\boldsymbol{\beta})} - \bar{x}, \boldsymbol{H}(y^{(\boldsymbol{\beta})} - \bar{x})\right\rangle$$

and

$$\nabla f(y^{(\boldsymbol{\beta})}) = \nabla f(\bar{x}) + \boldsymbol{H}(y^{(\boldsymbol{\beta})} - \bar{x})\,.$$

We plug these equations into the objective in (3), use $\boldsymbol{D\beta} = y^{(\boldsymbol{\beta})} - \bar{x}$, and obtain

$$\begin{aligned}
\ell_f^g(x; y^{(\boldsymbol{\beta})}) + \frac{1}{2}\|x - y^{(\boldsymbol{\beta})}\|_{\boldsymbol{T}}^2 = \; & g(x) + f(\bar{x}) + \left\langle \nabla f(\bar{x}), y^{(\boldsymbol{\beta})} - \bar{x} \right\rangle + \frac{1}{2}\left\langle \boldsymbol{D\beta}, \boldsymbol{HD\beta} \right\rangle \\
& + \left\langle \nabla f(\bar{x}), x - y^{(\boldsymbol{\beta})} \right\rangle + \left\langle x - y^{(\boldsymbol{\beta})}, \boldsymbol{HD\beta} \right\rangle \\
& + \frac{1}{2}\|x - \bar{x}\|_{\boldsymbol{T}}^2 + \frac{1}{2}\|\boldsymbol{D\beta}\|_{\boldsymbol{T}}^2 - \langle x - \bar{x}, \boldsymbol{D\beta}\rangle_{\boldsymbol{T}} \\
= \; & g(x) + f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \frac{1}{2}\|x - \bar{x}\|_{\boldsymbol{T}}^2 \\
& - \frac{1}{2}\langle \boldsymbol{D\beta}, \boldsymbol{HD\beta}\rangle + \langle x - \bar{x}, \boldsymbol{HD\beta}\rangle \\
& + \frac{1}{2}\langle \boldsymbol{D\beta}, \boldsymbol{TD\beta}\rangle - \langle x - \bar{x}, \boldsymbol{TD\beta}\rangle \\
= \; & g(x) + f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \frac{1}{2}\|x - \bar{x}\|_{\boldsymbol{T}}^2 \\
& + \frac{1}{2}\langle \boldsymbol{D\beta}, \boldsymbol{MD\beta}\rangle - \langle x - \bar{x}, \boldsymbol{MD\beta}\rangle \; .
\end{aligned}$$

Since this function is convex and differentiable with respect to $\boldsymbol{\beta}$, the optimal choice for $\boldsymbol{\beta}$ can be found by the first order optimality condition, which reads:

$$\boldsymbol{D}^\top \boldsymbol{MD\beta} = \boldsymbol{D}^\top \boldsymbol{M}(x - \bar{x})\,. \tag{5}$$

As the columns of D are linearly independent, this optimality condition has the unique solution $\boldsymbol{\beta}^*$ as stated. Using this optimal $\boldsymbol{\beta}^*$, the objective in (3) reads as follows:

$$\begin{aligned}
\ell_f^g(x; y^{(\boldsymbol{\beta}^*)}) + \frac{1}{2}\|x - y^{(\boldsymbol{\beta}^*)}\|_{\boldsymbol{T}}^2 &= \ell_f^g(x; \bar{x}) + \frac{1}{2}\|x - \bar{x}\|_{\boldsymbol{T}}^2 - \frac{1}{2}\langle \boldsymbol{D\beta}^*, \boldsymbol{MD\beta}^*\rangle \\
&= \ell_f^g(x; \bar{x}) + \frac{1}{2}\|x - \bar{x}\|_{\boldsymbol{Q}}^2\,,
\end{aligned}$$

and the representation in the statement follows directly. The rank of $\boldsymbol{U}$ is obviously m and the inversion follows from the rank-m generalization of the Sherman–Morrison–Woodbury formula. □

Remark 3. We could consider different points for the linearization and the proximal center. Let $\boldsymbol{\gamma} \in \mathbb{R}^m$ and $z^{(\boldsymbol{\gamma})} = \bar{x} + \boldsymbol{D\gamma}$. Instead of the objective in (3), we consider

$$\ell_f^g(x; y^{(\boldsymbol{\beta})}) + \frac{1}{2}\|x - z^{(\boldsymbol{\gamma})}\|_{\boldsymbol{T}}^2\,.$$

Requiring that $\boldsymbol{H} \in \mathbb{S}_{++}(N)$ and using the optimal $\boldsymbol{\beta}^*$ and $\boldsymbol{\gamma}^*$, it is equivalent to

$$\ell_f^g(x; \bar{x}) + \frac{1}{2}\|x - \bar{x}\|_{\boldsymbol{T} - \boldsymbol{U}^\top\boldsymbol{U} - \boldsymbol{V}^\top\boldsymbol{V}}^2\,,$$

where

$$\boldsymbol{U} := (\boldsymbol{D}^\top \boldsymbol{HD})^{-\frac{1}{2}}\boldsymbol{DH} \quad \text{and} \quad \boldsymbol{V} := (\boldsymbol{D}^\top \boldsymbol{TD})^{-\frac{1}{2}}\boldsymbol{DT}\,.$$

The rank 1 case. In practical applications, the case where $\boldsymbol{\beta} = \beta$ is 1-dimensional is most interesting, since the resulting quasi-Newton method is of type identity minus rank 1, for which the update steps can be evaluated efficiently. In fact, it turns out that in this case (3) is equivalent to a generalization of the SR1 quasi Newton method with a non-smooth term, i.e. a proximal SR1 quasi Newton method. Therefore, we provide the formulas explicitly for this case in the following corollary.

Corollary 4. Consider the problem in (3). Suppose that $f(x) = \frac{1}{2}\langle x, \boldsymbol{H}x\rangle + \langle b, x\rangle + c$ is quadratic with Hessian $\boldsymbol{H}$, $b \in \mathbb{R}^N$, $c \in \mathbb{R}$, the matrix $\boldsymbol{T} = \alpha^{-1}\boldsymbol{I}$ is a multiple of the identity with $\alpha > 0$ such that $\boldsymbol{M} := \boldsymbol{T} - \boldsymbol{H} \in \mathbb{S}_{++}(N)$, and $\boldsymbol{D} = d \in \mathbb{R}^N$. Then, the inner optimization problem w.r.t. $\boldsymbol{\beta}$ is solved by

$$\beta^* = \frac{\langle d, x - \bar{x}\rangle_{\boldsymbol{M}}}{\langle d, d\rangle_{\boldsymbol{M}}}\,.$$

The optimization problem in (3) is equivalent to the following:

$$\hat{x} = \operatorname*{argmin}_{x\in\mathbb{R}^N}\; g(x) + \frac{1}{2}\|x - \bar{x} + \boldsymbol{Q}^{-1}\nabla f(\bar{x})\|^2_{\boldsymbol{Q}}\,, \tag{6}$$

where

$$\boldsymbol{Q} := \boldsymbol{T} - uu^\top \qquad \text{with} \qquad u := \frac{\boldsymbol{M}d}{\|d\|_{\boldsymbol{M}}} \tag{7}$$

and

$$\boldsymbol{Q}^{-1} = \boldsymbol{T}^{-1} + \frac{\boldsymbol{T}^{-1}uu^\top\boldsymbol{T}^{-1}}{1 - u^\top\boldsymbol{T}^{-1}u} = \alpha\cdot\boldsymbol{I} + \frac{\alpha^2 uu^\top}{1 - \alpha u^\top u}\,. \tag{8}$$

