An Average Curvature Accelerated Composite Gradient Method for Nonconvex Smooth Composite Optimization Problems

Jiaming Liang ∗ Renato D.C. Monteiro ∗

September 9, 2019 (1st revision: October 18, 2019; 2nd revision: May 19, 2020)

Abstract

This paper presents an accelerated composite gradient (ACG) variant, referred to as the AC-ACG method, for solving nonconvex smooth composite minimization problems. As opposed to well-known ACG variants that are either based on a known Lipschitz gradient constant or a sequence of maximum observed curvatures, the current one is based on the average of all past observed curvatures. More specifically, AC-ACG uses a positive multiple of the average of all observed curvatures until the previous iteration as a way to estimate the “function curvature” at the current point and then two resolvent evaluations to compute the next iterate. In contrast to other variable Lipschitz estimation variants, e.g., the ones based on the maximum curvature, AC-ACG always accepts the aforementioned iterate regardless how poor the Lipschitz estimation turns out to be. Finally, computational results are presented to illustrate the efficiency of AC-ACG on both randomly generated and real-world problem instances.

1 Introduction

In this paper, we study an ACG-type algorithm for solving a nonconvex smooth composite optimization (SCO) problem

\[ \phi^* := \min \{ \phi(z) := f(z) + h(z) : z \in \mathbb{R}^n \} \]

where \( f \) is a real-valued differentiable (possibly nonconvex) function with an \( M \)-Lipschitz continuous gradient on \( \text{dom} \ h \) and \( h : \mathbb{R}^n \rightarrow (-\infty, \infty] \) is a proper lower semicontinuous convex function with a bounded domain.

A large class of algorithms for solving (1) sets the next iterate \( y_{k+1} \) as the unique optimal solution \( y(\tilde{x}_k; M_k) \) of the the linearized prox subproblem

\[ y(\tilde{x}_k; M_k) := \arg\min \left\{ \ell_f(x; \tilde{x}_k) + h(x) + \frac{M_k}{2} \| x - \tilde{x}_k \|^2 : x \in \mathbb{R}^n \right\} \]

where \( \ell_f(x; \tilde{x}_k) := f(\tilde{x}_k) + \langle \nabla f(\tilde{x}_k), x - \tilde{x}_k \rangle \), the prox-center \( \tilde{x}_k \) is chosen as either the current iterate \( y_k \) (as in unaccelerated algorithms) or a convex combination of \( y_k \) and another auxiliary iterate \( x_k \) (as in accelerated algorithms), and \( M_k \) is good upper curvature of \( f \) at \( \tilde{x}_k \), i.e., \( M_k > 0 \) and satisfies

\[ C(y(\tilde{x}_k; M_k); \tilde{x}_k) \leq M_k \]
where
\[ \mathcal{C}(y; \bar{x}) := \frac{2[f(y) - \ell f(y; \bar{x})]}{\|y - \bar{x}\|^2}. \] (4)

Regardless of the choice of \( \bar{x}_k \), it is well-known 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 computed as the local curvature of \( f \) at \( \bar{x}_k \), is the smallest value of \( M_k \) satisfying (3). However, since finding \( \bar{M}_k \) is generally time-consuming, alternative strategies that upper estimate \( \bar{M}_k \) are used. A common one is a backtracking procedure that initially sets \( M_k \) to be the maximum of all the observed curvatures \( \mathcal{C}_1, \ldots, \mathcal{C}_{k-1} \) where \( \mathcal{C}_i := \mathcal{C}(y_{i+1}; \bar{x}_i) \) for every \( i \geq 1 \). It then checks whether \( M_k \) is a good curvature of \( f \) at \( \bar{x}_k \); if so, it sets \( y_{k+1} = y(\bar{x}_k; M_k) \); otherwise, it updates \( M_k \leftarrow \eta M_k \) for some parameter \( \eta > 1 \), and then repeats this same step again. Such an approach has been used extensively in the literature dealing with composite gradient methods both in the context of convex and nonconvex SCO (N-SCO) problems (see for example [3, 7, 16, 21]) and can be efficient particularly for those SCO instances where a sharp upper bound \( M \) on the smallest Lipschitz constant \( M \) of \( \nabla f \) on \( \text{dom} \, h \) is not available.

This paper investigates an ACG variant for solving the N-SCO problem where \( M_k \) is computed as a positive multiple of the average of all observed curvatures up to the previous iteration. As opposed to ACG variants based on the scheme outlined above as well as other ACG variants, AC-ACG always computes a new step regardless of whether \( M_k \) overestimates or underestimates \( \mathcal{C}_k \). More specifically, if \( M_k \) overestimates \( \mathcal{C}_k \) then a composite step as in (2) is taken; otherwise, \( y_{k+1} \) is set to be a convex combination of \( y_k \) and an auxiliary iterate \( x_{k+1} \), which is obtained by a resolvent evaluation of \( h \). It is worth noting that both of these steps are used in previous ACG variants but only one of them is used at a time. The main result of the paper establishes a convergence rate for AC-ACG. More specifically, it states that \( k \) iterations of the AC-ACG method generate a pair \((y, v)\) satisfying \( v \in \nabla f(y) + \partial h(y) \) and \( \|v\|^2 = O(M_k/k) \) where \( M_k \) is as in the beginning of this paragraph. Since \( M_k \) is usually much smaller than \( M \) or even \( M_k \), this convergence rate bound explains the efficiency of AC-ACG to solve both randomly generated and real-world problem instances of (1) used in our numerical experiments. Finally, it is shown that AC-ACG also has similar iteration-complexity as previous ACG variants (e.g., [6, 10, 15, 16]).

**Related works.** The first complexity analysis of an ACG algorithm for solving (1) under the assumption that \( f \) is a nonconvex differentiable function whose gradient is Lipschitz continuous and that \( h \) is a simple lower semicontinuous convex function is established in the novel work [6]. Inspired by [6], many papers have proposed other ACG variants for solving (1) under the aforementioned assumptions (see e.g., [5, 7, 16]) or even under the relaxed assumption that \( h \) is nonconvex (see e.g., [13, 14, 25]). It is worth mentioning that: i) in contrast to [6, 16], the other works deal with hybrid-type accelerated methods that resort to unaccelerated composite gradient steps whenever a certain descent property is not satisfied; and ii) in contrast to the methods of [7, 13, 16] that choose \( M_k \) adaptively in a manner similar to that described in the second paragraph in Section 1, the methods in [5, 6, 14, 25] works with a constant sequence \( \{M_k\} \), namely, \( M_k = M \) for some \( M > \bar{M} \). Section 3 provides a more detailed overview of ACG variants for solving both convex and nonconvex SCO problems which includes most of the ones just mentioned.

Other approaches towards solving (1) use an inexact proximal point scheme where each prox subproblem is constructed to be (possibly strongly) convex and hence efficiently solvable by a convex ACG variant. Papers [4, 10, 22] propose a descent unaccelerated inexact proximal-type method, which works with a larger prox stepsize and hence has a better outer iteration-complexity than the approaches in the previous paragraph. Paper [15] presents an accelerated inexact proximal point method that performs an accelerated step with a large prox stepsize in every outer iteration and
requires a prox subproblem to be approximately solved by an ACG variant in the same way as in the algorithms presented in [4, 10].

Definitions and notations. The set of real numbers is denoted by \( \mathbb{R} \). The set of non-negative real numbers and the set of positive real numbers are denoted by \( \mathbb{R}_+ \) and \( \mathbb{R}_{++} \), 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 inner product and Frobenius norm in \( \mathbb{R}^{m \times n} \) are denoted by \( \langle \cdot, \cdot \rangle_F \) and \( \| \cdot \|_F \), respectively. The set of real \( n \times n \) symmetric matrices is denoted by \( \mathcal{S}^n \), and we define \( \mathcal{S}^n_+ \) to be the subset of \( \mathcal{S}^n \) consisting of the positive semidefinite matrices. The indicator function \( I_S \) of a set \( S \subset \mathbb{R}^n \) is defined as \( I_S(z) = 0 \) for every \( z \in S \), and \( I_S(z) = \infty \), otherwise. The cardinality of a finite set \( A \) is denoted by \( |A| \). Let \( O_1(\cdot) \) denote \( O(\cdot + 1) \) where \( O \) is the big \( O \) notation.

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

\[
\psi(\alpha z + (1 - \alpha)u) \leq \alpha \psi(z) + (1 - \alpha)\psi(u) - \frac{\alpha(1 - \alpha)\mu}{2} \| z - u \|^2
\]

for every \( z, u \in \operatorname{dom} \psi \) and \( \alpha \in [0, 1] \). If \( \psi \) is differentiable at \( \bar{z} \in \mathbb{R}^n \), then its affine approximation \( \ell_{\psi}(\cdot; \bar{z}) \) at \( \bar{z} \) is defined as

\[
\ell_{\psi}(z; \bar{z}) := \psi(\bar{z}) + \langle \nabla \psi(\bar{z}), z - \bar{z} \rangle \quad \forall z \in \mathbb{R}^n.
\]

The subdifferential of \( \psi \) at \( z \in \mathbb{R}^n \) is denoted by \( \partial \psi(z) \). The set of all proper lower semi-continuous convex functions \( \psi : \mathbb{R}^n \to (-\infty, +\infty] \) is denoted by \( \operatorname{Conv}(\mathbb{R}^n) \).

Organization of the paper. Section 2 describes the N-SCO problem and the assumptions made on it. It also presents the AC-ACG method for solving the N-SCO problem and describes the main result of the paper, which establishes a convergence rate bound for AC-ACG in terms of the average of observed curvatures. Section 3 contains three subsections. The first subsection reviews three ACG variants for solving convex SCO (C-SCO) problems. The second (resp. third) one reviews pure (resp. hybrid) ACG variants for solving N-SCO problems. Section 4 provides the proof of the main result stated in Section 2. Section 5 presents computational results illustrating the efficiency of the AC-ACG method. Section 6 presents some concluding remarks. Finally, the appendix contains a technical result.

2 The AC-ACG method for solving the N-SCO problem

This section presents the main algorithm studied in this paper, namely, an ACG method based on a sequence of average curvatures, and derives a convergence rate for it expressed in terms of this sequence. More specifically, it describes the N-SCO problem and the assumptions made on it, presents the AC-ACG method and states the main result of the paper, i.e., the convergence rate of the AC-ACG method.

The problem of interest in this paper is the N-SCO problem (1), where the following conditions are assumed to hold:

(A1) \( h \in \operatorname{Conv}(\mathbb{R}^n) \);

(A2) \( f \) is a nonconvex differentiable function on \( \operatorname{dom} h \) and there exist scalars \( m \geq 0, M \geq 0 \) such that for every \( u, u' \in \operatorname{dom} h \),

\[
-\frac{m}{2} \| u - u' \|^2 \leq f(u) - \ell_f(u; u'), \quad \| \nabla f(u) - \nabla f(u') \| \leq M \| u - u' \|;
\]

(5)
(A3) the diameter $D := \sup\{\|u - u'\| : u, u' \in \text{dom } h\}$ is bounded.

Throughout the paper, we let $\bar{m}$ (resp., $\bar{M}$) denote the smallest scalar $m \geq 0$ (resp., $M \geq 0$) satisfying the first (resp., second) inequality in (5).

