

FISTA Review

Hongda Li

December 1, 2022

Abstract

We prove the proximal gradient and accelerated proximal gradient algorithm convergence rate under convexity assumptions. We proved the convergence for the proximal gradient without the Nesterov Acceleration differently compared to the original work by Beck[2]; we extract a lemma and then use it to prove the convergence rate under the accelerated case. Additionally, we provide thorough context for the proximal gradient method by incorporating the majorization via envelope idea. We did numerical experiments differently from Beck by keeping track of the norm of the fixed point error on the proximal gradient step during our numerical experiment to expose a 2 phase descent property. Finally, we implemented the accelerated proximal gradient method for the LASSO problem on a big image in Julia.

1 Introduction

We are concerned with the problem type

$$\min_x g(x) + h(x), \tag{1.0.1}$$

where the objective is the sum of two functions. Algorithms developed for solving optimization problems of this format. We will list some algorithms with their convergence rate under different assumptions.

1. The projected subgradient algorithm solves $h = \delta_Q$ where Q is a closed convex set, and g is closed and convex with $Q \subseteq \text{ri} \circ \text{dom}(g)$. The algorithm generates a sequence of $x^{(k)}$ where the weighted average of the sequence by step size has a convergence rate of $\mathcal{O}(1/\sqrt{k})$ in terms of the optimality; this result is from the convex analysis class I took before. For a more thorough exposition of the matter regarding the convergence of the optimality for the subgradient method under the choice of Polyak Step Size, refer to Theorem 8.13 of Beck's work[1].
2. We use the proximal Gradient algorithm when g is strongly smooth, convex, closed, and proper function and h has an easy-to-compute proximal oracle. Assuming h is closed convex, proper, and coersive, the optimal and the minimizer exists, and the convergence rate is $\mathcal{O}(1/k)$. We will prove this result in our report.

3. The Accelerated Proximal Algorithm is a modified version of the proximal gradient that uses Nesterov Momentum and converges with $\mathcal{O}(1/k^2)$ with an additional convexity assumption on g . The convergence can be even faster when more additional assumptions on g . We will prove the convergence results using the convexity assumption in this report.

In this report, we introduce the setup for the proximal gradient algorithms. We give proofs for the convergence results of the Proximal Gradient algorithms with fixed step sizes with and without the Nesterov Momentum. Finally, we also implemented some nontrivial examples of the algorithm in Julia[3]. The code that produces the plots for the numerical experiments is on my GitHub [here](#).

For this report, some of the materials are in Amir Beck’s textbook [1], and the proof for the convergence of the Accelerated Proximal Gradient method closely follows the original paper for FISTA[2] by Amir and Marc Teboulle. Other additional and specific materials might get used during the expression as well.

In the [section 2](#), we introduce the minimum mathematical background needed to understand the proximal gradient method. We make use of the forward-backward envelope to introduce the proximal gradient method in [section 3](#). In addition, we introduce several fundamental lemmas related to the monotone property of the Proximal Gradient method and the choice of step size and extract out the important [lemma 4.1.1](#) for the proof of the accelerated case, which is in the appendix. In [section 4](#), we prove the convergence of the proximal gradient method under convexity and smoothness assumption with a fixed step size. Moreover, in [section 5](#), we state the proximal gradient algorithm with Nesterov acceleration (also referred to as FISTA). In the [appendix A](#), we prove the better convergence of FISTA with meticulous detail without explicitly assuming that momentum term. Finally, for applications in [section 6](#), we consider the convergence of the optimality and the norm of the gradient mapping for the LASSO problem, and then we apply the LASSO algorithm for the task of deblurring images with high gaussian noise. In that section, we introduce other typical applications and extensions of the algorithm.

1.1 Contributions

Nesterov first created gradient descent with momentum in 1983; Beck’s contribution involves proving that the Nesterov accelerations work for the proximal gradient method. Beck popularized the use of momentum in the broader context to improve the convergence of algorithms such as the proximal gradient method. In our report, we made the following contributions:

1. We prepare the theoretical context and background for the proximal gradient algorithm. We use the majorizations and minimizations interpretations to derive the proximal gradient algorithm.
2. We organize the proof for the proximal gradient method without momentum and present it; we extract a lemma that is fundamental to the proof of the proximal gradient method with the Nesterov accelerations term.
3. We prove the accelerated proximal gradient algorithm without prior assumptions on the sequence which produces the momentum term. Instead, we assume a template

algorithm with a momentum term and derive some desired properties for the sequence t_k instead of assuming it in advance.

4. We implemented the proximal gradient algorithm in Julia and applied the algorithm to the LASSO problem for denoising images. We use the algorithm for a larger image and collect the norm of the fixed point error of the proximal gradient operator while tuning the algorithm. We present the results and show that the convergence of the fixed point error follows a 2 phase descend pattern.

2 Preliminaries

The proximal operator is a crucial component of the algorithm, and its non-expansive property is relevant to the convergence of the Proximal Gradient under the non-convex case. We will not go into detail about the non-convex case. Under the assumption of convexity for f , the property of the strongly smooth function is more relevant.

2.1 The proximal operator

Definition 1 (Proximal Operator and Moreau Envelope). A Moreau Envelope $\text{env}_{\alpha,f}(x)$, $\text{prox}_{\alpha,f}$ the proximal operator are defined for some function f :

$$\begin{aligned}\text{env}_{f,\alpha}(x) &:= \min_y \left\{ f(y) + \frac{1}{2\alpha} \|y - x\|^2 \right\}, \\ \text{prox}_{f,\alpha}(x) &:= \arg \min_y \left\{ f(y) + \frac{1}{2\alpha} \|y - x\|^2 \right\}.\end{aligned}$$

The proximal operator is a singleton when the function f is convex, proper, and closed due to the strong convexity of $f(y) + 1/(2\alpha)\|y - x\|^2$. Observe that $\text{env}_{\alpha,f}(x) = (f \square \frac{1}{2\alpha} \|\cdot\|^2)(x)$, hence the infimal convolution gives us the interpretation that the epigraphs of the envelope are adding between the epigraph of these two functions. This conceptualization will help with the intuitive understanding of many proximal algorithms. In addition, please observe the following identities:

$$\begin{aligned}\text{prox}_{f/\alpha,1} &= \text{prox}_{f,\alpha} \\ \alpha^{-1} \text{env}_{\alpha,f,1}(x) &= \text{env}_{f,\alpha}(x).\end{aligned}$$

Lemma 2.1.1 (Proximal Operator as resolvent of Scaled Subgradient). When the function f is convex closed and proper, we view $\text{prox}_{\alpha,f}$ as the resolvent $(I + \alpha\partial f)^{-1}$, which is also, a single-valued operator that sometimes has a nice closed form solution to it.

