Average Curvature FISTA for Nonconvex Smooth Composite Optimization Problems

May 13, 2021
first revision: March 10, 2022
second revision: February 3, 2023

Jiaming Liang [1] and Renato D.C. Monteiro [2]

Abstract

A previous authors' paper introduces an accelerated composite gradient (ACG) variant, namely AC-ACG, for solving nonconvex smooth composite optimization (N-SCO) problems. In contrast to other ACG variants, AC-ACG estimates the local upper curvature of the N-SCO problem by using the average of the observed upper-Lipschitz curvatures obtained during the previous iterations, and uses this estimation and two composite resolvent evaluations to compute the next iterate. This paper presents an alternative FISTA-type ACG variant, namely AC-FISTA, which has the following additional features: i) it performs an average of one composite resolvent evaluation per iteration; and ii) it estimates the local upper curvature by using the average of the previously observed upper (instead of upper-Lipschitz) curvatures. These two properties acting together yield a practical AC-FISTA variant which substantially outperforms earlier ACG variants, including the AC-ACG variants discussed in the aforementioned authors' paper.

Key words. nonconvex smooth composite optimization, average curvature, accelerated composite gradient methods, FISTA, first-order methods, line search free methods.

AMS subject classifications. 49M05, 49M37, 65K05, 68Q25, 90C26, 90C30.

1 Introduction

This paper studies a FISTA-type accelerated composite gradient (ACG) algorithm, namely the AC-FISTA method, for solving the nonconvex smooth composite optimization (N-SCO) problem

$$\phi_* := \min \{\phi(z) := f(z) + h(z) : z \in \mathbb{R}^n\} \tag{1}$$

where $h : \mathbb{R}^n \to (-\infty, \infty]$ is a proper lower semicontinuous convex function and f is a real-valued differentiable (possibly nonconvex) function with an L-Lipschitz continuous gradient on a compact

[1] Department of Computer Science, Yale University, New Haven, CT 06511. (email: `jiaming.liang@yale.edu`). This author was partially supported by ONR Grant N00014-18-1-2077 and NSF grant CCF-1740776 through IDEaS-TRIAD Research Scholarship from Georgia Tech.

[2] School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. (email: `monteiro@isye.gatech.edu`). This author was partially supported by ONR Grant N00014-18-1-2077 and AFOSR Grant FA9550-22-1-0088.

convex set containing $\operatorname{dom} h$. The N-SCO problem (1) has a wide range of real-world applications including support vector machine [4], sparse PCA [6], matrix completion [16], and nonnegative matrix factorization [5, 9]. ACG methods are widely used optimization approaches for solving N-SCO problems. A critical issue related to the practical performance of these methods lies on the development of efficient stepsize selection strategies.

More specifically, a key step in ACG methods for solving the N-SCO problem (1) is to compute an iterate y_{k+1} as the solution of a proximal subproblem of the form

$$y_{k+1} = y(\tilde{x}_k; M_k) := \operatorname{argmin} \left\{ \ell_f(x; \tilde{x}_k) + h(x) + \frac{M_k}{2} \|x - \tilde{x}_k\|^2 : x \in \mathbb{R}^n \right\} \tag{2}$$

where $\ell_f(x; \tilde{x}_k) := f(\tilde{x}_k) + \langle \nabla f(\tilde{x}_k), x - \tilde{x}_k \rangle$, $\tilde{x}_k$ is a convex combination of y_k and another auxiliary iterate x_k, and M_k is a positive scalar such that

$$\mathcal{C}(y(\tilde{x}_k; M_k); \tilde{x}_k) \le \tau M_k \tag{3}$$

where $\tau \in (0, 1]$ and

$$\mathcal{C}(y; \tilde{x}) := \frac{2\,[f(y) - \ell_f(y; \tilde{x})]}{\|y - \tilde{x}\|^2}. \tag{4}$$

It can be shown that the smaller the sequence $\{M_k\}$ is, the faster the convergence rate of the method becomes. Hence, it is desirable to choose $M_k = \bar{M}_k$ where $\bar{M}_k$ is the smallest value of M_k satisfying (3). Since $\bar{M}_k$ is hard to compute, a large class of ACG methods for solving either convex or nonconvex SCO problems simply computes a scalar M_k satisfying (3) either by setting it to be a sufficiently large constant or by using a line search procedure. Works dealing with ACG methods for solving nonconvex SCO problems based on this idea are discussed in "Other related works" below.

In contrast to the ACG methods based on the above ideas, the AC-ACG methods of [13] do not require the next iterate y_{k+1} to satisfy (3). They instead follow the natural geometrical viewpoint of choosing M_k, and hence the local approximate model in (2), by means of average curvature information. More specifically, the theoretical version of AC-ACG in [13] computes y_{k+1} as in (2) with M_k set to be a positive multiple of the average of all observed curvatures $\tilde{\mathcal{C}}_0, \ldots, \tilde{\mathcal{C}}_{k-1}$, where $\tilde{C}_i := \tilde{\mathcal{C}}(y_{i+1}; \tilde{x}_i)$ for every i, and

$$\tilde{\mathcal{C}}(y; \tilde{x}) := \max\left\{\mathcal{C}(y; \tilde{x}), \mathcal{L}(y; \tilde{x})\right\}, \quad \mathcal{L}(y; \tilde{x}) := \frac{\|\nabla f(y) - \nabla f(\tilde{x})\|}{\|y - \tilde{x}\|}. \tag{5}$$

It is shown in Theorem 2.1 of [13] that, for every k, the theoretical version of AC-ACG generates a pair $(\hat{y}_k, \hat{v}_k)$ satisfying $\hat{v}_k \in \nabla f(\hat{y}_k) + \partial h(\hat{y}_k)$ and $\|\hat{v}_k\|^2 = \mathcal{O}(M_k/k)$, and hence that its convergence rate directly depends on the magnitude of M_k. Paper [13] also presents a practical aggressive variant of AC-ACG, which computes M_k using the average of the $\mathcal{C}_i$'s, $i = 0, \ldots, k-1$, where $C_i := \mathcal{C}(y_{i+1}; \tilde{x}_i)$ for every i, instead of the usually much larger $\tilde{C}_i$'s. Most likely due to smaller size of the generated sequence $\{M_k\}$, the practical AC-ACG variant computationally outperforms other ACG variants, including its theoretical variant, but its convergence rate analysis is left open in [13].

This paper presents the AC-FISTA method for solving (1) which is a FISTA-type variant of AC-ACG, and establishes a convergence rate for it similar to the one described above but with M_k obtained in the same way as in the practical AC-ACG variant. AC-FISTA has the following advantages compared to the theoretical AC-ACG variant. It computes about half the number of composite resolvent evaluations as that performed by AC-ACG, and hence its iterations are computationally cheaper. It uses C_k (in place of $\tilde{C}_k$) to compute M_k, and hence generates a sequence of smaller curvature estimates $\{M_k\}$. (It is worth noting that, even though this last property is established in the context of AC-FISTA, it indirectly addresses the aforementioned open question of [13] posed in the context of AC-ACG.) Finally, computational results are presented in this paper to demonstrate that AC-FISTA substantially outperforms previous ACG variants as well as the theoretical and practical AC-ACG variants, both in terms of CPU time and computed solution quality.

Other related works. The first convergence analysis of an ACG algorithm based on (3) for solving the N-SCO problem (1) under the same assumptions as in this paper appears in [3]. Inspired by [3], many papers have proposed other ACG variants based on (3). Algorithms presented in [2, 11, 14, 17] choose M_k constant as in [3], i.e., $M_k = L/\tau$ for every k, where L is the Lipschitz constant of ∇f and $\tau \in (0, 1]$. Moreover, algorithms discussed in [4, 10, 14] use line search procedures to compute a relatively small scalar M_k satisfying (3).

In addition to the ACG methods mentioned above, it is worth discussing other approaches for solving (1) that use an inexact proximal point method where each proximal subproblem is constructed to be (possibly strongly) convex and hence solved by a convex ACG variant. Papers [1, 7, 15] describe a descent unaccelerated inexact proximal-type method that works with a large prox stepsize and approximately solves a proximal subproblem by an ACG variant. Finally, [12] proposes an accelerated inexact proximal point method, which in each outer iteration performs an accelerated step with a large prox stepsize and follows the same way as in the algorithms presented in [1, 7] to solve a proximal subproblem.

Basic definitions and notation. Let $\mathbb{R}$ and $\mathbb{R}_+$ denote the set of real numbers and the set of non-negative real numbers, respectively. Let $\mathbb{R}^n$ denote the standard n-dimensional Euclidean space with inner product and norm denoted by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$, respectively. The Frobenius norm in $\mathbb{R}^{m\times n}$ is denoted by $\|\cdot\|_F$. The set of real symmetric positive semidefinite matrices in $\mathbb{R}^{n\times n}$ is denoted by $\mathcal{S}^n_+$. Let $\lceil\cdot\rceil$ denote the ceiling function. The cardinality of a finite set A is denoted by $|A|$. The indicator function I_X of a set $X \subset \mathbb{R}^n$ is defined as $I_X(z) = 0$ for every $z \in X$, and $I_X(z) = \infty$, otherwise. The diameter of a compact set $X \subset \mathbb{R}^n$ is $D_X := \sup\{\|z - \bar{z}\| : z, \bar{z} \in X\}$. If X is a nonempty closed convex set, the orthogonal projection $P_X : \mathbb{R}^n \to \mathbb{R}^n$ onto X is defined as

$$P_X(z) := \operatorname{argmin}_{\bar{z}\in X}\|\bar{z} - z\| \quad \forall z \in \mathbb{R}^n.$$

Let $\Psi : \mathbb{R}^n \to (-\infty, +\infty]$ be given. The effective domain of Ψ is denoted by $\operatorname{dom}\Psi := \{x \in \mathbb{R}^n : \Psi(x) < \infty\}$ and Ψ is proper if $\operatorname{dom}\Psi \neq \emptyset$. Moreover, a proper function $\Psi : \mathbb{R}^n \to (-\infty, +\infty]$ is μ-strongly convex for some $\mu \geq 0$ if

$$\Psi(\beta z + (1-\beta)\bar{z}) \leq \beta\Psi(z) + (1-\beta)\Psi(\bar{z}) - \frac{\beta(1-\beta)\mu}{2}\|z - \bar{z}\|^2$$

for every $z, \bar{z} \in \operatorname{dom}\Psi$ and $\beta \in [0, 1]$. Let $\partial\Psi(z)$ denote the subdifferential of Ψ at $z \in \operatorname{dom}\Psi$. If Ψ is differentiable at $\bar{z} \in \mathbb{R}^n$, then its affine approximation $\ell_\Psi(\cdot;\bar{z})$ at $\bar{z}$ is defined as $\ell_\Psi(z;\bar{z}) := \Psi(\bar{z}) + \langle\nabla\Psi(\bar{z}), z - \bar{z}\rangle$ for every $z \in \mathbb{R}^n$. The set of all proper lower semi-continuous convex functions $\Psi : \mathbb{R}^n \to (-\infty, +\infty]$ is denoted by $\overline{\operatorname{Conv}}(\mathbb{R}^n)$.