We now make some remarks about the above assumptions. First, the set of optimal solutions $X_*$ is nonempty and compact in view of (A1)-(A3). Second, the second inequality in (5) implies

$$-\frac{M}{2}\|u - u'\|^2 \leq f(u) - \ell_f(u; u') \leq \frac{M}{2}\|u - u'\|^2 \quad \forall u, u' \in \text{dom } h.$$  

(6)

Third, the last remark together with the fact that $f$ is nonconvex on dom $h$ due to assumption (A2) implies that $0 < \bar{m} \leq \bar{M}$.

A necessary condition for $\hat{y}$ to be a local minimum of (1) is that $0 \in \nabla f(\hat{y}) + \partial h(\hat{y})$, i.e, $\hat{y}$ be a stationary point of (1). More generally, given a tolerance $\hat{\rho} > 0$, a pair $(\hat{y}, \hat{v})$ is called a $\hat{\rho}$-approximate stationary pair of (1) if

$$\hat{v} \in \nabla f(\hat{y}) + \partial h(\hat{y}), \quad \|\hat{v}\| \leq \hat{\rho}.$$  

(7)

We are now ready to state the AC-ACG method, which stops when a $\hat{\rho}$-approximate stationary pair of (1) is computed. AC-ACG requires as input a scalar $M \geq \bar{M}$ where $\bar{M}$ is defined in the paragraph following (A3).

**Average Curvature - Accelerated Composite Gradient (AC-ACG)**

0. Let a parameter $\gamma \in (0, 1)$, a scalar $M \geq \bar{M}$, a tolerance $\hat{\rho} > 0$ and an initial point $y_0 \in \text{dom } h$ be given and set $A_0 = 0, x_0 = y_0, M_0 = \gamma M, k = 0$ and

$$\alpha = \frac{0.9}{8} \left(1 + \frac{1}{0.9\gamma}\right)^{-1};$$  

(8)

1. compute

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

(9)

2. set $y_{k+1}^g = y(\tilde{x}_k; M_k)$ where $y(\cdot; \cdot)$ is as in (2) and compute

$$x_{k+1} = \arg\min_{u \in \mathbb{R}^n} \left\{a_k \left[\ell_f(u; \tilde{x}_k) + h(u)\right] + \frac{1}{2}\|u - x_k\|^2\right\},$$  

(10)

$$v_{k+1} = M_k (\tilde{x}_k - y_{k+1}^g) + \nabla f(y_{k+1}^g) - \nabla f(\tilde{x}_k);$$  

(11)

3. if $\|v_{k+1}\| \leq \hat{\rho}$ then output $(\hat{y}, \hat{v}) = (y_{k+1}^g, v_{k+1})$ and stop; otherwise, compute

$$C_k = \max \left\{C(y_{k+1}^g; \tilde{x}_k), \frac{\|\nabla f(y_{k+1}^g) - \nabla f(\tilde{x}_k)\|}{\|y_{k+1}^g - \tilde{x}_k\|}\right\},$$  

(12)

$$C_{k+1}^{\text{avg}} = \frac{1}{k+1} \sum_{j=0}^{k} C_j,$$  

(13)

$$M_{k+1} = \max \left\{\frac{1}{\alpha} C_{k+1}^{\text{avg}}, \gamma M\right\}$$  

(14)

where $C(\cdot; \cdot)$ is as in (4);
We add a few observations about the AC-ACG method. First, the first two identities in (9) imply that
\[ A_{k+1} = M_k a_k^2. \] (16)
Second, the AC-ACG method evaluates two gradients of \( f \) and exactly two resolvents of \( h \), i.e., an evaluation of \((I + \lambda\partial h)^{-1}(\cdot)\) for some \( \lambda > 0 \) per iteration, namely, one in (2) and the other one in (10). Third, Theorem 2.1 below guarantees that AC-ACG always terminates and outputs a \( \hat{\rho} \)-approximate solution \((\hat{y}, \hat{v})\) (see step 3). Fourth, \( C_k \) is the most recent observed curvature, \( C_k^{\text{avg}} \) is the average of all observed curvatures obtained so far and \( M_{k+1} \) is a modified average curvature that will be used in the next iteration to compute \( y_{k+2}^0 \). Fifth, the observed curvature \( C_k \) used here is different from the one mentioned in the Introduction (see (3)) and it is more suitable for our theoretical analysis. Sixth, every iteration starts with a triple \((A_k, x_k, y_k)\) and obtains the next one \((A_{k+1}, x_{k+1}, y_{k+1})\) as in (9), (10) and (2). The iterate \( y_{k+1} \) is chosen to be either \( y_{k+1}^b = y(\tilde{x}_k; M_k) \) obtained in (2) or the convex combination \( y_{k+1}^b \) defined in (15) depending on whether the current curvature \( C_k \) is smaller than or equal to a multiple (e.g., 0.9) of the modified average curvature \( M_k \) or not, respectively. Seventh, in the iterations for which \( C_k \leq 0.9M_k \) (called the good ones), \( M_k \) is clearly a good upper curvature of \( f \) at \( \tilde{x}_k \) in view of the definitions of \( C_k \) and \( y_{k+1}^b \) in (12) and step 2 of AC-ACG, respectively, and the definition of a good curvature in (3). Thus, assuming that the frequency of good iterations is relatively high, it is reasonable to expect that the smaller the sequence \( \{M_k\} \) is, the faster the convergence rate of AC-ACG will be (see the discussion after (3) in the Introduction). Eighth, it follows as a consequence of the results of Section 4 that the number of good iterations is relatively large (see Lemma 4.5) and that the overall effect of the bad ones are nicely under control (see Lemma 4.4). Moreover, Theorem 2.1 below states that the convergence rate of AC-ACG is directly proportional to \( \sqrt{M_k} \) in that \( \min\{\|v_i\| : i \leq k\} = O(\sqrt{M_k}/\sqrt{k}) \).

We now discuss the likelihood of \( M_{k+1} \), or equivalently, \( \gamma_{k+1} := M_{k+1}/M \), being small. First observe that (14) implies that \( \gamma_{k+1} \geq \gamma \). Hence, let us examine the situation in which \( \gamma_{k+1} = \gamma \), i.e., \( \gamma_{k+1} \) reaches its lowest possible value for a fixed \( \gamma \in (0,1) \). Clearly, it follows from (14) that \( \gamma_{k+1} = \gamma \) if and only if
\[ \frac{C_k^{\text{avg}}}{M} \leq \alpha \gamma. \] (17)
Moreover, in view of (8) and the fact that \( \gamma < 1 \), it follows that \( \alpha = \Theta(\gamma) \), and hence (17) implies that \( C_k^{\text{avg}} / M = O(\gamma^2) \). In conclusion, under the restrictive choice of \( \alpha \) in (8), \( \gamma_{k+1} = \gamma \) can only happen when the computed average curvature ratio \( C_k^{\text{avg}} / M \) is \( O(\gamma^2) \). However, choice (8) for \( \alpha \) is too conservative in practice. Indeed, it follows from the proof of Lemma 4.5 and the arguments in the paragraph following it that in practice \( \alpha \in (0,1) \) can be chosen as \( \Theta(1) \) instead of \( \Theta(\gamma) \) as above. Clearly, with such a choice of \( \alpha \), (17) implies that the ratio \( C_k^{\text{avg}} / M \) is \( O(\gamma) \) instead of \( O(\gamma^2) \) as above. In summary, if \( \gamma \in (0,1) \) is relatively small and \( \alpha \) is chosen as \( (0,1) \ni \alpha = \Theta(1) \) instead of (8), then the chances of having \( \gamma_{k+1} = \gamma \) increases. In view of the aforementioned observation, the two AC-ACG variants which are computationally profiled in Section 5 relax the choice of \( \alpha \) from (8) to one satisfying \((0,1) \ni \alpha = \Theta(1) \).
We now state the main result of the paper which describes how fast one of the iterates $y_1, \ldots, y_k$ approaches the stationary condition $0 \in \nabla f(y) + \partial h(y)$. A remarkable feature of its convergence rate bound is that it is expressed in terms of $M_k$ rather than a scalar $M \geq \bar{M}$.

**Theorem 2.1.** The following statements hold:

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

(b) for every $k \geq 12$, we have

$$\min_{1 \leq i \leq k} \|v_i\|^2 \leq O \left( \frac{M_k^2 D^2}{\gamma_k^2} + \frac{\theta_k \bar{m} M_k D^2}{k} \right)$$

where

$$\theta_k := \max \left\{ \frac{M_k}{\bar{M}_i} : 0 \leq i \leq k \right\} \geq 1. \quad (18)$$

We now make two remarks about Theorem 2.1. First, it immediately leads to a worst-case iteration-complexity bound as follows. In view of the second inequality in (5), the second inequality in (6), the definition of $\bar{M}$ in the paragraph following (A3), and relation (12), it follows that for every $k \geq 0$, $C_k \leq \bar{M}$, and hence that $C_k^{avg} \leq \bar{M}$ in view of (13). The latter inequality, (14), and the fact that $\alpha = \Theta(\gamma)$ (see the line following (17)), then imply that $\gamma \leq M_{k+1}/M$ and

$$\frac{M_{k+1}}{M} = O \left( \frac{\bar{M}}{\alpha M} + \gamma \right) = O \left( \frac{M}{\gamma M} + \gamma \right). \quad (19)$$

These two estimates and the definition of $\theta_k$ in (18) then easily imply that

$$\theta_k = O \left( \frac{\bar{M}}{\gamma^2 M} + 1 \right).$$

Thus, it follows from (19), the last observation and Theorem 2.1(b), that the iteration-complexity for AC-ACG to obtain a $\hat{\rho}$-approximate stationary pair $(\hat{y}, \hat{v})$ is

$$O_1 \left[ \frac{D}{\gamma^{1/2} \hat{\rho}} + \left( \frac{\bar{M}}{\gamma^2 M} + 1 \right) \frac{\bar{m} D^2}{\hat{\rho}^2} \right] \left( \frac{\bar{M}}{\gamma} + \gamma M \right) = O_1 \left( \frac{MD}{\gamma^{3/2} \hat{\rho}} + \frac{\bar{m} MD^2}{\gamma^3 \hat{\rho}^2} \right). \quad (20)$$

Hence, for small values of $\gamma$, the worst-case iteration-complexity of AC-ACG is high but, if $\gamma$ is viewed as a constant, i.e., $1/\gamma = O(1)$, then the above complexity is as good as any other ACG method found in the literature for solving the N-SCO problem as long as the second term in (20) is the dominant one. In particular, in terms of $\hat{\rho}$ only, its worst-case iteration-complexity for solving an N-SCO problem is $O(1/\hat{\rho}^2)$, which is identical to that of any other known ACG method (see e.g., [6, 10, 15, 16]).

Second, the dependence of the worst-case iteration-complexity (20) on $\gamma$ is not good because it is obtained using the conservative estimate (19). We will now examine the iteration-complexity bound under the assumption that $\gamma_{k+1} = M_{k+1}/M = \gamma$, or equivalently, (17) holds, for every $k \geq 0$. In this case, $\theta_k = 1$ for every $k \geq 0$ and hence the convergence rate bound in Theorem 2.1(b) yields the iteration-complexity bound

$$O_1 \left( \frac{\gamma^{1/2} MD}{\hat{\rho}} + \frac{\gamma \bar{m} MD^2}{\hat{\rho}^2} \right)$$

for AC-ACG, which improves as $\gamma$ decreases. This contrasts with bound (20), which becomes worse as $\gamma$ decreases.
3 Comparison with other accelerated type methods

This section gives a brief overview of existing ACG methods for solving convex and nonconvex SCO problems. It contains three subsections. The first subsection reviews three ACG variants for solving C-SCO problems. The second one discusses pure ACG variants for solving N-SCO problems, i.e., ACG variants which perform only accelerated steps similar to the ones of the variants of the first subsection. The third one discusses hybrid ACG variants which, in addition to accelerated composite gradient steps, may also perform unaccelerated ones.