Proof. The statement is an obvious consequence of Theorem 2. □

Corollary 5. Consider the situation in Corollary 4. Let $d = x_k - x_{k-1}$ be the difference between the current and the previous iterate, define $y := \boldsymbol{H}d = \nabla f(x_k) - \nabla f(x_{k-1})$. Then

$$\boldsymbol{Q} = \boldsymbol{T} + \frac{(y - \boldsymbol{T}d)(y - \boldsymbol{T}d)^\top}{\langle d, y - \boldsymbol{T}d\rangle} \quad \text{and} \quad \boldsymbol{Q}^{-1} = \boldsymbol{T}^{-1} + \frac{(d - \boldsymbol{T}^{-1}y)(d - \boldsymbol{T}^{-1}y)^\top}{\langle d - \boldsymbol{T}^{-1}y, y\rangle}\,.$$

Proof. The formulas follow directly from Corollary 3 by simple algebraic manipulations. □

Remark 6. The naming "proximal SR1 quasi-Newton method" is justified as follows: In (3), using standard notation of quasi-Newton methods, let $\boldsymbol{T} = \boldsymbol{B}_k$ be the current approximation of the Hessian matrix, then the formula for $\boldsymbol{B}_{k+1} = \boldsymbol{Q}$ in Corollary 5 is exactly the update formula for the SR1 quasi Newton method [40, Eq. 8.24], and $\boldsymbol{H}_{k+1} = \boldsymbol{Q}^{-1}$ the formula for the approximation of the inverse Hessian matrix [40, Eq. 8.25].

Unfortunately, the SR1 update formula does not preserve the positive definiteness, which makes the convergence analysis for line search based algorithms difficult. For more details on this discussion, we refer to [40, Section 8.2].

In case we restart the approximation of the Hessian matrix in each iteration with a multiple of the identity matrix (with a Barzilai–Borwein rule), we obtain the recently proposed zero memory version of the SR1 quasi-Newton method [5]. If the chosen multiplicity is larger than the Lipschitz constant of ∇f, the following section proves the convergence of this method.

3.2 Convergence analysis

In this section, we analyze the convergence of the method with exact updates (3) and inexact updates (4) for solving the optimization problem (1). The main convergence result (Theorem 7), which shows convergence to a stationary point of the objective, does not require the function f or g to be convex. In subsequent sections, we provide two hybrid algorithms involving (3), which yield an accelerated convergence rate of $O(1/k^2)$ in the convex case. Note that, when f is a quadratic function, the convergence results also apply to the zero memory SR1 proximal quasi-Newton method.

For the convergence analysis, we make use of the following notation:

$$\hat{x}^{(\boldsymbol{\beta})} := \operatorname*{argmin}_{x\in\mathbb{R}^N} \ell_f^g(x; y^{(\boldsymbol{\beta})}) + \frac{1}{2}\|x - y^{(\boldsymbol{\beta})}\|_{\boldsymbol{T}}^2 ,$$

which denotes the proximal point given a specific choice of $\boldsymbol{\beta}$. The required concepts from non-smooth analysis are introduced in Section A.

3.2.1 The non-convex case

Theorem 7. Let $(x_k)_{k\in\mathbb{N}}$, $(\boldsymbol{\beta}_k)_{k\in\mathbb{N}}$, and $(\boldsymbol{D}_k)_{k\in\mathbb{N}}$ be sequences satisfying the condition (4), i.e. given $\bar{x} = x_k$ we set $x_{k+1} = \tilde{x}$, $\boldsymbol{\beta}_k = \tilde{\boldsymbol{\beta}}$, and $\boldsymbol{D}_k = \boldsymbol{D}$ from (4), with starting point $x_0 \in \mathbb{R}^N$. If $\boldsymbol{T} - L\cdot\boldsymbol{I} \in \mathbb{S}_{++}(N)$, then

- the objective values are non-increasing and converging, and
- $x_{k+1} - y_k^{(\boldsymbol{\beta}_k)} \to 0$ as $k\to\infty$.

Moreover, if the sequence $(x_k)_{k\in\mathbb{N}}$ is bounded, and $\|\partial f^g(x_{k+1})\|_- \le C\|x_{k+1} - y_k^{(\boldsymbol{\beta}_k)}\|$ for all k, then $x_k \overset{K}{\to} x^*$ for some $K\subset\mathbb{N}$ (i.e. $x_k\to x^*$ for $k\to\infty$ with $k\in K$), where x^* is a critical point of f^g, i.e. $0\in\partial f^g(x^*)$.

Proof. Combining the quadratic upper bound from the Lipschitz continuity of ∇f with (4), we obtain

$$\begin{aligned} f^g(\tilde{x}) &\le \ell_f^g(\tilde{x}; y^{(\tilde{\boldsymbol{\beta}})}) + \frac{1}{2}\|\tilde{x} - y^{(\tilde{\boldsymbol{\beta}})}\|_{L\cdot\boldsymbol{I}}^2 \\ &\le \ell_f^g(\tilde{x}; y^{(\tilde{\boldsymbol{\beta}})}) + \frac{1}{2}\|\tilde{x} - y^{(\tilde{\boldsymbol{\beta}})}\|_{\boldsymbol{T}}^2 - \frac{1}{2}\|\tilde{x} - y^{(\tilde{\boldsymbol{\beta}})}\|_{\boldsymbol{T}-L\cdot\boldsymbol{I}}^2 \\ &\le \ell_f^g(x; y^{(\boldsymbol{\beta})}) + \frac{1}{2}\|x - y^{(\boldsymbol{\beta})}\|_{\boldsymbol{T}}^2 - \frac{1}{2}\|\tilde{x} - y^{(\tilde{\boldsymbol{\beta}})}\|_{\boldsymbol{T}-L\cdot\boldsymbol{I}}^2 . \end{aligned} \tag{9}$$

The non-increasingness of the objective values is a direct consequence of (9) with $x = \bar{x}$ and $\boldsymbol{\beta} = 0$ and the positive definiteness of $\boldsymbol{T} - L \cdot \boldsymbol{I} \in \mathbb{S}_{++}(N)$. Summing both sides of (9) yields

$$\frac{1}{2}\sum_{i=0}^{k} \|x_{i+1} - y_i^{(\boldsymbol{\beta}_i)}\|^2_{\boldsymbol{T}-L\cdot\boldsymbol{I}} \leq f^g(x_0) - f^g(x_{k+1})\,,$$

hence, $x_{k+1} - y_k^{(\boldsymbol{\beta}_k)} \to 0$ as $k \to \infty$. Combining the lower semi-continuity of f^g with a consideration of the limit of (9) for $x = x^*$, we obtain

$$\begin{aligned}\limsup_{k \overset{K}{\to} \infty} f^g(x_{k+1}) \leq \limsup_{k \overset{K}{\to} \infty} \ell_f^g(x^*; y_k^{(\boldsymbol{\beta}_k)}) + \frac{1}{2}\|x^* - y_k^{(\boldsymbol{\beta}_k)}\|^2_{\boldsymbol{T}} - \frac{1}{2}\|x_{k+1} - y_k^{(\boldsymbol{\beta}_k)}\|^2_{\boldsymbol{T}-L\cdot\boldsymbol{I}} \\ = f^g(x^*) \leq \liminf_{x \overset{K}{\to} \infty} f^g(x_{k+1})\,,\end{aligned}$$

where we used to continuous differentiability of f and $y_k^{(\boldsymbol{\beta}_k)} \overset{K}{\to} x^*$. This shows the f^g-attentive convergence of $(x_{k+1})_{k\in\mathbb{N}}$. From $\|\partial f^g(x_{k+1})\|_- \leq C\|x_{k+1} - y_k^{(\boldsymbol{\beta}_k)}\|$ and the closedness property of the limiting subdifferential, we conclude $0 \in \partial f^g(x^*)$, which proves the statement. □

