An Adaptive Accelerated Proximal Gradient Method and its Homotopy Continuation for Sparse Optimization

Qihang Lin* Lin Xiao†

February 4, 2014

Abstract

We consider optimization problems with an objective function that is the sum of two convex terms: one is smooth and given by a black-box oracle, and the other is general but with a simple, known structure. We first present an accelerated proximal gradient (APG) method for problems where the smooth part of the objective function is also strongly convex. This method incorporates an efficient line-search procedure, and achieves the optimal iteration complexity for such composite optimization problems. In case the strong convexity parameter is unknown, we also develop an adaptive scheme that can automatically estimate it on the fly, at the cost of a slightly worse iteration complexity.

Then we focus on the special case of solving the ℓ_1-regularized least-squares problem in the high-dimensional setting. In such a context, the smooth part of the objective (least-squares) is not strongly convex over the entire domain. Nevertheless, we can exploit its restricted strong convexity over sparse vectors using the adaptive APG method combined with a homotopy continuation scheme. We show that such a combination leads to a global geometric rate of convergence, and the overall iteration complexity has a weaker dependency on the restricted condition number than previous work.

1 Introduction

Exploiting problem structure has become an important theme in recent advances in convex optimization. It is well known that proper use of problem structure at the numerical linear algebra level may dramatically improve the efficiency of an optimization method. More recently, it has become clear that exploiting problem structure can also lead to more efficient optimization methods in terms of their iteration complexity, sometimes significantly surpassing the limitations of the black-box complexity theory (see [Nes08] for an excellent discussion). Such examples start with the theory of self-concordant functions for interior-point methods [NN94], to the more recent development of smoothing technique [Nes05], minimization of composite objective functions [Nes13], and acceleration via manifold identification (e.g., [Wri12]).

In this paper, we first develop an adaptive accelerated proximal gradient method for minimizing composite objective functions that are strongly convex, without the knowledge of their convexity parameters or any lower bound. Then we employ this method in a homotopy continuation scheme for sparse optimization (with ℓ_1-regularization), and show that it achieves an improved iteration complexity than previous methods for solving the sparse least-squares problem.

*Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, 15213. Email: qihang-lin@uiowa.edu

†Machine Learning Groups, Microsoft Research, Redmond, WA 98052. Email: lin.xiao@microsoft.com

1.1 Minimizing composite objective functions

We consider first-order methods for minimizing *composite* objective functions, i.e., the problem of

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \Big\{\phi(x) \triangleq f(x) + \Psi(x)\Big\}, \tag{1}$$

where $f(x)$ and $\Psi(x)$ are lower-semicontinuous, proper convex functions [Roc70, Section 7]. We assume that $\operatorname{dom}\Psi$ is closed, and f is differentiable on an open set containing $\operatorname{dom}\Psi$. We also assume that ∇f is Lipschitz continuous on $\operatorname{dom}\Psi$, i.e., there exists a constant L_f such that

$$\|\nabla f(x) - \nabla f(y)\|_2 \;\le\; L_f\|x-y\|_2, \quad \forall\, x, y \in \operatorname{dom}\Psi. \tag{2}$$

In general, $\Psi(x)$ can be nondifferentiable, hence so does $\phi(x)$. Further assumptions on f and Ψ, which correspond to richer structure, will be added as we narrow down the potential applications and aim for more efficient first-order methods.

If we ignore the structure in (1), and simply treat $\phi(x)$ as a black-box first-order oracle, then classical complexity theory states that the *information complexity* for finding an ϵ-optimal solution (whose objective value is within ϵ of the optimum) is $O(1/\epsilon^2)$ [NY83]. Here information complexity is the *lower* estimate for the number of oracle calls that is necessary for any first-order method to obtain an ϵ-solution to any problem from the problem class. In this paper we will mainly use *iteration complexity* to estimate the efficiency of a first-order method, which is an *upper* bound on the number of iterations for the method to find an ϵ-optimal solution. For example, the subgradient method has an iteration complexity $O(1/\epsilon^2)$ for minimizing $\phi(x)$, which matches the lower complexity bound [NY83, Nes04]. All methods we discuss in this paper implement a line search procedure within each iteration, so there can be multiple oracle calls per iteration. However, the average number of oracle calls per iteration is always bounded by a small constant.

Many applications that motivate the model in (1) have additional structure. The one that is responsible for most recent developments is that $\Psi(x)$ being *simple* [Nes13], meaning that for any $y \in \operatorname{dom}\Psi$, the following auxiliary optimization problem can be solved efficiently or in closed-form:

$$T_L(y) = \arg\min_x \left\{ f(y) + \nabla f(y)^T(x-y) + \frac{L_f}{2}\|x-y\|_2^2 + \Psi(x) \right\}. \tag{3}$$

This is the case, e.g., when $\Psi(x) = \tau\|x\|_1$ for any $\tau > 0$, or $\Psi(x)$ is the indicator function of a closed convex set that admits an easy projection from any point in $\mathbb{R}^n$. The so-called *proximal gradient* (PG) method simply uses (3) as its update rule:

$$x^{(k+1)} = T_L(x^{(k)}), \quad k = 0, 1, 2, \ldots, \tag{4}$$

where L is set to L_f or determined by a linear search procedure. The iteration complexity for the PG method is $O(L_f/\epsilon)$, which is better than the lower bound when using a black-box model for ϕ. Of course this is not a contradiction, since each iteration of the PG method relies on the entire structure of Ψ, rather than a mere subgradient of it. In fact, a far better iteration complexity, $O\big(\sqrt{L_f/\epsilon}\big)$, can be obtained by accelerated proximal gradient (APG) methods [Nes13, BT09, Tse08], with the same order of computational cost per iteration.