Organization of the paper. Section 2 consists of two subsections. Subsection 2.1 describes the assumptions made on the N-SCO problem and presents the AC-FISTA method for solving it. It also describes the main result of the paper, which establishes a convergence rate bound for AC-FISTA in terms of the average of observed curvatures. Subsection 2.2 discusses a special case of AC-FISTA which is quite efficient in practice, and provides reasons behind its good performance. Section 3 provides the proof of the main result stated in Subsection 2.1. Section 4 presents computational results demonstrating the efficiency of AC-FISTA. Finally, Section 5 provides some concluding remarks.

2 AC-FISTA and the main result

This section consists of two subsections. The first one describes the assumptions made on the N-SCO problem (1) and presents the AC-FISTA method for solving it. It also presents results describing the global convergence rate of AC-FISTA in terms of the iteration count and some parameters associated with AC-FISTA and the problem instance. The second subsection discusses a special case of AC-FISTA and the practical consequences of the above results to this context.

2.1 AC-FISTA and its theoretical guarantees

Throughout this paper, we consider the N-SCO problem (1) and make the following assumptions on it:

(A1) $h \in \overline{\text{Conv}}\,(\mathbb{R}^n)$;

(A2) there exist scalar $L \geq 0$ and a compact convex set $\Omega \supset \mathcal{H} := \text{dom}\, h$ such that f is nonconvex and differentiable on Ω, and

$$\|\nabla f(u) - \nabla f(u')\| \leq L\|u - u'\| \qquad \forall u, u' \in \Omega. \tag{6}$$

We now make some remarks about the above assumptions. First, it follows from (A1) and (A2) that the set of optimal solutions X_* is nonempty and compact. Second, if L satisfies (6) then the pair $(M, m) = (L, L)$ satisfies

$$-\frac{m}{2}\|u - u'\|^2 \leq f(u) - \ell_f(u; u') \leq \frac{M}{2}\|u - u'\|^2 \quad \forall u, u' \in \Omega. \tag{7}$$

Throughout this paper, $\bar{L}$ denotes the smallest L satisfying (6), and $\bar{m}$ (resp., $\bar{M}$) denotes the smallest m (resp., M) satisfying the first (resp., second) inequality in (7). Clearly, in view of (A2) and the second remark above, we have $0 < \bar{m} \leq \bar{L}$ and $0 \leq \bar{M} \leq \bar{L}$.

A necessary condition for y to be a local minimum of (1) is that y is a stationary point of (1), i.e., $0 \in \nabla f(y) + \partial h(y)$. The goal of AC-FISTA described below is to find an approximate stationary point defined as follows.

Definition 2.1. *Given a tolerance $\hat{\rho} > 0$, a pair $(\hat{y}, \hat{v}) \in \mathbb{R}^n \times \mathbb{R}^n$ is called a $\hat{\rho}$-approximate stationary point of* (1) *if it satisfies $\hat{v} \in \nabla f(\hat{y}) + \partial h(\hat{y})$ and $\|\hat{v}\| \leq \hat{\rho}$.*

We are now ready to state AC-FISTA.

AC-FISTA

---

0. Let parameters $\alpha, \gamma \in (0, 1]$, scalar M such that $0.9M \geq \bar{M}$, tolerance $\hat{\rho} > 0$, and initial point $y_0 \in \mathcal{H}$ be given, and set $A_0 = 0$, $x_0 = y_0$, $M_0 = \gamma M$ and $k = 0$;

1. compute
$$a_k = \frac{1+\sqrt{1+4M_kA_k}}{2M_k}, \quad A_{k+1} = A_k + a_k, \quad \tilde{x}_k = \frac{A_k y_k + a_k x_k}{A_{k+1}}; \tag{8}$$

2. compute
$$y^g_{k+1} = y(\tilde{x}_k; M_k), \quad C_k = \mathcal{C}(y^g_{k+1}; \tilde{x}_k), \tag{9}$$
$$v_{k+1} = M_k(\tilde{x}_k - y^g_{k+1}) + \nabla f(y^g_{k+1}) - \nabla f(\tilde{x}_k) \tag{10}$$

   where $y(\cdot;\cdot)$ and $\mathcal{C}(\cdot;\cdot)$ are as in (2) and (4), respectively; if $\|v_{k+1}\| \le \hat{\rho}$ then output $(\hat{y}, \hat{v}) = (y^g_{k+1}, v_{k+1})$ and **stop**;

3. if $C_k \le 0.9M_k$, then compute
$$x^g_{k+1} = P_\Omega\left(\frac{A_{k+1}}{a_k} y^g_{k+1} - \frac{A_k}{a_k} y_k\right), \tag{11}$$

   and set $x_{k+1} = x^g_{k+1}$ and $\tilde{y}_{k+1} = y^g_{k+1}$; otherwise, compute
$$x^b_{k+1} = \underset{u\in\mathbb{R}^n}{\operatorname{argmin}} \left\{ a_k[\ell_f(u;\tilde{x}_k) + h(u)] + \frac{1}{2}\|u - x_k\|^2 \right\}, \tag{12}$$
$$y^b_{k+1} = \frac{A_k y_k + a_k x^b_{k+1}}{A_{k+1}}, \tag{13}$$

   and set $x_{k+1} = x^b_{k+1}$ and $\tilde{y}_{k+1} = y^b_{k+1}$;

4. choose $y_{k+1} \in \mathcal{H}$ such that $\phi(y_{k+1}) \le \phi(\tilde{y}_{k+1})$, compute
$$M_{k+1} = \max\left\{ \gamma M \, , \, \frac{\sum_{j=0}^k C_j}{\alpha(k+1)} \right\}, \tag{14}$$

   set $k \leftarrow k+1$, and go to step 1.

The k-th iteration of AC-FISTA is called good (resp., bad) if the inequality at the beginning of step 3, which is identical to (3) with $\tau = 0.9$, is satisfied (resp., violated). Moreover, for the sake of future reference, we define the index sets for the good and bad iterations as
$$\mathcal{G} := \{k \ge 0 : C_k \le 0.9M_k\}, \quad \mathcal{B} := \{k \ge 0 : C_k > 0.9M_k\}, \tag{15}$$
respectively. For ACG methods that satisfy (3) for every $k \ge 0$, it is well-known that the smaller the sequence $\{M_k\}$ is, the faster their practical performance is. One of the goals of this paper is to show that this observation also holds for AC-FISTA, even though it does not satisfy (3) at every $k \ge 0$. Hence, from the AC-FISTA point of view, it is desirable to choose α large, say $\alpha = 0.5$, and γ small, say $\gamma = 10^{-6}$, since this forces M_k in (14) to be small. It turns out that this is the AC-FISTA implemented in our benchmark of Section 4 and we refer to it as the $(0.5, 10^{-6})$-AC-FISTA.

The following paragraphs give some relevant comments about AC-FISTA.

It is shown below in Theorem 2.2(a) that the pair (y^g_{k+1}, v_{k+1}) satisfies the inclusion in Definition 2.1 for every $k \ge 0$. Hence, if the termination criterion $\|v_{k+1}\| \le \hat{\rho}$ in step 2 is satisfied, then AC-FISTA terminates with a $\hat{\rho}$-approximate stationary point of (1). The first two identities in (8) imply that
$$A_{k+1} = M_k a_k^2. \tag{16}$$

It follows from step 0 of AC-FISTA, (14), (4), (7) with $(m, M) = (\bar{m}, \bar{M})$, and the definition of C_k in (9), that

$$M_k \geq \gamma M, \quad C_k \in [-\bar{m}, \bar{M}], \quad \forall k \geq 0. \tag{17}$$

Two popular rules for choosing y_{k+1} in step 4 of AC-FISTA are: i) $y_{k+1} = \tilde{y}_{k+1}$ for all $k \geq 0$; and ii) y_{k+1} such that $\phi(y_{k+1}) = \min\{\phi(y_k), \phi(\tilde{y}_{k+1})\}$ for all $k \geq 0$. If rule (i) is chosen, then an iteration of AC-FISTA works as follows. Given a pair (x_k, y_k), the k-th iteration sets (x_{k+1}, y_{k+1}) as being the pair (x^g_{k+1}, y^g_{k+1}) obtained in (11) and (9) if $k \in \mathcal{G}$, or the pair (x^b_{k+1}, y^b_{k+1}) obtained in (12) and (13) if $k \in \mathcal{B}$. The condition on y_{k+1} in step 4 simply relaxes rule (i), and allows AC-FISTA to include as special case the monotone (i.e., satisfying $\phi(y_{k+1}) \leq \phi(y_k)$ for all k) variant in which, in place of (i), rule (ii) is used instead.

Following the same notation of this paper, Subsection 3.1 of [13] reviews three rules for performing an ACG iteration which have roots in works dealing with the convex version of SCO (1). They are referred there to as FISTA rule, AT rule, and LLM rule. Under the assumption that $\Omega = \mathbb{R}^n$, the two AC-FISTA iterations (i.e., good and bad) can be interpreted in terms of the three rules above as follows: a good (resp., bad) iteration of AC-FISTA performs an ACG iteration based on the FISTA (resp., AT) rule. Since the test to decide the type of iteration (i.e., good or bad) to perform depends on y^g_{k+1}, this point needs to be computed prior to a bad iteration (even though the iteration itself does not use it).

We now comment on the computational effort of an AC-FISTA iteration. A good iteration computes only one resolvent evaluation of ∂h, while a bad one computes two composite resolvent evaluations (one in (9) to compute y^g_{k+1} and another in (12) to compute x^b_{k+1}). Since Ω is usually chosen so that the projection onto Ω in (11) is considerably cheaper than a composite resolvent evaluation and the majority of iterations performed by AC-FISTA is assumed to be good ones (see Condition A below), it follows that the average number of resolvent evaluations per iteration is close to 1.

We now state the main result of the paper which describes how fast one of the iterates $y^g_1, \ldots, y^g_k$ approaches the stationary condition $0 \in \nabla f(y) + \partial h(y)$. Its main conclusion assumes the following condition.