Proof.

$$\begin{aligned}
\mathbf{0} &\in \partial \left[f(y) + \frac{1}{2\alpha} \|y - x\|^2 \right] (y^+) \\
\mathbf{0} &\in \partial f(y^+) + \frac{1}{\alpha} (y^+ - x) \\
\frac{x}{\alpha} &\in (\partial f + \alpha^{-1} I)(y^+) \\
x &\in (\alpha \partial f + I)(y^+) \\
y^+ &\in (\alpha \partial f + I)^{-1}(x).
\end{aligned}$$

□

2.2 The α smoothness

Definition 2 (α -Smoothness). A differentiable function g is called α -smooth with a constant α then it satisfies:

$$|g(y) - g(x) - \langle \nabla g(x), y - x \rangle| \leq \frac{\alpha}{2} \|x - y\|^2 \quad \forall x, y \in \mathbb{E}. \quad (2.2.1)$$

We remove the absolute value sign when g is convex.

Theorem 1 (Lipschitz Gradient and α Smoothness Equivalence Under Convexity). Suppose g is differentiable on the entire of \mathbb{E} and, it is closed convex proper then, it is α -smooth if and only if the gradient ∇g is globally Lipschitz continuous with a parameter of α and g .

Proof. For conciseness, we skip the proof here. To prove it, we consider applying generalized Cauchy Inequality to (iv) in Theorem 5.8 for Beck's textbook [1]. To prove that α -smoothness implies global Lipschitz gradient, for the converse, consider taking some line integrals on the gradient and then using the global Lipschitz property of the gradient from x to y . □

3 Proximal gradient and forward backward envelope

We introduce the algorithm through the forward-backward envelope; this helps with the intuitive understanding of this algorithm. We then state some of the essential properties. The forward-backward envelope name came from a numerical method that simulates a gradient system that is the summation of stiff and nonstiff dynamics by using forward Euler on the nonstiff part and backward Euler on the stiff part. We will not discuss this specific interpretation of the proximal gradient method. For more detail regarding this interpretation of the proximal gradient method, refers to Broyd's paper [5].

Assumption 1 (Convex Smooth Nonsmooth with Bounded Minimizers). We will assume that $g : \mathbb{E} \mapsto \mathbb{R}$ is strongly smooth with constant L_g and $h : \mathbb{E} \mapsto \bar{\mathbb{R}}$ is closed convex and proper. We define $f := g + h$ to be the summed function and $\text{ri} \circ \text{dom}(g) \cap \text{ri} \circ \text{dom}(h) \neq \emptyset$. We also assume that a set of minimizers exists for the function f and that the set is bounded. Denote the minimizer using \bar{x} .

3.1 Proximal gradient

First, we follow the intuitive idea of constructing an upper bounding function given a parameter β as $m_x(y|\beta)$; it can be interpreted as a surrogate function if one prefers for $g + h$ with $\beta \geq L_g$:

$$g(x) + h(x) \leq g(x) + \nabla g(x)^T(y - x) + \frac{\beta}{2}\|y - x\|^2 + h(y) =: m_x(y|\beta) \quad \forall y \in \mathbb{E},$$

this function $m_x(y|\beta)$ is a strongly convex function and it's equal to $g + h$ at x , and larger than it on every other points. The *envelope function*, defined as $m^+(y|\beta) := \min_y \{m_x(y|\beta)\}$ minimizes the upper bounding function, and the function m^+ is lower than $g + h$ on all points and its minimizer takes the following form:

$$\arg \min_y \{m_x(y|\beta)\} = \arg \min_y \left\{ g(x) + \nabla g(x)^T(y - x) + \frac{\beta}{2}\|y - x\|^2 + h(y) \right\}.$$

Theorem 2 (Minimizer of $m_x(y|\beta)$). The minimizer for the envelope has a closed form, and it is $\text{prox}_{h,\beta^{-1}}(x + \beta^{-1}\nabla g(x))$, with [assumption 1](#).

Proof. We consider minimizing the envelope; zero is in the subgradient of the upper bounding function $m_x(y|\beta)$.

$$\begin{aligned} \mathbf{0} &\in \nabla g(x) + \beta(y - x) + \partial h(y) \\ \nabla g(x) + \beta y &\in \beta y + \partial h(y) \\ -\beta^{-1}\nabla g(x) + x &\in y + \beta^{-1}\partial h(y) \\ -\beta^{-1}\nabla g(x) + x &\in [I + \beta^{-1}\partial h](y) \\ \implies [I + \beta^{-1}\partial h]^{-1}(-\beta^{-1}\nabla g(x) + x) &\ni y, \end{aligned}$$

using [lemma 2.1.1](#), the RHS is the operator $\text{prox}_{h,\beta^{-1}}(x + \beta^{-1}\nabla g(x))$. □

Remark 3.1.1. The minimizer of the envelope at x : $\text{prox}_{h,\beta^{-1}}(x + \beta^{-1}\nabla g(x))$ is what we call *prox step* for short, it makes the envelope $m_x(y|\beta)$ strictly lower than $f(x)$ for any point x that is not the minimizer of $h + g$.