3.1 Review of convex ACG methods

This subsection reviews three ACG variants for solving C-SCO problems, i.e., SCO problems of the form (1) where (A1)-(A3) hold with \( m = 0 \), and hence \( f \) is convex. All the ACG methods reviewed here are described in terms of the notation introduced in the AC-ACG method or the ACG framework described below. This approach has the advantage that all the ACG methods are viewed under the same notation and hence their similarities/differences become more apparent.

The accelerated gradient method for solving unconstrained C-SCO problems (i.e., (1) with \( h = 0 \)) were originally developed by Nesterov in his celebrated work [18]. Subsequently, several variants of his method (see for example [1, 3, 11, 17, 19, 20, 21, 23]) have been developed for solving C-SCO problems.

Before reviewing ACG variants for solving C-SCO, we first describe a common ACG framework underlying them.

ACG framework

0. Let an initial point \( y_0 \in \text{dom} h \) be given, and set \( x_0 = y_0, A_0 = 0 \) and \( k = 0 \);
1. compute \( a_k, A_{k+1} \) and \( \tilde{x}_k \) as in (9);
2. compute \( x_{k+1} \) and \( y_{k+1} \) using one of the rules listed below;
3. set \( k \leftarrow k + 1 \), and go to step 1.

We will now describe three possible rules for computing the iterates \( x_{k+1} \) and \( y_{k+1} \) in step 2 of the above framework.

i) (FISTA rule) This rule sets \( y_{k+1}^a = y(\tilde{x}_k; M_k) \) where \( y(\tilde{x}_k; M_k) \) is defined in (2) and \( M_k > 0 \) is a good upper curvature of \( f \) at \( \tilde{x}_k \), then chooses \( y_{k+1} \) to be any point satisfying \( \phi(y_{k+1}) \leq \phi(y_{k+1}^a) \) and computes \( x_{k+1} \) as

\[
x_{k+1} = y_{k+1}^a + \frac{A_k}{a_k} (y_{k+1}^a - y_k).
\]  

FISTA rule with \( y_{k+1} = y_{k+1}^a \) was first introduced in [19] and further studied in [2, 3].

ii) (AT rule) This rule computes \( x_{k+1} \) as (10) and chooses \( y_{k+1} \) to be any point satisfying \( \phi(y_{k+1}) \leq \phi(y_{k+1}^a) \) where

\[
y_{k+1}^a = \frac{A_k y_k + a_k x_{k+1}}{A_{k+1}}.
\]  

This rule with \( y_{k+1} = y_{k+1}^a \) was introduced by Auslender and Teboulle in [1], which explains the name “AT” adopted here.
iii) (LLM rule) This rule sets \( y_{k+1} \) as in the FISTA rule and and \( x_{k+1} \) as in the AT rule. LLM rule was introduced by Lu, Lan and Monteiro in [11], which explains the name “LLM” adopted here.

We now make a few remarks on the three ACG variants based on the above three rules. First, the ACG variant based on the LLM rule performs two resolvent evaluations of \( h \) per iteration, while the variants based on the AT and FISTA rules perform exactly one resolvent evaluation. Second, two popular choices of an upper curvature sequence \( \{M_k\} \) are as follows: 1) for some \( M \geq \bar{M} \), \( M_k = M \) for every \( k \geq 0 \); and 2) for every \( k \geq 0 \), \( M_k \) is computed by a backtracking procedure such as the one outlined in the second paragraph of Section 1. While papers [1, 11, 19] consider only the ACG variant based on the LLM rule performs two resolvent evaluations of \( y \) as the one outlined in the second paragraph of Section 1. While papers [1, 11, 19] consider only the first choice, [3, 21] analyze the FISTA variant for both choices of \( \{M_k\} \). Third, the AC-ACG method studied in this paper uses the LLM rule and works with a sequence \( \{M_k\} \) such that \( M_k \) is not necessarily a good upper curvature of \( f \) at \( \tilde{x}_k \).

We now comment on the monotonicity of the three aforementioned ACG variants. The three ACG variants based on the identity \( y_{k+1} = y^a_{k+1} \) are not necessarily monotone (i.e., it satisfies \( \phi(y_{k+1}) \leq \phi(y_k) \) for every \( k \geq 0 \)), even if every \( \bar{M}_k \) is a good upper curvature of \( f \) at \( \tilde{x}_k \). However, they can be made monotone by invoking an idea introduced in [20] which sets \( \bar{y}_{k+1} = \arg\min \{ \phi(y) : y \in \{y_k, y^a_{k+1}\} \} \), where \( y^a_{k+1} \) is as described in each of the rules above. Another alternative way of forcing monotonicity, which requires an extra resolvent evaluation of \( h \), is to choose \( y_{k+1} \) as

\[
y_{k+1} = \arg\min \{ \phi(y) : y \in \{y_k, y^a_{k+1}, y^{na}_{k+1}\} \}
\]  

(23)

where \( y^{na}_{k+1} = y(y_k; M^{na}_k) \) and \( M^{na}_k \) is a good upper curvature of \( f \) at \( y_k \). We remark that \( y_k \) can actually be removed from the right hand side of (23). This is due to the fact that \( M^{na}_k \) being a good upper curvature of \( f \) at \( y_k \) implies that \( \phi(y^{na}_{k+1}) \leq \phi(y_k) \) in view of Lemma A.1 in the Appendix with \( (M_k, \tilde{x}_k, y_{k+1}) = (M^{na}_k, y_k, y^{na}_{k+1}) \).

### 3.2 Pure accelerated variants

This subsection discusses pure ACG variants for solving the N-SCO problem (1). More specifically, we discuss three methods, namely: the AG method proposed in [6], the NC-FISTA of [16], and its adaptive variant ADAP-NC-FISTA also described in [16]. The iteration-complexity of all three methods are analyzed under the assumption that \( \text{dom} \ h \) is bounded, but in practice all three methods can successfully solve many problems with unbounded \( \text{dom} \ h \).

AG is a direct extension of the ACG variant, based on the LLM rule and the constant choice of \( M_k \), to the N-SCO context. Clearly, AG performs two resolvent evaluations of \( h \) per iteration.

NC-FISTA requires as input a pair \((M, m)\) such that \( M > \bar{M} \) and \( M \geq m \geq \bar{m} \). It is an extension of the version of FISTA with \( y_{k+1} = y^a_{k+1} \) from the C-SCO to the N-SCO context, and it reduces to the latter one when \( m = \bar{m} = 0 \). More specifically, NC-FISTA sets \( y_{k+1} = y(\tilde{x}_k; M_k) \) where \( M_k = M + \kappa_0 m/(M a_k) \), and computes \( x_{k+1} \) as in (21) with \( A_k/a_k \) replaced by \((\kappa_0 M/m + 1)^{-1}(A_k/a_k) \) where \( \kappa_0 \) is a positive universal constant. In contrast to an iteration of the AG method, every iteration of NC-FISTA performs exactly one resolvent evaluation of \( h \).

One drawback of NC-FISTA is its required input pair \((M, m)\), which is usually hard to obtain or is often poorly estimated. On the other hand, ADAP-NC-FISTA remedies this drawback in that it only requires as input an arbitrary initial pair \((M_0, m_0)\) such that \( M_0 \geq m_0 > 0 \), which is dynamically updated by means of two separate backtracking search procedures.
3.3 Hybrid accelerated variants

This subsection discusses hybrid ACG variants for solving the N-SCO problem (1). More specifically, we discuss three methods, namely: a non-monotone variant as well as a monotone one both described in [13], which we refer to as NM-APG and M-APG, respectively, and UPFAG proposed in [7]. To the best of our knowledge, the convergence of these hybrid ACG variants is guaranteed due to the possibility of performing an extra unaccelerated composite gradient step. Whether their convergence can be shown without this optional step is an open question even for the case in which $\text{dom } h$ is bounded.

M-APG is exactly the instance of the ACG variant based on the FISTA rule in which $y_{k+1}$ is computed by means of (23) which, as already mentioned above, guarantees its monotonicity property due to the fact that $M^a_k$ is chosen as a good upper curvature of $f$ at $y_k$. NM-APG is a variant of M-APG, which either sets $y_{k+1} = y^a_{k+1}$ or computes $y_{k+1}$ as in (23) depending on whether or not, respectively, $y^a_{k+1}$ satisfies a key inequality, which ensures convergence of the method but not necessarily its monotonicity.

UPFAG is an ACG variant based on the AT rule in which the next iterate $y_{k+1}$ is chosen as in (23) except that $(M^a_k, M^{na}_k)$ is computed by line searches so that $M^a_k$ closely approximates a good curvature of $f$ at $\tilde{x}_k$ and $M^{na}_k$ satisfies a relaxed version of the descent condition (50) with $(M_k, \tilde{x}_k, y_k) = (M^{na}_k, y_k, y^{na}_{k+1})$.

4 Proof of Theorem 2.1

This section presents the proof of Theorem 2.1. We start with the following technical result, which assumes that all sequences start with $k = 0$.

Lemma 4.1. The following statements hold:

(a) the sequences $\{x_k\}$, $\{y_k\}$, $\{y^g_{k+1}\}$, $\{y^b_{k+1}\}$ and $\{\tilde{x}_k\}$ are all contained in $\text{dom } h$;

(b) for every $u \in \text{dom } h$ and $k \geq 0$, we have

$$A_k \|y_k - \tilde{x}_k\|^2 + a_k \|u - \tilde{x}_k\|^2 \leq a_k D^2;$$

(c) for every $k \geq 0$, $C_k \leq \bar{M}$ and $F_k \leq \bar{M}$, where

$$F_k := C(y_{k+1}; \tilde{x}_k)$$

and $C(\cdot; \cdot)$ is defined in (4);

(d) for every $k \geq 0$, we have

$$v_{k+1} \in \nabla f(y^g_{k+1}) + \partial h(y^g_{k+1}), \quad \|v_{k+1}\| \leq (M_k + C_k) \|y^g_{k+1} - \tilde{x}_k\|. \quad (25)$$

Proof: (a) The sequences $\{x_k\}$ and $\{y^g_{k+1}\}$ are contained in $\text{dom } h$ in view of (10), (2) and step 0 of AC-ACG. Hence, using step 0 of AC-ACG again, (15) and the convexity of $\text{dom } h$, we easily see by induction that $\{y_k\}$ and $\{y^b_{k+1}\}$ are contained in $\text{dom } h$. Finally, $\{\tilde{x}_k\} \subset \text{dom } h$ follows from the third identity in (9) and the convexity of $\text{dom } h$.

(b) First note that for every $A, a \in \mathbb{R}_+$ and $x, y \in \mathbb{R}^n$, we have

$$A\|y\|^2 + a\|x\|^2 = (A + a) \left\| \frac{Ay + ax}{A + a} \right\|^2 + \frac{Aa}{A + a} \|y - x\|^2.$$
Applying the above identity with $A = A_k$, $a = a_k$, $y = y_k - \bar{x}_k$ and $x = u - \bar{x}_k$, and using both the second and the third identities in (9), we have

\[
A_k \|y_k - \bar{x}_k\|^2 + a_k \|u - \bar{x}_k\|^2 = A_{k+1} \left| \frac{A_k y_k + a_k u}{A_{k+1}} - \bar{x}_k \right|^2 + \frac{A_k a_k}{A_{k+1}} \|y_k - u\|^2
\]

\[
= \frac{a_k}{A_{k+1}} \left( a_k \|u - x_k\|^2 + A_k \|u - y_k\|^2 \right) \leq a_k D^2
\]

where the inequality follows from Lemma 4.1(a), the definition of $D$ in (A3) and the second equality in (9).