Condition A: There exist $k_0 \in \mathbb{N}_+$ such that $|\mathcal{B}_k| \leq k/3$ for every $k \geq k_0$ where

$$\mathcal{B}_k := \{i \in \mathcal{B} : i \leq k-1\} \qquad \forall k \geq 1. \tag{18}$$

It is worth noting that the factor 1/3 in Condition A is not particularly important for our analysis to hold. Even though this factor can be replaced by any scalar less than one, we have chosen a specific value for it in order to keep the number of constants used in our analysis small.

We now discuss some choices of (α, γ) which guarantee that Condition A holds. First, step 0 of AC-FISTA and (17) imply that $C_k \leq \bar{M} \leq 0.9M \leq 0.9M_k/\gamma$ for every $k \geq 0$. Thus, if $\gamma = 1$ (and $\alpha \in (0, 1]$ is arbitrary) then every iteration of AC-FISTA is good, and Condition A trivially holds with $k_0 = 0$. Second, it is shown in Lemma 4.5 of [13] that Condition A holds with $k_0 = 12$ whenever $\alpha, \gamma \in (0, 1]$ are chosen so that

$$\alpha \leq \frac{0.9}{8}\left(1 + \frac{1}{0.9\gamma}\right)^{-1}. \tag{19}$$

Rule (19) for choosing (α, γ) results in $\alpha = \mathcal{O}(\gamma)$ so that a small choice of γ implies that α is also small. Hence, it excludes the practical choice $(\alpha, \gamma) = (0.5, 10^{-6})$ mentioned in the paragraph following AC-FISTA. However, since the proof of Lemma 4.5 of [13] is based on quite conservative bounds, Subsection 2.2 below reexamines its proof and gives strong evidence (validated by our computational results) that Condition A holds for our practical choice of $(\alpha, \gamma) = (0.5, 10^{-6})$.

Instead of specifying values for (α, γ), the two main results below simply assume that Condition A holds, and establishes a global convergence rate and iteration-complexity for AC-FISTA.

Theorem 2.2. *Define the harmonic mean of the sequence $\{M_i\}$ and the average of the curvature sequence $\{\mathcal{L}(y^g_{i+1}; \tilde{x}_i)\}$ defined in (5) as*

$$M^{hm}_k := \frac{k}{\sum_{i=0}^{k-1} \frac{1}{M_i}}, \quad L^{avg}_k := \frac{1}{k} \sum_{i=0}^{k-1} \mathcal{L}(y^g_{i+1}; \tilde{x}_i), \tag{20}$$

respectively, and let

$$\theta_k := \frac{M_k}{M^{hm}_k}, \quad \tau_k := \frac{L^{avg}_k}{M_k}. \tag{21}$$

Then, the following statements hold:

(a) for every $k \geq 1$, we have $v_k \in \nabla f(y^g_k) + \partial h(y^g_k)$;

(b) if Condition A holds, then for every $k \geq \max\{12, k_0\}$,

$$\min_{1 \leq i \leq k} \|v_i\| = \mathcal{O}\left((1 + \tau_k) \left[\frac{M_k d_0}{k^{3/2}} + \left(\sqrt{\bar{M}} + \sqrt{\bar{m}} \right) \frac{\sqrt{M_k \theta_k} D_\Omega}{k} + \frac{\sqrt{\bar{m} M_k \theta_k} D_\mathcal{H}}{\sqrt{k}} \right] \right) \tag{22}$$

where d_0 denotes the distance of the initial point x_0 to the set of optimal solutions of (1), and D_Ω and $D_\mathcal{H}$ denote the diameters of Ω and $\mathcal{H}$, respectively;

(c) for every $k \geq 1$, we have

$$M_k = \mathcal{O}\left(\frac{M}{\alpha} \right), \quad \frac{k-1}{2k} \leq \theta_k \leq \frac{M_k}{\gamma M}, \quad \tau_k \leq \frac{\bar{L}}{\gamma M}; \tag{23}$$

as a consequence, $\theta_k = \mathcal{O}(1/(\alpha\gamma))$.

The corollary below describes the worst-case behavior of AC-FISTA under the assumption that Condition A holds.

Corollary 2.3. *If Condition A holds, then for every $k \geq \max\{12, k_0\}$,*

$$\min_{1 \leq i \leq k} \|v_i\| = \mathcal{O}\left(\left(1 + \frac{\bar{L}}{\gamma M} \right) \left[\frac{M d_0}{\alpha k^{3/2}} + \left(\sqrt{\bar{M}} + \sqrt{\bar{m}} \right) \frac{\sqrt{M} D_\Omega}{\alpha \sqrt{\gamma} k} + \frac{\sqrt{\bar{m} M} D_\mathcal{H}}{\alpha \sqrt{\gamma} \sqrt{k}} \right] \right). \tag{24}$$

As a consequence, the iteration-complexity to find a $\hat{\rho}$-approximate stationary point of (1) is

$$\mathcal{O}\left(\left(1 + \frac{\bar{L}}{\gamma M} \right)^{2/3} \left(\frac{M d_0}{\alpha \hat{\rho}} \right)^{2/3} + \left(1 + \frac{\bar{L}}{\gamma M} \right) \frac{(\sqrt{\bar{M}} + \sqrt{\bar{m}}) \sqrt{M} D_\Omega}{\alpha \sqrt{\gamma} \hat{\rho}} + \left(1 + \frac{\bar{L}}{\gamma M} \right)^2 \frac{\bar{m} M D^2_\mathcal{H}}{\alpha^2 \gamma \hat{\rho}^2} \right).$$

Proof: The first conclusion immediately follows from Theorem 2.2(b) and (c). The second one follows from (24) and Definition 2.1. ■

We finally make a comment about the dependence of the convergence rate bound (24) in terms of α and γ only. If M is chosen so as to satisfy $M \geq \bar{L}/0.9$, then the bottleneck term in the worst-case convergence rate bound (24) becomes $\mathcal{O}(\sqrt{\bar{m} M} D_\mathcal{H}/(\alpha \gamma^{3/2} \sqrt{k}))$, which is equal to $1/(\alpha \gamma^{3/2})$ times the convergence rate bounds of some other ACG variants derived in the literature (see e.g., [3, 14]). Hence, the above two convergence rate bounds are similar whenever α and γ are both close to one.

2.2 A practical AC-FISTA

This subsection discusses in more details the AC-FISTA with $(\alpha, \gamma) = (0.5, 10^{-6})$, i.e., the $(0.5, 10^{-6})$-AC-FISTA as defined in the paragraph immediately after AC-FISTA.

Although the dependence of the dominant term in (24) with respect to α and γ, i.e., $\mathcal{O}(1/(\alpha\gamma^{3/2}))$, is large for $(0.5, 10^{-6})$-AC-FISTA, it should be noted that this dependence factor was obtained using the conservative estimates in (23). In practice, the quantity θ_k, although sometimes initially large, quickly approaches one and stays close to one thereafter, and τ_k never exceeds 4 and, in many cases, is within the interval $[0, 3]$ (see Tables 3, 7, 8, 12 and 13). We can then conclude that, in terms of γ and α, both θ_k and τ_k are $\mathcal{O}(1)$, instead of $\mathcal{O}(1/(\alpha\gamma))$ and $\mathcal{O}(1/\gamma)$ as in Theorem 2.2(c). Second, under the (often observed) condition that θ_k and τ_k are $\mathcal{O}(1)$, the convergence rate bound reduces to

$$\min_{1\le i\le k} \|v_i\| = \mathcal{O}\left(\frac{M_k d_0}{k^{3/2}} + \left(\sqrt{\bar{M}} + \sqrt{\bar{m}}\right)\frac{\sqrt{M_k} D_\Omega}{k} + \frac{\sqrt{\bar{m} M_k} D_{\mathcal{H}}}{\sqrt{k}}\right),$$

and hence does not depend on $\gamma^{-1} = 10^6$. Third, the latter bound clearly shows that the practical behavior of this $(0.5, 10^{-6})$-AC-FISTA improves as the ratio M_k/M becomes small, which is what has been observed in our computational experiments.

We will now argue that in practice the $(0.5, 10^{-6})$-AC-FISTA is likely to satisfy Condition A. Indeed, letting

$$\eta_k := \frac{\sum_{i\in\mathcal{B}_k} C_i}{\sum_{i\in\mathcal{B}_k,\, i\le|\mathcal{B}_k|/2} C_i}$$

where $\mathcal{B}_k$ is as in (18), and examining the proof of Lemma 4.5 of [13], it can be easily seen that

$$k\alpha\eta_k \ge \frac{0.9|\mathcal{B}_k|}{2}.$$

Lemma 4.5 of [13] then uses (14), and the facts that $M \ge \bar{M} \ge C_i$ and $C_i > 0.9M_i$ for $i \in \mathcal{B}_k$, to conclude that η_k is bounded by $1 + 1/(0.9\gamma)$. However, such bound on η_k is quite conservative and it is observed computationally that η_k quickly approaches two and remains close to two thereafter. Hence, it follows from the latter observation and the above inequality that any choice of α in $(0, 1/8]$ makes Condition A very likely to hold in practice. Although our choice of $\alpha = 0.5$ does not lie in this (still conservative) range, we have observed that it works quite well in our computational experiments.

2.3 Comparison with the AC-ACG method of [13]

We start by giving an overview of the AC-ACG method of [13] using the description of AC-FISTA in Subsection 2.1. More specifically, AC-ACG is similar to the variant of AC-FISTA where $y_{k+1} = \tilde{y}_{k+1}$ for every $k \ge 0$ but differs in the following two aspects:

- it sets x_{k+1} to the right-hand side of (12) regardless of whether the iteration is good or bad;
- it computes M_{k+1} as in (14) but with C_j replaced by $\tilde{\mathcal{C}}(y^g_{j+1}; \tilde{x}_j)$ where $\tilde{C}(\cdot;\cdot)$ is as in (5).

Another not-so-crucial difference is that ACG chooses the parameters $\alpha, \gamma \in (0, 1]$ so that (19) holds as equality while AC-FISTA discards the latter condition on α and γ and instead simply makes the weaker assumption that Condition A holds (see the third remark in the paragraph containing (19)).

In terms of the three ACG rules described in Subsection 3.1 of [13], a good (resp., bad) iteration of AC-ACG performs an ACG iteration based on the LLM (resp., AT) rule. Hence, while a good iteration AC-ACG uses the LLM rule, the one for AC-FISTA uses the FISTA rule.

From a computational point of view, while AC-ACG always performs two resolvent evaluations at every iteration, AC-FISTA performs one resolvent evaluation in a good iteration and two resolvent evaluations in a bad one. Since in practice most of the iterations of AC-FISTA are good, its average cost per iteration is relatively lower than that of AC-ACG.