3.2 Fixed point of the prox step

Denote the prox step $\mathcal{P}_{\beta^{-1}}^{g,h}(x) = \text{prox}_{h,\beta^{-1}}(x - \beta^{-1}\nabla g(x))$, in most context without ambiguity it will be denoted as $\mathcal{P}x$. The fixed point of \mathcal{P} is a point x such that $x = \mathcal{P}x$ if and only if x is the minimizer of f when [assumption 1](#) is true. We denoted the fixed point as \bar{x} . To see how this is true consider any x^+ such that $x^+ = \mathcal{P}x$, using subgradient of the envelope:

$$\begin{aligned} \mathbf{0} &\in \nabla g(x) + \beta(x^+ - x) + \partial h(x^+) \\ \beta(x - x^+) &\in \partial h(x^+) + \nabla g(x^+) \\ x = x^+ &\iff \mathbf{0} \in \partial h(x^+) + \nabla g(x^+), \end{aligned}$$

therefore, if x^+ is a fixed point of \mathcal{P} if and only if it is one of the local minimizers of the function f . Conversely, if x^+ is not a fixed point of x , then it has to make the objective value

of the upper bounding function $m_x(y|\beta)$ decrease because it is a strongly convex function. However, this does not necessarily mean that the prox step can decrease the value of the function f ; we need more conditions for the parameter β so that the value of f at the prox step point is lower than the envelope. We will explain more about this in the next subsection.

Remark 3.2.1. The operator $\beta(x - \mathcal{P}x)$ is called the gradient mapping in Amiar's Book [1], and it has many more critical properties that are useful for the convergence proof of the proximal gradient method under many different contexts. Please observe that if the function $h \equiv 0$, the gradient mapping is simply the gradient of the function g .

3.3 Step-sizes that ensures monotone descent property

With [assumption 1](#), only a specific size of step-size can guarantee a decrease in the function value for the minimizers that minimize the envelope.

Theorem 3 (Stepsize that Ensures Monotone Decrease). The step size L^{-1} of the proximal gradient that guarantees a decrease in the objective value has to satisfy: $L \geq L_g$, where L_g is the Lipschitz constant for the gradient of the function g (recall [theorem 1](#)) and $\mathcal{P}x$ is $\mathcal{P}_{L^{-1}}^{g,h}(x)$.

Proof. Consider the fact that the envelope at the prox step is smaller than the point where the envelope is touching with the function f at x (recall [theorem 2](#)). By the minimizing property of \S on the upperbounding function we have $m_x(\mathcal{P}x|L_f) \leq m_x(x) = f(x)$ which gives:

$$\begin{aligned} m_x(\mathcal{P}x|L) &\leq m_x(\mathcal{P}x|L_f) \leq f(x) \\ \implies h(\mathcal{P}x) + \langle \nabla g(x), \mathcal{P}x - x \rangle + \frac{L}{2} \|\mathcal{P}x - x\|^2 &\leq h(x) \\ h(\mathcal{P}x) - h(x) + \langle \nabla g(x) - \mathcal{P}x - x \rangle &\leq \frac{-L}{2} \|\mathcal{P}x - x\|^2, \end{aligned} \quad (\Delta)$$

next, we also consider the strongly smooth property of g to obtain:

$$\begin{aligned} g(\mathcal{P}x) - g(x) - \langle \nabla g(x), \mathcal{P}x - x \rangle &\leq \frac{L_g}{2} \|\mathcal{P}x - x\|^2 \quad (\nabla) \\ \implies h(\mathcal{P}x) + g(\mathcal{P}x) - g(x) - h(x) &\leq \left(\frac{L_g}{2} - \frac{L}{2} \right) \|\mathcal{P}x - x\|^2 \quad (**) \\ f(\mathcal{P}x) - f(x) &\leq \left(\frac{L_g}{2} - \frac{L}{2} \right) \|\mathcal{P}x - x\|^2, \end{aligned}$$

where $(**)$ is $(\nabla) + (\Delta)$. Observe that the objective decrease is asserted on the last line if $L_g \leq L$. Additionally, using [theorem 2](#), we have L^{-1} being the step sizes inside of the proximal gradient operator. See Beck's paper [2] for more details about line search conditions employed for the proximal gradient algorithm. \square

Remark 3.3.1. The monotone decrease property of step size is useful for engineering the backtracking routine for the proximal gradient method. More specifically, as long as the step size L^{-1} satisfies $m_x(\mathcal{P}x|L) \leq f(x)$, then it is an acceptable step size.

3.4 Proximal gradient algorithm

Algorithm 1 Proximal Gradient With Fixed Step-sizes

```

1: Input:  $g, h$ , smooth and nonsmooth,  $L$  stepsize,  $x^{(0)}$  an initial guess of solution.
2: for  $k = 1, 2, \dots, N$  do
3:    $x^{(k+1)} = \mathcal{P}_{L^{-1}}^{g,h} x^{(k)}$ 
4:   if  $x^{(k+1)}, x^{(k)}$  close enough then
5:     Break
6:   end if
7: end for

```

Remark 3.4.1. The [Proximal Gradient With Fixed Step Size](#) algorithm terminates either it reaches the iteration limit N or the fixed point iterations on the operator \mathcal{P} have converged. In some cases, we can obtain the Lipschitz constant for g in advance; in other cases, it is not easy to obtain.

4 Convergence of proximal gradient

Here, we give analysis for the convergence behaviors of the algorithm in [1](#) with fixed stepsizes and assumption [1](#) is true.

4.1 Convergence under the convex case

Before the proof, we state some of the quantities that are involved in the proof.

1. Recall from [section 3.2](#) where $G_\beta(x) - \nabla g(x) \in \partial h(x^+)$ with $x^+ \in \mathcal{P}_{\beta^{-1}}^{g,h}(x)$, and this general condition is true for all values of x . We refer to $G_\beta(x)$ as the residual of the proximal gradient algorithm. Finally, $G_\beta(x) = \beta(x - x^+)$
2. By choosing the stepsize $\beta^{-1} \leq L^{-1}$, we assert a strict decrease of the value of the objective function, $f(x^+) \leq f(x)$.
3. We denote \bar{f} to be $f(\bar{x})$ where \bar{x} is one of the minimizers of f .