If the function ϕ is also strongly convex, i.e., either f or Ψ or both of them are strongly convex, then the PG method and variants of APG methods can have geometric convergence. In this paper,

we focus on the case when f is strongly convex, i.e., there exists a constant $\mu_f > 0$ (called the *convexity parameter*) such that

$$f(x) \;\geq\; f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu_f}{2}\|x - y\|_2^2, \quad \forall\, x, y \in \operatorname{dom} \Psi. \tag{5}$$

Throughout this paper, we use $\kappa_f = L_f/\mu_f$ to denote the *condition number* of the function f. In this case, the PG method has iteration complexity $O(\kappa_f \log(1/\epsilon))$, and APG methods can achieve $O(\sqrt{\kappa_f}\log(1/\epsilon))$ [Nes04, Nes13]. However, in order to obtain the improved complexity, the APG methods need to use the convexity parameter μ_f, or a lower bound of it, explicitly in their updates. In many applications, an effective lower bound of μ_f can be hard to estimate.

Our first contribution in this paper is a new variant of APG method for solving problem (1) when f is strongly convex. It incorporates an efficient line-search procedure, and achieves the optimal iteration complexity $O(\sqrt{\kappa_f}\log(1/\epsilon))$. In case the strong convexity parameter is unknown, we develop an adaptive scheme that can automatically estimate it on the fly. This method achieves the iteration complexity $O\left(\sqrt{\kappa_f}\log\kappa_f \cdot \log(\kappa_f/\epsilon)\right)$, where the extra $\log\kappa_f$ factors are due to the adaptive scheme for estimating μ_f.

This adaptive scheme is similar to the one proposed by Nesterov for his accelerated dual gradient (ADG) method [Nes13], which has the same iteration complexity. The ADG method works directly with a model where the strong convexity lies in Ψ instead of f. Therefore, in order to use it under our assumption (that f is strongly convex), one needs to relocate a strong convexity term from f to Ψ whenever the adaptive scheme changes its estimation of μ_f. Our method demonstrates that the same adaption idea can be applied successfully without resorting to the relocation of strongly convex terms. Moreover, the adaptive ADG method [Nes13, Section 5.3] requires two line-searches in each iteration while our method requires only one, which reduces the computational cost per iteration by half. Our analysis also clears the theoretical reasoning why one line-search is enough.

There exist other adaptive schemes for parameters search in APG methods [MOS12, GK13]. However, different from our method, these search schemes are devoted to accelerate APG methods in practice rather than search the unknown parameter μ_f. In particular, the method by [GK13] has to assume either μ_f or its non-trivial lower bound is known while [MOS12] does not consider the strongly convex case and thus only achieves a sublinear convergence rate.

In the next, we will discuss the sparse least-squares problem to show that, even in the case of $\mu_f = 0$, additional structure of the problem may still allow the development of first-order methods with geometric convergence (same as *linear convergence* in this paper).

1.2 Homotopy continuation for sparse optimization

An important special case of (1) is the ℓ_1-regularized least-squares (ℓ_1-LS) problem:

$$\underset{x}{\text{minimize}} \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1, \tag{6}$$

where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ are the problem data, and $\lambda > 0$ is a regularization parameter. Here $\|x\|_1 = \sum_i |x_i|$ is the ℓ_1 norm of x. In terms of the model in (1), we have

$$f(x) = (1/2)\|Ax - b\|_2^2, \qquad \Psi(x) = \lambda\|x\|_1.$$

Since the ℓ_1 term promotes sparse solutions, we also refer (6) as the *sparse least-squares* problem.

The ℓ_1-LS problem has important applications in machine learning, signal processing, and statistics; see, e.g., [Tib96, CDS98, BDE09]. It received revived interests in recent years due to the emergence of *compressed sensing* theory, which builds upon the fundamental idea that a finite-dimensional signal having a sparse or compressible representation can be recovered from a small set of linear, nonadaptive measurements [CRT06, CT06, Don06]. We are especially interested in solving the ℓ_1-LS problem in the context of high-dimensional sparse recovery. More specifically, we focus on the case when $m < n$ (i.e., the linear system $Ax = b$ is underdetermined) and the solution $x^\star(\lambda)$ is sparse (which requires the parameter λ to be sufficiently large).

The function $f(x) = (1/2)\|Ax - b\|_2^2$ has a constant Hessian $\nabla^2 f(x) = A^T A$, and we have

$$\begin{aligned} L_f &= \rho_{\max}(A^T A), \\ \mu_f &= \rho_{\min}(A^T A), \end{aligned}$$

where $\rho_{\max}(\cdot)$ and $\rho_{\min}(\cdot)$ denote the largest and smallest eigenvalues, respectively, of a symmetric matrix. Under the assumption $m < n$, the matrix $A^T A$ is singular, hence $\mu_f = 0$ (i.e., f is not strongly convex). Therefore, we only expect sublinear convergence rates (at least globally) when using first-order optimization methods. For example, we have the iteration complexity $O(L_f/\epsilon)$ when using the PG method, and $O(\sqrt{L_f/\epsilon})$ for the APG methods.

