Monotonicity and Restart in Fast Gradient Methods

Pontus Giselsson and Stephen Boyd

***Abstract*— Fast gradient methods are known to be non-monotone algorithms, and oscillations typically occur around the solution. To avoid this behavior, we propose in this paper a fast gradient method with restart, and analyze its convergence rate. The proposed algorithm bears similarities to other algorithms in the literature, but differs in a key point that enables theoretical convergence rate results. The efficiency of the proposed method is demonstrated by two numerical examples.**

I. Introduction

In his seminal paper from the mid 80's [8], Nesterov presents a fast gradient method that achieves the (up to a constant) optimal convergence rate as described in [7]. The fast gradient method by Nesterov was largely unrecognized for two decades, even though Nesterov presented another optimal gradient method in [9]. From the mid 00's, fast gradient methods have been extended and generalized in several directions. Contributions include [10], where among other things a projected fast gradient method is presented, and [2], where composite objective functions are considered, and [16], that presents a unified framework for many of the fast gradient methods presented in the literature. For more on fast gradient methods and other methods suitable for solving composite optimization problems, the reader is referred to [14].

One characteristic of fast gradient methods, is that the iterates are not monotone. That is, the function values may increase for a couple of iterates, before decreasing again. This is sometimes an undesirable property that has been addressed in [1], [11], where monotone versions of different fast gradient methods have been proposed. In these methods, the function values of the current and the previous iterates are compared, and appropriate measures are taken to ensure monotonicity. These methods share the $O(1/k^2)$ convergence rate of the fast gradient method when applied to smooth functions, and the performance is often similar to the performance of the corresponding non-monotone algorithm. In [12], an algorithm that exploits the non-monotonicity to achieve a faster convergence is presented. They observed that if the algorithm is restarted when a non-monotone behavior is detected, the restarted algorithm often proceeds in a good direction in terms of function value progress. Besides using a monotonicity-test to restart the algorithm, the authors in [12] also propose a gradient-based test that indicates if the iterates tend to oscillate away from the solution. Both these restart schemes often perform well in practice, especially when medium to high accuracy is desired. In [12], these adaptive restart fast gradient methods are shown to converge linearly for strongly convex and smooth functions, but convergence rate results for non-strongly convex and smooth functions are still missing.

In this paper, we will analyze a restart scheme for fast gradient methods that is a slight variation of the method in [12]. This modification of the algorithm allows for proving a $O(1/k^2)$ convergence rate for the restarted fast gradient method when solving smooth problems. We show that the constant in the $O(1/k^2)$ convergence rate is the same as in the standard fast gradient method, except for one term that is added and one term that is subtracted every time the algorithm is restarted. We will argue that the net addition typically is negative when restarted at non-monotonicity. Thus, to restart the algorithm at non-monotonicity, typically improves constant in the convergence rate bound of the algorithm.

We evaluate the performance of the proposed restart scheme by applying it to one randomly generated lasso optimization problem and one model predictive control optimization problem, where the pitch angle of an aircraft is controlled. The numerical examples show that by restarting the algorithm at non-monotonicity, the convergence of the algorithm is often improved.

A. Notation

We denote by $\mathbf{R}$, $\mathbf{R}^n$, and $\mathbf{R}^{m\times n}$ the sets of real numbers, real column vectors of size n, and real matrices of size $m \times n$ respectively. Further, $\overline{\mathbf{R}} = \mathbf{R} \cup \{\infty\}$ denotes the extended real line. Moreover, $\mathbf{S}^n$ ($\mathbf{S}^n_{++}$) [$\mathbf{S}^n_+$] denote the sets of (positive [semi] definite) symmetric matrices of size $n \times n$. We consider Euclidean spaces with inner product $\langle x, y\rangle = x^T y$ and norm $\|x\|_2 = \sqrt{x^T x}$. We also consider Euclidean spaces with scaled norms, i.e. spaces with inner product $\langle x, y\rangle = x^T y$ and scaled norm $\|x\|_L = \sqrt{x^T L x}$. These spaces are denoted by $\mathbb{E}_L$. Finally, we define strong convexity and smoothness.

Definition 1: A function $f : \mathbf{R}^n \to \overline{\mathbf{R}}$ is *β-strongly convex* w.r.t. $\mathbb{E}_L$ if

$$f(x) \geq f(y) + \langle u, x - y\rangle + \tfrac{\beta}{2}\|x - y\|_L^2$$

holds for all $x, y \in \mathbf{R}^n$ and $u \in \partial f(y)$.

Definition 2: A function $f : \mathbf{R}^n \to \mathbf{R}$ is β-smooth w.r.t. $\mathbb{E}_L$ if it is convex, differentiable and if $L \in \mathbf{S}^n_{++}$ is such that

$$f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + \tfrac{\beta}{2}\|x - y\|_L^2$$

holds for all $x, y \in \mathbf{R}^n$.