Theorem 4 (Convergence Under Convexity). With [assumption 1](#), execute the algorithm for N steps, we have:

$$f(x^{(N+1)}) - \bar{f} \leq \frac{\beta(\|x^{(0)} - \bar{x}\|^2 - \|x^{(N+1)} - \bar{x}\|^2)}{2(N+1)}.$$

Proof. This proof is standard and does not entirely resemble the proof showed in [\[2, Aimir, Teboulle\]](#); nonetheless, we will extract a lemma out of this proof and use that as the foundation for the proof in the Nesterov Accelerated case of the proximal gradient algorithm.

Firstly by choice of step size and the strong smoothness of the function g , we have the inequality:

$$g(x^+) \leq g(x) - \beta^{-1} \langle \nabla g(x), G_\beta(x) \rangle + \underbrace{\frac{L}{2\beta^2} \|G_\beta(x)\|^2}_{\leq \frac{1}{2\beta} \|G_\beta(x)\|^2}, \quad (*)$$

next, by the convexity of f, g , we have inequalities:

$$\begin{aligned} g(x) &\leq g(z) - \langle \nabla g(x), x - z \rangle \\ h(x^+) &\leq h(z) + \langle \partial h(x^+), x^+ - z \rangle, \end{aligned}$$

where we abuse the notation $\partial h(x^+)$ to denote some vector in the subgradient of h at point x^+ . Next, we substitute the above results into (*):

$$\begin{aligned} g(x^+) + h(x^+) &\leq g(x) + \beta^{-1} \langle \nabla g(x), G_\beta(x) \rangle + \frac{1}{2\beta} \|G_\beta(x)\|^2 + h(x^+) \\ &\leq g(z) + \underbrace{\langle \nabla g(x), x - z \rangle}_{[1]} - \underbrace{\beta^{-1} \langle \nabla g(x), G_\beta(x) \rangle}_{[2]} \\ &\quad + \frac{1}{2\beta} \|G_\beta(x)\|^2 + h(z) + \underbrace{\langle \partial h(x^+), x^+ - z \rangle}_{[4]}, \end{aligned} \quad (\nabla)$$

and we consider the summation for each of these numerically labeled terms to obtain

$$\begin{aligned} [3] &:= [1] + [2] \\ [3] &= \langle \nabla g(x), x - z - x + x^+ \rangle = \langle \nabla g(x), x^+ - z \rangle \\ [3] + [4] &= \langle \nabla g(x), x^+ - z \rangle + \langle G_\beta(x) - \nabla g(x), x^+ - z \rangle \quad (**) \\ &= \langle G_\beta(x), x^+ - z \rangle \\ &= \langle G_\beta(x), x - z - (x - x^+) \rangle \\ &= \langle G_\beta, x - z \rangle - \langle G_\beta, \underbrace{x - x^+}_{=\beta^{-1}G_\beta(x)} \rangle \\ &= \langle G_\beta(x), x - z \rangle - \beta^{-1} \|G_\beta(x)\|^2, \end{aligned}$$

where at (*) we applied the substitution $G_\beta(x) - \nabla f(x) \in \partial h(x^+)$. Continued from (∇) we obtain

$$\begin{aligned} \underbrace{g(x^+) + h(x^+)}_{f(x^+)} &\leq \underbrace{g(z) + h(z)}_{f(z)} - \frac{1}{2\beta} \|G_\beta(x)\|^2 + \langle G_\beta, x - z \rangle \\ f(x^+) - f(z) &\leq \langle G_\beta(x), x - z \rangle - \frac{1}{2\beta} \|G_\beta(x)\|^2. \end{aligned} \quad (\star)$$

Next, we make the simplifications using algebra and get

$$\begin{aligned}
f(x^+) - f(\bar{x}) &\leq \frac{-1}{2\beta} \|G_\beta(x)\|^2 + \langle G_\beta, x - \bar{x} \rangle \\
&= -\frac{\beta}{2} (\|x - x^+\|^2 - 2\langle x - x^+, x - \bar{x} \rangle) \\
[5] \implies &= \frac{-\beta}{2} (\|x^+ - \bar{x}\|^2 - \|x - \bar{x}\|^2) \\
&= \frac{\beta}{2} (\|x - \bar{x}\|^2 - \|x^+ - \bar{x}\|^2),
\end{aligned}$$

and since the step-size assert a non-decreasing sequence of number, we perform the telescoping sum by considering the substitution $x^+ = x^{(k+1)}$, $x = x^{(k)}$ we get:

$$\begin{aligned}
f(x^{(k+1)}) - \bar{f} &\leq \frac{\beta}{2} (\|x^{(k)} - \bar{x}\|^2 - \|x^{(k+1)} - \bar{x}\|^2) \\
\implies \left(\sum_{i=0}^N f(x^{(i+1)}) - \bar{f} \right) &\leq \frac{\beta}{2} (\|x^{(0)} - \bar{x}\|^2 - \|x^{(N+1)} - \bar{x}\|^2) \\
f(x^{(N+1)}) - \bar{f} &= \min_{i=0, \dots, N} \{f(x^{(i+1)}) - \bar{f}\} \leq \left(\frac{1}{N+1} \sum_{i=0}^N f(x^{(i+1)}) \right) - \bar{f} \\
\implies f(x^{(N+1)}) - \bar{f} &\leq \frac{\beta(\|x^{(0)} - \bar{x}\|^2 - \|x^{(N+1)} - \bar{x}\|^2)}{2(N+1)} \\
&\leq \frac{\beta\|x^{(0)} - \bar{x}\|^2}{2(N+1)}.
\end{aligned}$$

□

Remark 4.1.1. We can extract one important lemma from this proof that will later be important for the proof of the accelerated case. It came from the tagged expression (\star) during the above derivation. We will refer to this as the “Prox Step 2 Points” lemma. Expression (\star) is equivalent to the lemma 2.3 in the FISTA paper[2].