(c) The conclusion follows from definitions of $C_k$, $F_k$ and $C(\cdot, \cdot)$ in (12), (24) and (4), respectively, and the fact that $M$ satisfies both the second inequality in (5) and (6).

(d) The inclusion in (25) follows from the fact $y_{k+1} = y(\bar{x}_k; M_k)$, the optimality condition of (2) and the definition of $v_{k+1}$ in (11). Moreover, the inequality in (25) follows from definitions of $C_k$ in (12) and $v_{k+1}$, and the triangle inequality.

The next result provides an important recursive formula involving a certain potential function $\eta_k$ and the quantity $\|y_{k+1} - \bar{x}_k\|$ that will later be related to the residual vector $\|v_{k+1}\|$ (see the proof of Lemma 4.3(a)).

**Lemma 4.2.** For every $k \geq 0$ and $u \in \text{dom } h$, we have

\[
\frac{M_k - F_k}{2} A_{k+1} \|y_{k+1} - \bar{x}_k\|^2 \leq \eta_k(u) - \eta_{k+1}(u) + \frac{1}{2} a_k D^2
\]

where $M_k$ is as in (14) and

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

**Proof:** Let $k \geq 0$ and $u \in \text{dom } h$ be given and define $\gamma_k(u) := \ell_f(u; \bar{x}_k) + h(u)$. Using the fact $x_{k+1}$ is an optimal solution of (10) and $\gamma_k$ is a convex function, the second and third identities in (9), and relations (15) and (16), we conclude that

\[
A_k \gamma_k(y_k) + a_k \gamma_k(u) + \frac{1}{2} \|u - x_k\|^2 - \frac{1}{2} \|u - x_{k+1}\|^2 \geq A_k \gamma_k(y_k) + a_k \gamma_k(x_{k+1}) + \frac{1}{2} \|x_{k+1} - x_k\|^2
\]

\[
\geq A_{k+1} \gamma_k(y_{k+1}) + \frac{1}{2} \frac{A_{k+1}^2}{a_k^2} \|y_{k+1} - \bar{x}_k\|^2
\]

\[
= A_{k+1} \left[ \gamma_k(y_{k+1}) + \frac{M_k - F_k}{2} \|y_{k+1} - \bar{x}_k\|^2 \right].
\]

Moreover, relations (2), (15) and (24), and the fact that $\{y_{k+1}^b\} \subset \text{dom } h$ imply that

\[
\gamma_k(y_{k+1}) + \frac{M_k}{2} \|y_{k+1} - \bar{x}_k\|^2 \geq \gamma_k(y_{k+1}) + \frac{M_k}{2} \|y_{k+1} - \bar{x}_k\|^2 = \phi(y_{k+1}) + \frac{M_k - F_k}{2} \|y_{k+1} - \bar{x}_k\|^2.
\]

Using the above two inequalities, the definition of $\eta_k$ in (26) and the first inequality in (5), we easily see that

\[
\frac{M_k - F_k}{2} A_{k+1} \|y_{k+1} - \bar{x}_k\|^2 - \eta_k(u) + \eta_{k+1}(u) \leq A_k (\gamma_k(y_k) - \phi(y_k)) + a_k (\gamma_k(u) - \phi(u))
\]

\[
\leq \frac{m}{2} \left( A_k \|y_k - \bar{x}_k\|^2 + a_k \|u - \bar{x}_k\|^2 \right),
\]
which, together with Lemma 4.1(b), then immediately implies the lemma.

For the purpose of stating the next results, we define the set of good and bad iterations as

$$\mathcal{G} := \{k \geq 0 : C_k \leq 0.9M_k\}, \quad \mathcal{B} := \{k \geq 0 : C_k > 0.9M_k\},$$  \hfill (27)

respectively. The following result specializes the bound derived in Lemma 4.2 to the two exclusive cases in which $k \in \mathcal{G}$ and $k \in \mathcal{B}$. More specifically, it derives a controllable bound on the residual vector $v_{k+1}$ and the potential function difference $\eta_{k+1}(u) - \eta_k(u)$ in the good iterations and a controllable bound only on $\eta_{k+1}(u) - \eta_k(u)$ in the bad iterations.

**Lemma 4.3.** The following statements hold for every $u \in \text{dom } h$ and $k \geq 0$:

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

$$\frac{(M_k - F_k)A_{k+1}}{72.2M_k} \|v_{k+1}\|^2 \leq \eta_k(u) - \eta_{k+1}(u) + \frac{1}{2} \bar{m} \eta_k D^2;$$ \hfill (28)

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

$$0 \leq \eta_k(u) - \eta_{k+1}(u) + \frac{1}{2} \bar{m} \eta_k D^2 + \frac{1 - \gamma}{2\gamma} D^2.$$

**Proof:** (a) Let $k \in \mathcal{G}$ be given and note that (27) and (15) imply that $0.9M_k \geq C_k$ and $y_{k+1} = y^b_{k+1}$ where $y^{g}_{k+1} = y(\tilde{x}_k; M_k)$ is as in (2). Hence, using the inequality in (25), and the definitions of $C_k$ and $F_k$ in (12) and (24), respectively, we conclude that $\|v_{k+1}\| \leq 1.9M_k \|y_{k+1} - \tilde{x}_k\|$ and $F_k \leq C_k \leq 0.9M_k$. The latter two conclusions and Lemma 4.2 then immediately imply that (28) holds.

(b) Let $k \in \mathcal{B}$ be given and note that (15) and (27) imply that $y_{k+1} = y^b_{k+1}$. Using the latter observation, Lemma 4.2, Lemma 4.1(c), the last equality in (9), and relation (16), we conclude that

$$\eta_k(u) - \eta_{k+1}(u) + \frac{1}{2} \bar{m} \eta_k D^2 \geq \frac{(M_k - F_k)A_{k+1}}{2} \|y^b_{k+1} - \tilde{x}_k\|^2$$

$$= \frac{(M_k - F_k)A_{k+1}}{2} \|A_k y_k + a_k x_{k+1} - A_k y_k + a_k x_{k+1}\|^2$$

$$= \frac{(M_k - F_k)A_{k+1}}{2A_{k+1}} \|x_{k+1} - x_k\|^2 = \frac{1}{2} \left(1 - \frac{F_k}{M_k}\right) \|x_{k+1} - x_k\|^2$$

and hence that (29) holds in view of Lemma 4.1(a) and (A3).

As a consequence, the next lemma provides the result of the summation of inequalities for $k \in \mathcal{G}$ and $k \in \mathcal{B}$ in Lemma 4.3.

**Lemma 4.4.** For every $u \in \text{dom } h$ and $k \geq 1$, we have

$$\left(\frac{1}{36.1} \sum_{i \in \mathcal{G}_k} \frac{A_{i+1}}{M_i}\right) \min_{1 \leq i \leq k} \|v_i\|^2 \leq \|u - x_0\|^2 - 2\eta_k(u) + \bar{m} D^2 A_k + \frac{1 - \gamma}{\gamma} D^2 \sum_{i \in \mathcal{B}_k} A_k |\mathcal{B}_k|,$$

where $\mathcal{G}_k$ and $\mathcal{B}_k$ are defined as

$$\mathcal{G}_k := \{i \in \mathcal{G} : i \leq k - 1\}, \quad \mathcal{B}_k := \{i \in \mathcal{B} : i \leq k - 1\}.$$ \hfill (31)
Proof: First, note that

\[ \sum_{i \in G_k} \frac{A_i}{M_i} \|v_i+1\|^2 \geq \left( \sum_{i \in G_k} \frac{A_i}{M_i} \right) \min_{i \in G_k} \|v_i+1\|^2 \geq \left( \sum_{i \in G_k} \frac{A_i}{M_i} \right) \min_{1 \leq i \leq k} \|v_i\|^2. \]

The conclusion follows by adding (28) and (29) both with \( k = i \) as \( i \) varies in \( G_k \) and \( B_k \), respectively, and using the above inequality, the definition of \( \eta_k \) in (26), and the facts that \( A_k = A_0 + \sum_{i=0}^{k-1} a_i \) and \( A_0 = 0 \), which are due to (9) and step 0 of the AC-ACG method, respectively.

Note that the left hand side of (30) is actually zero when \( G_k = \emptyset \), and hence (30) is meaningless in this case. The result below, which plays a major role in our analysis, uses for the first time the fact that \( M_k \) is chosen as in (14) and shows that \( G_k \) is nonempty and well-populated. This fact in turn implies that the term inside the parenthesis in the left hand side of (30) is sufficiently large (see Lemma 4.8 below). The proof of Theorem 2.1 will then follow by combining these observations.

**Lemma 4.5.** For every \( k \geq 12 \), \( |B_k| \leq k/3 \) where \( B_k \) is as defined in (31).

**Proof:** Let \( k \geq 12 \) be given and, for the sake of this proof, define \( C_{avg} = 0 \). In view of (14) and the definition of \( B_k \) in (31), it follows that for every \( i \in B_k \),

\[ \frac{\alpha}{0.9} C_i > \alpha M_i \geq C_{avg}^{i-1}, \]

and hence that

\[ \frac{\alpha}{0.9} \sum_{i \in B_k} C_i > \sum_{i \in B_k} C_{avg}^{i-1}. \tag{32} \]

Using Lemma 4.1(c) and the facts that \( C_i > 0.9M_i \) for every \( i \in B_k \) and that \( M_i \geq \gamma M \geq \gamma \bar{M} \) (see (14) and step 0 of the AC-ACG method) for every \( i \geq 0 \), we have

\[ 0.9\gamma \bar{M} \leq C_i \leq \bar{M} \quad i \in B_k. \tag{33} \]

Let \( l := |B_k| \) and let \( i_1 < \cdots < i_l \) denote the indices in \( B_k \). Clearly, in view of (13) and the fact that \( i_j \leq k \) for every \( j = 1, \ldots, l \), we have

\[ C_{avg}^{i_{j-1}} \geq 0, \quad C_{avg}^{i_j} \geq \frac{1}{l} C_i, \quad \cdots, \quad C_{avg}^{i_{l-1}} \geq \frac{1}{l} (C_{i_1} + \cdots + C_{i_{l-1}}). \]

Summing these inequalities, we obtain

\[ \sum_{i \in B_k} C_{avg}^{i-1} \geq \frac{1}{l} \sum_{j=1}^{l} (l-j) C_{i_j} \geq \frac{1}{k} \sum_{j=1}^{\lceil l/2 \rceil} (l-j) C_{i_j} \geq \frac{1}{k} \left \lfloor \frac{l}{2} \right \rfloor \sum_{j=1}^{\lceil l/2 \rceil} C_{i_j}. \]

Combining (32) and the last inequality, we then conclude that

\[ \frac{\alpha(S_1 + S_2)}{0.9} \geq \frac{1}{k} \left \lfloor \frac{l}{2} \right \rfloor S_1 \]

where

\[ S_1 := \sum_{j=1}^{\lceil l/2 \rceil} C_{i_j}, \quad S_2 := \sum_{j=\lceil l/2 \rceil + 1}^{l} C_{i_j}. \tag{34} \]
Since (33) and the above definitions of $S_1$ and $S_2$ immediately imply that $S_2/S_1 \leq 1/(0.9\gamma)$, we then conclude from the above inequality that

$$|B_k| = l \leq \left( \frac{2\alpha k}{0.9} \right) \left( 1 + \frac{S_2}{S_1} \right) + 1 \leq \left( \frac{2\alpha k}{0.9} \right) \left( 1 + \frac{1}{0.9\gamma} \right) + 1$$

(35)

and hence that $|B_k| \leq k/4 + 1 \leq k/3$ in view of the definition of $\alpha$ in (8) and the fact that $k \geq 12$. The last conclusion of the lemma follows straightforwardly from the first one.