P. Giselsson and S. Boyd are with the Electrical Engineering Department at Stanford University. E-mail: {pontusg,boyd}@stanford.edu.

II. Fast gradient method with restart

Fast proximal gradient methods can be applied to solve problems of the form

$$\text{minimize } f(x) + g(x) \tag{1}$$

where $x \in \mathbf{R}^n$, and $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R}^n \to \overline{\mathbf{R}}$ satisfy the following assumptions:

Assumption 1:

a) The function $f : \mathbf{R}^n \to \mathbf{R}$ is 1-smooth w.r.t. $\mathbb{E}_L$.
b) The (extended-valued) function $g : \mathbf{R}^n \to \overline{\mathbf{R}}$ is proper, closed and convex.

Fast proximal gradient methods can be applied on different spaces. Here, we consider applying fast gradient methods on the space $\mathbb{E}_L$. The standard Euclidean fast proximal gradient method is obtained by letting $L = \beta I$, where β is a Lipschitz constant to ∇f. The fast gradient method on $\mathbb{E}_L$ is stated below:

Algorithm 1: **Fast proximal gradient method**

Set: $x^0 = x^{-1} \in \mathrm{dom} g, \theta_0 = \theta_{-1} = 1$
For $k \geq 0$

$$\begin{aligned} y^k &= x^k + \theta_k(\theta_{k-1}^{-1} - 1)(x^k - x^{k-1}) \\ x^{k+1} &= \mathrm{prox}_g\left(y^k - L^{-1}\nabla f(y^k)\right) \end{aligned}$$

where on the space $\mathbb{E}_L$, the prox operator is defined as:

$$\mathrm{prox}_g(y) := \operatorname*{argmin}_x \left\{ g(x) + \tfrac{1}{2}\|x - y\|_L^2 \right\}$$

In the first step of Algorithm 1, an auxiliary variable y^k is computed that is a specific linear combination, described by the θ_k-sequence, of the two previous x^k iterates. To guarantee fast convergence of the algorithm, the θ_k:s must satisfy (see [16])

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

This is, e.g., satisfied for $\theta_k = \frac{2}{k+2}$. Another option that decays slightly faster towards zero than $\theta_k = \frac{2}{k+2}$, is obtained by solving (2) with equality:

$$\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2} < \frac{2}{k+3} \tag{3}$$

see [16]. A straight-forward generalization of [16, Corollary 2], gives that if Assumption 1 holds, then Algorithm 1 with θ_k satisfying (3) converges with the rate

$$f_g(x^k) - f_g(x^\star) \leq \frac{2\left\|x^\star - x^0\right\|_L^2}{(k+2)^2} \tag{4}$$

where $f_g := f + g$ and $x^\star$ is an optimal solution to (1). In this paper, we will analyze the convergence behavior of the following generalized fast gradient method with restart. In this algorithm, which is inspired by the adaptive restart method in [12], the algorithm iterates are restarted whenever a certain restart condition holds. The objective of the restart scheme is to improve the performance compared to Algorithm 1 by avoiding a non-monotone behavior.

Algorithm 2:
Generalized fast proximal gradient method w/ restart

Set: $x^0 = x^{-1} \in \mathbf{R}^n, \theta_0 = \theta_{-1} = 1$
For $k \geq 0$

$$\begin{aligned} y^k &= x^k + \theta_k(\theta_{k-1}^{-1} - 1)(x^k - x^{k-1}) \\ x^{k+1} &= \mathrm{prox}_g\left(y^k - L^{-1}\nabla f(y^k)\right) \end{aligned}$$

if restart test holds

$$\begin{aligned} y^k &= x^k \\ x^{k+1} &= \mathrm{prox}_g\left(y^k - L^{-1}\nabla f(y^k)\right) \end{aligned}$$

In the convergence rate analysis of Algorithm 2, we do not rely on a specific restart test, we only assume that the restart test is satisfied for a finite number of iterates. A discussion on different restart tests can be found in Section III. Before we state the convergence rate result of Algorithm 2, we introduce the following function

$$\ell(y, x) = f(y) + \langle \nabla f(y), x - y \rangle + g(x)$$

and note that the x^{k+1}-update can be written as

$$\begin{aligned} x^{k+1} &= \mathrm{prox}_g\left(y^k - L^{-1}\nabla f(y^k)\right) \\ &= \operatorname*{argmin}_x \left\{ g(x) + \langle \nabla f(y^k), x \rangle + \tfrac{1}{2}\|x - y^k\|_L^2 \right\} \\ &= \operatorname*{argmin}_x \left\{ \ell(y^{k+1}, x) + \tfrac{1}{2}\|x - y^k\|_L^2 \right\}. \end{aligned} \tag{5}$$

In the proof of the convergence rate result, we also need the following lemma. A general version of this lemma is stated in [16, Property 1], but here, a different proof is provided.