Lemma 4.1.1 (Prox Step 2 Points). With [assumption 1](#), and $\beta^{-1} > L_g$ still being our stepsize for [algorithm 1](#), let $y \in \mathbb{E}$ and define $y^+ = \mathcal{P}_{\beta^{-1}}^{g,h}(y)$ we have for any $x \in \mathbb{E}$:

$$f(x) - f(y^+) \geq \frac{\beta}{2} \|y^+ - y\|^2 + \beta \langle y - x, y^+ - y \rangle.$$

Proof. The proof is continued from expression (\star) :

$$\begin{aligned}
f(x^+) - f(z) &\leq \langle G_\beta(x), x - z \rangle - \frac{1}{2\beta} \|G_\beta(x)\|^2. \\
f(x^+) - f(z) &\leq \beta \langle x - x^+, x - z \rangle - \frac{1}{2\beta} \|\beta(x - x^+)\|^2 \\
f(z) - f(x^+) &\geq \frac{\beta}{2} \|x - x^+\|^2 + \beta \langle x^+ - x, x - z \rangle,
\end{aligned}$$

by substituting $x := y$ and $z := x$ in the last line, we completed the proof of the lemma. □

5 Accelerated proximal gradient

Here we state the Accelerated Proximal algorithm in Beck's Paper [paper\[2\]](#) and prove the convergence in the appendix. The convergence proof follows what is in the paper but with more details in the appendix. The FISTA algorithm stands for Fast Iterative Shrinkage-Thresholding Algorithm, which is the specific case of the Proximal Gradient with Nesterov momentum applied to the LASSO problem. FISTA is an accelerated case of the ISTA algorithm, and it is the same as FISTA but without the Nesterov momentum.

5.1 Accelerated proximal gradient algorithm

Algorithm 2 FISTA With Constant Step Size

```

1: Input: the step size  $\beta^{-1}$ , and  $x^{(0)}$  the initial guess.
2:  $y^{(1)} = x^{(0)}$ 
3: for  $k = 1, \dots, N$  do
4:    $x^{(k)} := \mathcal{P}y^{(k)}$ 
5:   if  $y^{(k)} - x^{(k)}$  small enough then
6:     Break
7:   end if
8:    $t_{k+1} := \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ 
9:    $y^{(k+1)} := x^{(k)} + \frac{t_k - 1}{t_{k+1}}(x^{(k)} - x^{(k-1)})$ 
10: end for

```

5.2 Convergence under the convex case

Theorem 5 (FISTA Convergence under Convexity). If [assumption 1](#) is satisfied, then the FISTA algorithm has convergence result of:

$$f(x^{(k)}) - f(\bar{x}) \leq \frac{2\beta^{-1}\|x^{(0)} - \bar{x}\|^2}{(k+1)^2},$$

where \bar{x} is one of the optimizers, the convergence rate is $\mathcal{O}(1/k^2)$.

Proof. For a proof see [appendix A](#) □

6 Numerical experiments

This section uses a simple LASSO algorithm to demonstrate convergence. We demonstrate a bigger application of image deblurring with noises using the FISTA algorithm.

Simple LASSO

As the name suggested, we consider the overuse example problem of:

$$\min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \right\}$$

For a brief background, this problem appears in the context of regression for a generalized linear model where we want sparse coefficients on the regression parameters. Theoretically, it corresponds to having a prior Laplace distribution for the regression parameters (See 11.4.1 in Murphy’s[4] book for more details.). The implementation is simple, and the proximal gradient oracle for $\|\cdot\|_1$ is given as:

$$(\text{prox}_{\lambda\|\cdot\|_1,t}(x))_i = \text{sign}(x_i) \max(|x_i| - t\lambda, 0),$$

which can be interpreted the sign function as the projection onto the interval $[-1, 1]$, and the $\max(|x| - t\lambda, 0)$ as the distance of x to the set $[-t\lambda, t\lambda]$. The quantity t is the stepsize in the proximal gradient method, which is less than the reciprocal of the maximum absolute eigenvalue of $A^T A$.

For this simple lasso problem, we make A a diagonal 128 by 128 matrix, whose diagonals are points equally spaced in the interval $[0, 2]$. Observe that this is quadratic but not strongly convex. The right-hand side vector b is the same as the diagonal of matrix A , but with every odd index replaced with a gaussian random noise on the level of 10^{-3} . we performed the experiment with both ISTA and FISTA with $\lambda = 10^{-2}$. We use a step size of 0.2 for both (It is there to prevent triggering the line search routine in the implementation). The initial guess vector $x^{(0)}$ is a vector of 3, which is the same for both FISTA and ISTA.

For the experiment, we record and present the objective values f for each of the iterations and the norm of the proximal mapping $\|x^{(k+1)} - x^{(k)}\|_\infty$ in the none accelerated case and $\|y^{(k)} - x^{(k+1)}\|_\infty$ in the accelerated case for each iteration. See [figure 1](#) for an illustration. Both algorithms terminate whenever the norm of the proximal gradient mapping is less than 10^{-10} during the iteration. We plotted the norm on a log scale. Observe that the type of convergence for FISTA is very different compared to the ISTA case. In the case of ISTA, the norm of the proximal mapping on the log plot resemble a curve at the start and quickly changes its behaviors in the later iterations. The convergence of iteration after 2000 is a straight line. This phenomenon indicates first-order convergence for ISTA. In FISTA, the overall convergence rate is slower than ISTA for 2000 iterations, and then it starts to slow down; nonetheless, it converges faster. We plot the objective value on the left of [figure 1](#); the plot is on a log scale too. The optimal value $f(\bar{x})$ is assumed to be whenever the gradient mapping has a norm within 10^{-10} . Observe that FISTA already reaches the optimal around 476 iterations, disregarding that the norm of the gradient mapping is not within the tolerance.