We now make some remarks about choosing $\alpha$ more aggressively, i.e., larger than the value in (8) (recall the discussion in the second paragraph following the AC-ACG method). First, in view of their definitions in (34), the quantities $S_1$ and $S_2$ are actually quantities that depend on the iteration index $k$ and hence should have been denoted by $S_k^1$ and $S_k^2$. Second, it follows from the first inequality in (35) that

$$|B_k| \leq \left( \frac{2\alpha k}{0.9} \right) (1 + \bar{\gamma}_k) + 1$$

where $\bar{\gamma}_k := S_k^2 / S_k^1$. Third, we have used in the proof of Lemma 4.5 that $\bar{\gamma}_k$ is bounded above by $1/(0.9\gamma)$, which is a very conservative bound for this quantity. In practice though, $\bar{\gamma}_k$ behaves as $O(1)$ (if not for all $k$, then at least for a substantial number of iterations). Fourth, in order to conclude that $|B_k| \leq k/3$ as in the proof of Lemma 4.5, it suffices to choose

$$\alpha = \frac{0.9}{8(1 + \bar{\gamma})}$$

where $\bar{\gamma} := \max\{\bar{\gamma}_k : k \geq 1\}$. Observe that the above choice of $\alpha$ is $\Theta(1)$ if $\bar{\gamma}$ behaves as $O(1)$.

Before presenting Lemma 4.8, we first state two technical results about the sequences $\{M_k\}$ and $\{A_k\}$.

**Lemma 4.6.** For every $1 \leq i < k$, we have

$$M_k \geq \frac{i}{k} M_i.$$  

**Proof:** From the definition of $C_k^{avg}$ in (13), for every $i = 1, \ldots, k - 1$, we have

$$kC_{k-1}^{avg} - IC_{i-1}^{avg} = C_i + \ldots + C_{k-1}$$

and thus

$$\frac{C_{k-1}^{avg}}{C_{i-1}^{avg}} = \frac{i}{k} + \frac{C_i + \ldots + C_{k-1}}{kC_{i-1}^{avg}} \geq \frac{i}{k}.$$  

The conclusion follows from the above inequality, the definition of $M_k$ in (14) and the fact that $\max\{a, c\} \geq \max\{b, d\}$ for $a, b, c, d \in \mathbb{R}$ such that $a \geq b$ and $c \geq d$.

The following result describes bounds on $A_k$ in terms of the first $k$ elements of the sequence $\{M_i\}$ and also in terms of $M_k$ alone.

**Lemma 4.7.** Consider the sequences $\{A_k\}$ and $\{M_i\}$ defined in (9) and (14), respectively. For every $k \geq 12$, we have

$$A_k \leq \left( \sum_{i=0}^{k-1} \frac{1}{\sqrt{M_i}} \right)^2 \leq k \sum_{i=0}^{k-1} \frac{1}{M_i} \leq k^2 \frac{\theta_k}{M_k}$$

(36)
and
\[ A_k \geq \frac{1}{4} \left( \sum_{i=0}^{k-1} \frac{1}{\sqrt{M_i}} \right)^2 \geq \frac{k^2}{12M_k} \tag{37} \]

where \( \theta_k \) is as in (18).

**Proof:** We first establish the inequalities in (36). Using the first two identities in (9) and the fact
\[ \sqrt{b_1 + b_2} \leq \sqrt{b_1} + \sqrt{b_2} \]
for any \( b_1, b_2 \in \mathbb{R}_+ \), we conclude that for any \( i \geq 0 \),
\[ \sqrt{A_{i+1}} = \left( A_i + \frac{1 + \sqrt{1 + 4M_iA_i}}{2M_i} \right)^{1/2} \leq \left( A_i + \frac{1 + \sqrt{M_iA_i}}{M_i} \right)^{1/2} \leq \sqrt{A_i} + \frac{1}{\sqrt{M_i}}. \]

Now, the first inequality in (36) follows by summing the above inequality from \( i = 0 \) to \( k-1 \) and using the assumption that \( A_0 = 0 \). Moreover, the second and third inequalities in (36) follow straightforwardly from the Cauchy-Schwarz inequality and the definition of \( \theta_k \) in (18), respectively.

We now establish the inequalities in (37). Using the first two identities in (9), we have
\[ \sqrt{A_{i+1}} = \left( A_i + \frac{1 + \sqrt{1 + 4M_iA_i}}{2M_i} \right)^{1/2} \geq \left( A_i + \frac{1 + 2\sqrt{M_iA_i}}{2M_i} \right)^{1/2} \geq \sqrt{A_i} + \frac{1}{2\sqrt{M_i}}. \]

The first inequality in (37) now follows by summing the above inequality from \( i = 0 \) to \( k-1 \) and using the assumption that \( A_0 = 0 \). For every \( k \geq 12 \), we have
\[ \sum_{i=0}^{k-1} i \geq \int_0^{k-1} \sqrt{x} \, dx = \frac{2}{3} (k-1)^{3/2} \geq \frac{2}{3} \left( \frac{11}{12} k \right)^{3/2} \geq 0.58k^{3/2}, \]
which, together with Lemma 4.6, then implies that
\[ \sum_{i=1}^{k-1} \frac{1}{\sqrt{M_i}} \geq \frac{1}{\sqrt{kM_k}} \sum_{i=1}^{k-1} \sqrt{i} \geq \frac{0.58k}{\sqrt{M_k}}. \]

The second inequality in (37) now follows immediately from the one above.

The following result provides a lower bound on the term inside the parenthesis of the left hand side of (30).

**Lemma 4.8.** For every \( k \geq 12 \), we have
\[ \sum_{i \in \mathcal{G}_k} A_{i+1} \frac{i}{M_i} \geq \frac{k^3}{3402M_k^2}. \]

**Proof:** Let \( k \geq 12 \) be given and define
\[ \tilde{\mathcal{G}}_k := \{ i \in \mathcal{G}_k : i \geq \lfloor k/3 \rfloor \}, \quad \tilde{\mathcal{B}}_k := \{ i \in \mathcal{B}_k : i \geq \lfloor k/3 \rfloor \}. \tag{38} \]

Using Lemma 4.6, the facts that \( \tilde{\mathcal{G}}_k \subset \mathcal{G}_k \), \( \{ A_k \} \) is strictly increasing, and \( i/k \geq 2/7 \) for any \( i \in \tilde{\mathcal{G}}_k \) and \( k \geq 12 \), and inequality (37), we conclude that
\[ \sum_{i \in \mathcal{G}_k} A_{i+1} \frac{i}{M_i} \geq \sum_{i \in \mathcal{G}_k} i A_{i+1} \frac{i}{kM_k} \geq \sum_{i \in \tilde{\mathcal{G}}_k} i A_{i+1} \frac{i}{kM_k} \geq \frac{2|\tilde{\mathcal{G}}_k|}{7M_k} A_{\lfloor k/3 \rfloor + 1} \]
\[ \geq \frac{2|\tilde{\mathcal{G}}_k|}{7M_k} A_{\lfloor k/3 \rfloor} \geq \frac{|\tilde{\mathcal{G}}_k|((k/3)!)^2}{42M_kM_{\lfloor k/3 \rfloor}} \geq \frac{|\tilde{\mathcal{G}}_k|k^2}{378M_kM_{\lfloor k/3 \rfloor}}. \tag{39} \]
On the other hand, Lemma 4.6 with $i = \lceil k/3 \rceil$ implies that
\[ M_k \geq \frac{\lceil k/3 \rceil}{k} M_{\lceil k/3 \rceil} \geq \frac{1}{3} M_{\lceil k/3 \rceil}. \]

Moreover, the definition of $\tilde{G}_k$ in (38), the fact that $\tilde{B}_k \subset B_k$ and Lemma 4.5 imply that
\[ |\tilde{G}_k| = k - \lfloor k/3 \rfloor - |\tilde{B}_k| \geq k - \lfloor k/3 \rfloor - |B_k| \geq k/3. \]

The conclusion of the lemma now follows by combining (39) with the last two observations.

We are now ready to prove the main result of our paper.

**Proof of Theorem 2.1:** (a) The conclusion immediately follows from Lemma 4.1(d).

(b) Letting $x_\star \in X_\star$ be given and noting that $\eta_k(x_\star) \geq 0$ in view of the definition of $\eta_k$ in (26) and using the above inequality, Lemma 4.4 with $u = x_\star$, Lemma 4.5 and relation (36), we conclude that
\[
\left( \frac{1}{36.1} \sum_{i \in \tilde{G}_k} \frac{A_{i+1}}{M_i} \right) \min_{1 \leq i \leq k} \|v_i\|^2 \leq \|x_0 - x_\star\|^2 + \bar{m}D^2A_k + \frac{1 - \gamma}{\gamma} D^2|B_k|
\leq D^2 + \bar{m}D^2A_k + \frac{(1 - \gamma)D^2k}{3\gamma}
\leq D^2 + \frac{\bar{m}D^2k^2\theta_k}{M_k} + \frac{(1 - \gamma)D^2k}{3\gamma}.
\]

Statement b) of the theorem now follows by combining the above inequality and Lemma 4.8.

5 Numerical results

This section presents computational results to illustrate the performance of two variants of the AC-ACG method against five other state-of-the-art algorithms on a collection of nonconvex optimization problems that are either in the form of or can be easily reformulated into (1). It contains five subsections, with each one reporting computational results on one of following classes of nonconvex optimization problems: (a) quadratic programming (Subsection 5.1); (b) support vector machine (SVM, Subsection 5.2); (c) sparse PCA (Subsection 5.3); (d) matrix completion (Subsection 5.4); and (e) nonnegative matrix factorization (NMF, Subsection 5.5). Note that sparse PCA and NMF are problems for which $\text{dom } h$ is unbounded.

We start by describing the two AC-ACG variants considered in our computational experiments, both of which do not impose the restrictive condition (8) on the choice of $\alpha$ and $\gamma$. The first variant, which we refer to as ACT throughout this section, preserves all steps in the AC-ACG method except that $\gamma$ and $\alpha$ are provided as input by the user without necessarily satisfying (8). In our implementation, we set $\gamma = 0.01$ for every problem class listed above but the one in (b) for which $\gamma$ is set to 0.002. The latter choice of $\gamma$ prevents the percentage of good iterations from being 100% all the time and instead keeps it within a range of about 65% to 75% (see Subsection 5.2). The choice of the scalar $\alpha$ varies per problem class and is described in each one of the subsections below. The second variant, referred to as AC throughout this section, sets $M_0 = 0.01M$, and computes $M_{k+1}$ as in (14) with $\gamma = 10^{-6}$ and $C_k$ as
\[
C_k = \max\{C(y_{k+1}^0; \hat{x}_k), 0\}
\]
where $C(\cdot,\cdot)$ is defined in (4). Our implementation of AC sets $\alpha$ to values that depend on the problem class under consideration and are specified in the subsections below. Clearly, among the two variants described above, ACT is the closest to AC-ACG.