Lemma 1: Suppose that $\psi : \mathbf{R}^n \to \overline{\mathbf{R}}$ is closed, proper, and convex, that $L \in \mathbf{S}_{++}^n$, and that

$$x^+ = \operatorname*{argmin}_x \{ \psi(x) + \tfrac{1}{2}\|x - z\|_L^2 \}. \tag{6}$$

Then for all $x \in \mathrm{dom}\psi$:

$$\psi(x) + \tfrac{1}{2}\|x - z\|_L^2 \geq \psi(x^+) + \tfrac{1}{2}\|x^+ - z\|_L^2 + \tfrac{1}{2}\|x^+ - x\|_L^2.$$

Proof. Denote by $h_z(x) := \psi(x) + \frac{1}{2}\|x - z\|_L^2$. Then 1-strong convexity of h_z w.r.t $\mathbb{E}_L$ implies that the following holds for all $u \in \partial h_z(x^+)$:

$$\begin{aligned} h_z(x) &\geq h_z(x^+) + \langle u, x - x^+ \rangle + \tfrac{1}{2}\|x^+ - x\|_L^2 \\ &\geq h_z(x^+) + \tfrac{1}{2}\|x^+ - x\|_L^2 \end{aligned}$$

where the second inequality holds due to first order optimality conditions of (6): $\langle u, x - x^+ \rangle \geq 0$ for all $x \in \mathrm{dom} h_z = \mathrm{dom}\psi$ and $u \in \partial h_z(x^+)$. Recalling the definition of h_z gives the result. □

Now, we are ready to state the convergence rate result, the proof of which is inspired by the convergence rate result in [16] for fast gradient method applied on Hilbert spaces.

Theorem 1: Suppose that Assumption 1 holds and that the restart test is satisfied at iterations $k = k_1, \ldots, k_p$. Then

Algorithm 2 with θ_k satisfying (3), converges with the rate

$$f_g(x^{k+1}) - f_g(x^\star) \leq \frac{2}{(k+2)^2}\Bigg(\|x^\star - x^0\|_L^2 + \sum_{k_i \leq k} \left\{\|x^\star - x^{k_i}\|_L^2 - \|x^\star - z^{k_i}\|_L^2\right\}\Bigg) \quad (7)$$

where $z^k = x^{k-1} + \theta_{k-1}^{-1}(x^k - x^{k-1})$.

Proof. Following the proof of [16, Proposition 2], we let $y = (1-\theta_k)x^k + \theta_k x$ and conclude for $k \neq k_i$ that

$$\begin{aligned}
f_g(x^{k+1}) &\leq \ell(y^k, x^{k+1}) + \tfrac{1}{2}\|y^k - x^{k+1}\|_L^2 \\
&\leq \ell(y^k, y) + \tfrac{1}{2}\|y^k - y\|_L^2 - \tfrac{1}{2}\|x^{k+1} - y\|_L^2 \\
&= \ell(y^k, y) + \tfrac{1}{2}\|(1-\theta_k)x^k + \theta_k x - y^k\|_L^2 \\
&\quad - \tfrac{1}{2}\|(1-\theta_k)x^k + \theta_k x - x^{k+1}\|_L^2 \\
&= \ell(y^k, y) + \theta_k^2 \tfrac{1}{2}\|x + \theta_k^{-1}(x^k - y^k) - x^k\|_L^2 \\
&\quad - \theta_k^2 \tfrac{1}{2}\|x + \theta_k^{-1}(x^k - x^{k+1}) - x^k\|_L^2 \\
&= \ell(y^k, y) + \theta_k^2 \tfrac{1}{2}\|x - z^k\|_L^2 - \theta_k^2 \tfrac{1}{2}\|x - z^{k+1}\|_L^2 \\
&= \ell(y^k, (1-\theta_k)x^k + \theta_k x) + \theta_k^2 \tfrac{1}{2}\|x - z^k\|_L^2 \\
&\quad - \theta_k^2 \tfrac{1}{2}\|x - z^{k+1}\|_L^2 \\
&\leq (1-\theta_k)\ell(y^k, x^k) + \theta_k \ell(y^k, x) + \theta_k^2 \tfrac{1}{2}\|x - z^k\|_L^2 \\
&\quad - \theta_k^2 \tfrac{1}{2}\|x - z^{k+1}\|_L^2 \\
&\leq (1-\theta_k) f_g(x^k) + \theta_k f_g(x) + \theta_k^2 \tfrac{1}{2}\|x - z^k\|_L^2 \\
&\quad - \theta_k^2 \tfrac{1}{2}\|x - z^{k+1}\|_L^2 \qquad (8)
\end{aligned}$$

where the first inequality holds since f is 1-smooth w.r.t. $\mathbb{E}_L$, the second equality is due to (5) and Lemma 1, the first and second equalities insert y and rearrange, the third equality inserts y^k and identifies z^k, the fourth equality inserts y, the third inequality uses convexity of $\ell(\cdot, x)$ and that $\theta_k \in (0, 1]$, and the last inequality uses convexity of f which implies that $f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle$, and that $\theta_k \in (0, 1]$.