Remark 8. (*i*) The boundedness assumption of $(x_k)_{k\in\mathbb{N}}$ in Theorem 7 is not restrictive, and is, in fact implied here, when f^g is coercive ($f^g(x) \to \infty$ when $\|x\| \to \infty$).

(*ii*) The relative error condition (cf. [1]) $\|\partial f^g(x_{k+1})\|_- \leq C\|x_{k+1} - y_k^{(\boldsymbol{\beta}_k)}\|$ is automatically satisfied when exact update steps (3) instead of (4) are considered. In fact, optimality of x_{k+1} requires the following condition

$$0 \in \partial g(x_{k+1}) + \nabla f(y_k^{(\boldsymbol{\beta}_k)}) + \boldsymbol{T}(x_{k+1} - y_k^{(\boldsymbol{\beta}_k)})\,,$$

which, together with $\|\nabla f(y_k^{(\boldsymbol{\beta}_k)}) - \nabla f(x_{k+1})\| \leq L\|x_{k+1} - y_k^{(\boldsymbol{\beta}_k)}\|$, implies the relative error condition.

3.3 Variants with $O(1/k^2)$-convergence rate

In this section, we introduce two variants of our method (with $\boldsymbol{\beta} = \beta \in \mathbb{R}$) for convex optimization problems (1), which have a convergence rate of $O(1/k^2)$. The two methods are variants in the sense that the update step (3) (or an inexact version) is embedded into another algorithmic strategy. Both cases are related to the standard FISTA method [4, 50].

The final rate of convergence is dictated by a sequence $(\theta_k)_{k\in\mathbb{N}}$ with $\theta_0 = \theta_{-1} \in (0, 1]$ and $\theta_{k+1} \in (0, 1]$ such that

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2} \leq \frac{1}{\theta_k^2} \tag{10}$$

holds. As a particularly simple and enlightening choice of this sequence is $\theta_k = 2/(k + 2)$, for which the left hand side above, which is the inverse of the convergence rate $\theta_k^2/(1 - \theta_k)$, becomes $k(k + 2)/4$.

3.3.1 Monotone FISTA version

We use the idea of [33] (see also [3]) to obtain an acceleration. We generate a sequence $(z_k)_{k\in\mathbb{N}}$ starting at some $z_0 \in \mathbb{R}^N$ that obeys the following update scheme:

$$\tilde{y}_k = z_k + \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}(z_k - z_{k-1}) + \frac{\theta_k}{\theta_{k-1}}(\hat{x}_k - z_k) \tag{11}$$

$$\tilde{x}_{k+1} = \operatorname*{argmin}_x \ell_f^g(x;\tilde{y}_k) + \frac{1}{2}\|x-\tilde{y}_k\|^2 \tag{12}$$

$$y_k^{(\beta)} = z_k + \beta(z_k - z_{k-1}) \tag{13}$$

$$\hat{x}_{k+1} = \operatorname*{argmin}_x \min_\beta \ \ell_f^g(x; y_k^{(\beta)}) + \frac{1}{2}\|x - y_k^{(\beta)}\|^2 \tag{14}$$

$$z_{k+1} = \begin{cases} \hat{x}_{k+1}\,, & \text{if } f^g(\hat{x}_{k+1}) \le f^g(\tilde{x}_{k+1}) \\ \tilde{x}_{k+1}\,, & \text{if } f^g(\hat{x}_{k+1}) > f^g(\tilde{x}_{k+1}) \end{cases} \tag{15}$$

Remark 9. (14) is an update step as in our method (3) with $d_k = z_k - z_{k-1}$.

This scheme obeys an accelerated rate of convergence as the following proposition shows.

Proposition 10. The sequence $(z_k)_{k\in\mathbb{N}}$ obeys the following rate of convergence with respect to the objective values:

$$f^g(z_k) - f^g(x^*) \le \frac{\theta_k^2}{1-\theta_k}\left(\frac{L}{2}\|z_0 - x^*\|^2 + \frac{1-\theta_0}{\theta_0}(f^g(z_0) - f^g(x^*))\right) \in O(1/k^2)\,.$$

Proof. We make the following estimation:

$$\begin{aligned}
f^g(z_{k+1}) &\overset{(i)}{\le} f^g(\tilde{x}_{k+1}) \\
&\overset{(ii)}{\le} \ell_f^g(\tilde{x}_{k+1}, \tilde{y}_k) + \frac{L}{2}\|\tilde{x}_{k+1} - \tilde{y}_k\|^2 \\
&\overset{(iii)}{\le} \ell_f^g(y, \tilde{y}_k) + \frac{L}{2}\|y - \tilde{y}_k\|^2 - \frac{L}{2}\|y - \tilde{x}_{k+1}\|^2 \\
&\overset{(iv)}{\le} \ell_f^g((1-\theta_k)z_k + \theta_k x, \tilde{y}_k) + \frac{L}{2}\|(1-\theta_k)z_k + \theta_k x - \tilde{y}_k\|^2 - \frac{L}{2}\|(1-\theta_k)z_k + \theta_k x - \tilde{x}_{k+1}\|^2 \\
&\overset{(v)}{\le} (1-\theta_k)\ell_f^g(z_k, \tilde{y}_k) + \theta_k \ell_f^g(x, \tilde{y}_k) + \theta_k^2\frac{L}{2}\|(1-\theta_k)/\theta_k z_k + x - \tilde{y}_k/\theta_k\|^2 \\
&\quad - \theta_k^2\frac{L}{2}\|(1-\theta_k)/\theta_k z_k + x - \tilde{x}_{k+1}/\theta_k\|^2 \\
&\overset{(vi)}{\le} (1-\theta_k)f^g(z_k) + \theta_k f^g(x) + \theta_k^2\frac{L}{2}(\|U_k(x)\|^2 - \|U_{k+1}(x)\|^2)
\end{aligned}$$

where (i) uses (15), (ii) uses the quadratic (Lipschitz) upper bound, (iii) holds for all y by using L-strong convexity of $x \mapsto \ell_f^g(x, \tilde{y}_k) + \frac{L}{2}\|x - \tilde{y}_k\|^2$ and optimality of $\tilde{x}_{k+1}$, (iv)

holds for all x by the change of variables $y = (1-\theta_k)z_k + \theta_k x$, (v) holds by convexity of ℓ_f^g in the first argument and a simple algebraic manipulation, and (vi) holds by defining $U_{k+1}(x) := (1-\theta_k)/\theta_k z_k + x - \tilde{x}_{k+1}/\theta_k$ and the definition of[1] $\tilde{y}_k$.