We compare the two variants of AC-ACG with five other methods, namely: (i) the AG method proposed in [6]; (ii) the NC-FISTA of [16]; (iii) the ADAP-NC-FISTA also described in [16]; (iv) the NM-APG method proposed in [13]; and (v) the UPFAG method in [7]. We remark that methods (i)-(iii) are the three pure ACG variants that have been outlined in Subsection 3.2 and methods (iv) and (v) are two among the three hybrid ACG variants that have been discussed in Subsection 3.3. For the sake of simplicity, we use the abbreviations NM, UP, NC and AD to refer to the NM-APG method proposed in [13]; (ii) the NC-FISTA of [16]; (iii) the ADAP-NC-FISTA also described in [16]; (iv) the two variants described above, ACT is the closest to AC-ACG.

All seven methods terminate with a pair $(z,v)$ satisfying

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

where $\hat{\rho} = 5 \times 10^{-4}$ in the matrix completion problem and $\hat{\rho} = 10^{-7}$ in all the other problems. All the computational results were obtained using MATLAB R2017b on a MacBook Pro with a quad-core Intel Core i7 processor and 16 GB of memory.

### 5.1 Quadratic programming

This subsection discusses the performance of the AC-ACG method for solving a class of quadratic programming problems.

More specifically, it considers the problem

$$\min \left\{ f(Z) := -\frac{\alpha_1}{2}\|DB(Z)\|^2 + \frac{\alpha_2}{2}\|A(Z) - b\|^2 : Z \in P_n \right\}$$

(41)
where \((\alpha_1, \alpha_2) \in \mathbb{R}_+^2, b \in \mathbb{R}^l\) is a vector with entries sampled from the uniform distribution \(U[0, 1]\), \(D \in \mathbb{R}^{n \times n}\) is a diagonal matrix whose diagonal entries are sampled from the discrete uniform distribution \(U\{1, 1000\}\), \(P_n := \{Z \in S^n_+ : \text{tr}(Z) = 1\}\) denotes the spectraplex, and \(A : S^n_+ \rightarrow \mathbb{R}^l\) and \(B : S^n_+ \rightarrow \mathbb{R}^n\) are linear operators given by

\[
[A(Z)]_i = \langle A_i, Z \rangle_F \quad \forall 1 \leq i \leq l,
\]

\[
[B(Z)]_j = \langle B_j, Z \rangle_F \quad \forall 1 \leq j \leq n,
\]

with \(A_i \in S^n_+\) and \(B_j \in S^n_+\) all being sparse matrices having the same density (i.e., percentage of nonzeros) \(d\) and nonzero entries uniformly sampled from \([0, 1]\).

The quadratic programming problem (41) is an instance of (1) where \(h\) is the indicator function of the spectraplex \(P_n\). For chosen curvature pairs \((M,m) \in \mathbb{R}_+^2\), the scalars \(\alpha_1\) and \(\alpha_2\) are chosen so that \(\lambda_{\max}(\nabla^2 f) = M\) and \(\lambda_{\min}(\nabla^2 f) = -m\) where \(\lambda_{\max}(\cdot)\) and \(\lambda_{\min}(\cdot)\) denote the largest and smallest eigenvalue functions, respectively.

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 \(P_n\). The parameter \(\alpha\) is set to 1 in AC and 0.5 in ACT.

Numerical results for the seven methods are given in Tables 1, 3 and 5, with each table addressing a collection of instances with the same dimension pair \((l,n)\) and density \(d\). Specifically, each row of Tables 1, 3 and 5 corresponds to an instance of (41), their first column specifies the pair \((M,m)\) for the corresponding instance, their second to sixth (resp., seventh to eleventh) columns provide numbers of iterations (resp., running times) for the seven methods. The best objective function values obtained by all seven methods are not reported since they are essentially the same on all instances. The number of resolvent evaluations is 1 in NC, 2 in AG, AC and ACT, 1 or 2 in NM, 1 on average in AD, and 3 on average in UP. The bold numbers highlight the method that has the best performance in an instance of the problem.

Some statistic measures for AC and ACT to solve the instances in Tables 1, 3 and 5 are given in Tables 2, 4 and 6, respectively. The first column in these tables is the same as that of Tables 1, 3 and 5, the second (resp., fifth) column provides the maximum of all observed curvatures \(C_k\) in AC (resp. ACT), the third (resp., sixth) column provides the average of all observed curvatures \(C_k\) in AC (resp. ACT), and the fourth (resp. seventh) column gives the percentage of good iterations (see (27)) in AC (resp. ACT).

In Tables 1-2, the density \(d = 2.5\%\) and the dimension pair \((l,n) = (50, 200)\).

<table>
<thead>
<tr>
<th>((M,m))</th>
<th>Iteration Count</th>
<th>Running Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>((10^6, 10^6))</td>
<td>AG 46 NM 80 UP 9 NC/AD 33/12 ACT/AC 23/8</td>
<td>AG 1.6 NM 2.1 UP 0.7 NC/AD 0.8/0.7 ACT/AC 1.4/0.6</td>
</tr>
<tr>
<td>((10^6, 10^5))</td>
<td>3089 6242 2633 3384/2206 1099/883</td>
<td>130 191 261 94/89 57/39</td>
</tr>
<tr>
<td>((10^6, 10^4))</td>
<td>5400 10404 7203 1236/2591 1820/1760</td>
<td>188 328 705 30/104 109/73</td>
</tr>
<tr>
<td>((10^6, 10^3))</td>
<td>4621 11053 5429 5139/2637 1712/1508</td>
<td>176 360 540 122/109 118/68</td>
</tr>
<tr>
<td>((10^6, 10^2))</td>
<td>4476 11271 6891 11838/2639 1610/1472</td>
<td>176 312 653 283/116 103/65</td>
</tr>
<tr>
<td>((10^6, 10))</td>
<td>4461 11253 6479 14851/2640 1599/1485</td>
<td>171 311 613 362/116 155/66</td>
</tr>
</tbody>
</table>

Table 1: Numerical results for AG, NM, UP, NC, AD, ACT and AC
In Table 3-4, the density $d = 0.5\%$ and the dimension pair $(l, n) = (50, 400)$.

### Table 3: Numerical results for AG, NM, UP, NC, AD, ACT and AC

<table>
<thead>
<tr>
<th>$(M, m)$</th>
<th>$l/n$</th>
<th>AG</th>
<th>NM</th>
<th>UP</th>
<th>NC/AD</th>
<th>ACT/AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(10^6, 10^6)$</td>
<td>$(50, 400)$</td>
<td>44</td>
<td>75</td>
<td>10</td>
<td>33/12</td>
<td>17/8</td>
</tr>
<tr>
<td>$(10^6, 10^5)$</td>
<td>$(40, 800)$</td>
<td>1411</td>
<td>3151</td>
<td>56</td>
<td>610/530</td>
<td>403/131</td>
</tr>
<tr>
<td>$(10^6, 10^4)$</td>
<td>$(30, 1200)$</td>
<td>1963</td>
<td>5071</td>
<td>105</td>
<td>1212/868</td>
<td>599/237</td>
</tr>
<tr>
<td>$(10^6, 10^3)$</td>
<td>$(20, 2400)$</td>
<td>1935</td>
<td>5172</td>
<td>115</td>
<td>4415/900</td>
<td>564/245</td>
</tr>
<tr>
<td>$(10^6, 10^2)$</td>
<td>$(10, 4800)$</td>
<td>1934</td>
<td>5056</td>
<td>113</td>
<td>7527/904</td>
<td>561/246</td>
</tr>
</tbody>
</table>

#### Table 4: AC and ACT statistics

In Table 5-6, the density $d = 0.1\%$ and the dimension pair $(l, n) = (50, 800)$.

<table>
<thead>
<tr>
<th>$(M, m)$</th>
<th>$l/n$</th>
<th>AC</th>
<th>ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(10^6, 10^6)$</td>
<td>$(50, 800)$</td>
<td>Max</td>
<td>Avg</td>
</tr>
<tr>
<td>$(10^6, 10^5)$</td>
<td>$(40, 1600)$</td>
<td>2.40E5</td>
<td>3.22E4</td>
</tr>
<tr>
<td>$(10^6, 10^4)$</td>
<td>$(30, 3200)$</td>
<td>1.53E5</td>
<td>1.98E4</td>
</tr>
<tr>
<td>$(10^6, 10^3)$</td>
<td>$(20, 6400)$</td>
<td>2.03E5</td>
<td>2.50E4</td>
</tr>
<tr>
<td>$(10^6, 10^2)$</td>
<td>$(10, 12800)$</td>
<td>2.07E5</td>
<td>2.55E4</td>
</tr>
<tr>
<td>$(10^6, 10)$</td>
<td>$(5, 25600)$</td>
<td>2.08E5</td>
<td>2.51E4</td>
</tr>
</tbody>
</table>

#### Table 5: Numerical results for AG, NM, UP, NC, AD, ACT and AC
Table 6: AC and ACT statistics

<table>
<thead>
<tr>
<th>((M, m))</th>
<th>AC</th>
<th>ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>((10^6, 10^6))</td>
<td>1.28E5</td>
<td>3.65E5</td>
</tr>
<tr>
<td></td>
<td>1.70E4 88%</td>
<td>5.37E4 94%</td>
</tr>
<tr>
<td>((10^6, 10^5))</td>
<td>1.80E4</td>
<td>1.78E5</td>
</tr>
<tr>
<td></td>
<td>2.84E3 86%</td>
<td>2.66E4 96%</td>
</tr>
<tr>
<td>((10^6, 10^4))</td>
<td>3.26E4</td>
<td>1.78E5</td>
</tr>
<tr>
<td></td>
<td>3.89E3 91%</td>
<td>2.99E4 98%</td>
</tr>
<tr>
<td>((10^6, 10^3))</td>
<td>3.41E4</td>
<td>1.78E5</td>
</tr>
<tr>
<td></td>
<td>3.73E3 92%</td>
<td>2.64E4 98%</td>
</tr>
<tr>
<td>((10^6, 10^2))</td>
<td>3.42E4</td>
<td>1.78E5</td>
</tr>
<tr>
<td></td>
<td>3.75E3 92%</td>
<td>2.58E4 98%</td>
</tr>
<tr>
<td>((10^6, 10))</td>
<td>3.43E4</td>
<td>1.78E5</td>
</tr>
<tr>
<td></td>
<td>3.75E3 92%</td>
<td>2.57E4 98%</td>
</tr>
</tbody>
</table>

In summary, computational results demonstrate that: i) the computed average curvature of AC is small compared with \(M\) and the computed maximum curvature; ii) the percentage of good iterations of AC lies in a suitable range; and iii) AC has the best performance in terms of running time.