Remark 6.0.1. The convergence rate of the optimality gap is linear when the smooth function g is strongly convex; see theorem 10.29 of Beck’s book[1] for the linear convergence of proximal gradient without acceleration. There are variants of FISTA, for example, the Restarted FISTA. Beck’s book, Theorem 10.41 [1] discusses their convergence behaviors.

Image deblurring

We reproduced some of the experiments conducted in Beck’s FISTA paper [2] but using larger images with colors. We consider a cartoon image of a pink unicorn (I own the image) of 500 by 500 pixels, three color channels, blurred. The matrix A does the blurring. It is performing a convolution using discretized 15×15 pixels gaussian kernel with a variance of

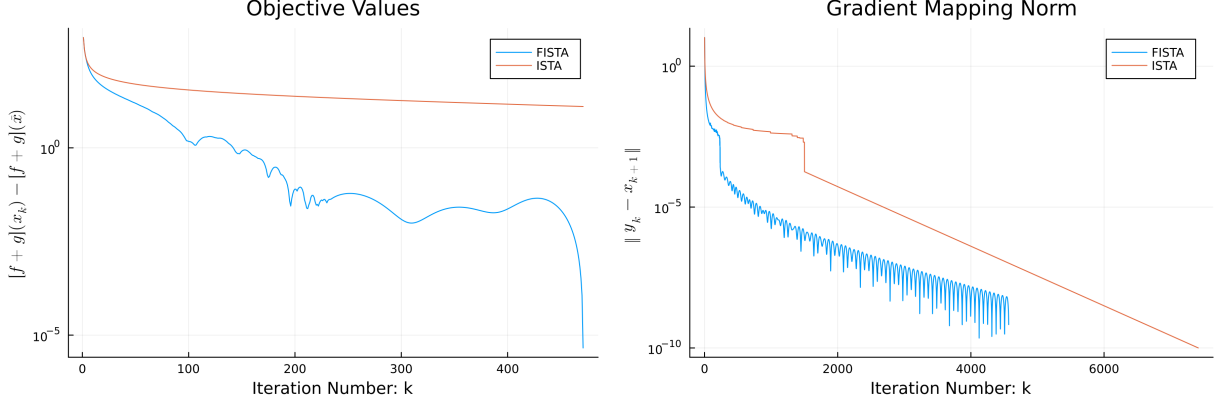


Figure 1: The left is the objective value of the function during all iterations, and the right side is the norm of the gradient mapping for all the iterations.

4 on all pixels with a periodic boundary condition across all color channels independently. The implementation for A is a for loop that constructs a sparse matrix A . The 750000 by 750000 sparse matrix A acts independently on the vectorized image across all three color channels. More efficient ways exist, such as using the Kronecker product. However, for the sake of demonstration, the alternatives are unexplored because the explicit A matrix allows us to reuse the code that made the previous demonstration.

The vector $b = Ax^+ + \epsilon$ where x^+ is the flattened array of the original image of all three color channels normalized to $[0, 1]$ using float64. The quantity ϵ is a zero mean gaussian noise vector with zero mean and variance of 2×10^{-2} . Here we define $\lambda = \alpha \times (3 \times 500^2)^{-1}$, and we make 3 experiments with $\alpha = 0, 0.01, 0.1$. The initial guess vector $x^{(0)}$ is a random zero mean gaussian vector of unit variance. The blurred image is showed in [figure 2](#). Moreover, the results of the deblurring algorithm for different value of α is shown in [figure 3](#), observes that with $\lambda = 0$, the solution contains much noise. However, just a tiny amount of λ prevents the noises on the black background on the recovered image.

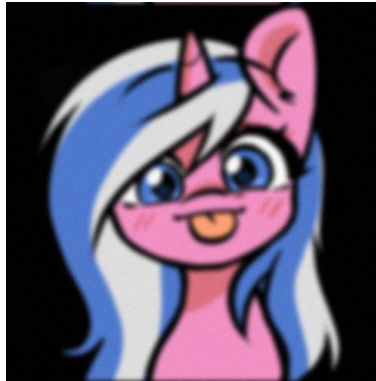


Figure 2: The image is blurred by the Gaussian Blurred matrix A with a tiny amount of noise on the level of 2×10^{-2} that is barely observable. Zoom in to observe the tiny amount of Gaussian noise on top of the blur.



Figure 3: (a) $\alpha = 0$, without any one norm penalty, is not robust to the additional noise. (b) $\alpha = 0.01$, there is a tiny amount of λ . (c) $\alpha = 0.1$, it is more penalty compared to (a).

A A slightly better proof For convergence of FISTA

In this section, we will go through a proof I made, and it does not require the momentum sequence t_k for the FISTA algorithm under [assumption 1](#). We prepare the following template algorithm:

Algorithm 3 Template Proximal Gradient Method With Momentum

- 1: **Input:** $x^{(0)}, x^{(-1)}, L, h, g$; 2 initial guesses and stepsize L
 - 2: $y^{(0)} = x^{(0)} + \theta_k(x^{(0)} - x^{(-1)})$
 - 3: **for** $k = 1, \dots, N$ **do**
 - 4: $x^{(k)} = \text{prox}_{h, l^{-1}}(y^{(k)} + l^{-1}\nabla g(y^{(k)})) =: \mathcal{P}y^{(k)}$
 - 5: $y^{(k+1)} = x^{(k)} + \theta_k(x^{(k)} - x^{(k-1)})$
 - 6: **end for**
-

A.1 Preparations

[Algorithm 3](#) is a template algorithm without any specific assumptions about θ_k , and it is up to ourselves to find out the best update sequences for the momentum parameters θ_k . To make the proof more intuitive than Beck's proof [2], we consider the following list of quantities that is more informative:

1. $v^{(k)} = x^{(k)} - x^{(k-1)}$ is the velocity term.
2. $\bar{v}^{(k)} = \theta_k v^{(k)}$ is the weighed velocity term.
3. $e^{(k)} := x^{(k)} - \bar{x}$, where $\bar{x} \in \arg \min_x (f(x))$, where \bar{x} might not be unique.
4. $\Delta_k := f(x^{(k)}) - f(\bar{x})$ which represent the optimality gap at step k .