Rearranging this inequality and setting $x = x^*$ yields

$$\frac{1}{\theta_k^2}(f^g(z_{k+1}) - f^g(x^*) - \frac{(1-\theta_k)}{\theta_k^2}(f^g(z_k) - f^g(x^*)) \leq \frac{L}{2}(\|U_k(x^*)\|^2 - \|U_{k+1}(x^*)\|^2)\,,$$

which by standard arguments shows the result. □

3.3.2 Tseng-like version

The following variant does not require a comparison of the objective value, it relies on a comparison of the value of the proximal linearization. This algorithm is closely connected to [50, Algorithm 1]. Here, we need an auxiliary sequence $(z_k)_{k\in\mathbb{N}}$ and consider the following update scheme:

$$\tilde{y}_k = (1-\theta_k)\hat{x}_k + \theta_k\tilde{x}_k \tag{16}$$

$$\tilde{x}_{k+1} = \arg\min_x \ \ell_f^g(x, \tilde{y}_k) + \theta_k\frac{L}{2}\|x - \tilde{x}_k\|^2 \tag{17}$$

$$z_{k+1} = (1-\theta_k)\hat{x}_k + \theta_k\tilde{x}_{k+1} \tag{18}$$

$$\text{find } (\hat{x}_{k+1}, \hat{y}_k) \ \ s.t. \ \ \ell_f^g(\hat{x}_{k+1}, \hat{y}_k) + \frac{L}{2}\|\hat{x}_{k+1} - \hat{y}_k\|^2 \leq \ell_f^g(z_{k+1}, \tilde{y}_k) + \frac{L}{2}\|z_{k+1} - \tilde{y}_k\|^2 \tag{19}$$

Remark 11. Our scheme (3) is hidden in (19). We can choose any point $\hat{y}_k$. In fact, if we use the exact version of our update scheme (3) with $d_k = \tilde{x}_k - \hat{x}_k$, there is no need to check this inequality. In this case, unlike in Section 3.3.1, no additional comparison of objective values is required.

Remark 12. The original version of the Algorithm [50, Algorithm 1] allows for a certain class of Bregman distances in the proximal update steps. We could also include this extension in theory, but left it for future work, since it is not clear which "rank-1 Bregman proximal mappings" can be solved efficiently.

Proposition 13. The sequence $(\hat{x}_k)_{k\in\mathbb{N}}$ obeys the following rate of convergence with respect to the objective values:

$$f^g(\hat{x}_k) - f^g(x^*) \leq \frac{\theta_k^2}{1-\theta_k}\left(\frac{L}{2}\|x_0 - x^*\|^2 + \frac{1-\theta_0}{\theta_0}(f^g(\hat{x}_0) - f^g(x^*))\right) \in O(1/k^2)\,.$$

[1] This equality was actually used to define $\tilde{y}_k$.

Proof. We make the following estimation:

$$
\begin{aligned}
f^g(\hat{x}_{k+1}) &\overset{(i)}{\leq} \ell_f^g(\hat{x}_{k+1}, \hat{y}_k) + \frac{L}{2}\|\hat{x}_{k+1} - \hat{y}_k\|^2 \\
&\overset{(ii)}{\leq} \ell_f^g(z_{k+1}, \tilde{y}_k) + \frac{L}{2}\|z_{k+1} - \tilde{y}_k\|^2 \\
&\overset{(iii)}{\leq} \ell_f^g((1-\theta_k)\hat{x}_k + \theta_k\tilde{x}_{k+1}, \tilde{y}_k) + \frac{L}{2}\|(1-\theta_k)\hat{x}_k + \theta_k\tilde{x}_{k+1} - \tilde{y}_k\|^2 \\
&\overset{(iv)}{\leq} (1-\theta_k)\ell_f^g(\hat{x}_k, \tilde{y}_k) + \theta_k(\ell_f^g(\tilde{x}_{k+1}, \tilde{y}_k) + \theta_k\frac{L}{2}\|\tilde{x}_{k+1} - \tilde{x}_k\|^2) \\
&\overset{(v)}{\leq} (1-\theta_k)\ell_f^g(\hat{x}_k, \tilde{y}_k) + \theta_k(\ell_f^g(x, \tilde{y}_k) + \theta_k\frac{L}{2}\|x - \tilde{x}_k\|^2 - \theta_k\frac{L}{2}\|x - \tilde{x}_{k+1}\|^2) \\
&\overset{(vi)}{\leq} (1-\theta_k)f^g(\hat{x}_k) + \theta_k f^g(x) + \theta_k^2\frac{L}{2}(\|U_k(x)\|^2 - \|U_{k+1}(x)\|^2)
\end{aligned}
$$

where (i) uses the quadratic (Lipschitz) upper bound, (ii) uses (19), (iii) uses (18), (iv) uses convexity of ℓ_f^g and the definition of[2] $\tilde{y}_k$, (v) holds for any x thanks to the optimality of $\tilde{x}_{k+1}$ in (17) and the L-strong convexity of $x \mapsto \ell_f^g(x, \tilde{y}_k) + \theta_k\frac{L}{2}\|x - \tilde{x}_k\|^2$, (vi) holds by defining $U_k(x) := x - \tilde{x}_k$ and by convexity of ℓ_f^g. The statement follows analogously to Proposition 10. □

4 Numerical Experiments

For optimization problems (1) where f is a quadratic function and g is convex, adaptive FISTA (aFISTA) is equivalent to the zero memory SR1 quasi-Newton method with $\alpha < 1/L$. Algorithm 1 provides the details of aFISTA when f is quadratic and g may be non-convex. For the experiments in this paper, we assume that $\boldsymbol{\beta} = \beta$ is one-dimensional. In Section 4.1 and 4.2, we focus on a comparison between aFISTA and the accelerated variants aMFISTA proposed in Section 3.3.1 and aTseng in Section 3.3.2, standard FISTA [4], and its monotone variant (MFISTA) [3].

Remark 14. The matrix $\boldsymbol{Q}$ and $\boldsymbol{Q}^{-1}$ are not explicitly constructed. It is more efficient to work with the identity minus rank 1 decomposition $\boldsymbol{T} - uu^\top$.

In Section 4.3, we consider a “highly” non-convex and non-smooth optimization problem. We compare our proposed method (aFISTA), for which an overview of the implementation is provided in Algorithm 2, with other solvers with theoretical convergence guarantees for such problems: Forward–Backward Splitting (FBS) [1], iPiano [42], and a monotone variant of FISTA (MFISTA) [33].

[2] Again, $\tilde{y}_k$ is actually chosen such that this equality holds.

Algorithm 1 (Exact Adaptive Fista for partially quadratic Problems).