Nevertheless, even in the case of $m < n$, when the solution $x^\star(\lambda)$ is sparse, the PG method often exhibits fast convergence when it gets close to the optimal solution. Indeed, local linear convergence can be established for the PG method provided that the active submatrix (columns of A corresponding to the nonzero entries of the sparse iterates) is well conditioned [LT92, HYZ08, BL08]. Here, local linear convergence means that the algorithm converges linearly only when the number of iterations is sufficiently large and the solution is close enough to optimality. To explain the reason for the local linear convergence more formally, we define the *restricted eigenvalues* of A at the sparsity level s as

$$\begin{aligned} \rho_+(A, s) &= \sup\left\{ \frac{x^T A^T A x}{x^T x} : x \neq 0,\ \|x\|_0 \leq s \right\}, \\ \rho_-(A, s) &= \inf\left\{ \frac{x^T A^T A x}{x^T x} : x \neq 0,\ \|x\|_0 \leq s \right\}, \end{aligned} \tag{7}$$

where s is a positive integer and $\|x\|_0$ denotes the number of nonzero entries of a vector $x \in \mathbb{R}^n$. From the above definitions, we have

$$\mu_f \ \leq\ \rho_-(A, s) \ \leq\ \rho_+(A, s) \ \leq\ L_f, \qquad \forall\, s > 0.$$

As discussed before, we have $\mu_f = 0$ for $m < n$. But it is still possible that $\rho_-(A, s) > 0$ holds for some $s < m$. In this case, we say that the matrix A satisfies the *restricted eigenvalue condition* at the sparsity level s. Let $\mathrm{supp}(x) = \{j : x_j \neq 0\}$, and assume that $x, y \in \mathbb{R}^n$ satisfy $|\mathrm{supp}(x) \cup \mathrm{supp}(y)| \leq s$. Then it can be shown [XZ12, Lemma 3] that

$$f(x) \ \geq\ f(y) + \langle \nabla f(y), x - y \rangle + \frac{\rho_-(A, s)}{2}\|x - y\|_2^2.$$

The above inequality gives the notion of *restricted strong convexity* (cf. strong convexity defined in (5)). Intuitively, if the iterates $x^{(k)}$ of the PG method (4) become sparse and their supports

do not fluctuate much from each other, then restricted strong convexity leads to (local) linear convergence. This is exactly what happens when the PG method speeds up while getting close to the optimal solution.

Moreover, such a local linear convergence can be exploited by a homotopy continuation strategy to obtain much faster convergence in practice [HYZ08, WNF09, XZ12]. The basic idea is to solve the ℓ_1-LS problem (6) with a large value of λ first, and then gradually decreases the value of λ until the target regularization is reached. For each value of λ, we employ the PG method to solve (6) up to an adequate precision, and then use the resulting approximate solution to warm start the PG method for the next value of λ. The hope is to engage each stage within their local linear convergence zone only. It is shown in [XZ12] that under suitable assumptions for sparse recovery (mainly the restricted eigenvalue condition), an appropriate homotopy strategy can ensure all iterates of the PG method be sparse, hence linear convergence at each stage can be established. As a result, the overall iteration complexity of such a proximal-gradient homotopy (PGH) method is $\widetilde{O}\big(\kappa_s \log(1/\epsilon)\big)$ where κ_s denotes the *restricted condition number* at some sparsity level $s > 0$, i.e.,

$$\kappa_s \;\triangleq\; \kappa(A, s) \;=\; \frac{\rho_+(A, s)}{\rho_-(A, s)}, \tag{8}$$

and the notation $\widetilde{O}(\cdot)$ hides some additional $\log(\kappa_s)$ factors. We note that in order to accommodate the fluctuations of $\|x^{(k)}\|_0$ generated by the PG method, the sparsity level s in the above complexity bound needs to be much larger than $\|x^\star(\lambda)\|_0$.

Our second contribution in this paper is to show that, by using the adaptive APG method developed in this paper in a homotopy continuation scheme, we can further improve the iteration complexity for solving the ℓ_1-LS problem to $\widetilde{O}\big(\sqrt{\kappa_{s'}} \log(1/\epsilon)\big)$, where the sparsity level s' is slightly larger than the one for the PGH method.

We note that this result is not an trivial extension from the convergence results for the PGH method in [XZ12]. In particular, the proof of the convergence results of [XZ12] strongly depends on the fact that PG monotonically decreases the objective values. However, it is well known that APG does not have the property of monotone decreasing. In order to overcome this difficulty, we had to show a "non-blowout" property (Lemma 10) of our adaptive APG method first and prove that this can guarantee a sparse solution path without monotonicity. Furthermore, this property is in itself interesting to the users of our adaptive APG method. For a problem that is strongly convex with an unknown strong convexity parameter, "non-blowout" property guarantees that the adaptive APG method will at least not diverge even if we overestimate the strong convexity parameter. This provides safeguard for tuning the algorithms.