3 Proof of Theorem 2.2

We start by providing a straightforward technical result which is then used to outline our analysis in this section.

Lemma 3.1. *For every $k \geq 1$, we have $v_k \in \nabla f(y_k^g) + \partial h(y_k^g)$.*

Proof: The inclusion follows from the optimality condition of (2), and the definitions of y_{k+1}^g and v_{k+1} in (9) and (10), respectively. ■

The most technical and difficult part of Theorem 2.2 is its statement (b) where a convergence rate bound on $\min_{1\leq i\leq k}\|v_i\|$ is claimed. A rough outline of the proof of this statement is as follows. First, Lemmas 3.2-3.6 are used to prove that

$$\sum_{i\in\mathcal{G}_k}\left(A_{i+1}M_i\|y_{i+1}^g-\tilde{x}_i\|^2\right)=\mathcal{O}\left(d_0^2+\frac{\bar{m}+\bar{M}}{M_k^{hm}}D_\Omega^2 k+\frac{\bar{m}}{M_k^{hm}}D_\mathcal{H}^2k^2\right) \tag{25}$$

where $\mathcal{G}_k = \{0,\dots,k-1\}\setminus\mathcal{B}_k$. Next, using Condition A and some nontrivial technical results, namely, Lemmas 3.7-3.9, it is shown within the proof of Theorem 2.2(b) that

$$\min_{1\leq i\leq k}\|v_i\|=\mathcal{O}\left(\frac{M_k+L_k^{avg}}{k^{3/2}}\left(\sum_{i\in\mathcal{G}_k}A_{i+1}M_i\|y_{i+1}^g-\tilde{x}_i\|^2\right)^{1/2}\right). \tag{26}$$

A direct combination of the above two claims then immediately gives us Theorem 2.2(b). The proof of Theorem 2.2(c) does not require any technical result.

The first lemma below states a few basic properties of AC-FISTA.

Lemma 3.2. *For every $k \geq 0$, we define*

$$\tilde{\gamma}_k(u) := \ell_f(u;\tilde{x}_k)+h(u), \tag{27}$$
$$\gamma_k(u) := \tilde{\gamma}_k(y_{k+1}^g)+M_k\langle\tilde{x}_k-y_{k+1}^g,u-y_{k+1}^g\rangle. \tag{28}$$

Then the following statements hold for every $k \geq 0$:

(a) γ_k minorizes $\tilde{\gamma}_k$, $\tilde{\gamma}_k(y_{k+1}^g)=\gamma_k(y_{k+1}^g)$,

$$\min_u\left\{\tilde{\gamma}_k(u)+\frac{M_k}{2}\|u-\tilde{x}_k\|^2\right\}=\min_u\left\{\gamma_k(u)+\frac{M_k}{2}\|u-\tilde{x}_k\|^2\right\},$$

and these minimization problems have y_{k+1}^g as unique optimal solution;

(b) for every $u\in\mathcal{H}$, $\tilde{\gamma}_k(u)-\phi(u)\leq\bar{m}\|u-\tilde{x}_k\|^2/2$;

(c) $x^g_{k+1} = argmin\left\{a_k\gamma_k(u) + \|u - x_k\|^2/2 : u \in \Omega\right\}$;

(d) $\{x^b_k\}$, $\{y_k\}$, $\{y^g_k\}$ *and* $\{\tilde{y}_k\}$ *are contained in* $\mathcal{H}$, *while* $\{x^g_k\}$, $\{x_k\}$ *and* $\{\tilde{x}_k\}$ *lie in* Ω;

(e) *for every* $u \in \mathcal{H}$, *we have*

$$A_k\|y_k - \tilde{x}_k\|^2 + a_k\|u - \tilde{x}_k\|^2 \le \frac{1}{M_k}D^2_\Omega + a_kD^2_{\mathcal{H}}.$$

Proof: (a) This statement follows from Lemma 2.2(a) of [14] with $(\kappa_0, \lambda, y_{k+1}) = (0, 1/M_k, y^g_{k+1})$.

(b) This statement immediately follows from the first inequality in (7) and the definition of $\tilde{\gamma}_k(u)$ in (27).

(c) It follows from the definitions of $\tilde{x}_k$ and γ_k in (8) and (28), respectively, and relation (16) that the (unique) global minimizer of the function $a_k\gamma_k(u) + \|u - x_k\|^2/2$ over $\mathbb{R}^n$ is

$$x_k + a_kM_k(y^g_{k+1} - \tilde{x}_k) = x_k + \frac{A_{k+1}}{a_k}\left(y^g_{k+1} - \frac{A_ky_k + a_kx_k}{A_{k+1}}\right) = \frac{A_{k+1}}{a_k}y^g_{k+1} - \frac{A_k}{a_k}y_k.$$

This observation and the definition of x^g_{k+1} in (11) then imply that the conclusion of (c) holds.

(d) First, it is by definition that $\{y_k\}$ is contained $\mathcal{H}$. In view of (2) (resp., (12)), it is clear that the sequence $\{y^g_k\}$ (resp., $\{x^b_k\}$) is contained in $\mathcal{H}$. Hence, using the fact that $y_0 \in \mathcal{H}$ (see step 0 of AC-FISTA), (13) and the convexity of $\mathcal{H}$, we easily see that $\{y^b_k\} \subset \mathcal{H}$ and hence $\{\tilde{y}_k\} \subset \mathcal{H}$. It is also easy to see that $\{x^g_k\} \subset \Omega$ from its definition in (11). Hence, it follows from the fact that $\{x^b_k\} \subset \mathcal{H} \subset \Omega$ and step 0 of AC-FISTA that $\{x_k\}$ lie in Ω. Finally, $\{\tilde{x}_k\} \subset \Omega$ follows from the third identity in (8) and the convexity of Ω.

(e) It is easy to see that for every $x, y \in \mathbb{R}^n$ and $a, A \in \mathbb{R}_+$,

$$A\|y\|^2 + a\|x\|^2 = (A + a)\left\|\frac{Ay + ax}{A + a}\right\|^2 + \frac{Aa}{A + a}\|y - x\|^2.$$

Using the above identity with $x = u - \tilde{x}_k$, $y = y_k - \tilde{x}_k$, $a = a_k$ and $A = A_k$, and the second and third identities in (8), we have

$$A_k\|y_k - \tilde{x}_k\|^2 + a_k\|u - \tilde{x}_k\|^2 = \frac{a_k^2}{A_{k+1}}\|u - x_k\|^2 + \frac{A_ka_k}{A_{k+1}}\|u - y_k\|^2.$$

This statement now follows from the above inequality, statement (d), the definitions of D_Ω and $D_{\mathcal{H}}$, and relation (16). ∎

The next result introduces a crucial potential function and provides an important recursive formula based on it.

Lemma 3.3. *For every* $u \in \mathcal{H}$ *and* $k \ge 0$, *we have*

$$\frac{M_k - F_k}{2}A_{k+1}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2 \le \eta_k(u) - \eta_{k+1}(u) + \frac{\bar{m}}{2}\left(\frac{1}{M_k}D^2_\Omega + a_kD^2_{\mathcal{H}}\right) \tag{29}$$

where M_k *is as in* (14), $F_k := \mathcal{C}(\tilde{y}_{k+1}; \tilde{x}_k)$ *and*

$$\eta_k(u) := A_k[\phi(y_k) - \phi(u)] + \frac{1}{2}\|u - x_k\|^2. \tag{30}$$

Proof: We first note that in order to prove the lemma, it suffices to show

$$\frac{M_k - F_k}{2} A_{k+1}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2 - \eta_k(u) + \eta_{k+1}(u) \le A_k(\tilde{\gamma}_k(y_k) - \phi(y_k)) + a_k(\tilde{\gamma}_k(u) - \phi(u)). \quad (31)$$

Indeed, it follows from the above inequality and Lemma 3.2(b) that

$$\frac{M_k - F_k}{2} A_{k+1}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2 - \eta_k(u) + \eta_{k+1}(u) \le \frac{\bar{m}}{2}\left(A_k\|y_k - \tilde{x}_k\|^2 + a_k\|u - \tilde{x}_k\|^2\right),$$

which, together with Lemma 3.2(e), then immediately implies (29).

We now prove (31) holds for $k \in \mathcal{G}$. Let $k \in \mathcal{G}$ and $u \in \mathcal{H}$ be given. Noting that $x_{k+1} = x^g_{k+1}$, and using Lemma 3.2(c), relations (8) and (16), and the fact that $a_k\gamma_k + \|\cdot - x_k\|^2/2$ is 1-strongly convex, we conclude that

$$\begin{aligned} A_k\gamma_k(y_k) + a_k\gamma_k(u) + \frac{1}{2}\|u - x_k\|^2 - \frac{1}{2}\|u - x_{k+1}\|^2 &\ge A_k\gamma_k(y_k) + a_k\gamma_k(x_{k+1}) + \frac{1}{2}\|x_{k+1} - x_k\|^2 \\ &\ge A_{k+1}\gamma_k(\hat{y}_{k+1}) + \frac{1}{2}\frac{A_{k+1}^2}{a_k^2}\|\hat{y}_{k+1} - \tilde{x}_k\|^2 = A_{k+1}\left[\gamma_k(\hat{y}_{k+1}) + \frac{M_k}{2}\|\hat{y}_{k+1} - \tilde{x}_k\|^2\right] \end{aligned}$$

where $\hat{y}_{k+1} = (A_k y_k + a_k x_{k+1})/A_{k+1}$. It follows from Lemma 3.2(a) and the fact that $\tilde{y}_{k+1} = y^g_{k+1}$ for every $k \in \mathcal{G}$ that

$$\begin{aligned} \gamma_k(\hat{y}_{k+1}) + \frac{M_k}{2}\|\hat{y}_{k+1} - \tilde{x}_k\|^2 &\ge \gamma_k(\tilde{y}_{k+1}) + \frac{M_k}{2}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2 \\ &= \tilde{\gamma}_k(\tilde{y}_{k+1}) + \frac{M_k}{2}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2 = \phi(\tilde{y}_{k+1}) + \frac{M_k - F_k}{2}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2 \\ &\ge \phi(y_{k+1}) + \frac{M_k - F_k}{2}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2. \end{aligned}$$

where the last identity is due to the definitions of F_k and $\tilde{\gamma}_k$ in (27), and the last inequality is due to the fact that $\phi(y_{k+1}) \le \phi(\tilde{y}_{k+1})$. Using the above two inequalities and the definition of η_k in (30), we have

$$\frac{M_k - F_k}{2} A_{k+1}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2 - \eta_k(u) + \eta_{k+1}(u) \le A_k[\gamma_k(y_k) - \phi(y_k)] + a_k[\gamma_k(u) - \phi(u)],$$

which together with the fact that $\gamma_k \le \tilde{\gamma}_k$ (see Lemma 3.2(a)) implies that (31) holds.