- **Optimization Problem**: $\min_{x \in \mathbb{R}^N} f(x) + g(x)$ where
  - f is quadratic with L-Lipschitz gradient, and
  - g is proper lower semi-continuous with simple proximal mapping.
- **Parameters**:
  - Initialization: Set $x_0 = x_{-1} = \bar{x}$ for some $\bar{x} \in \mathbb{R}^N$.
  - Step size: $\boldsymbol{T} = \alpha^{-1}\boldsymbol{I}$ with $\alpha < 1/L$.
- **Update for** $k = 0, \dots, n$
  - Set $\bar{x} = x_k$ and $d = x_k - x_{k-1}$ in (3) and (2).
  - Compute $\boldsymbol{Q}$ and $\boldsymbol{Q}^{-1}$ from (7) and (8).
  - Compute the solution $\hat{x}$ of the (identity minus rank 1) proximal mapping (6).
  - Set $x_{k+1} = \hat{x}$.

Algorithm 2 (Adaptive Fista for Non-convex Problems).

- **Optimization Problem**: $\min_{x \in \mathbb{R}^N} f(x) + g(x)$ where
  - f is continuously differentiable with L-Lipschitz gradient, and
  - g is proper lower semi-continuous with simple rank-1 proximal mapping.
- **Parameters**:
  - Initialization: Set $x_0 = x_{-1} = \bar{x}$ for some $\bar{x} \in \mathbb{R}^N$.
  - Maximal number of backtracking line-search steps: $m \in \mathbb{N}$.
  - Backtracking line-search scaling factor: $\delta > 0$.
  - Step size: $\boldsymbol{T} = \alpha^{-1}\boldsymbol{I}$ with $\alpha < 1/L$.
- **Update for** $k = 0, \dots, n$
  - Set $\bar{x} = x_k$ and $d = x_k - x_{k-1}$ in (4) and (2).
  - Fix $\bar{\beta}_k$ (e.g. $\bar{\beta}_k = \theta_k(\theta_{k-1}^{-1} - 1)$ with $(\theta_k)_{k \in \mathbb{N}}$ as in (10)).
  - Select $\tilde{\beta}$ in $\{\bar{\beta}_k, \bar{\beta}_k\delta^1, \bar{\beta}_k\delta^2, \dots, \bar{\beta}_k\delta^m, 0\}$ in (4).
  - Set $x_{k+1} = \tilde{x}$ and with $\tilde{x}$ from (4) with $\tilde{\beta}$.

Remark 15. (i) Unlike classical (backtracking) line-search strategies, which are applied to find the step size, the line-search in Algorithm 2 seeks for the overrelaxation parameter, for which $\tilde{\beta} = 0$ is always a feasible choice that satisfies (4). Therefore, the line-search can trivially be terminated after a finite number of steps, which is sometimes hard to verify for classical line-search strategies, especially, in non-smooth optimization.

(ii) If the Lipschitz constant of ∇f is unknown, an additional line-search for the Lipschitz constant is required, which has the same termination issues as classical line-search strategies.

(iii) The most obvious way to generate $\tilde{x}$ for a given $\tilde{\beta}$ in (4), is to perform an exact proximal gradient step from the point $y^{(\tilde{\beta})}$.

(iv) Note the choice of the restriction of the scaling factor for the backtracking line-search is $\delta > 0$. Since the termination of the line-search is always satisfied, the trial values for the extrapolation parameter can also be increased. In fact, there is no need to bound the choice of $\tilde{\beta}$ to $[0, 1)$ (see Section 4.3).

4.1 Nesterov's worst case functions

The smooth problem to be minimized is the following Nesterov's worst case function [38]:

$$\min_{x\in\mathbb{R}^N} f(x)\,, \quad f(x) = \frac{L}{4}\Big(\frac{1}{2}(x_1^2 + \sum_{i=1}^{p-1}(x_{i+1} - x_i)^2 + x_p^2) - x_1\Big)\,, \tag{20}$$

where, exceptionally, the sub-index refers to a coordinate of $x \in \mathbb{R}^N$, L is a parameter, which coincides with the Lipschitz constant of the gradient of the objective, and we set $p = 2k + 1$ with $1 \leq k \leq \frac{N-1}{2}$. In [38], this objective is constructed such that its minimization is hard for any first-order algorithm and it is used to show that after k iterations there is no such algorithm[3] that generates a point $x_k \in \mathbb{R}^N$ with $f(x_k) - \min f \leq 3L\|x_0 - x^*\|^2/(32(k + 1)^2)$ where x_0 is the initialization and x^* is an optimal point of f, i.e. the right hand side is a lower bound for the functional residual.

We set $L = 1$, $k = 100$, and $p = N = 201$. In Figure 1, we compare our accelerated methods, which are optimal (the convergence rate is proportional to the lower bound), with other optimal methods FISTA and MFISTA. Moreover, we also incorporate the pure aFISTA method (here equivalent to the ZeroSR1 method). The convergence plot suggests that also the aFISTA converges in $O(1/k^2)$, which we could not prove. All other accelerated methods show a very similar performance with respect to the number of iterations and the actual computation time. While aTseng and aMFISTA perform equally w.r.t. the number of iteration, aTseng is slightly faster as it requires one comparison of objective values less than

[3] To be more precise, the consideration is restricted to algorithms that generate the next iterate on an affine space spanned by the gradients evaluated at the current and all previous iterates.

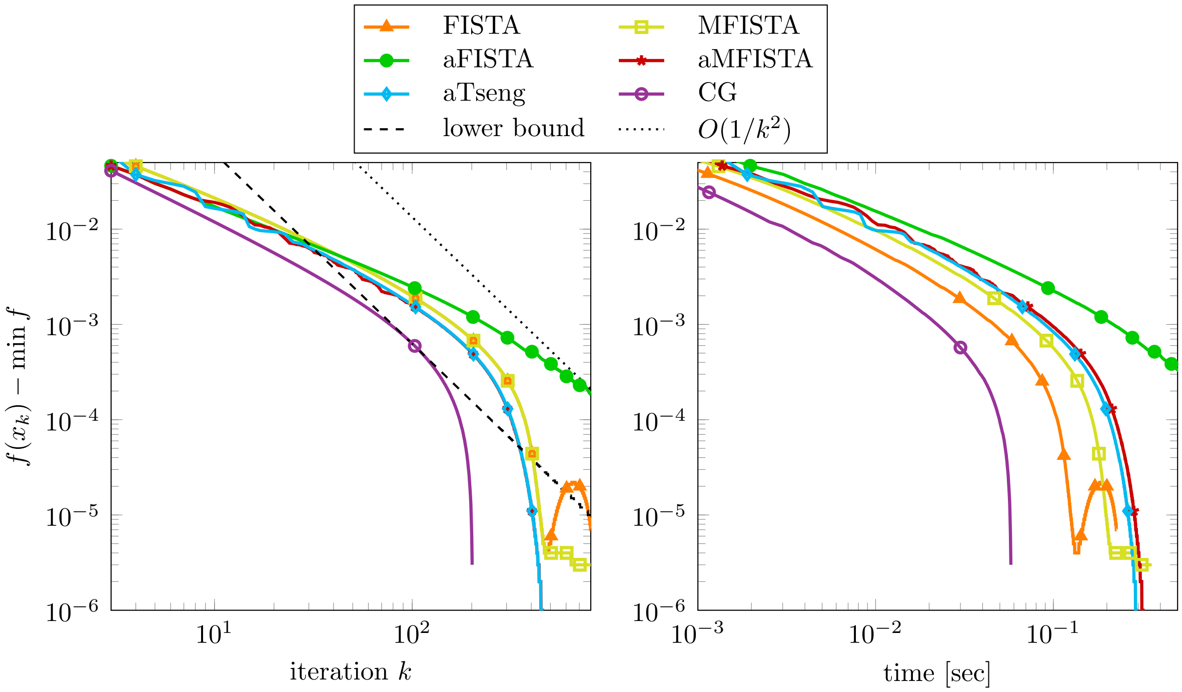

Figure 1: Convergence plots for the methods described in Section 4.1 for solving (20). The vertical axis is the same for both plots. The coordinate system is set such that differences between the methods are visible. The proposed acceleration strategies for the ZeroSR1 method in Section 3.3 perform similarly to the state-of-the-art.

aMFISTA. The proposed acceleration strategies are competitive with the state-of-the-art for Nesterov's worst case problem.