A.2 The momentum magic

We are looking for the right place to insert momentum in this part. We start by considering the prox 2 point lemma (lemma 4.1.1) and substitute $x = x^{(k)}, y = y^{(k+1)}$ gives:

$$\begin{aligned}
f(x^{(k)}) - f \circ \mathcal{P}y^{(k+1)} &\geq \frac{L}{2} \|\mathcal{P}y^{(k+1)} - y^{(k+1)}\|^2 + L \langle y^{(k+1)} - x^{(k)}, \mathcal{P}y^{(k+1)} - y^{(k+1)} \rangle \\
[*1] \implies 2L^{-1}(\Delta_k - \Delta_{k+1}) &\geq \|x^{(k+1)} - y^{(k+1)}\|^2 + 2 \langle x^{(k+1)} - y^{(k+1)}, y^{(k+1)} - x^{(k)} \rangle \\
[*2] \implies 2L^{-1}(\Delta_k - \Delta_{k+1}) &\geq \|v^{(k+1)} - \bar{v}^{(k)}\|^2 + 2 \langle v^{(k+1)} - \bar{v}^{(k)}, \bar{v}^{(k)} \rangle \quad (*)
\end{aligned}$$

where we make use of the fact that $x^{(k+1)} = \mathcal{P}y^{(k+1)}$ at [*1], and using $x^{(k+1)} - y^{(k+1)} = x^{(k+1)} - x^{(k)} - \bar{v}^{(k)} = v^{(k+1)} - \bar{v}^{(k)}$ at [*2]. Similarly we can use the prox 2 points lemma (lemma 4.1.1) and substitute $x = \bar{x}, y = y^{(k+1)}$, giving us:

$$\begin{aligned}
-2L^{-1}\Delta_{k+1} &\geq \|x^{(k+1)} - y^{(k+1)}\|^2 + 2 \langle y^{(k+1)} - \bar{x}, x^{(k+1)} - y^{(k+1)} \rangle \\
-2L^{-1}\Delta_{k+1} &\geq \|v^{(k+1)} - \bar{v}^{(k)}\|^2 + 2 \langle v^{(k+1)} - \bar{v}^{(k)}, e^{(k)} + \bar{v}^{(k)} \rangle. \quad (\star)
\end{aligned}$$

we make use of the fact that $y^{(k+1)} = x^{(k)} - \bar{v}^{(k)}$, then $y^{(k+1)} - \bar{x} = x^{(k)} - \bar{v}^{(k)} - \bar{x} = e^{(k)} - \bar{v}^{(k)}$. Without acceleration, we considered the expression (\star) . We did some algebra to sum it up like a telescoping series, similar to the proof we did in theorem 4. As an alternative, we consider the linear combination of $(*)$, (\star) such that it leaves $v^{(k)} - \bar{v}^{(k)}$ inside of the cross term with a multiplier t_{k+1} , let us call it t_k (This is a generic sequence that will contribute to the engineering of the algorithm). As a result $(t_{k+1} - 1)(*) + (\star)$ with $(t_k - 1) \geq 0$ for all k gives

$$\begin{aligned}
2L^{-1}((t_{k+1} - 1)\Delta_k - t_{k+1}\Delta_{k+1}) \\
\geq t_{k+1} \|v^{(k+1)} - \bar{v}^{(k)}\|^2 + 2 \langle t_{k+1}(v^{(k+1)} - \bar{v}^{(k)}), e^{(k)} + t_{k+1}\bar{v}^{(k)} \rangle, \quad (**)
\end{aligned}$$

unfortunately, at the current step, we will not be able to trigger the monotone property and sum it up like in the case without any momentum due to the term t_{k+1} . Instead, we need to consider a new approach. In the next section, we highlight a format for two bounded sequences and reduce the above expression with some conditions on the sequence t_k .

A.3 Two bounded sequences

For the sake of idealization, we may assume that there might exist a way to write $(**)$ in the format of $a_k - a_{k+1} \geq b_{N+1} - b_k$. Therefore we introduce the following lemma:

Lemma A.3.1. Two bounded sequences Consider the sequences $a_k, b_k \geq 0$ for $k \in \mathbb{N}$ with $a_1 + b_1 \leq c$. Inductively the two sequences satisfy $a_k - a_{k+1} \leq b_{k+1} - b_k$, which describes a sequence with oscillations bounded by the difference of another sequence. Consider the

telescoping sum:

$$\begin{aligned}
& a_k - a_{k+1} \geq b_{k+1} - b_k \quad \forall k \in \mathbb{N} \\
\implies & -\sum_{k=1}^N a_{k+1} - a_k \geq \sum_{k=1}^N b_{k+1} - b_k \\
& -(a_{N+1} - a_1) \geq b_{N+1} - b_1 \\
& c \geq a_1 + b_1 \geq b_{N+1} + a_{N+1} \\
\implies & c \geq a_{N+1}.
\end{aligned}$$

Remark A.3.1. If we make the expression $(\star\star)$ the same form as the two sequences, then there is a way to restrain the value of Δ_k . Intuitively we are thinking of bounding the changes in the sequence. If the initial $a_1 + b_1$ is bounded by c , and a_k has oscillations bounded by changes in b_k , then given both a_k, b_k are non-negative, the total amount of changes of a_k will be bounded by the total amount of changes in the sequence b_k as well.

Additionally, we may consider adding a residual term for the sequences $a_k - a_{k+1} \geq b_{k+1} - b_k + r_k$ with a residual term, then the results a_{N+1} would be bounded by a larger quantity. This consideration allows for different sequences of t_k, θ_k is not necessarily satisfying $(\star\star\star)$ in the next section by introducing some error terms, opening up possibilities for more sequences that could potentially have different convergence properties.