We finally prove (31) holds for $k \in \mathcal{B}$. Let $k \in \mathcal{B}$ and $u \in \mathcal{H}$ be given. Noting that $x_{k+1} = x^b_{k+1}$ and $\tilde{y}_{k+1} = y^b_{k+1}$, and using the definitions of $\tilde{\gamma}_k$, x^b_{k+1}, $\tilde{y}_{k+1}$ and $\tilde{x}_k$ in (27), (12), (13) and (8), respectively, the fact that $a_k\tilde{\gamma}_k + \|\cdot - x_k\|^2/2$ is 1-strongly convex, and relation (16), we conclude that

$$\begin{aligned} A_k\tilde{\gamma}_k(y_k) + a_k\tilde{\gamma}_k(u) + \frac{1}{2}\|u - x_k\|^2 - \frac{1}{2}\|u - x_{k+1}\|^2 &\ge A_k\tilde{\gamma}_k(y_k) + a_k\tilde{\gamma}_k(x^b_{k+1}) + \frac{1}{2}\|x^b_{k+1} - x_k\|^2 \\ &\ge A_{k+1}\tilde{\gamma}_k(y^b_{k+1}) + \frac{1}{2}\frac{A_{k+1}^2}{a_k^2}\|y^b_{k+1} - \tilde{x}_k\|^2 = A_{k+1}\left[\tilde{\gamma}_k(\tilde{y}_{k+1}) + \frac{M_k}{2}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2\right] \\ &= A_{k+1}\left[\phi(\tilde{y}_{k+1}) + \frac{M_k - F_k}{2}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2\right] \ge A_{k+1}\left[\phi(y_{k+1}) + \frac{M_k - F_k}{2}\|\tilde{y}_{k+1} - \tilde{x}_k\|^2\right] \end{aligned}$$

where the last identity is due to the definitions of F_k and $\tilde{\gamma}_k$ in (27), and the last inequality is due to the fact that $\phi(y_{k+1}) \le \phi(\tilde{y}_{k+1})$. Using the above inequality and the definition of η_k in (30), and rearranging the terms, we obtain (31). ∎

The following result discusses the consequences of Lemma 3.3 when k is a good iteration and also when k is a bad one.

Lemma 3.4. *The following statements hold for every $u \in \mathcal{H}$:*

(a) if $k \in \mathcal{G}$ then

$$\frac{1}{20} A_{k+1} M_k \|\tilde{y}_{k+1} - \tilde{x}_k\|^2 \leq \eta_k(u) - \eta_{k+1}(u) + \frac{\bar{m}}{2}\left(\frac{1}{M_k} D_\Omega^2 + a_k D_\mathcal{H}^2\right); \tag{32}$$

(b) if $k \in \mathcal{B}$ then

$$0 \leq \eta_k(u) - \eta_{k+1}(u) + \frac{\bar{m}}{2}\left(\frac{1}{M_k} D_\Omega^2 + a_k D_\mathcal{H}^2\right) + \frac{\bar{M}}{2M_k} D_\Omega^2. \tag{33}$$

Proof: (a) Let $k \in \mathcal{G}$ be given. It is easy to see that $F_k \leq 0.9 M_k$ due to the fact that $F_k = C_k$ when $k \in \mathcal{G}$ and (15), and hence (32) immediately follows from this observation and (29).

(b) Let $k \in \mathcal{B}$ be given. Noting that $x_{k+1} = x_{k+1}^b$ and $\tilde{y}_{k+1} = y_{k+1}^b$, and using relations (29) and (16), and the definitions of y_{k+1}^b and $\tilde{x}_k$ in (13) and (8), respectively, we conclude that

$$\begin{aligned} \eta_k(u) - \eta_{k+1}(u) + \frac{\bar{m}}{2}\left(\frac{1}{M_k} D_\Omega^2 + a_k D_\mathcal{H}^2\right) &\geq \frac{M_k - F_k}{2} A_{k+1} \|\tilde{y}_{k+1} - \tilde{x}_k\|^2 \\ = \frac{M_k - F_k}{2} A_{k+1} \left\| \frac{A_k y_k + a_k x_{k+1}}{A_{k+1}} - \frac{A_k y_k + a_k x_k}{A_{k+1}} \right\|^2 &= \frac{(M_k - F_k) a_k^2}{2A_{k+1}} \|x_{k+1} - x_k\|^2 \\ = \frac{1}{2}\left(1 - \frac{F_k}{M_k}\right) \|x_{k+1} - x_k\|^2 &\geq \frac{1}{2}\left(1 - \frac{\bar{M}}{M_k}\right) \|x_{k+1} - x_k\|^2 \end{aligned}$$

where the last inequality is due to the fact that $F_k \leq \bar{M}$, and hence that (33) holds in view of Lemma 3.2(d). ■

We now state a technical result which will be used to derive a consequence of Lemma 3.4.

Lemma 3.5. *The sequences $\{A_k\}$ and $\{M_k\}$ generated by AC-FISTA satisfy the following statements:*

a) for every $k \geq 1$, we have $A_k M_k^{hm} \leq k^2$;

b) for every $k \geq 4$, we have $A_k M_k \geq k^2/12$;

c) for every $i \in \{1, \ldots, k\}$, we have $i M_i \leq k M_k$.

Proof: a) This statement follow from Lemma 4.7 of [13] and the definition of M_k^{hm} in (20).

b) This statement can be proved by following an argument similar to the proof of (37) of [13]. Note that the only difference is for every $k \geq 4$, we have

$$\sum_{i=1}^{k-1} \sqrt{i} \geq \int_0^{k-1} \sqrt{x} dx = \frac{2}{3}(k-1)^{3/2} \geq \frac{2}{3}\left(\frac{3}{4}k\right)^{3/2} \geq \frac{1}{3} k^{3/2}.$$

c) This is Lemma 4.6 of [13]. ■

The following result follows by combining the conclusions (a) and (b) of Lemma 3.4, and using Lemma 3.5(a).

Lemma 3.6. *For every $u \in \mathcal{H}$ and $k \geq 1$, we have*

$$\sum_{i \in \mathcal{G}_k} \left(A_{i+1} M_i \|\tilde{y}_{i+1} - \tilde{x}_i\|^2\right) \leq 10\left(d_0^2 + \frac{\bar{m} + \bar{M}}{M_k^{hm}} D_\Omega^2 k + \frac{\bar{m}}{M_k^{hm}} D_\mathcal{H}^2 k^2\right). \tag{34}$$

Proof: Let $x_* \in X_*$ be such that $x_* = \operatorname{argmin}\{\|x_0 - u\| : u \in X_*\}$ be given and denote $\|x_0 - x_*\|$ by d_0. In view of the definition of η_k in (30), we observe that $\eta_k(x_*) \geq 0$ for every $k \geq 0$ and $\eta_0(x_*) = d_0^2/2$. Adding (32) and (33) with $u = x_*$ as k varies in $\mathcal{G}_k \cup \mathcal{B}_k$, and using the previous observation and the definition of η_k in (30), we have that for $k \geq 1$,

$$\sum_{i \in \mathcal{G}_k} \left(A_{i+1} M_i \|\tilde{y}_{i+1} - \tilde{x}_i\|^2\right) \leq 10 \left(d_0^2 + (\bar{m} + \bar{M}) D_\Omega^2 \sum_{i=0}^{k-1} \frac{1}{M_i} + \bar{m} D_{\mathcal{H}}^2 A_k\right).$$

Inequality (34) now follows from the above conclusion, the definition of M_k^{hm} in (20), and the second inequality in Lemma 3.5(a). ■

The two following technical results require Condition A to hold. Recall that sufficient conditions for Condition A to hold have been discussed in the paragraph containing (19). Moreover, Subsection 2.2 discusses the likelihood that Condition A holds in the practical setting of AC-FISTA.

Recall from the discussion on the line above (19) that Condition A always holds with $k_0 = 12$ if α is chosen so as to satisfy (19). However, our analysis may also hold for α's that do not satisfy the restrictive condition (19) as long as the resulting sequence $\{|\mathcal{B}_k|\}$ satisfy Condition A (e.g., see the last paragraph in Subsection 2.2 which argues that this condition practically holds for the $(0.5, 10^{-6})$-AC-FISTA).

Lemma 3.7. *Assume that Condition A holds and, for every $k \geq 1$, define*

$$\bar{\mathcal{G}}_k := \{i \in \mathcal{G}_k : i \geq \lceil k/3 \rceil\}.$$

Then, $|\bar{\mathcal{G}}_k| \geq k/4$ for every $k \geq \max\{12, k_0\}$.

Proof: Using the definitions of $\mathcal{B}_k$ in (18) and $\bar{\mathcal{G}}_k$ above, we have

$$\bar{\mathcal{G}}_k \cup \mathcal{B}_k \supset \left\{\left\lceil \frac{k}{3} \right\rceil, \ldots, k-1\right\},$$

and hence that

$$|\bar{\mathcal{G}}_k| + |\mathcal{B}_k| = |\bar{\mathcal{G}}_k \cup \mathcal{B}_k| \geq k - \left\lceil \frac{k}{3} \right\rceil \geq \frac{2k}{3} - 1.$$

This observation and Condition A then imply that for every $k \geq \max\{12, k_0\}$,

$$|\bar{\mathcal{G}}_k| \geq \frac{2k}{3} - 1 - |\mathcal{B}_k| \geq \frac{k}{3} - 1 \geq \frac{k}{4},$$

where the last inequality is due to the fact that $k \geq 12$. ■

We now present an important inequality of our analysis that connects other key ingredients, i.e., Lemmas 3.6, 3.7, and 3.9 below, for proving Theorem 2.2.

Lemma 3.8. *Under Condition A, we have for every $k \geq \max\{12, k_0\}$,*

$$\min_{1 \leq i \leq k} \|v_i\| \leq \frac{8}{k^{3/2}} \left(\sum_{i=\lceil k/3 \rceil}^{k-1} \frac{M_i + L_i}{\sqrt{A_{i+1} M_i}}\right) \left(\sum_{i \in \mathcal{G}_k} A_{i+1} M_i \|\tilde{y}_{i+1} - \tilde{x}_i\|^2\right)^{1/2} \tag{35}$$

where $L_i := \mathcal{L}(y_{i+1}^g; \tilde{x}_k)$ and $\mathcal{L}(\cdot;\cdot)$ is as in (5).