Similar homotopy continuation schemes have also been applied to PG method by [HYZ08, WNF09] and to active-set method by [WYGZ10] and further accelerate these algorithms in practice. However, they do not analyze the convergence properties under the homotopy schemes. We also note that [GLW13] develop an APG method for solving the constrained version of (6), i.e., $\min_{\|x\|_1 \le \lambda} \frac{1}{2}\|Ax - b\|_2^2$, without using a homotopy continuation scheme. However, they only achieve local linear convergence while our method guarantees a global linear convergence rate.

1.3 Outline of the paper

In Section 2, we define the notations and review some preliminaries used in this paper. In Section 3, we present an accelerated proximal gradient method (scAPG) for minimizing strongly convex functions with a known convexity parameter, and prove that its iteration complexity is $O(\sqrt{\kappa_f} \log(1/\epsilon))$.

In Section 4, we incorporate an estimation scheme for the convexity parameter into scAPG, to develop an adaptive APG (AdapAPG) method that works with an unknown convexity parameter. We prove that this algorithm has an iteration complexity $O(\sqrt{\kappa_f}\log(\kappa_f)\log(\kappa_f/\epsilon))$. Then a homotopy continuation version of the AdapAPG method is proposed in Section 5 for sparse optimization, and we give its complexity analysis for solving the ℓ_1-LS problems under a restricted eigenvalue condition. Numerical experiments and conclusions are provided in Section 6 and Section 7 respectively.

2 Preliminaries and notations

In this section, we first defined the notion of *optimality residue* for minimizing composite objective functions, then review the definition and properties of *composite gradient mapping*, as well as a proximal-gradient method with line search developed in [Nes13].

Consider the optimization problem (1) where the function f is convex and differentiable, and Ψ is closed and convex on $\mathbb{R}^n$. The optimality condition of (1) states that $x^\star$ is a solution if and only if there exists $\xi \in \partial\Psi(x^\star)$ such that

$$\nabla f(x^\star) + \xi = 0$$

(see, e.g., [Roc70, Section 27]). Therefore, a good measure of accuracy for any x as an approximate solution is the quantity

$$\omega(x) \triangleq \min_{\xi\in\partial\Psi(x)} \|\nabla f(x) + \xi\|_\infty. \tag{9}$$

We call $\omega(x)$ the *optimality residue* of x. We will use it in the stopping criterion of the proximal gradient methods discussed in this paper.

2.1 Composite gradient mapping

Composite gradient mapping was introduced by Nesterov in [Nes13]. For any fixed point y and a given constant $L > 0$, we define a local model of $\phi(x)$ around y using a quadratic approximation of f but keeping Ψ intact:

$$\psi_L(y; x) = f(y) + \nabla f(y)^T(x - y) + \frac{L}{2}\|x - y\|_2^2 + \Psi(x).$$

Let

$$T_L(y) = \arg\min_x \ \psi_L(y; x). \tag{10}$$

Then the *composite gradient mapping* of f at y is defined as

$$g_L(y) = L(y - T_L(y)).$$

In the case $\Psi(x) = 0$, it is easy to verify that $g_L(y) = \nabla f(y)$ for any $L > 0$, and $1/L$ can be considered as the step-size from y to $T_L(y)$ along the direction $-g_L(y)$.

The first-order optimality condition of (10) states that there exists $\xi_L(y) \in \partial\Psi(T_L(y))$ such that

$$\nabla f(y) + L(T_L(y) - y) + \xi_L(y) = 0.$$

In the rest of this paper, we denote

$$\phi'(T_L(y)) \ = \ \nabla f(T_L(y)) + \xi_L(y) \ = \ L\big(y - T_L(y)\big) + \nabla f(T_L(y)) - \nabla f(y).$$

Apparently, $\phi'(T_L(y))$ is a subgradient of ϕ at $T_L(y)$.

Throughout this paper, we assume that f has Lipschitz continuous gradient, i.e., it satisfies (2). A direct consequence is the following inequality (see, e.g., [Nes04, Theorem 2.1.5]):

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L_f}{2}\|x - y\|_2^2, \quad \forall\, x, y \in \mathbb{R}^n. \tag{11}$$

Therefore, whenever $L \geq L_f$, we have for any y,

$$\phi(T_L(y)) \leq \psi_L(y; T_L(y)). \tag{12}$$

However, the condition $L \geq L_f$ is not necessary for the above inequality to hold for any particular y. In fact, the line search procedures discussed in this paper try to find an L that is as small as possible to satisfy (12), which corresponds to a larger step size $1/L$ in the gradient direction.

Following [Nes13], we also define a local Lipschitz parameter

$$S_L(y) = \frac{\|\nabla f(T_L(y)) - \nabla f(y)\|_2}{\|T_L(y) - y\|_2}.$$

We will need the following three properties of composite gradient mapping shown in [Nes13]:

Lemma 1. *(Part of [Nes13, Theorem 2]) For any $y \in \operatorname{dom}\Psi$ and any $L > 0$,*

$$\psi_L(y; T_L(y)) \leq \phi(y) - \frac{1}{2L}\|g_L(y)\|_2^2.$$

Lemma 2. *(Part of [Nes13, Theorem 1]) For any $x, y \in \operatorname{dom}\Psi$ and any $L > 0$, we have*

$$\left\langle \phi'(T_L(y)), x - T_L(y) \right\rangle \geq -\left(1 + \frac{1}{L}S_L(y)\right) \cdot \|g_L(y)\|_2 \cdot \|T_L(y) - x\|_2.$$

Lemma 3. *([Nes13, Lemma 2]) Suppose ϕ is strongly convex with convexity parameter $\mu > 0$, and let $x^\star$ be the unique minimizer of ϕ. Then for any $y \in \operatorname{dom}\Psi$ and any $L > 0$, we have*

$$\|T_L(y) - x^\star\|_2 \leq \frac{1}{\mu}\left(1 + \frac{1}{L}S_L(y)\right)\|g_L(y)\|_2.$$

The next lemma shows that we can measure how close $T_L(y)$ is from satisfying the optimality condition by using the norm of the composite gradient mapping at y.