5.2 Support Vector Machine

This subsection presents the performance of AC-ACG for solving a support vector machine problem. Given data points \(\{(x_i, y_i)\}_{i=1}^p\), where \(x_i \in \mathbb{R}^n\) is a feature vector and \(y_i \in \{-1, 1\}\) denotes the corresponding label, we consider the SVM problem defined as

\[
\min_{z \in \mathbb{R}^n} \frac{1}{p} \sum_{i=1}^p \ell(x_i, y_i; z) + \frac{\lambda}{2} \|z\|^2 + I_{B_r}(z)
\]

for some \(\lambda, r > 0\), where \(\ell(x_i, y_i; \cdot) = 1 - \tanh(y_i \langle \cdot, x_i \rangle)\) is a nonconvex sigmoid loss function and \(I_{B_r}(\cdot)\) is the indicator function of the ball \(B_r := \{z \in \mathbb{R}^n : \|z\| \leq r\}\). The SVM problem (42) is an instance of nonconvex SCO problems (1) where

\[
f(z) = \frac{1}{p} \sum_{i=1}^p \ell(x_i, y_i; z) + \frac{\lambda}{2} \|z\|^2, \quad h(z) = I_{B_r}(z).
\]

Clearly, \(f\) is differentiable everywhere and its gradient is \(M\)-Lipschitz continuous where

\[
M = \frac{1}{p} \sum_{i=1}^p L_i + \lambda, \quad L_i = \frac{4\sqrt{3}}{9} \|x_i\|^2 \quad \forall i = 1, \ldots, p.
\]

Since no sharper \(m < M\) satisfying the first inequality in (5) is known, we simply set \(m = M\).

We generate synthetic data sets as follows: for each data point \((x_i, y_i)\), \(x_i\) is drawn from the uniform distribution on \([0, 1]^n\) and is sparse with 5% nonzero components, and \(y_i = \text{sign}(\langle \bar{z}, x_i \rangle)\) for some \(\bar{z} \in B_r\). We consider four different problem sizes \((n, p)\), i.e., \((1000, 500)\), \((2000, 1000)\), \((3000, 1000)\) and \((4000, 500)\). We set \(\lambda = 1/p\) and \(r = 50\).

We start all seven methods from the same initial point \(z_0\) that is chosen randomly from the uniform distribution within the ball \(B_r\). The parameter \(\alpha\) is set to 0.5 in both AC and ACT.

Numerical results of the seven methods are given in Table 7 and some statistic measures of AC and ACT are given in Table 8. The explanation of their columns excluding the first one is the same as those of Tables 1-6 (see the two paragraphs preceding Table 1). Their first columns differ from those of Tables 1-6 in that they only list the value of \(M\) computed according to (43). The best objective function values obtained by all seven methods are not reported since they are essentially
the same on all instances. The number of resolvent evaluations is 1 in NC, 2 in AG, AC and ACT, 1 or 2 in NM, 1 on average in AD, and 3 on average in UP. The bold numbers highlight the method that has the best performance in an instance of the problem. The numbers marked with * indicate that the maximum number of iterations has been reached.

<table>
<thead>
<tr>
<th>$M$</th>
<th>Iteration Count</th>
<th>Running Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AG  NM  UP  NC/AD  ACT/AC</td>
<td>AG  NM  UP  NC/AD  ACT/AC</td>
</tr>
<tr>
<td>13</td>
<td>37384 42532 130 42533/12274 583/546</td>
<td>639 649 8 233/188 9/6</td>
</tr>
<tr>
<td>25</td>
<td>112562 123551 278 174845/21127 1017/1131</td>
<td>4419 4486 39 5833/1836 93/60</td>
</tr>
<tr>
<td>38</td>
<td>155503 163197 401 500000*/71991 1208/1032</td>
<td>12636 12101 97 26258*/8957 168/95</td>
</tr>
<tr>
<td>50</td>
<td>79752 79064 247 172535/12450 730/615</td>
<td>4406 5264 44 5503/1033 65/39</td>
</tr>
</tbody>
</table>

Table 7: Numerical results for AG, NM, UP, NC, AD, ACT and AC

<table>
<thead>
<tr>
<th>$M$</th>
<th>AC</th>
<th>ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Avg</td>
</tr>
<tr>
<td>13</td>
<td>0.25</td>
<td>0.05</td>
</tr>
<tr>
<td>25</td>
<td>0.47</td>
<td>0.06</td>
</tr>
<tr>
<td>38</td>
<td>0.34</td>
<td>0.07</td>
</tr>
<tr>
<td>50</td>
<td>0.18</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Table 8: AC and ACT statistics

In summary, computational results demonstrate that: i) the computed average curvature of AC is small compared with $M$ and the computed maximum curvature; ii) the percentage of good iterations of AC lies in a suitable range; and iii) AC is either the best method or close to the best one in terms of running time.

5.3 Sparse PCA

This subsection considers a penalized version of the sparse PCA problem, namely,

$$
\min_{X, Y \in \mathbb{R}^{p \times p}} -\langle \hat{\Sigma}, X \rangle_F + \frac{\mu}{2} \|X\|_F^2 + Q_{\lambda, b}(Y) + \lambda \|Y\|_1 + \frac{\beta}{2} \|X - Y\|_F^2 + I_{F^r}(X),
$$

(44)

where the dataset consists of an empirical covariance matrix $\hat{\Sigma} \in \mathbb{R}^{p \times p}$, two regularization parameters $\mu > 0$ and $\lambda > 0$, a penalty parameter $\beta > 0$ and two scalars $b > 0$ and $r \in \mathbb{N}_+$. Moreover, $\|\cdot\|_1$ and $Q_{\lambda, b}(\cdot)$ are the matrix 1-norm and a decomposable nonconvex penalty function defined as

$$
\|Y\|_1 := \sum_{i,j=1}^p |Y_{ij}|, \quad Q_{\lambda, b}(X) := \sum_{i,j=1}^p q_{\lambda, b}(X_{ij})
$$

where

$$
q_{\lambda, b}(t) := \begin{cases} 
-\frac{t^2}{2b}, & \text{if } |t| \leq b\lambda; \\
\frac{b^2 \lambda^2}{2} - \lambda|t|, & \text{otherwise}
\end{cases}
$$

and $I_{F^r}(\cdot)$ is the indicator function of the Fantope

$$
F^r := \{X \in S^n : 0 \preceq X \preceq I \text{ and } \text{tr}(X) = r\}.
$$
Clearly, problem (44) is an instance of the nonconvex SCO problem (1) where
\[ f(X,Y) = -\langle \hat{\Sigma}, X \rangle_F + \frac{\mu}{2} \|X\|^2_F + Q_{\lambda,b}(Y) + \frac{\beta}{2} \|X-Y\|^2_F, \quad h(X,Y) = I_{F^r}(X) + \lambda \|Y\|_1. \]
Moreover, it is easy to see that the pair
\[ (M,m) = \left( \max \left\{ \mu + 2\beta, \frac{1}{b} \right\}, \frac{1}{b} \right) \]  
(45)
satisfies assumption (A2).

We discuss how synthetic datasets are generated. Let \( \Sigma \in \mathbb{R}^{p \times p} \) be an unknown covariance matrix and \( X^* \) be the projection matrix onto the \( r \)-dimensional principal subspace of \( \Sigma \). In the sparse PCA problem, we seek an \( s \)-sparse approximation \( X \) of \( X^* \) in the sense that \( \|\text{diag}(X)\|_0 \leq s \), where \( s \in \mathbb{N}_+ \). We generate four datasets by designing four covariance matrices \( \Sigma \) as described in [9] and list all required parameters in Table 9. For each covariance matrix \( \Sigma \), we sample \( n = 80 \) i.i.d. observations from the normal distribution \( \mathcal{N}(0, \Sigma) \) and then calculate the sample covariance matrix \( \hat{\Sigma} \).

<table>
<thead>
<tr>
<th>dataset</th>
<th>( s )</th>
<th>( r )</th>
<th>( p )</th>
<th>( b )</th>
<th>( \beta )</th>
<th>( \mu )</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>10</td>
<td>5</td>
<td>1200</td>
<td>3</td>
<td>0.33</td>
<td>1.67</td>
<td>0.25</td>
</tr>
<tr>
<td>II</td>
<td>10</td>
<td>5</td>
<td>1200</td>
<td>3</td>
<td>0.33</td>
<td>3.33</td>
<td>1</td>
</tr>
<tr>
<td>III</td>
<td>5</td>
<td>1</td>
<td>1200</td>
<td>3</td>
<td>30</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>IV</td>
<td>5</td>
<td>1</td>
<td>1200</td>
<td>3</td>
<td>30</td>
<td>0.67</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 9: Synthetic datasets for the sparse PCA problem

All seven methods are started from the same initial point \((X_0, Y_0)\) that are chosen as follows. For datasets I and II, we set \( X_0 = Y_0 \) to be a diagonal matrix with the first five diagonal entries equal to 1 and the other entries equal zero. For datasets III and IV, we set \( X_0 = Y_0 \) with the first diagonal entry being 1 and any other entries being 0. We observe that the initial points were chosen differently so as to guarantee that they are feasible (i.e., lie in \( \text{dom} h \)) for their respective instances. The parameter \( \alpha \) is set to 0.5 in both AC and ACT.

Numerical results of the seven methods are given in Table 10 and some statistic measures of AC and ACT are given in Table 11. The explanation of their columns excluding the first one is the same as those of Tables 7 and 8, respectively. Their first columns differ from those of Tables 7 and 8 in that the value of \( M \) is computed according to (45). The best objective function values obtained by all seven methods are not reported since they are essentially the same on all instances. The number of resolvent evaluations is 1 in NC, 2 in AG, AC and ACT, 1 or 2 in NM, 1 on average in AD, and 3 on average in UP. The bold numbers highlight the method that has the best performance in an instance of the problem.

<table>
<thead>
<tr>
<th>( M )</th>
<th>AG</th>
<th>NM</th>
<th>UP</th>
<th>NC/AD</th>
<th>ACT/AC</th>
<th>AG</th>
<th>NM</th>
<th>UP</th>
<th>NC/AD</th>
<th>ACT/AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.33</td>
<td>21</td>
<td>18</td>
<td>7</td>
<td>15/31</td>
<td>18/15</td>
<td>8.63</td>
<td>4.96</td>
<td>6.71</td>
<td>4.50/10.70</td>
<td>9.70/7.33</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>13/12</td>
<td>9/7</td>
<td>10.08</td>
<td>2.73</td>
<td>7.55</td>
<td>4.42/4.01</td>
<td>4.66/3.94</td>
</tr>
<tr>
<td>63</td>
<td>32</td>
<td>43</td>
<td>18</td>
<td>81/48</td>
<td>43/27</td>
<td>19.91</td>
<td>12.06</td>
<td>17.61</td>
<td>22.54/16.05</td>
<td>24.08/12.04</td>
</tr>
<tr>
<td>60.67</td>
<td>35</td>
<td>46</td>
<td>17</td>
<td>84/52</td>
<td>48/31</td>
<td>19.01</td>
<td>14.28</td>
<td>16.97</td>
<td>24.31/17.05</td>
<td>26.70/12.51</td>
</tr>
</tbody>
</table>

Table 10: Numerical results for AG, NM, UP, NC, AD, ACT and AC
Table 11: AC and ACT statistics

<table>
<thead>
<tr>
<th>M</th>
<th>AC</th>
<th>ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Avg</td>
</tr>
<tr>
<td>2.33</td>
<td>2.00</td>
<td>0.72</td>
</tr>
<tr>
<td>4</td>
<td>3.67</td>
<td>3.41</td>
</tr>
<tr>
<td>63</td>
<td>44.41</td>
<td>31.12</td>
</tr>
<tr>
<td>60.67</td>
<td>36.00</td>
<td>28.26</td>
</tr>
</tbody>
</table>

In summary, computational results demonstrate that: i) the computed average curvature of AC is close to the computed maximum curvature; ii) the percentage of good iterations of AC lies in a suitable range; and iii) AC is either the best method or close to the best one in terms of running time.