Proof: It follows from the definitions of L_k and v_{k+1} in this lemma and (10), respectively, and the triangle inequality that

$$\|v_{k+1}\| \le (M_k + L_k)\|y^g_{k+1} - \tilde{x}_k\|.$$

Using the above inequality, and the facts that $\tilde{y}_{i+1} = y^g_{i+1}$ for $i \in \mathcal{G}_k$ and $\bar{\mathcal{G}}_k \subset \mathcal{G}_k$, we have

$$\begin{aligned} \min_{1\le i\le k} \|v_i\| \le \min_{i\in\bar{\mathcal{G}}_k} \|v_i\| &\le \min_{i\in\bar{\mathcal{G}}_k} \left(\frac{M_i + L_i}{\sqrt{A_{i+1}M_i}}\right)\left(\sqrt{A_{i+1}M_i}\|\tilde{y}_{i+1} - \tilde{x}_i\|\right) \\ &\le |\bar{\mathcal{G}}_k|^{-3/2}\left(\sum_{i\in\bar{\mathcal{G}}_k} \frac{M_i + L_i}{\sqrt{A_{i+1}M_i}}\right)\left(\sum_{i\in\bar{\mathcal{G}}_k} A_{i+1}M_i\|\tilde{y}_{i+1} - \tilde{x}_i\|^2\right)^{1/2} \end{aligned} \tag{36}$$

where the last inequality is due to Lemma 9 of [8] with $k = |\bar{\mathcal{G}}_k|$, $p = 3/2$, and

$$a_i = \frac{M_i + L_i}{\sqrt{A_{i+1}M_i}}, \quad b_i = \sqrt{A_{i+1}M_i}\|\tilde{y}_{i+1} - \tilde{x}_i\|.$$

The conclusion of the proposition now follows from (36), the facts that $\bar{\mathcal{G}}_k \subset \{\lceil k/3\rceil, \ldots, k-1\}$ and $\bar{\mathcal{G}}_k \subset \mathcal{G}_k$, and Lemma 3.7. ∎

In view of Lemmas 3.6 and 3.8, it is sufficient to develop a bound on the first summation in (35) to obtain a bound on $\min_{1\le i\le k} \|v_i\|$. Hence, we present the following lemma.

Lemma 3.9. *For every $k \ge 12$, we have*

$$\sum_{i=\lceil k/3\rceil}^{k-1} \frac{M_i + L_i}{\sqrt{A_{i+1}M_i}} \le 6\sqrt{3}\left(2M_k + L^{avg}_k\right) \tag{37}$$

where L_i is defined in Lemma 3.8 and L^{avg}_k is as in (20).

Proof: In view of the assumption that $k \ge 12$, it is easy to see that $i \ge 4$ for $i \ge \lceil k/3\rceil$. This observation, the fact that $A_{i+1} \ge A_i$ and Lemma 3.5(b) imply that for every $k \ge 12$,

$$\sum_{i=\lceil k/3\rceil}^{k-1} \frac{M_i + L_i}{\sqrt{A_{i+1}M_i}} \le \sum_{i=\lceil k/3\rceil}^{k-1} \frac{M_i + L_i}{\sqrt{A_iM_i}} \le 2\sqrt{3}\sum_{i=\lceil k/3\rceil}^{k-1} \frac{M_i + L_i}{i}.$$

Using Lemma 3.5(c), we have

$$\sum_{i=\lceil k/3\rceil}^{k-1} \frac{M_i}{i} \le \sum_{i=\lceil k/3\rceil}^{k-1} \frac{kM_k}{i^2} \le kM_k\frac{2k/3}{(k/3)^2} = 6M_k.$$

It is easy to see from the definition of L^{avg}_k in (20) that

$$\sum_{i=\lceil k/3\rceil}^{k-1} \frac{L_i}{i} \le \frac{3}{k}\sum_{i=\lceil k/3\rceil}^{k-1} L_i \le \frac{3}{k}\sum_{i=0}^{k-1} L_i = 3L^{avg}_k.$$

Inequality (37) immediately follows from the above three inequalities. ∎

We are now ready to prove Theorem 2.2.

Proof of Theorem 2.2: (a) See Lemma 3.1.

(b) Putting together Lemmas 3.6, 3.8, and 3.9, and using the inequality $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ for every $a, b \ge 0$, we have

$$\min_{1\le i\le k} \|v_i\| \le \frac{48\sqrt{30}}{k^{3/2}} \left(2M_k + L_k^{avg}\right) \left(d_0 + \frac{\sqrt{\bar{m}} + \sqrt{\bar{M}}}{\sqrt{M_k^{hm}}} D_\Omega \sqrt{k} + \frac{\sqrt{\bar{m}}}{\sqrt{M_k^{hm}}} D_{\mathcal{H}} k \right).$$

Statement b) now follows from the above inequality and the definitions of θ_k and τ_k in (21).

(c) Using the definition of M_{k+1} in (14), and the facts that $\gamma < 1$ and $C_k \le \bar{M} \le M$ for every $k \ge 0$, we have for every $k \ge 0$,

$$M_{k+1} = \mathcal{O}\left(\gamma M + \frac{\bar{M}}{\alpha}\right) = \mathcal{O}\left(\frac{M}{\alpha}\right),$$

and hence the inequality on M_k in (23) holds. Using the definition of M_k^{hm} in (20) and Lemma 3.5(c), we have

$$M_k^{hm} = \frac{k}{\sum_{i=0}^{k-1} \frac{1}{M_i}} \le \frac{k}{\sum_{i=0}^{k-1} \frac{i}{kM_k}} = \frac{2k}{k-1} M_k,$$

and hence the first inequality on θ_k in (23) holds in view of the definition of θ_k. The second inequality on θ_k in (23) immediately follows from the definition of θ_k in (21) and the fact that $M_i \ge \gamma M$ for every $i \ge 0$ (see (14)). The bound on τ_k in (23) is a direct consequence of the definition of τ_k in (21), and the facts that $L_k^{avg} \le \bar{L}$ and $M_k \ge \gamma M$. Finally, the bound $\mathcal{O}(1/(\alpha\gamma))$ on θ_k immediately follows from the bound on M_k and the second inequality on θ_k in (23). ■

4 Numerical results

This section reports computational results of AC-FISTA and a corresponding restart variant against five other state-of-the-art algorithms on three instances of N-SCO problems: support vector machine (Subsection 4.1), quadratic programming (Subsection 4.2) and matrix completion (Subsection 4.3).

We start by describing the implementation of AC-FISTA and its restart variant used in our computational benchmark. Our implementation of AC-FISTA sets $M_0 = 0.01M$, computes M_{k+1} according to (14) with $(\alpha, \gamma) = (0.5, 10^{-6})$, and chooses $y_{k+1} = \tilde{y}_{k+1}$. The restart variant of AC-FISTA uses the same parameters as AC-FISTA but rejects y_{k+1} whenever $k \in \mathcal{G}$ and $\phi(y_{k+1}) \ge \phi(y_k)$ in which case it sets $x_k = y_k$ and $A_k = 0$, and repeats the k-th iteration.

We compare our methods with five other ACG variants, namely: (i) the UPFAG method in [4]; (ii) the ADAP-NC-FISTA described in [14]; (iii) the theoretical AC-ACG method proposed in [13] (referred to as ACT in its Section 5); and (iv) restart variants of the methods in (ii) and (iii) which are described in the paragraph below. For the sake of simplicity, we use the abbreviations UP, AD, AC and AF to refer to UPFAG, ADAP-NC-FISTA, AC-ACG and AC-FISTA, respectively, both in the discussions and tables below. Moreover, we use AD(R), AC(R) and AF(R) to denote the restart variants of AD, AC and AF, respectively.

This paragraph provides details about the five other ACG variants used in our benchmark. UP is described in Algorithm 1 of [4] and the code for it was provided by the authors of [4]. In particular, we have used their choice of parameters but have slightly modified the code to accommodate for our termination criterion, i.e., Definition 2.1. More specifically, the input parameters $(\hat{\lambda}_0, \hat{\beta}_0, \gamma_1, \gamma_2, \gamma_3, \delta, \gamma)$ of UP were set to $(1/L, 1/L, 0.4, 0.4, 1, 10^{-3}, 10^{-10})$. AD was implemented by the authors according to its description in Section 3 in [14]. The input triple (M_0, m_0, θ) of

AD was set to $(1, 1000, 1.25)$ in Subsections 4.1 and 4.2, and $(1, 1, 1.25)$ in Subsection 4.3. Method AC is exactly the theoretical AC-ACG method of [13] with parameter pair (α, γ) set to $(0.5, 0.01)$. Moreover, the restart variant AC(R) (resp., AD(R)) uses the same set of parameters as AC (resp., AD) and restarts in the same way as AF(R) does.

All seven methods terminate when a pair (z, v) satisfying a relative termination criterion

$$v \in \nabla f(z) + \partial h(z), \qquad \frac{\|v\|}{\|\nabla f(z_0)\| + 1} \leq \hat{\rho}$$

is obtained, where z_0 is the initial point, $\hat{\rho} = 10^{-7}$ in Subsections 4.1 and 4.2, and $\hat{\rho} = 5 \times 10^{-4}$ in Subsection 4.3. We run all numerical experiments using MATLAB R2017b on a MacBook Pro with a quad-core Intel Core i7 processor and 16 GB of memory.

4.1 Support Vector Machine

This subsection discusses the performance of the methods in our computational benchmark for solving a support vector machine (SVM) problem (see (4.1) in [4]). For given data points $\{(u_i, v_i)\}_{i=1}^p$, where $u_i \in \mathbb{R}^n$ is a feature vector and $v_i \in \{-1, 1\}$ denotes the corresponding label, we formulate the SVM problem as

$$\min \left\{ f(z) := \frac{1}{p} \sum_{i=1}^{p} \ell(u_i, v_i; z) + \frac{\lambda}{2} \|z\|^2 : z \in B_r \right\} \tag{38}$$

where $\ell(u_i, v_i; \cdot) = 1 - \tanh(v_i \langle \cdot, u_i \rangle)$ is a nonconvex sigmoid loss function, $\lambda > 0$ is a regularization parameter and $B_r := \{z \in \mathbb{R}^n : \|z\| \leq r\}$ is a ball with radius $r > 0$ and centered at the origin.

The SVM problem (38) is an instance of (1) where h is the indicator function of the ball B_r. We set $\lambda = 1/p$, $r = 50$ and $\Omega = B_r$, where the set Ω is introduced in (A2). It can be shown that f is differentiable everywhere and satisfies

$$m = M = L = \frac{1}{p} \sum_{i=1}^{p} \frac{4\sqrt{3}}{9} \|u_i\|^2 + \lambda, \quad \forall i = 1, \ldots, p.$$

We now describe the datasets *SVM-1*, *SVM-2*, *SVM-3* and *SVM-4* considered in the numerical experiments. Each dataset contains data points $\{(u_i, v_i)\}_{i=1}^p$ where u_i is a sparse vector with density d and its nonzero entries are drawn from the uniform distribution $\mathcal{U}[0, 1]$, and $v_i = \text{sign}(\langle \bar{z}, u_i \rangle)$ for some $\bar{z} \in B_r$. Table 1 lists basic statistics of the datasets.