A.4 The sequence grows linearly

Lemma A.4.1. The sequence $t_k = \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}$ with basecase $t_0 = 1$ satisfies

$$t_k \geq \frac{1 + k}{2}.$$

Proof. The case when $k = 1, t_1 = 1$ holds true, and it is not hard to see that if for some k it is true, then it is valid for $k + 1$ because:

$$\begin{aligned}
t_k & \geq \frac{k + 1}{2} \\
4t_k^2 & \geq 4 \left(\frac{k + 1}{2} \right)^2 \\
\implies t_{k+1} & = \frac{1}{2} \left(1 + \sqrt{1 + 4t_k^2} \right) \\
& = \frac{1}{2} + \frac{\sqrt{1 + (k + 1)^2}}{2} \\
& \geq \frac{2 + k}{2},
\end{aligned}$$

because $\sqrt{1 + (k + 1)^2} \geq k + 1$. □

A.5 Form matching

Our goal is to match the terms in the previous expression $(\star\star)$ to the form: $a_k - a_{k+1} \leq b_{k+1} - b$. To accomplish that, we simplify $(\star\star)$ by multiplying both sides by t_{k+1} (so that we can move the constant to the inside of the norm instead of letting it dangle outside) and we assume it to be a positive quantity larger than one:

$$\begin{aligned}
& 2L^{-1}((t_{k+1}^2 - t_{k+1})\Delta_k - t_{k+1}^2\Delta_{k+1}) \\
& \geq t_{k+1}^2\|v^{(k+1)} - \bar{v}^{(k)}\|^2 + 2\langle t_{k+1}^2(v^{(k+1)} - \bar{v}^{(k)}), e^{(k)} + t_{k+1}\bar{v}^{(k)} \rangle \\
& = \|t_{k+1}(v^{(k+1)} - \bar{v}^{(k)})\|^2 + 2\langle t_{k+1}(v^{(k+1)} - \bar{v}^{(k)}), e^{(k)} + t_{k+1}\bar{v}^{(k)} \rangle \\
& = \|t_{k+1}v^{(k+1)} - t_{k+1}\bar{v}^{(k)} + e^{(k)} + t_{k+1}\bar{v}^{(k)}\|^2 - \|e^{(k)} - t_{k+1}\bar{v}^{(k)}\|^2 \\
& = \|t_{k+1}v^{(k+1)} + e^{(k)}\|^2 - \|e^{(k)} - t_{k+1}\bar{v}^{(k)}\|^2 \\
[1] \implies & = \|t_{k+1}v^{(k+1)} + e^{(k)}\|^2 - \|v^{(k)} + e^{(k-1)} + t_{k+1}\bar{v}^{(k)}\|^2 \\
& = \|t_{k+1}v^{(k+1)} + e^{(k)}\|^2 - \|e^{(k-1)} + (t_{k+1}\theta_k + 1)v^{(k)}\|^2, \tag{\star\star}
\end{aligned}$$

where at [1], we use the fact that $e^{(k)} = x^{(k)} - \bar{x} = x^{(k)} - x^{(k-1)} + x^{(k-1)} - \bar{x} = v^{(k)} - e^{(k)}$ and to match the form, we would need the sequence of t_k, θ_k to satisfies

$$\begin{cases} t_{k+1}^2 - t_{k+1} = t_k^2, \\ t_k = t_{k+1}\theta_k + 1. \end{cases} \tag{\star\star\star}$$

One of the options is the sequence suggested in the FISTA paper, stated as follows:

$$\begin{aligned} t_k &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \\ \theta_k &= \frac{t_k - 1}{t_{k+1}}, \end{aligned} \tag{\star\star\star}$$

moreover, with these properties for the sequences in mind, we can express $(\star\star)$ in the form of:

$$\underbrace{2L^{-1}t_k^2\Delta_k}_{a_k} - \underbrace{2L^{-1}t_{k+1}^2\Delta_{k+1}}_{a_{k+1}} \geq \underbrace{\|t_{k+1}v^{(k+1)} + e^{(k)}\|^2}_{b_{k+1}} - \underbrace{\|e^{(k-1)} + t_kv^{(k)}\|^2}_{b_k},$$

which has $a_k = 2L^{-1}\Delta_{k+1}$ finally, we observe that setting $k = 1$ on (\star) gives:

$$\begin{aligned} -2L^{-1}\Delta_1 &\geq \|v^{(1)} - \bar{v}^{(0)}\|^2 + 2\langle e^{(0)} - \bar{v}^{(0)}, v^{(1)} - \bar{v}^{(0)} \rangle \\ &\geq \|v^{(1)} - \bar{v}^{(0)} + e^{(0)} - \bar{v}^{(0)}\|^2 - \|e^{(0)} - \bar{v}^{(0)}\|^2 \\ \|e^{(0)} - v^{(0)}\|^2 &\geq \|v^{(1)} + e^{(0)}\|^2 + 2L^{-1}\Delta_1, \end{aligned}$$

now we let $a_1 = 2L^{-1}\Delta_1$, which implies $t_1 = 1$, and hence we also have $b_1 = \|v^{(1)} + e^{(0)}\|^2$ with $c = \|e^{(0)} - v^{(0)}\|^2$, and this completes the base case for using the sequence lemma. Applying the lemma, we obtain

$$\begin{aligned} a_{N+1} &\leq c \\ 2L^{-1}t_{N+1}^2\Delta_{N+1} &\leq \|e^{(0)} - v^{(0)}\|^2, \end{aligned}$$

by [lemma A.4.1](#) asserts convergence for the above expression. We skip the proof for the sequence lower bound here.

References

- [1] Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.
- [2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems, 2009.
- [3] Jeff Bezanson, Stefan Karpinski, Viral B Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*, 2012.
- [4] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.
- [5] Neal Parikh and Stephen Boyd. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, jan 2014.