5.4 Matrix Completion

This subsection focuses on a constrained version of the nonconvex low-rank matrix completion problem. Before stating the problem, we first give a few definitions. Let $\Omega$ be a subset of $\{1, \ldots, l\} \times \{1, \ldots, n\}$ and let $\Pi_{\Omega}$ denote the linear operator that maps a matrix $A$ to the matrix whose entries in $\Omega$ have the same values of the corresponding ones in $A$ and whose entries outside of $\Omega$ are all zero. Also, for given parameters $\beta > 0$ and $\theta > 0$, let $p : \mathbb{R} \to \mathbb{R}_+$ denote the log-sum penalty defined as

$$p(t) = p_{\beta, \theta}(t) := \beta \log \left(1 + \frac{|t|}{\theta}\right).$$

The constrained version of the nonconvex low-rank matrix completion problem considered in this subsection is

$$\min_{Z \in \mathbb{R}^{l \times n}} \left\{ \frac{1}{2} \|\Pi_{\Omega}(Z - O)\|_F^2 + \mu \sum_{i=1}^{r} p(\sigma_i(Z)) : Z \in B_R \right\}$$

(46)

where $R$ is a positive scalar, $B_R := \{Z \in \mathbb{R}^{l \times n} : \|Z\|_F \leq R\}$, $O \in \mathbb{R}^{\Omega}$ is an incomplete observed matrix, $\mu > 0$ is a parameter, $r := \min\{l, n\}$ and $\sigma_i(Z)$ is the $i$-th singular value of $Z$. The above problem differs from the one considered in [24] in that it adds the constraint $\|Z\|_F \leq R$ to the latter one.

The matrix completion problem in (46) is equivalent to

$$\min_{Z \in \mathbb{R}^{l \times n}} f(Z) + h(Z),$$

(47)

where

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

$$h(Z) = \mu p_0 \|Z\|_* + I_{B_R}(Z), \quad p_0 = p'(0) = \frac{\beta}{\theta}$$

and $\| \cdot \|_*$ denotes the nuclear norm defined as $\| \cdot \|_* := \sum_{i=1}^{r} \sigma_i(\cdot)$. It is proved in [24] that the second term in the definition of $f$, i.e., $\mu \sum_{i=1}^{r} [p(\sigma_i(\cdot)) - p_0 \sigma_i(\cdot)]$, is concave and $2\mu\tau$-smooth where $\tau = \beta/\theta^2$, so $f$ is nonconvex and smooth. Since $h$ is convex and nonsmooth, the problem in (47) falls into the general class of nonconvex SCO problems (1). It is easy to see that the pair

$$(M, m) = (\max\{1, 2\mu\tau\}, 2\mu\tau)$$

(48)
satisfies assumption (A2).

We use the MovieLens dataset\(^1\) to obtain the observed index set \(\Omega\) and the incomplete observed matrix \(O\). The dataset includes a sparse matrix with 100,000 ratings of \(\{1,2,3,4,5\}\) from 943 users on 1682 movies, namely \(l = 943\) and \(n = 1682\). The radius \(R\) is chosen as the Frobenius norm of the matrix of size \(943 \times 1682\) containing the same entries as \(O\) in \(\Omega\) and 5 in the entries outside of \(\Omega\).

We start all seven methods from the same initial point \(Z_0\) that is sampled from the standard Gaussian distribution and is within \(B(R)\). The parameter \(\alpha\) is set to 0.5 in AC and 0.1 in ACT.

Numerical results of the seven methods are given in Table 12 and some statistic measures of AC and ACT are given in Table 13. The format of Table 12 is similar to that of Table 10 with the exception that the second to sixth columns provide the function values of (46) at the last iteration and the numbers of iterations for all seven methods. Note that the first columns of Tables 12 and 13 give the value of \(M\) computed according to (48). The number of resolvent evaluations is 1 in NC, 2 in AG, AC and ACT, 1 or 2 in NM, 1 on average in AD, and 3 on average in UP. The bold numbers highlight the method that has the best performance in an instance of the problem.

<table>
<thead>
<tr>
<th>(M)</th>
<th>Function Value / Iteration Count</th>
<th>Running Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AG</td>
<td>NM</td>
</tr>
<tr>
<td>4.4</td>
<td>2257</td>
<td>1809</td>
</tr>
<tr>
<td></td>
<td>3856</td>
<td>1036</td>
</tr>
<tr>
<td>8.9</td>
<td>3886</td>
<td>3359</td>
</tr>
<tr>
<td></td>
<td>9158</td>
<td>1617</td>
</tr>
<tr>
<td>20</td>
<td>4282</td>
<td>3635</td>
</tr>
<tr>
<td></td>
<td>22002</td>
<td>2875</td>
</tr>
<tr>
<td>30</td>
<td>5967</td>
<td>5237</td>
</tr>
<tr>
<td></td>
<td>37032</td>
<td>3717</td>
</tr>
</tbody>
</table>

Table 12: Numerical results for AG, NM, UP, NC, AD, ACT and AC

<table>
<thead>
<tr>
<th>(M)</th>
<th>AC</th>
<th>ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Avg</td>
</tr>
<tr>
<td>4.4</td>
<td>1.00</td>
<td>0.31</td>
</tr>
<tr>
<td>8.9</td>
<td>1.00</td>
<td>0.28</td>
</tr>
<tr>
<td>20</td>
<td>0.99</td>
<td>0.25</td>
</tr>
<tr>
<td>30</td>
<td>0.97</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Table 13: AC and ACT statistics

In summary, computational results demonstrate that: i) the computed average curvature for AC is small compared with \(M\) and the computed maximum curvature; ii) the percentage of good iterations of AC lies in a suitable range; and iii) AC has the best performance in terms of running time. Although AC uses the least amount of time to terminate, NM finds solutions with the smallest objective function values.

\(^1\)http://grouplens.org/datasets/movielens/
5.5 Nonnegative Matrix Factorization

This subsection focuses on the following NMF problem

\[
\min \left\{ f(X,Y) := \frac{1}{2} \| A - XY \|_F^2 : X \geq 0, Y \geq 0 \right\},
\]

where \( A \in \mathbb{R}^{n \times l}, X \in \mathbb{R}^{n \times p} \) and \( Y \in \mathbb{R}^{p \times l} \), which have been thoroughly studied in the literature (see e.g., [8, 12]).

This subsection reports the efficiency of directly using all seven methods to solve (49) without making use of its two-block structure. We use the facial image dataset provided by AT&T Laboratories Cambridge\(^2\) to construct the matrix \( A \). More specifically, this dataset consists of 400 images, and each of those contains \( 92 \times 112 \) pixels with \( 256 \) gray levels per pixel. It results in an \( n \times l = 10,304 \times 400 \) matrix \( A \) whose columns are the vectorized images. The dimension \( p \) is set to 20.

We start all seven methods from the same initial point \((X_0, Y_0) = (1^{n \times p} / (np), 1^{p \times l} / (pl))\), where \( 1^{n \times p} \) and \( 1^{p \times l} \) are matrices of all ones of sizes \( n \times p \) and \( p \times l \), respectively. We estimate \( M \) in (5) as \( M = 100 \times C((X_0, Y_0), (0, 0)) \) where \( C(\cdot, \cdot) \) is defined in (4). Since no sharper \( m < M \) satisfying the first inequality in (5) is known, we simply set \( m = M \). The parameter \( \alpha \) is set to 0.7 in both AC and ACT.

Numerical results for the seven methods are given in Table 14. The bold numbers highlight the method that has the best performance in the problem. The best objective function values obtained by all seven methods are not reported since they are essentially the same.

<table>
<thead>
<tr>
<th>Method</th>
<th>Iteration Count</th>
<th>Running time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG</td>
<td>786</td>
<td>73.03</td>
</tr>
<tr>
<td>NM</td>
<td>162</td>
<td>14.91</td>
</tr>
<tr>
<td>UP</td>
<td>37</td>
<td>11.12</td>
</tr>
<tr>
<td>NC</td>
<td>656</td>
<td>41.67</td>
</tr>
<tr>
<td>AD</td>
<td>44</td>
<td>5.21</td>
</tr>
<tr>
<td>ACT</td>
<td>41</td>
<td>6.54</td>
</tr>
<tr>
<td>AC</td>
<td>36</td>
<td>4.70</td>
</tr>
</tbody>
</table>

Table 14: Numerical results for AG, NM, UP, NC, AD, ACT and AC

6 Concluding remarks

This paper presents an average curvature accelerated composite gradient method, namely, the AC-ACG method, for solving the N-SCO problem which is based on the average of all observed curvatures. More specifically, as opposed to other ACG variants, which use a known Lipschitz constant or a backtracking procedure that searches for a good upper curvature \( M_k \), AC-ACG uses the average of all observed curvatures to compute \( M_k \) (see (14)) and always accepts the first computed iterate according to (2) no matter whether \( M_k \) is good or not. A nice feature of AC-ACG is that its convergence rate bound is expressed in terms of \( M_k \) rather than an upper curvature \( M \geq M \).

We now discuss some possible extensions of this paper. First, numerical results show that the AC variant, which computes \( C_k \) as in (40), performs substantially better than previous ACG

\(^2\)https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
variants as well as the ACT variant, which is closer to the main method analyzed in this paper, namely, AC-ACG. However, convergence rate analysis of AC (possibly with $\gamma$ and $\alpha$ satisfying (8)) is an interesting open problem. Second, the AC-ACG method performs two resolvent evaluations of $h$ per iteration. It would be desirable to develop AC-ACG variants which only perform one resolvent evaluation of $h$ per iteration.

7 Acknowledgements

We are grateful to Guanghui Lan and Saeed Ghadimi for providing the code for the UPFAG method of their paper [7].

References


A technical result

Recall the definition of a good upper curvature of $f$ given above (3).

**Lemma A.1.** If $M_k$ is a good upper curvature of $f$ at $\tilde{x}_k$ and $y_{k+1} = y(\tilde{x}_k; M_k)$ where $y(\cdot; \cdot)$ is defined in (2), then

$$\phi(y_{k+1}) \leq \phi(\tilde{x}_k) - \frac{M_k}{2} \|y_{k+1} - \tilde{x}_k\|^2.$$ (50)
Proof: Using the fact that $M_k$ is a good upper curvature of $f$ at $\tilde{x}_k$ and (3), we have
\[
\phi(y_{k+1}) \leq \ell_f(y_{k+1}; \tilde{x}_k) + h(y_{k+1}) + \frac{M_k}{2} \|y_{k+1} - \tilde{x}_k\|^2.
\] (51)

It follows from the definition of $y_{k+1}$, (2) and the fact that the objective function in (2) is $M_k$-strongly convex that for every $u \in \text{dom } h$,
\[
\ell_f(u; \tilde{x}_k) + h(u) + \frac{M_k}{2} \|u - \tilde{x}_k\|^2 \geq \ell_f(y_{k+1}; \tilde{x}_k) + h(y_{k+1}) + \frac{M_k}{2} \|y_{k+1} - \tilde{x}_k\|^2 + \frac{M_k}{2} \|u - y_{k+1}\|^2,
\]
which together with $u = \tilde{x}_k$ implies that
\[
\phi(\tilde{x}_k) \geq \ell_f(y_{k+1}; \tilde{x}_k) + h(y_{k+1}) + M_k \|y_{k+1} - \tilde{x}_k\|^2.
\]

Now inequality (50) immediately follows from (51) and the above inequality. ■