For $k = k_i$, we have $y^k = x^k$, which implies that in the third equality in (8), we get

$$\|x + \theta_k^{-1}(x^k - y^k) - x^k\|_L^2 = \|x - x^k\|_L^2 \qquad (9)$$

instead of $\|x - z^k\|_L^2$. Performing the computations in (8) again, using (9) in the third equality, gives for $k = k_i$:

$$f_g(x^{k+1}) \leq (1-\theta_k) f_g(x^k) + \theta_k f_g(x) + \theta_k^2 \tfrac{1}{2}\|x - x^k\|_L^2 - \theta_k^2 \tfrac{1}{2}\|x - z^{k+1}\|_L^2. \quad (10)$$

Letting $x = x^\star$, dividing (8) by θ_k^2, and rearranging the terms give

$$\tfrac{1}{\theta_k^2}(f_g(x^{k+1}) - f_g(x^\star)) \leq \tfrac{1-\theta_k}{\theta_k^2}(f_g(x^k) - f_g(x^\star)) + \tfrac{1}{2}\|x^\star - z^k\|_L^2 - \tfrac{1}{2}\|x^\star - z^{k+1}\|_L^2 \quad (11)$$

for $k \neq k_i$. Similarly, letting $x = x^\star$, dividing (10) by θ_k^2, and rearranging the terms give

$$\tfrac{1}{\theta_k^2}(f_g(x^{k+1}) - f_g(x^\star)) \leq \tfrac{1-\theta_k}{\theta_k^2}(f_g(x^k) - f_g(x^\star)) + \tfrac{1}{2}\|x^\star - x^k\|_L^2 - \tfrac{1}{2}\|x^\star - z^{k+1}\|_L^2 \quad (12)$$

for $k = k_i$. Further, due to (2) and (3) we have $\frac{1}{\theta_k^2} = \frac{1-\theta_{k+1}}{\theta_{k+1}^2}$, which enables telescope summation of (11) and (12). Summing from $k = k_i$ to $k_i + 1 \leq k \leq k_{i+1} - 1$ for any $i = 0, \ldots, p$ (where we define $k_0 = 0$ and $k_{p+1} = \infty$), gives

$$\tfrac{1}{\theta_k^2}(f_g(x^{k+1}) - f_g(x^\star)) \leq \tfrac{1-\theta_{k_i}}{\theta_{k_i}^2}(f_g(x^{k_i}) - f_g(x^\star)) + \tfrac{1}{2}\|x^\star - x^{k_i}\|_L^2 - \tfrac{1}{2}\|x^\star - z^{k+1}\|_L^2 \quad (13)$$

and especially, if summing from k_i to $k_{i+1} - 1$, we get

$$\tfrac{1}{\theta_{k_{i+1}-1}^2}(f_g(x^{k_{i+1}}) - f_g(x^\star)) \leq \tfrac{1-\theta_{k_i}}{\theta_{k_i}^2}(f_g(x^{k_i}) - f_g(x^\star)) + \tfrac{1}{2}\|x^\star - x^{k_i}\|_L^2 - \tfrac{1}{2}\|x^\star - z^{k_{i+1}}\|_L^2. \quad (14)$$

Again performing telescope summation of (13) and (14) gives for any $k \geq 0$ that

$$\begin{aligned}
&f_g(x^{k+1}) - f_g(x^\star) \\
&\leq \frac{\theta_k^2}{2}\Bigg(\|x - x^0\|_L^2 + \sum_{k_i \leq k}\left\{\|x^\star - x^{k_i}\|_L^2 - \|x^\star - z^{k_i}\|_L^2\right\}\Bigg) \\
&\leq \frac{2}{(k+2)^2}\Bigg(\|x^\star - x^0\|_L^2 + \sum_{k_i \leq k}\left\{\|x^\star - x^{k_i}\|_L^2 - \|x^\star - z^{k_i}\|_L^2\right\}\Bigg)
\end{aligned}$$

since $\theta_0 = 1$, $z^0 = x^0$, and $\theta_k \leq \frac{2}{k+2}$ (as noted in (3)). This concludes the proof. □

Remark 1: If Algorithm 2 never enters the if-clause, the sum in (7) is zero, and the theoretical convergence rate for Algorithm 2 reduces to the convergence rate (4) for Algorithm 1, which is Algorithm 2 without restart.

The convergence result in Theorem 1 suggests how to improve the performance of fast gradient methods using a restart scheme; if the algorithm is restarted for iterates k_i that satisfy $\|x^\star - x^{k_i}\|_L^2 - \|x^\star - z^{k_i}\|_L^2 < 0$ then the constant in the theoretical convergence rate expression of Algorithm 2 is improved. However, this test cannot be used in practice since it involves the optimal solution $x^\star$. In the following section, we will analyze typical situations when $\|x^\star - x^{k_i}\|_L^2 - \|x^\star - z^{k_i}\|_L^2 < 0$. This analysis will guide us in developing tests that typically imply $\|x^\star - x^{k_i}\|_L^2 - \|x^\star - z^{k_i}\|_L^2 < 0$. Before we proceed to discuss possible restart tests, we conclude this section with a remark on similarities and differences between Algorithm 2 and other similar algorithms from the literature.