Dataset	n	p	Density d	λ	r	M
SVM-1	5000	500	5%	0.002	50	13
SVM-2	10000	1000	5%	0.001	50	25
SVM-3	15000	1000	5%	0.001	50	38
SVM-4	20000	500	5%	0.002	50	50

Table 1: SVM datasets

We start all seven methods from the same initial point z_0 that is generated randomly and uniformly within the ball B_r.

Numerical results of the seven methods for solving (38) with datasets *SVM-1*, *SVM-2*, *SVM-3* and *SVM-4* are given in Table 2. Specifically, the second to eighth columns provide numbers of iterations and running times for the seven methods. We do not report the best objective function values obtained by all seven methods, since they are essentially the same on each instance. The bold numbers highlight the method that has the best performance in an instance of (38).

Dataset	Iteration Count / Running Time (s)						
	UP	AD	AC	AF	AD(R)	AC(R)	AF(R)
SVM-1	250	2333	1678	604	440	339	160
	18	52	62	14	13	15	**5**
SVM-2	254	3996	4801	1352	549	605	230
	81	396	772	144	67	110	**29**
SVM-3	284	3499	6023	1563	503	695	200
	137	529	1505	248	93	187	**38**
SVM-4	156	1701	4136	823	377	630	151
	50	175	661	86	46	113	**19**

Table 2: Numerical results for solving (38) with *SVM-1, 2, 3, & 4*

Recall that we have commented on the practical behavior of the ratios θ_k, τ_k and $|\mathcal{B}_k|/k$ in Subsection 2.2. We now present the statistics of the three ratios of AF and AF(R) for solving the SVM problem (38).

Dataset	AF			AF(R)		
	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$
SVM-1	1.34	0.55	31%	2.16	0.55	37%
SVM-2	1.16	0.60	32%	1.24	0.61	35%
SVM-3	1.04	0.58	26%	1.35	0.62	32%
SVM-4	0.93	0.55	21%	1.25	0.60	37%

Table 3: Statistics of $\bar{\theta}_k$, $\bar{\tau}_k$ and $|\mathcal{B}_k|$

In Table 3, $\bar{\theta}_k$ and $\bar{\tau}_k$ are defined as

$$\bar{\theta}_k := \max\{\theta_k : k \geq 100\}, \quad \bar{\tau}_k := \max\{\tau_k : k \geq 100\}.$$

The ratio $|\mathcal{B}_k|/k$ represents the the percentage of bad iterations at the last iteration of each method.

In summary, computational results demonstrate that: i) AF(R) is the best method in terms of running time; ii) AF(R) (resp., AD(R) an AC(R)) improves the results of AF (resp., AD and AC); and iii) $\bar{\theta}_k$ and $\bar{\tau}_k$ are small and $|\mathcal{B}_k|/k$ is no more than 37%.

4.2 Quadratic Programming

In this subsection, we consider solving a class of nonconvex quadratic programming (QP) problems. More specifically, the QP problem reads as

$$\min\left\{f(Z) := -\frac{\alpha_1}{2}\|D\mathcal{P}(Z)\|^2 + \frac{\alpha_2}{2}\|\mathcal{Q}(Z) - b\|^2 : Z \in O_n\right\} \tag{39}$$

where $(\alpha_1, \alpha_2) \in \mathbb{R}^2_{++}$, $b \in \mathbb{R}^l$, $D \in \mathbb{R}^{n\times n}$, $O_n := \{Z \in \mathcal{S}^n_+ : \mathrm{tr}(Z) = 1\}$ denotes the spectraplex, and $\mathcal{P} : \mathcal{S}^n_+ \to \mathbb{R}^n$ and $\mathcal{Q} : \mathcal{S}^n_+ \to \mathbb{R}^l$ are linear operators given by

$$[\mathcal{P}(Z)]_i = \langle P_i, Z\rangle_F \quad \forall\, 1 \leq i \leq n,$$
$$[\mathcal{Q}(Z)]_j = \langle Q_j, Z\rangle_F \quad \forall\, 1 \leq j \leq l,$$

with $P_i \in \mathcal{S}^n_+$ and $Q_j \in \mathcal{S}^n_+$.

We now describe the datasets *QP-1* and *QP-2* considered in the numerical experiments. Each dataset contains b, D, P_i for $1 \le i \le n$ and Q_j for $1 \le j \le l$. The entries of b are sampled from the uniform distribution $\mathcal{U}[0,1]$. The diagonal entries of the diagonal matrix D are generated from the discrete uniform distribution $\mathcal{U}\{1,1000\}$. Sparse matrices P_i and Q_j have the same density (i.e., percentage of nonzeros) d and their nonzero entries are generated from $\mathcal{U}[0,1]$. Table 4 lists basic statistics of the datasets.

Dataset	l	n	Density d
QP-1	50	200	2.5%
QP-2	50	400	0.5%

Table 4: Quadratic programming datasets

The nonconvex QP problem (39) is an instance of (1) where h is the indicator function of the spectraplex O_n. The set Ω introduced in (A2) is chosen as $\Omega = \{Z \in \mathcal{S}^n_+ : \|Z\|_F = 1\}$. It is easy to see that $\Omega \supset \mathcal{H}$, which is required in (A2), since $\Omega \supset O_n = \mathcal{H}$. For given curvature pairs $(M,m) \in \mathbb{R}^2_{++}$, we L set to $\max\{M,m\}$ and choose scalars α_1 and α_2 so that $(\lambda_{\max}(\nabla^2 f), \lambda_{\min}(\nabla^2 f)) = (M, -m)$, where $\lambda_{\max}(\cdot)$ (resp., $\lambda_{\min}(\cdot)$) denotes the largest (resp., smallest) eigenvalue function.

We start all seven methods from the same initial point $Z_0 = I_n/n$ where I_n is an $n \times n$ identity matrix, namely Z_0 is the centroid of O_n.

Numerical results of the seven methods for solving (39) with datasets *QP-1* and *QP-2* are given in Tables 5 and 6, respectively. Each table addresses a collection of instances with the same dataset and $M = 10^6$. Specifically, each table contains six instances of (39), their first column specifies m for the instances. The explanation of columns in Tables 5 and 6 excluding the first one is the same as that of Table 2 (see the paragraphs preceding Table 2). We do not report the best objective function values obtained by all seven methods, since they are essentially the same on each instance. The bold numbers highlight the method that has the best performance in an instance of (39).

m	Iteration Count / Running Time (s)						
	UP	AD	AC	AF	AD(R)	AC(R)	AF(R)
10^5	2633	2206	1009	947	787	966	419
	261	89	55	30	33	55	**14**
10^4	7203	2591	1820	1744	1573	1777	601
	705	104	98	55	66	99	**20**
10^3	5429	2637	1712	2000	1552	1709	773
	540	109	92	63	65	100	**26**
10^2	6891	2639	1610	1687	1666	1600	736
	653	116	95	52	69	96	**25**
10	6479	2640	1599	1804	1675	1593	785
	613	116	95	56	69	96	**26**

Table 5: Numerical results for solving (39) with *QP-1*

m	Iteration Count / Running Time (s)						
	UP	AD	AC	AF	AD(R)	AC(R)	AF(R)
10^5	56	530	403	140	292	414	140
	13	56	58	**12**	30	60	**12**
10^4	105	868	599	195	364	599	182
	26	93	85	**17**	38	86	**17**
10^3	115	900	564	187	384	557	80
	29	103	81	16	40	80	**15**
10^2	119	904	559	216	385	554	179
	32	103	80	19	40	82	**16**
10	113	904	561	221	385	554	177
	31	104	86	19	40	84	**16**

Table 6: Numerical results for solving (39) with *QP-2*

We now present the statistics of $\bar{\theta}_k$, $\bar{\tau}_k$ and $|\mathcal{B}_k|/k$ of AF and AF(R) for solving the nonconvex QP problem (39).

m	AF			AF(R)		
	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$
10^5	0.92	1.22	13%	1.04	1.24	15%
10^4	1.07	1.05	7%	1.07	1.05	13%
10^3	0.99	1.14	5%	0.99	1.14	13%
10^2	1.02	1.07	5%	1.02	1.07	18%
10	1.00	1.10	5%	1.00	1.10	10%

Table 7: Statistics of $\bar{\theta}_k$, $\bar{\tau}_k$ and $|\mathcal{B}_k|$ for *QP-1*

m	AF			AF(R)		
	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$
10^5	0.60	3.08	13%	0.60	3.08	13%
10^4	0.68	2.29	18%	0.72	2.16	15%
10^3	0.69	2.38	16%	0.74	2.14	15%
10^2	0.69	2.40	14%	0.73	2.17	15%
10	0.69	2.40	14%	0.73	2.17	15%

Table 8: Statistics of $\bar{\theta}_k$, $\bar{\tau}_k$ and $|\mathcal{B}_k|$ for *QP-2*

In summary, computational results demonstrate that: i) AF(R) is the best method in terms of running time; ii) AF(R) (resp., AD(R)) improves the results of AF (resp. AD), while AC(R) has similar performance as AC; and iii) $\bar{\theta}_k$ and $\bar{\tau}_k$ are small and $|\mathcal{B}_k|/k$ is no more than 18%.

4.3 Matrix Completion

This subsection considers a constrained version of the nonconvex low-rank matrix completion (NL-RMC) problem.

We start by giving a few definitions. Given parameters $\beta > 0$ and $\tau > 0$, let $p : \mathbb{R} \to \mathbb{R}_+$ denote the log-sum penalty defined as $p(t) = p_{\beta,\tau}(t) := \beta \log(1 + |t|/\tau)$. Let $\mathcal{Q}$ denote a subset of $\{1,\ldots,l\}\times\{1,\ldots,n\}$. Let $\Pi_{\mathcal{Q}}:\mathbb{R}^{l\times n}\to\mathbb{R}^{l\times n}$ denote a linear operator such that, for given $A\in\mathbb{R}^{l\times n}$, $\Pi_{\mathcal{Q}}(A)_{ij}=A_{ij}$ if $(i,j)\in\mathcal{Q}$, and $\Pi_{\mathcal{Q}}(A)_{ij}=0$ otherwise.