As a reference, also the convergence of the Conjugate Gradient (CG) method is shown in the convergence plots, which is known as the "best" method for minimizing quadratic functions with a finite-time convergence property. It shows that Nesterov's lower bound is sharp with respect to the iteration count $k = 100$.

4.2 Lasso

We consider the example of sparsity regularized linear regression (also known as LASSO or Basis Pursuit problem):

$$\min_{x\in\mathbb{R}^N} h(x)\,, \quad h(x) = \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1\,, \tag{21}$$

where $\|x\|_1 := \sum_{i=1}^N |x_i|$ is the ℓ_1-norm, $A \in \mathbb{R}^{M\times N}$ and $b \in \mathbb{R}^M$, $M, N \in \mathbb{N}$. In this case our algorithm aFISTA (Algorithm 1) is closely related to the zero memory SR1 proximal quasi-Newton method from [5] with step size $1/\|A^\top A\|$. The favorable performance for this special instance was already demonstrated in [5, 30]. Moreover, efficient solution strategies of the ℓ_1-proximal mapping with respect to a metric of type "identity plus/minus rank 1" are

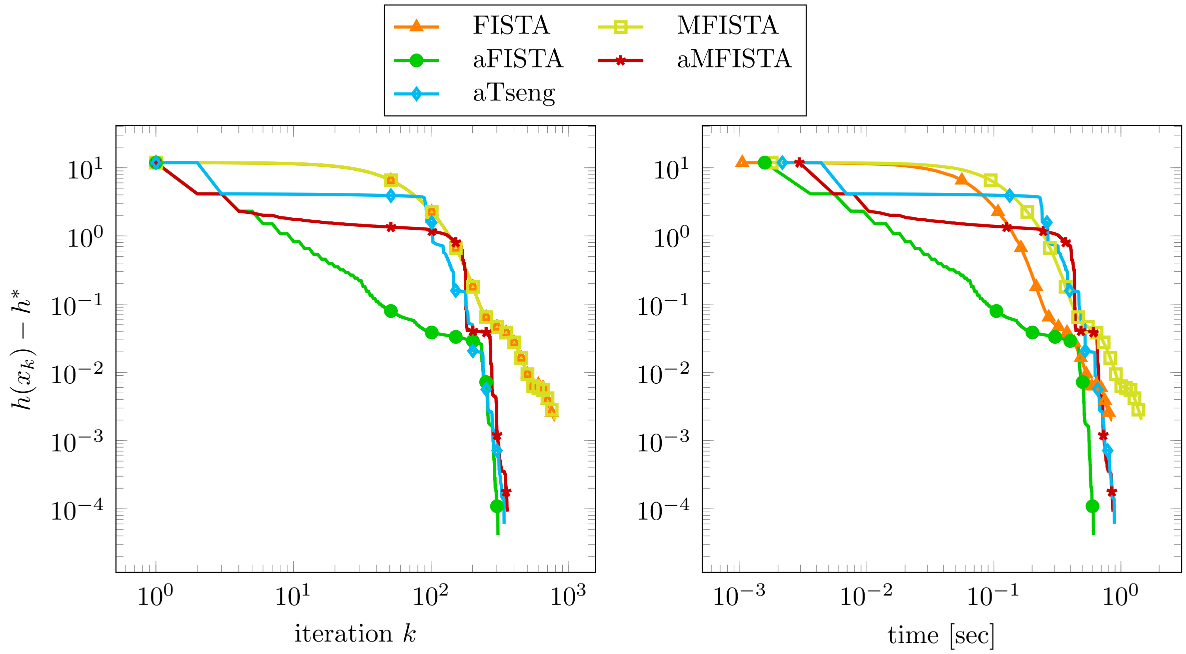

Figure 2: Convergence plots for the methods described in Section 4.2 for solving (21). The vertical axis is the same for both plots, where h^* is the numeric optimal value computed with 10^6 Forward–Backward iterations. The pure aFISTA method initially outperforms all accelerated variants for this problem, and the accelerations from Section 3.3 outperform standard versions of FISTA. Using rank-1 proximal mappings significantly speeds-up the convergence.

also given in [5, 30], which include the ℓ_1-norm. Therefore, we perform a quick comparison with the accelerated variants as in Section 4.1 for this non-smooth problem only. We use $M = 800$, $N = 350$, $\lambda = 0.1$, and the entries of A and b are drawn from a uniform distribution in $[0, 1]$.

The convergence plot in Figure 2 shows that aFISTA and its accelerations outperform standard variants of FISTA. The usage of the proximal mapping with respect to metrics of type "identity plus/minus rank 1", significantly accelerated the convergence. Surprisingly, the convergence of pure aFISTA outperforms its accelerated variants from Section 3.3 with respect to the number of iterations and the actual computation time.

4.3 Sparsity regularized non-linear inverse problem

In this experiment, we consider a neural network formulation of a one dimensional regression problem. We are given $N = 80$ noisy samples $(X, \tilde{Y}) \in (\mathbb{R}^{1\times N})^2$ of the function $F\colon [-3, 3] \to \mathbb{R}$, $x \mapsto x^3 + \cos(5x)$, arranged as corresponding columns of the matrices X and $\tilde{Y}$, i.e.,

$$\tilde{Y}_{1,i} := F(X_{1,i}) + E_{1,i} \,,$$

where $E \in \mathbb{R}^{1\times N}$ is an additive noise matrix that models Gaussian noise with standard deviation $\frac{3}{2}$ and 20 randomly scaled outliers. The neural network optimization problem,