Remark 2: As mentioned, Algorithm 2 bears similarities to the algorithm in [12]. The main difference is that after a restart test holds in [12], $\theta_k = \theta_{k-1} = 1$, while in our algorithm, the θ_k is unaffected. This difference makes it possible to prove a $O(1/k^2)$ convergence rate of Algorithm 2 when applied to composite problems with one smooth and one non-smooth component. Such results are not available for the algorithm in [12]. Further, numerical experiments suggest that, when using the same restart rule, Algorithm 2 perform slightly better in practice than the algorithm in [12].

Algorithm 2 also has similarities with the monotone fast gradient method in [1]. In [1], the y^k-update after non-monotonicity is constructed such the third equality in (8)

holds. This implies that the sum in the convergence rate result of Theorem 1 disappears. Thus, the monotone fast gradient method in [1] has the same theoretical convergence rate (4) as the non-monotone version. Numerical experiments also verify that the monotone algorithm in [1] often performs similarly to its non-monotone counterpart, and that it is often outperformed by Algorithm 2 if medium to high accuracy is desired.

III. Restart conditions

In this section, we present some tests that can be used for restarting Algorithm 2. These tests should ideally imply that $\|x^\star - x^k\|_L^2 < \|x^\star - z^k\|_L^2$, which, if holds, improves the constant in the convergence rate expression of Algorithm 2, see Theorem 1. To gain some intuition on when $\|x^\star - x^k\|_L^2 < \|x^\star - z^k\|_L^2$ might hold, Figure 1 shows the $\{x^k\}$ and $\{z^k\}$-sequences when solving a 2-d QP using Algorithm 1 (i.e. the proximal fast gradient method without restart) with $L = \beta I$, and Figure 2 shows the corresponding function value progress. In Figure 2, we see an almost non-monotone behavior starting at around iteration 11 and a non-monotone behaviour starting at iteration 49. These iteration numbers are also marked in Figure 1. We see around iteration 11, that $\|z^{11} - x^\star\|_2 \approx \|x^{11} - x^\star\|_2$ (note that the level curves in Figure 1 are level curves for the function value, not for the Euclidean distance, and note that we compare in Euclidean distances since $L = \beta I$). However, for iteration 49 we clearly have $\|z^{49} - x^\star\|_2 > \|x^{49} - x^\star\|_2$. This implies that restarting the algorithm at iteration 49, would improve the theoretical convergence rate. This simple example suggests that $\|z^k - x^\star\|_2$ is greater than $\|x^k - x^\star\|_2$ for k where non-monotonicity is detected. Also, the more non-monotone the behavior, the greater the difference. Although this is only a simple unconstrained 2-dimensional example, we have observed that the same pattern emerges also in higher dimensions and on problems with an additional non-smooth term.

Figures 1 and 2 also indicate that the non-monotone behavior of $f(x^k)$ and the large over-shoot for the z^k iterates, occur when the x^k trajectory is changing its principal direction. This clearly happens at around iteration 11, in Figure 1, but also at iteration 49 when the x^k sequence starts to decelerate to get back towards the optimal solution. Here, the momentum term pushes the x^k trajectory past the optimal solution, and a restart of the momentum at this point could have great positive impact on the convergence.

The preceding discussion suggests that a good detector for $\|x^\star - x^k\|_L^2 < \|x^\star - z^k\|_L^2$ is to detect non-monotonicity. Sometimes, for instance in duality based optimization, function evaluations are expensive, and other methods might be better suited for the restart test. Next, we propose a computationally inexpensive method that implies non-monotonicity. The method relies on a generalization of a result in [2, Lemma 2.3], namely:

Lemma 2: Suppose that Assumption 1 holds. Then for any

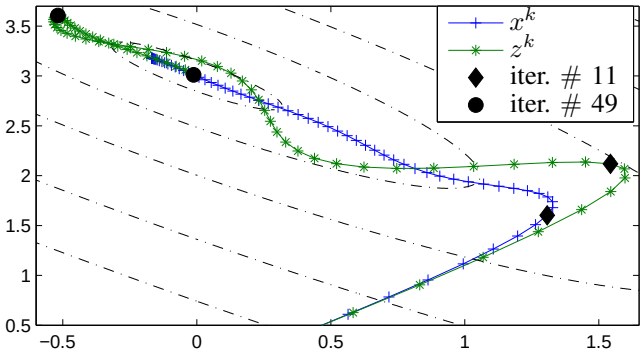

Fig. 1. x^k and z^k-trajectories for a 2-d QP without constraints.