Lemma 4. *([XZ12, Lemma 2]) If f has Lipschitz continuous gradients with Lipschitz constant L_f, then*

$$\omega(T_L(y)) \leq \left(1 + \frac{S_L(y)}{L}\right)\|g_L(y)\|_2 \leq \left(1 + \frac{L_f}{L}\right)\|g_L(y)\|_2.$$

We will also need the following result in the complexity analysis of our algorithms:

Lemma 5. *Suppose $\mu \leq \mu_f$ and the inequality (12) holds for y. Then, for any $x \in \mathbb{R}^n$, we have*

$$\phi(x) \geq \phi(T_L(y)) + \langle g_L(y), x - y \rangle + \frac{1}{2L}\|g_L(y)\|^2 + \frac{\mu}{2}\|x - y\|^2. \tag{13}$$

We omit the proof of this lemma since it is similar to that of [Nes04, Theorem 2.2.7], in which Ψ is restricted to be the indicator function of a closed convex set. A variant of this lemma corresponding to $\mu = 0$ appeared in [BT09, Lemma 2.3]

Algorithm 1: $\{x^{(k+1)}, M_k, g^{(k)}, S_k\} \leftarrow \texttt{LineSearch}(x^{(k)}, L_k)$

parameter: $\gamma_{\text{inc}} > 1$
$L \leftarrow L_k/\gamma_{\text{inc}}$
repeat
  $L \leftarrow L\gamma_{\text{inc}}$
  $x^{(k+1)} \leftarrow T_L(x^{(k)})$
until $\phi(x^{(k+1)}) \leq \psi_L(x^{(k)}; x^{(k+1)})$
$M_k \leftarrow L$
$g^{(k)} \leftarrow M_k(y^{(k)} - x^{(k+1)})$
$S_k \leftarrow S_L(y^{(k)})$

2.2 Proximal gradient method with line search

With the machinery of composite gradient mapping, Nesterov developed several variants of proximal gradient methods in [Nes13]. Here we describe the simple primal-gradient method in Algorithms 1 and 2, which correspond to (3.1) and (3.2) in [Nes13], respectively. To use this algorithm, we need to first choose an initial optimistic estimate $L_{\min}$ for the Lipschitz constant L_f:

$$0 < L_{\min} \leq L_f,$$

and two adjustment parameters $\gamma_{\text{dec}} \geq 1$ and $\gamma_{\text{inc}} > 1$. We note that Algorithm 2 does not need to calculate $g^{(k)}$ and S_k, so the last two steps in Algorithm 1 can be skipped to save computation. However, they will be necessary as part of the adaptive algorithms we develop later in Section 4.

Each iteration of the proximal gradient method generates the next iterate in the form of

$$x^{(k+1)} = T_{M_k}(x^{(k)}),$$

where M_k is chosen by the line search procedure in Algorithm (1). The line search procedure starts with an estimated Lipschitz constant L_k, and increases its value by the factor γ_{inc} until the stopping criteria is satisfied. The stopping criteria for line search ensures

$$\begin{aligned} \phi(x^{(k+1)}) &\leq \psi_{M_k}\left(x^{(k)}, x^{(k+1)}\right) = \psi_{M_k}\left(x^{(k)}, T_{M_k}(x^{(k)})\right) \\ &\leq \phi(x^{(k)}) - \frac{1}{2M_k}\left\|g_{M_k}(x^{(k)})\right\|_2^2, \end{aligned} \tag{14}$$

where the last inequality follows from Lemma 1. Therefore, we have the objective value $\phi_\lambda(x^{(k)})$ decrease monotonically with k, unless the gradient mapping $g_{M_k}(x^{(k)}) = 0$. In the latter case, according to Lemma 4, $x^{(k+1)}$ is an optimal solution. A key feature of this algorithm is the adaptive line search: it always tries to use a smaller Lipschitz constant at the beginning of each iteration by setting L_{k+1} to be $\min\{L_{\min}, M_k/\gamma_{\text{dec}}\}$, which corresponds to a larger step size.

The only difference between Algorithm 2 and Nesterov's gradient method [Nes13, (3.2)] is that Algorithm 2 has an explicit stopping criterion. This stopping criterion is based on the optimality residue $\omega(x^{(k+1)})$ being small. For the ℓ_1-LS problem, it can be computed with additional $O(n)$ flops given the gradient $\nabla f(x)$. For other problems, if the optimality residue $w(\cdot)$ cannot be computed efficiently, then Lemma 4 suggests that the norm of the gradient mapping $\left\|g_{M_k}(x^{(k)})\right\|_2$ can be a good replacement to serve as the optimality measure in the stopping criterion.