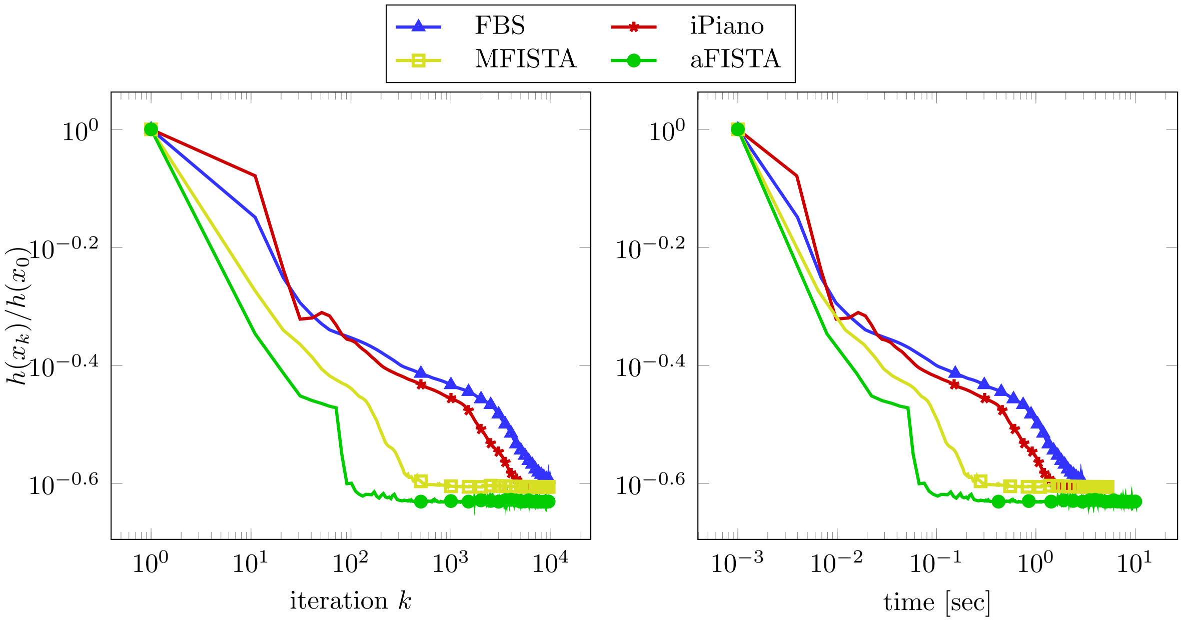

Figure 3: Convergence plots for the methods described in Section 4.3 for solving (22). The vertical axis is the same for both plots. In this experiment, our method aFISTA outperforms the other methods with respect to the actual computation time and the final objective value.

respectively the non-linear inverse problem, that we consider is formulated as follows:

$$\min_{\substack{W_0,W_1,W2\\b_0,b_1,b_2}} \sum_{i=1}^{N} \Big(\|(W_2\sigma_2(W_1\sigma_1(W_0X+B_0)+B_1)+B_2-\tilde{Y})_{1,i}\|^2+\varepsilon^2\Big)^{1/2}+\lambda\sum_{j=0}^{2}\|W_j\|_1\,, \tag{22}$$

$$B_j := b_j\mathbf{1}^\top\,,\quad \mathbf{1}^\top := (1,\dots,1)\,,\quad j=1,2,3\,, \tag{23}$$

where $D_0 = D_3 = 1$, $D_1 = D_2 = 10$, $W_j \in \mathbb{R}^{D_j\times D_{j-1}}$, $B_{j+1} \in \mathbb{R}^{D_j\times N}$, for $j = 1,2,3$, and $\sigma_j\colon \mathbb{R}^{D_j\times N} \to \mathbb{R}^{D_j\times N}$, $A \mapsto (\max(0,(A_{i,l}^2+\varepsilon^2)^{1/2}))_{i,l}$, for $j = 1,2$, with $\varepsilon = 0.1$. We used $\lambda = 1$, which led to a sparsity level of about 87% of the coordinates.

We compare our method aFISTA from Algorithm 2 with $\delta = \frac{1}{2}$, $m = 2$, and $\bar{\beta}_k = 2$, against Forward–Backward Splitting (FBS), iPiano with $\beta = 0.95$, and monotone FISTA (MFISTA). We used the same heuristic step size $\alpha = 5\cdot 10^{-5}$ for all methods. Throughout several experiments, the performance of aFISTA was on a par with MFISTA. The convergence for one problem instance is shown in Figure 3. In this experiment, aFISTA is slightly better and finds a lower objective value. Since aFISTA used a small number of backtracking iterations, the performance with respect to time is also competitive with MFISTA.

5 Conclusion

In this paper, we analyzed a non-convex variant of the well-known FISTA, where the extrapolation parameter is adaptively optimized in each iteration, which we call aFISTA. In the special case where the smooth part of the split objective function is quadratic, aFISTA is equivalent to a certain proximal quasi-Newton method, which unlike the general class of proximal quasi-Newton methods, allows for efficient solutions of the proximal mapping. This equivalence relies on a reformulation of the quadratic function and is not influenced by the non-smooth part of the objective, which may also be non-convex. It provides a different view on quasi-Newton methods, which allows for accelerated variants. We propose two accelerated variants of aFISTA for convex objective functions with convergence rate $O(1/k^2)$ where k is the iteration count.

The general convergence of aFISTA is studied for non-convex objective functions that are the sum of a continuously differentiable function with Lipschitz continuous gradient and a proper lower semi-continuous non-smooth function with simple proximal mapping. Subsequential convergence to a stationary point in terms of the limiting subdifferential is proved. In numerical experiments, aFISTA and its variants have been shown to be competitive with the state-of-the-art.

In future work, we will explore the relationship between the adaptive extrapolation strategy (aFISTA) with r linearly independent directions and other well known quasi-Newton methods such as BFGS. Of course, this requires efficient solution algorithms for rank-r proximal mappings. Moreover, our perspective of the SR1 quasi-Newton method might also lead to new convergence results when the construction of the metric uses memory of previous Hessian approximations. As a third direction of future research, the usage of Bregman distances needs to be investigated. For strongly convex Bregman functions, convergence should not be difficult to proof, however, it is not clear what "rank-1" Bregman proximal mappings can be solved efficiently.

A Concepts from non-smooth analysis

The **Fréchet subdifferential** of f at $\bar{x} \in \operatorname{dom} f := \{x \in \mathbb{R}^N \,|\, f(x) < +\infty\}$ is the set $\widehat{\partial} f(\bar{x})$ of those elements $v \in \mathbb{R}^N$ such that

$$\liminf_{\substack{x \to \bar{x} \\ x \neq \bar{x}}} \frac{f(x) - f(\bar{x}) - \langle v, x - \bar{x} \rangle}{\|x - \bar{x}\|} \geq 0\,.$$

For $\bar{x} \notin \operatorname{dom} f$, we set $\widehat{\partial} f(\bar{x}) = \emptyset$. The **(limiting) subdifferential** of f at $\bar{x} \in \operatorname{dom} f$ is defined by

$$\partial f(\bar{x}) := \{v \in \mathbb{R}^N \,|\, \exists\, (x_k, f(x_k)) \to (\bar{x}, f(\bar{x})),\ v_k \in \widehat{\partial} f(x_k),\ v_k \to v\}\,,$$

and $\partial f(\bar{x}) = \emptyset$ for $\bar{x} \notin \operatorname{dom} f$. A point $\bar{x} \in \operatorname{dom} f$ for which $0 \in \partial f(\bar{x})$ is a called a **critical points**. As a direct consequence, we have to following closedness property:

$$(x_k, f(x_k)) \to (\bar{x}, f(\bar{x})),\ v_k \to \bar{v},\ \text{and for all } k \in \mathbb{N}\colon v_k \in \partial f(x_k) \quad \Longrightarrow \quad \bar{v} \in \partial f(\bar{x})\,.$$

The **distance** of $\bar{x} \in \mathbb{R}^N$ to a set $\omega \subset \mathbb{R}^N$ as is given by $\operatorname{dist}(\bar{x}, \omega) := \inf_{x \in \omega} \|\bar{x} - x\|$. We introduce the **non-smooth slope** $\|\partial f(\bar{x})\|_- := \inf_{v \in \partial f(\bar{x})} \|v\| = \operatorname{dist}(0, \partial f(\bar{x}))$ at $\bar{x}$. Note that $\inf \emptyset := +\infty$ by definition. Furthermore, we have (see [23]):

Lemma 16. If $(x_k, f(x_k)) \to (\bar{x}, f(\bar{x}))$ and $\liminf_{k\to\infty} \|\partial f(x_k)\|_- = 0$, then $0 \in \partial f(\bar{x})$.