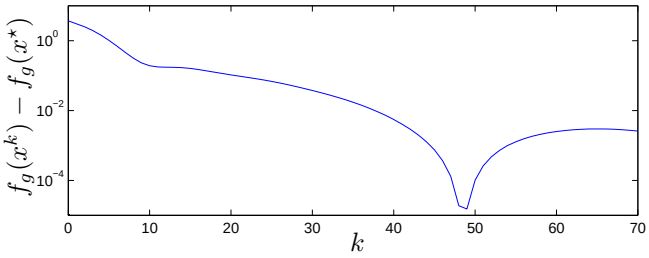

Fig. 2. Function value progress for the x^k trajectory in Figure 1.

$x, y \in \mathrm{dom} g$

$$f_g(x) - f_g(\bar{x}) \geq \frac{1}{2}\|\bar{x} - y\|_L^2 + \langle y - x, L(\bar{x} - y)\rangle \quad (15)$$

where $\bar{x} = \mathrm{prox}_g\left(y - L^{-1}\nabla f(y)\right)$.

The case where L is restricted to be a multiple of the identity matrix, i.e. the Euclidean fast proximal gradient method case, is considered in [2, Lemma 2.3]. The generalization to arbitrary $L \in \mathbf{S}_{++}^n$ is straight-forward and omitted here for space considerations.

Letting $x = x^{k+1}$ and $y = y^{k-1}$ in Lemma 2, where x^k and y^k are generated by Algorithm 2, gives

$$\begin{aligned} &f_g(x^{k+1}) - f_g(x^k) \\ &\quad \geq \frac{1}{2}\|x^k - y^{k-1}\|_L^2 + \langle y^{k-1} - x^{k+1}, L(x^k - y^{k-1})\rangle \\ &\quad = \langle L(y^{k-1} - x^k), x^{k+1} - \tfrac{1}{2}(x^k + y^{k-1})\rangle \end{aligned}$$

since $x^k = \mathrm{prox}_g\left(y^{k-1} - L^{-1}\nabla f(y^{k-1})\right)$. Thus

$$\langle L(y^{k-1} - x^k), x^{k+1} - \tfrac{1}{2}(x^k + y^{k-1})\rangle > 0$$

implies that $f(x^{k+1}) > f(x^k)$. This test cannot be used at the first iterate after a restart, since then $x^k \neq \mathrm{prox}_g\left(y^{k-1} - L^{-1}\nabla f(y^{k-1})\right)$, which is an assumption for Lemma 2 to hold.

Besides the exact monotonicity test, a gradient based test is also proposed in [12], namely

$$\langle L(y^k - x^{k+1}), x^{k+1} - x^k\rangle > 0.$$

This is referred to as a gradient-based test since $L(y^k - x^{k+1})$ is the gradient mapping (which is a generalization of the gradient) for the x^{k+1}-update. It is in [12] noted that

this approach can have advantageous numerical properties compared to the exact non-monotonicity test. Further, for problems where function evaluations are expensive, it is often a computationally cheaper method.

To summarize, we will use three tests to detect when it might be beneficial to restart the algorithm. The tests are:

T1: Exact non-monotonicity test:
$f(x^{k+1}) - f(x^k) > 0$

T2: Gradient-mapping based test:
$\langle L(y^k - x^{k+1}), x^{k+1} - x^k \rangle > 0$

T3: Non-monotonicity implying test:
$\langle L(y^{k-1} - x^k), x^{k+1} - \frac{1}{2}(x^k + y^{k-1}) \rangle > 0$

We will refer to these tests as T1, T2, and T3 respectively.

IV. Numerical examples

In this section, we evaluate the proposed restart scheme for fast gradient methods by applying it to optimization problems arising in compressed sensing and model predictive control.

A. Compressed sensing

In this section, Algorithm 2 is evaluated by applying it to the following lasso problem

$$\text{minimize } \tfrac{1}{2}\|Ax - b\|_2^2 + \gamma\|x\|_1 \qquad (16)$$

where $A \in \mathbf{R}^{100\times 1000}$ is a sparse matrix with approximate density of 0.07 and each non-zero element is drawn from a Gaussian distribution with zero mean and unit variance, $b \in \mathbf{R}^{100}$ has all elements drawn from a Gaussian distribution with zero mean and unit variance, $x \in \mathbf{R}^{1000}$ is the decision variable and $\gamma = 1$. Problem (16) satisfies Assumption 1 by letting $f(x) := \frac{1}{2}\|Ax - b\|_2^2$ and $g(x) := \gamma\|x\|_1$.

In Figure 3, Algorithm 2 is compared to the restart algorithm in [12], to MFISTA in [1] which is a monotone version of FISTA [2], and to Algorithm 1 which is a fast gradient method without restart. Algorithm 2 and the restart algorithm in [12] both use the exact function value test, i.e. test T1. Figure 3 reveals that MFISTA performs similarly to Algorithm 1 without restart, but exhibits a monotone behavior. Figure 3 also shows that Algorithm 2 and the restart algorithm in [12] perform much better than Algorithm 1 and MFISTA, at least in the medium to high accuracy range. The figure also indicates that the performance of Algorithm 2 is very similar to the performance of the restart algorithm in [12].