Algorithm 2: $\{\hat{x}, \hat{M}\} \leftarrow \texttt{ProxGrad}(x^{(0)}, L_0, \epsilon)$

parameters: $L_{\min} > 0$, $\gamma_{\text{dec}} \geq 1$
repeat for $k = 0, 1, 2, \ldots$
  $\{x^{(k+1)}, M_k\} \leftarrow \texttt{LineSearch}(x^{(k)}, L_k)$
  $L_{k+1} \leftarrow \max\{L_{\min}, M_k/\gamma_{\text{dec}}\}$
until $\omega(x^{(k+1)}) \leq \epsilon$
$\hat{x} \leftarrow x^{(k+1)}$
$\hat{M} \leftarrow M_k$

Algorithm 3: $\{x^{(k+1)}, M_k, \alpha_k, g^{(k)}, S_k\} \leftarrow \texttt{AccelLineSearch}(x^{(k)}, x^{(k-1)}, L_k, \mu, \alpha_{k-1})$

parameter: $\gamma_{\text{inc}} > 1$
$L \leftarrow L_k/\gamma_{\text{inc}}$
repeat
  $L \leftarrow L\gamma_{\text{inc}}$
  $\alpha_k \leftarrow \sqrt{\frac{\mu}{L}}$
  $y^{(k)} \leftarrow x^{(k)} + \frac{\alpha_k(1-\alpha_{k-1})}{\alpha_{k-1}(1+\alpha_k)}(x^{(k)} - x^{(k-1)})$
  $x^{(k+1)} \leftarrow T_L(y^{(k)})$
until $\phi(x^{(k+1)}) \leq \psi_L(y^{(k)}; x^{(k+1)})$
$M_k \leftarrow L$
$g^{(k)} \leftarrow M_k(y^{(k)} - x^{(k+1)})$
$S_k \leftarrow S_L(y^{(k)})$

Nesterov established the following iteration complexities of Algorithm 2 for finding an ϵ-optimal solution of the problem (1):

- If ϕ is convex but not strongly convex, then the convergence is sublinear, with an iteration complexity $O(1/\epsilon)$ [Nes13, Theorem 4];
- If ϕ is strongly convex, then the convergence is geometric, with an iteration complexity $O(\kappa_f \log(1/\epsilon))$ [Nes13, Theorem 5].

A nice property of this algorithm is that we do not need to know a priori if the objective function is strongly convex or not. It will automatically exploit the strong convexity whenever it holds. The algorithm is the same for both cases.

3 An APG method for minimizing strongly convex functions

In this section, we assume that the function f is strongly convex. We present an accelerated proximal gradient (APG) method with adaptive line search for solving the composite minimization problem (1); see Algorithm 3 and Algorithm 4. We name this method scAPG, where "sc" stands for "strongly convex." This method requires an input parameter $\mu > 0$, which is an estimate of the true convexity parameter μ_f. The line search procedure is very similar to the one used in

Algorithm 4: $\{\hat{x}, \hat{M}\} \leftarrow \texttt{scAPG}(x^{(0)}, L_0, \mu, \hat{\epsilon})$

parameters: $L_{\min} \geq \mu > 0$, $\gamma_{\text{dec}} \geq 1$
$x^{(-1)} \leftarrow x^{(0)}$
$\alpha_{-1} = 1$
repeat for $k = 0, 1, 2, \ldots$
  $\{x^{(k+1)}, M_k, \alpha_k\} \leftarrow \texttt{AccelLineSearch}(x^{(k)}, x^{(k-1)}, L_k, \mu, \alpha_{k-1})$
  $L_{k+1} \leftarrow \max\{L_{\min}, M_k/\gamma_{\text{dec}}\}$
until $\omega(x^{(k+1)}) \leq \hat{\epsilon}$
$\hat{x} \leftarrow x^{(k+1)}$
$\hat{M} \leftarrow M_k$

Algorithms 1 and 2. In particular, we need to give an initial lower estimate $L_{\min}$ for the Lipschitz constant L_f and two adjustment parameters $\gamma_{\text{dec}} \geq 1$ and $\gamma_{\text{inc}} > 1$. Each iteration of the scAPG method generates the following three sequences

$$\begin{aligned} \alpha_k &= \sqrt{\frac{\mu}{M_k}}, \\ y^{(k)} &= x^{(k)} + \frac{\alpha_k(1-\alpha_{k-1})}{\alpha_{k-1}(1+\alpha_k)}(x^{(k)} - x^{(k-1)}), \\ x^{(k+1)} &= T_{M_k}(y^{(k)}). \end{aligned} \tag{15}$$

We note that the scAPG method does not need to calculate $g^{(k)}$ and S_k, so the last two steps in Algorithm 3 can be skipped to save computation (they will become necessary in Section 4).

This method can be considered as an extension of Nesterov's constant step scheme [Nes04, (2.2.11)], integrated with a line-search procedure. In fact, if $\alpha_k = \alpha_{k-1} = \sqrt{\mu_f/L_f}$, then the update for $y^{(k)}$ in (15) becomes

$$y^{(k)} = x^{(k)} + \frac{\sqrt{L_f} - \sqrt{\mu_f}}{\sqrt{L_f} + \sqrt{\mu_f}}(x^{(k)} - x^{(k-1)}),$$

which is the same as in Algorithm (2.2.11) in [Nes04]. Note that, one can not directly apply Algorithm 4 to problems without strongly convexity by simply setting $\mu = 0$.