References

[1] H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

[2] H. Attouch and J. Peypouquet. The rate of convergence of Nesterov's accelerated forward-backward method is actually faster than $1/k^2$. *SIAM Journal on Optimization*, 26(3):1824–1834, 2016.

[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(11):2419–2434, Nov. 2009.

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Applied Mathematics*, 2(1):183–202, Mar. 2009.

[5] S. Becker and J. Fadili. A quasi-Newton proximal splitting method. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2618–2626. Curran Associates Inc., 2012.

[6] S. Bonettini, I. Loris, F. Porta, and M. Prato. Variable metric inexact line-search based methods for nonsmooth optimization. *SIAM Journal on Optimization*, 26(2):891–921, Jan. 2016.

[7] S. Bonettini, I. Loris, F. Porta, M. Prato, and S. Rebegoldi. On the convergence of variable metric line-search based proximal-gradient method under the Kurdyka–Łojasiewicz inequality. *ArXiv e-prints*, May 2016. arXiv:1605.03791.

[8] J. Bonnans, J. Gilbert, C. Lemaréchal, and C. Sagastizábal. A family of variable metric proximal methods. *Mathematical Programming*, 68(1-3):15–47, Jan. 1995.

[9] C. Broyden. Quasi-Newton methods and their application to function minimisation. *Mathematics of Computation*, 21(99):368–381, 1967.

[10] J. Burke and M. Qian. A variable metric proximal point algorithm for monotone operators. *SIAM Journal on Control and Optimization*, 37(2):353–375, Jan. 1999.

[11] R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited-memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1994.

[12] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[13] G. Chen and R. Rockafellar. Convergence rates in forward–backward splitting. *SIAM Journal on Optimization*, 7(2):421–444, May 1997.

[14] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, Nov. 2013.

[15] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. A block coordinate variable metric forward-backward algorithm. *Journal of Global Optimization*, 66(3):457–485, Feb. 2016.

[16] P. Combettes and B. Vũ. Variable metric quasi-Fejér monotonicity. *Nonlinear Analysis: Theory, Methods & Applications*, 78:17–31, Feb. 2013.

[17] P. Combettes and B. Vũ. Variable metric forward–backward splitting with applications to monotone inclusions in duality. *Optimization*, 63(9):1289–1318, Sept. 2014.

[18] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[19] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward–backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[20] J. Dennis and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics. SIAM Journal on Applied Mathematics, Jan. 1996.

[21] D. Drusvyatskiy and C. Kempton. An accelerated algorithm for minimizing convex compositions. *ArXiv e-prints*, Apr. 2016. arXiv:1605.00125.

[22] R. Fletcher. *Practical Methods of Optimization*. Wiley-Interscience, New York, NY, USA, 1987.

[23] P. Frankel, G. Garrigos, and J. Peypouquet. Splitting methods with variable metric for Kurdyka–Łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications*, 165(3):874–900, Sept. 2014.

[24] M. Friedlander and G. Goh. Efficient evaluation of scaled proximal operators. *ArXiv e-prints*, Mar. 2016. arXiv: 1603.05719.

[25] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, Feb. 2015.

[26] J. Guo and A. Lewis. BFGS convergence to nonsmooth minimizers of convex functions. *ArXiv e-prints*, Mar. 2017. arXiv: 1703.06690.

[27] E. Hairer, C. Lubich, and G. Wanner. Geometric numerical integration illustrated by the Störmer/Verlet method. *Acta Numerica*, 12:399–450, 2003.

[28] M. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, Dec. 1952.

[29] D. R. Hunter and K. Lange. A tutorial on MM algorithms. *Journal of the American Statistical Association*, 58(1):30–37, 2004.

[30] S. Karimi and S. Vavasis. IMRO: A proximal quasi-newton method for solving ℓ_1-regularized least squares problems. *SIAM Journal on Optimization*, 27(2):583–615, Jan. 2017.

[31] A. Lewis and M. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141(1-2):135–163, Oct. 2013.

[32] A. Lewis and S. Zhang. Nonsmoothness and a variable metric method. *Journal of Optimization Theory and Applications*, 165(1):151–171, Apr. 2015.

[33] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 379–387. Curran Associates, Inc., 2015.

[34] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Applied Mathematics*, 16(6):964–979, 1979.

[35] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[36] A. Nemirovsky and D. Yudin. *Problem Complexity and Method Efficiency in Optimization.* A Wiley-Interscience publication. Wiley, 1983.

[37] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

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

[39] Q. Nguyen. Forward–backward splitting with Bregman distances. *Vietnam Journal of Mathematics*, 45:519–539, Jan. 2017.

[40] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, New York, 2nd edition, 2006.

[41] P. Ochs. Unifying abstract inexact convergence theorems for descent methods and block coordinate variable metric iPiano. *ArXiv e-prints*, Feb. 2016. arXiv:1602.07283.

[42] P. Ochs, Y. Chen, T. Brox, and T. Pock. ipiano: Inertial proximal algorithm for non-convex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[43] P. Ochs, J. Fadili, and T. Brox. Non-smooth non-convex Bregman minimization: Unification and new algorithms. *ArXiv e-prints*, July 2017. arXiv: 1707.02278.

[44] L. Parente, P. Lotito, and M. Solodov. A class of inexact variable metric proximal point algorithms. *SIAM Journal on Optimization*, 19(1):240–260, Jan. 2008.

[45] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[46] M. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In *Nonlinear Programming (Proc. Sympos., New York, 1975)*, volume IX of *SIAM–SAMS Proc.* Amer. Math. Soc., 1976.

[47] S. Salzo. The variable metric forward-backward splitting algorithm under mild differentiability assumptions. *ArXiv e-prints*, May 2016. arXiv: 1605.00952.

[48] L. Stella, A. Themelis, and P. Patrinos. Forward–backward quasi-Newton methods for nonsmooth optimization problems. *Computational Optimization and Applications*, 67(3):443–487, July 2017.

[49] W. Su, S. Boyd, and E. Candes. A differential equation for modeling Nesterovs accelerated gradient method: Theory and insights. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 2510–2518. Curran Associates, Inc., 2014.

[50] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

[51] P. Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Mathematical Programming*, 125(2):263–295, Oct. 2010.

[52] J. Yu, S. Vishwanathan, S. Günter, and N. Nicol. A quasi-newton approach to nonsmooth convex optimization problems in machine learning. *Journal of Machine Learning Research*, 11:1145–1200, Mar. 2010.