In Figure 4, we compare the restart tests proposed in Section III. The figure reveals that the exact function value test, T1, and the gradient-mapping based test, T2, introduced in [12], perform better than the test that implies non-monotone behavior, T3. Obviously, the non-monotonicity implying test, T3, need not directly indicate when non-monotonicity occurs. This implies that the algorithm is restarted more often using the exact function value test, T1, than using the non-monotonicity implying test, T3. We have also observed that the gradient-mapping test, T2, tend to be satisfied slightly more often than the exact function value test, T1. This implies that Algorithm 2 with the gradient-mapping test, T2, is typically restarted most often, then

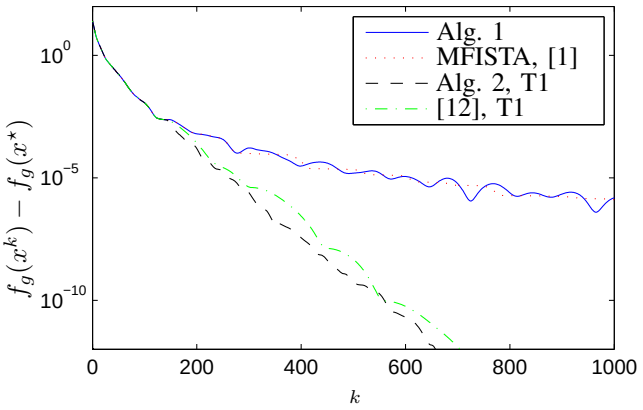

Fig. 3. Function value progress for Algorithm 2, Algorithm 1, and the algorithms in [12], [1]. Algorithm 2 and the algorithm in [12] use the exact monotonicity restart test T1.

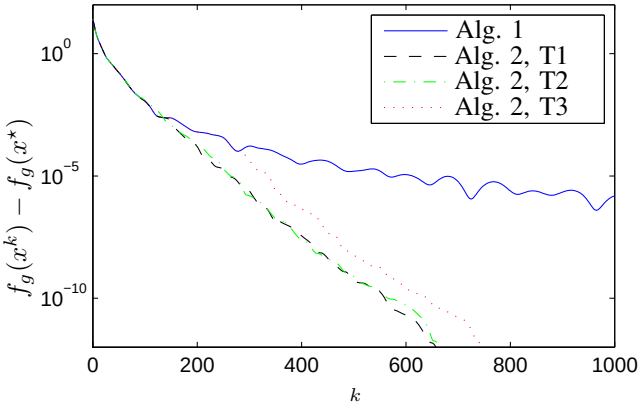

Fig. 4. Function value progress for Algorithm 1, and Algorithm 2 with restart tests T1, T2, and T3.

Algorithm 2 with the exact function value test, T1, and least often is Algorithm 2 with non-monotonicity implying test, T3, restarted. Often, the monotonicity test, T1, performs well. However, sometimes it is beneficial to restart slightly more often, i.e. to use T2, while sometimes it beneficial to restart slightly more seldom, i.e. to use T3. However, the most typical scenario we have encountered is depicted in Figure 4.

B. Model predictive control

We further evaluate Algorithm 2 by using it in model predictive control of the AFTI-16 aircraft model in [6], [3]. As in [3], the continuous time model from [6] is sampled using zero-order hold every 0.05 s. The system has four states $x = (x_1, x_2, x_3, x_4)$, two outputs $y = (y_1, y_2)$, two inputs $u = (u_1, u_2)$, and obeys the following dynamics

$$x^+ = \begin{bmatrix} 0.999 & -3.008 & -0.113 & -1.608 \\ -0.000 & 0.986 & 0.048 & 0.000 \\ 0.000 & 2.083 & 1.009 & -0.000 \\ 0.000 & 0.053 & 0.050 & 1.000 \end{bmatrix} x + \begin{bmatrix} -0.080 & -0.635 \\ -0.029 & -0.014 \\ -0.868 & -0.092 \\ -0.022 & -0.002 \end{bmatrix} u,$$
$$y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x$$

where x^+ denotes the state in the next time step. The dynamics, input, and output matrices are denoted by Φ, Γ, C respectively, i.e. we have $x^+ = \Phi x + \Gamma u, y = Cx$. The

TABLE I

COMPARISON BETWEEN DIFFERENT MONOTONICITY SCHEMES.