The sequence M_k is chosen by the line search procedure in Algorithm 3, which starts with an estimated Lipschitz constant L_k, and increases its value by the factor γ_{inc} until the stopping criteria is satisfied. Since f has Lipschitz constant L_f, the inequality (11) implies that the line search procedure is guaranteed to terminate if $L \geq L_f$. Therefore, we have

$$L_{\min} \leq L_k \leq M_k < \gamma_{\text{inc}} L_f. \tag{16}$$

Although there is no explicit bound on the number of repetitions in the line search procedure, it can be shown that the total number of line searches cannot be too big. More specifically, let N_k be the total number of operations $x^+ \leftarrow T_L(y)$ performed after k iterations in Algorithm 4. Then we have

$$N_k \;\leq\; \left(1 + \frac{\ln \gamma_{\text{dec}}}{\ln \gamma_{\text{inc}}}\right)(k+1) + \frac{1}{\ln \gamma_{\text{inc}}} \max\left\{\ln \frac{\gamma_{\text{inc}} L_f}{\gamma_{\text{dec}} L_{\min}}, 0\right\}.$$

The proof of the above bound follows the same arguments in [Nes13, Lemma 3]. For example, if we choose $\gamma_{\text{inc}} = \gamma_{\text{dec}} = 2$, then we always have $L_k \leq L_f$ and

$$N_k \leq 2(k+1) + \log_2 \frac{L_f}{L_{\min}}. \tag{17}$$

Thus, the performance of the scAPG method is well characterized by its iteration complexity (which bounds the number of iterations k).

The following theorem states that if μ is a nontrivial lower bound on the convexity parameter μ_f, then the scAPG converges geometrically and it has an iteration complexity $O(\sqrt{\kappa_f}\log(1/\epsilon))$.

Theorem 1. *Suppose $x^\star$ is the optimal solution of (1) and $0 < \mu \leq \mu_f$. Then Algorithm 4 guarantees that*

$$\phi(x^{(k)}) - \phi(x^\star) \;\leq\; \tau_k \left[\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right], \tag{18}$$

$$\frac{\mu}{2}\|y^{(k)} - x^\star\|_2^2 \;\leq\; \tau_k \left[\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right], \tag{19}$$

where

$$\tau_k = \begin{cases} 1 & k = 0, \\ \prod_{i=0}^{k-1}(1-\alpha_i) & k \geq 1. \end{cases} \tag{20}$$

Moreover,

$$\tau_k \leq \left(1 - \sqrt{\frac{\mu}{L_f \gamma_{\text{inc}}}}\right)^k. \tag{21}$$

In addition to the geometric convergence of $\phi(x^{(k)})$, this theorem states that the auxiliary sequence $y^{(k)}$ also converges to the unique optimizer $x^\star$ with a geometric rate.

3.1 Proof of Theorem 1

The proof of Theorem 1 is based on the notion of *estimate sequence* developed by Nesterov [Nes04]. We first give its definition and a few lemmas that are necessary for our proof.

Definition 1. *[Nes04, Definition 2.2.1] A pair of sequences $\{V_k(x)\}_{k\geq 0}$ and $\{\tau_k\}_{k\geq 0}$, $\tau_k \geq 0$, is called an estimate sequence of the function $\phi(x)$ if*

$$\tau_k \to 0$$

and for any $x \in \mathbb{R}^n$ and all $k \geq 0$, we have

$$V_k(x) \;\leq\; (1-\tau_k)\phi(x) + \tau_k V_0(x). \tag{22}$$

Lemma 6. *[Nes04, Lemma 2.2.1] Suppose $x^\star$ is an optimal solution to (1). Let the pair $\{V_k(x)\}_{k\geq 0}$ and $\{\tau_k\}_{k\geq 0}$ be an estimate sequence of $\phi(x)$. If we have some sequence $\{x_k\}_{k\geq 0}$ satisfying*

$$\phi(x^{(k)}) \leq V_k^\star := \min_{x\in\mathbb{R}^n} V_k(x), \tag{23}$$

then

$$\phi(x^{(k)}) - \phi(x^\star) \leq \tau_k \left[V_0(x^\star) - \phi(x^\star)\right]. \tag{24}$$

Lemma 7. *Assume that $f(x)$ has Lipschitz continuous gradient and is strongly convex with convexity parameter $\mu_f > 0$. Moreover, assume $0 < \mu \le \mu_f$ and*

1. *$\{y^{(k)}\}_{k\ge 0}$ is an arbitrary sequence in $\mathbb{R}^n$,*
2. *$\{M_k\}_{k\ge 0}$ is a sequence such that $\phi\big(T_{M_k}(y^{(k)})\big) \le \psi_{M_k}\big(y^{(k)}; T_{M_k}(y^{(k)})\big)$,*
3. *$\{\alpha_k\}_{k\ge 0}$ is a sequence that satisfies $\alpha_k \in (0,1)$ and $\sum_{k=0}^{\infty} \alpha_k = \infty$.*

Define the sequence $\{V_k(x)\}_{k\ge 0}$ by letting $V_0(x)$ be an arbitrary function on $\mathbb{R}^n$ and for $k \ge 0$,

$$\begin{aligned} V_{k+1}(x) \;&=\; (1-\alpha_k)V_k(x) \\ &\quad + \alpha_k\left[\phi\big(T_{M_k}(y^{(k)})\big) + \left\langle g_{M_k}(y^{(k)}), x - y^{(k)}\right\rangle + \frac{1}{2M_k}\big\|g_{M_k}(y^{(k)})\big\|_2^2 + \frac{\mu}{2}\|x - y^{(k)}\|_2^2\right], \end{aligned} \tag{25}$$

and define the sequence $\{\tau_k\}_{k\ge 0}$ by setting $\tau_0 = 1$ and

$$\tau_{k+1} = \tau_k(1-\alpha_k), \qquad k \ge 0. \tag{26}$$

Then the pair $\{V_k(x)\}_{k\ge 0}$ and $\{\tau_k\}_{k\ge 0}$ is an estimate sequence of $\phi(x)$.