Given radius $R>0$, penalty parameter $\mu>0$, and an incomplete observation matrix $O\in\mathbb{R}^{\mathcal{Q}}$, the constrained version of the NLRMC problem considered in this subsection is

$$\min\left\{\frac{1}{2}\|\Pi_{\mathcal{Q}}(Z-O)\|_F^2+\mu\sum_{i=1}^{r}p(\sigma_i(Z)):Z\in\mathcal{B}_R\right\} \tag{40}$$

where $r=\min\{l,n\}$, $\sigma_i(Z)$ is the i-th singular value of Z and $\mathcal{B}_R=\{Z\in\mathbb{R}^{l\times n}:\|Z\|_F\le R\}$.

It is discussed in [13] that (40) is an instance of the N-SCO problem (1) and can be rewritten as $\min\{f(Z)+h(Z):Z\in\mathbb{R}^{l\times n}\}$ where

$$f(Z)=\frac{1}{2}\|\Pi_{\mathcal{Q}}(Z-O)\|_F^2+\mu\sum_{i=1}^{r}[p(\sigma_i(Z))-p_0\sigma_i(Z)],$$

$$h(Z)=\mu p_0\|Z\|_*+I_{\mathcal{B}_R}(Z),\quad p_0=p'(0)=\frac{\beta}{\tau}$$

and $\|\cdot\|_*$ denotes the nuclear norm, i.e., $\|\cdot\|_*:=\sum_{i=1}^{r}\sigma_i(\cdot)$. It follows from (48) of [13] that the triple (m,M,L) satisfying (6) is

$$(m,M,L)=(2\mu\kappa,1,\max\{1,2\mu\kappa\}) \tag{41}$$

where $\kappa=\beta/\tau^2$.

We now describe the datasets *MovieLens 100K*[3] and *FilmTrust*[4] considered in the numerical experiments. Each dataset contains an observed index set $\mathcal{Q}$ and an incomplete observed matrix O with rows, columns and nonzero entries representing users, items and ratings, respectively, from some collaborative filtering systems. Table 9 lists basic statistics of the datasets.

Dataset	Users (l)	Items (n)	Ratings	Density	Scale
MovieLens 100K	943	1682	100000	6.30%	[1,5]
FilmTrust	1508	2071	35497	1.14%	[0.5,4.0]

Table 9: Matrix completion datasets

The radius R is chosen as the Frobenius norm of the matrix of size $l\times n$ containing the same entries as O in $\mathcal{Q}$ and entries outside of $\mathcal{Q}$ being maximum of the scale (i.e., 5 (resp., 4) in the case of *MovieLens 100K* (resp., *FilmTrust*)). The set Ω introduced in (A2) is set to be $\mathcal{B}_R$ with R as the aforementioned radius. It is easy to see that $\Omega\supset\mathcal{H}$, which is required in (A2), since $\Omega=\mathcal{B}_R=\mathcal{H}$.

We start all seven methods from the same initial point Z_0 that is sampled from the standard Gaussian distribution and is within $\mathcal{B}_R$.

Numerical results of the seven methods for solving (40) with datasets *MovieLens 100K* and *FilmTrust* are given in Tables 10 and 11, respectively. Each table addresses a collection of instances with the same dataset. The first columns in Tables 10 and 11 present the values of m of the four instances computed according to (41) with four different triples (μ,β,τ). In addition to the numbers of iterations and running times of all seven methods, the second to eighth columns of Tables 10 and 11 also provide the function values of (40) at the last iteration. The bold numbers highlight

[3]http://grouplens.org/datasets/movielens/

[4]http://guoguibing.github.io/librec/datasets.html#filmtrust

the method that has the best performance (smallest function value or least running time) in an instance of (40).

m	Function Value / Iteration Count / Running Time (s)						
	UP	AD	AC	AF	AD(R)	AC(R)	AF(R)
4.4	2605	2625	2296	**1836**	2625	2304	1912
	521	1674	1046	375	1674	904	305
	1545	1946	1242	287	1946	1087	**245**
8.9	4261	4203	3896	**3617**	4203	3914	3797
	576	1794	4773	291	1794	4511	241
	1621	1930	6519	233	1930	6245	**208**
20	4637	4582	4313	**4098**	4582	4312	4164
	676	2209	14892	260	2209	15708	304
	1914	2364	19948	**212**	2364	21666	267
30	6753	6293	6005	**5333**	6293	5952	5524
	606	1963	30815	505	1963	27986	413
	1628	2104	43172	417	2104	38644	**349**

Table 10: Numerical results for solving (40) with *MovieLens 100K*

m	Function Value / Iteration Count / Running Time (s)						
	UP	AD	AC	AF	AD(R)	AC(R)	AF(R)
4.4	1050	1069	981	849	1069	988	**804**
	584	2025	942	347	2025	1053	586
	6460	9063	4072	**991**	9063	4546	1753
8.9	1814	1854	1759	1538	1854	1738	**1516**
	634	2410	4312	469	2410	5461	753
	7130	11171	22187	**1334**	11171	29569	2198
20	2120	2064	1988	**1739**	2064	1993	1777
	630	2665	13957	676	2665	14379	528
	7214	12701	73023	1959	12701	77128	**1617**
30	2980	2917	2855	**2593**	2917	2853	**2593**
	559	2365	19419	533	2365	18515	533
	6244	11205	100580	**1582**	11205	96675	**1582**

Table 11: Numerical results for solving (40) with *FilmTrust*

We now present the statistics of $\bar{\theta}_k$, $\bar{\tau}_k$ and $|\mathcal{B}_k|/k$ of AF and AF(R) for solving the MC problem (40).

m	AF			AF(R)		
	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$
4.4	1.07	1.23	6%	1.12	1.20	4%
8.9	1.04	1.53	8%	1.02	1.48	10%
20	0.97	2.16	9%	1.00	1.88	13%
30	1.02	2.49	7%	1.02	2.40	11%

Table 12: Statistics of $\bar{\theta}_k$, $\bar{\tau}_k$ and $|\mathcal{B}_k|$ for *MovieLens 100K*

m	AF			AF(R)		
	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$	$\bar{\theta}_k$	$\bar{\tau}_k$	$\|\mathcal{B}_k\|/k$
4.4	1.09	1.25	10%	1.11	1.21	9%
8.9	1.02	1.55	6%	0.99	1.61	6%
20	1.04	2.07	8%	1.06	2.07	9%
30	1.04	2.59	11%	1.04	2.59	11%

Table 13: Statistics of $\bar{\theta}_k$, $\bar{\tau}_k$ and $|\mathcal{B}_k|$ for *FilmTrust*

In summary, computational results demonstrate that: i) AF and AF(R) are the best two methods; ii) AD(R) does not restart and has the same performance as AD; and iii) $\bar{\theta}_k$ and $\bar{\tau}_k$ are small and $|\mathcal{B}_k|/k$ is no more than 13%.

5 Concluding remarks

This paper studies the AC-FISTA method, which is a FISTA-type ACG variant of the AC-ACG method proposed in [13], for solving the N-SCO problem (1). At the k-th iteration, both methods compute $y(\tilde{x}_k; M_k)$ defined in (2) as a potential candidate for the next iterate where M_k is an estimation of the local upper curvature of (1) at $\tilde{x}_k$ obtained according to (14), and chooses as the next iterate either this point if it satisfies (3) or the convex combination in (13) otherwise. However, in contrast to AC-ACG, AC-FISTA computes M_k according to (14) using the average of the observed upper curvatures C_k's defined in (9) instead of the larger upper-Lipschitz curvatures $\tilde{C}_k$'s defined in the line above (5). In addition, AC-FISTA performs only one composite resolvent evaluation during the good iterations, and two composite resolvent evaluations in the bad ones, but has been observed to perform an average of about one composite resolvent evaluation per iteration in practice. These two features together lead to a practical AC-FISTA variant that substantially outperforms previous ACG variants as well as the theoretical and practical AC-ACG variants, both in terms of running time and solution quality.

We end this paper by discussing some possible extensions. First, even though we have not studied the convergence rate of the practical AC-ACG variant of [13], we believe that such analysis will follow by using similar arguments as the ones used in this paper to analyze AC-FISTA. Second, numerical results show that the restart variant of AC-FISTA greatly improves the empirical performance of its original variant but its convergence rate analysis has not been established anywhere in the literature and is an interesting research direction to pursue.

Data availability statements

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

[1] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

[2] D. Drusvyatskiy and C. Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, pages 1–56, 2018.

[3] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Programming*, 156:59–99, 2016.

[4] S. Ghadimi, G. Lan, and H. Zhang. Generalized uniformly optimal methods for nonlinear programming. *Journal of Scientific Computing*, 79(3):1854–1881, 2019.

[5] N. Gillis. The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, 12(257):257–291, 2014.

[6] Q. Gu, Z. Wang, and H. Liu. Sparse pca with oracle property. In *Advances in neural information processing systems*, pages 1529–1537, 2014.

[7] W. Kong, J. G. Melo, and R. D. C. Monteiro. Complexity of a quadratic penalty accelerated inexact proximal point method for solving linearly constrained nonconvex composite programs. *SIAM Journal on Optimization*, 29(4):2566–2593, 2019.

[8] W. Kong and R. D. C. Monteiro. Accelerated inexact composite gradient methods for nonconvex spectral optimization problems. *Computational Optimization and Applications*, pages 1–43, 2022.

[9] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

[10] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. *Adv. Neural Inf. Process. Syst.*, 28:379–387, 2015.

[11] Q. Li, Y. Zhou, Y. Liang, and P. K. Varshney. Convergence analysis of proximal gradient with momentum for nonconvex optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2111–2119, 2017.

[12] J. Liang and R. D. C. Monteiro. A doubly accelerated inexact proximal point method for nonconvex composite optimization problems. *Available on arXiv:1811.11378*, 2018.

[13] J. Liang and R. D. C. Monteiro. An average curvature accelerated composite gradient method for nonconvex smooth composite optimization problems. *SIAM Journal on Optimization*, 31(1):217–243, 2021.

[14] J. Liang, R. D. C. Monteiro, and C.-K. Sim. A FISTA-type accelerated gradient algorithm for solving smooth nonconvex composite optimization problems. *Computational Optimization and Applications*, pages 1–31, 2021.

[15] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst for gradient-based nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 613–622. PMLR, 2018.

[16] Q. Yao and J. T. Kwok. Efficient learning with a family of nonconvex regularizers by redistributing nonconvexity. *Journal of Machine Learning Research*, 18:179–1, 2017.

[17] Q. Yao, J. T. Kwok, F. Gao, W. Chen, and T.-Y. Liu. Efficient inexact proximal gradient algorithm for nonconvex problems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3308–3314. IJCAI, 2017.