		exec time (ms)		nbr iters	
Alg.	Test	avg.	max	avg.	max
Alg. 1	-	1.9	9.7	31.9	175
Alg. 2	T1	2.7	9.1	19.6	69
[12]	T1	3.1	11.4	22.2	84
Alg. 2	T2	1.7	5.7	19.9	72
[12]	T2	1.9	6.4	22.3	82
Alg. 2	T3	1.9	6.3	21.1	74
[12]	T3	2.1	6.9	23.3	80
MFISTA, [1]	-	4.3	20.0	31.3	144

system is unstable, the magnitude of the largest eigenvalue of the dynamics matrix is 1.313. The outputs are the attack and pitch angles, while the inputs are the elevator and flaperon angles. The inputs are physically constrained to satisfy $|u_i| \leq 25^\circ$, $i = 1, 2$. The outputs are soft constrained to satisfy $-s_1 - 0.5 \leq y_1 \leq 0.5 + s_2$ and $-s_3 - 100 \leq y_2 \leq 100 + s_4$ respectively, where $s = (s_1, s_2, s_3, s_4) \geq 0$ are slack variables. The cost in each time step is

$$\ell(x, u, s) = \frac{1}{2}\big((x - x_r)^T Q(x - x_r) + u^T R u + s^T S s\big)$$

where $Q = C^T Q_y C + Q_x$, where $Q_y = 10^2 I$ and $Q_x = \mathrm{diag}(10^{-4}, 0, 10^{-3}, 0)$, x_r is such that $y_r = C x_r$ where y_r is the output reference that can vary in each step, $R = 10^{-2} I$, and $S = 10^6 I$. This gives condition number 10^{10} of the full cost matrix. Further, the terminal cost is Q, and the control and prediction horizon is $N = 10$. The numerical data is obtained by following a reference trajectory on the output. The objective is to change the pitch angle from 0° to 10° and then back to 0° while the angle of attack satisfies the output constraints $-0.5^\circ \leq y_1 \leq 0.5^\circ$. The constraints on the angle of attack limits the rate on how fast the pitch angle can be changed.

We use the same splitting as in [15], [13] (which, among other splittings, is also used in [4]) to pose the optimal control problem for the pitch control. The dual to this optimization problem is preconditioned using the method in [5] and solved using Algorithm 1, Algorithm 2, and the algorithms in [12] and [1]. See [4] for details on how the problem is formulated. To create an easily transferable and fair termination criterion between algorithms, the optimal solution to each optimization problem $x^\star$ is computed to high accuracy using an interior point solver. Then, the termination criterion is $\|x^k - x^\star\|_2 / \|x^\star\|_2 \leq 0.001$, where x^k here denotes the primal iterate in the algorithm.

Table I reveals that MFISTA requires fewer iterations than Algorithm 1, but that the execution time is worse because of the function evaluations needed for the monotonicity tests. However, both these algorithms perform worse than the proposed Algorithm 2 and the algorithm in [12]. Table I also shows that for all tests (T1, T2, T3) Algorithm 2 performs very similarly to the algorithm in [12]. It further says that in terms of iteration count, the tests T1, T2, and T3 perform similarly, with a slight bias towards T1, then T2. However, the execution time of T1, which is the exact function evaluation test, is much higher. This is due to the high computational cost associated with computing the dual function value, since it is implicitly defined as the optimal value of an optimization problem.

V. CONCLUSIONS

We have proposed a restart method for fast gradient methods and proven a $O(1/k^2)$ convergence rate for composite optimization problems with one smooth and one non-smooth term. The proposed method is similar to the one in [12], but differs in a key point that enables for the $O(1/k^2)$ convergence rate to be proven. Two numerical examples are provided that demonstrate the efficiency of the proposed method when medium to high accuracy of the solution is desired.

REFERENCES

[1] 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, 2009.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[3] A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, 1997.

[4] P. Giselsson. Improved fast dual gradient methods for embedded model predictive control. In *Proceedings of 2014 IFAC World Congress*, Cape Town, South Africa, 2014. Accepted for publication. Available `https://www.control.lth.se/Staff/PontusGiselsson.html`.

[5] P. Giselsson and S. Boyd. Preconditioning in fast dual gradient methods. In *Conference on Decision and Control*, 2014. Submitted.

[6] P. Kapasouris, M. Athans, and G. Stein. Design of feedback control systems for unstable plants with saturating actuators. In *Proceedings of the IFAC Symposium on Nonlinear Control System Design*, pages 302–307. Pergamon Press, 1990.

[7] A Nemirovsky and D Yudin. *Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems*. Wiley, New York, NY, 1983.

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

[9] Y. Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ékonom. i. Mat. Metody*, 24:509–517, 1988.

[10] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Netherlands, 1st edition, 2003.

[11] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, May 2005.

[12] B. O'Donoghue and E. Candés. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, pages 1–18, 2013.

[13] B. O'Donoghue, G. Stathopoulos, and S. Boyd. A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 21(6):2432–2442, 2013.

[14] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[15] P. Patrinos and A. Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014.

[16] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report. Available: `http://www.csie.ntu.edu.tw/~b97058/tseng/papers/apgm.pdf`, May 2008.