Proof. First we show that the inequality (22) holds for all $k \ge 0$. It holds for $k = 0$ since $\tau_0 = 1$. Suppose it holds for some $k \ge 0$. Then the assumption on $\{M_k\}_{k\ge 0}$ and Lemma 5 imply

$$\begin{aligned} V_{k+1}(x) &\le (1-\alpha_k)V_k(x) + \alpha_k\phi(x) \\ &= \left(1 - (1-\alpha_k)\tau_k\right)\phi(x) + (1-\alpha_k)\left(V_k(x) - (1-\tau_k)\,\phi(x)\right) \\ &\le \left(1 - (1-\alpha_k)\tau_k\right)\phi(x) + (1-\alpha_k)\tau_k V_0(x) \\ &= (1-\tau_{k+1})\,\phi(x) + \tau_{k+1}V_0(x). \end{aligned}$$

In addition, we note that the sequence $\{\tau_k\}_{k\ge 0}$ defined by (26) is the same as the one given in (20), and the assumptions $\alpha_k \in (0,1)$ and $\sum_{k=0}^{\infty}\alpha_k = \infty$ ensures $\tau_k \to 0$. Therefore, by Definition 1, $\{V_k(x)\}_{k\ge 0}$ and $\{\tau_k\}_{k\ge 0}$ is an estimate sequence of $\phi(x)$. □

Lemma 8. *Let $V_0(x) = \phi(x^{(0)}) + \frac{\mu}{2}\|x - x^{(0)}\|_2^2$ where $x^{(0)}$ is an arbitrary point in $\mathbb{R}^n$. If we choose $\alpha_k = \sqrt{\frac{\mu}{M_k}}$ for $k \ge 0$, then the sequence $\{V_k(x)\}_{k\ge 0}$ defined by (25) can be written as*

$$V_k(x) = V_k^\star + \frac{\mu}{2}\|x - v^{(k)}\|_2^2, \tag{27}$$

where the sequences $\{v^{(k)}\}$ and $\{V_k^\star\}$ are defined as $v^{(0)} = x^{(0)}$, $V_0^\star = \phi(x^{(0)})$, and for $k \ge 0$,

$$v^{(k+1)} \;=\; (1-\alpha_k)v^{(k)} + \alpha_k y^{(k)} - \frac{1}{\alpha_k M_k}g_{M_k}(y^{(k)}), \tag{28}$$

$$\begin{aligned} V_{k+1}^\star \;&=\; (1-\alpha_k)V_k^\star + \alpha_k\phi\big(T_{M_k}(y^{(k)})\big) - \frac{1-\alpha_k}{2M_k}\big\|g_{M_k}(y^{(k)})\big\|_2^2 \\ &\quad + \alpha_k(1-\alpha_k)\left(\frac{\mu}{2}\|y^{(k)} - v^{(k)}\|_2^2 + \left\langle g_{M_k}(y^{(k)}), v^{(k)} - y^{(k)}\right\rangle\right). \end{aligned} \tag{29}$$

Proof. Follow similar algebraic derivations as in [Nes04, Lemma 2.2.3], omitted here. □

In order to prove Theorem 1, we first notice that the three sequences generated by the scAPG method (Algorithms 3 and 4), $\{y^{(k)}\}$, $\{M_k\}$ and $\{\alpha_k\}$, satisfy the assumptions in Lemma 7. More specifically, Lemma 7 does not have any restriction on $\{y^{(k)}\}$, the condition on $\{M_k\}$ is exactly the stopping criterion in Algorithm 3, and also

$$\alpha_k = \sqrt{\frac{\mu}{M_k}} \geq \sqrt{\frac{\mu}{\gamma_{\text{inc}} L_f}} \quad \Longrightarrow \quad \alpha_k \in (0,1), \quad \sum_{k=0}^{\infty} \alpha_k = \infty.$$

Therefore, we can use them to construct an estimate sequence as in (25) and (26). Next we need to show that the choice of $x^{(k+1)} = T_{M_k}(y^{(k)})$ guarantees the condition (23), so that we can invoke Lemma 6 to prove the convergence rate.

To proceed, we split the update of $y^{(k)}$ in (15) into the following two steps:

$$v^{(k)} = x^{(k)} + \frac{1-\alpha_{k-1}}{\alpha_{k-1}}(x^{(k)} - x^{(k-1)}), \tag{30}$$

$$y^{(k)} = \frac{\alpha_k v^{(k)} + x^{(k)}}{\alpha_k + 1}. \tag{31}$$

It is straightforward to check that substituting the expression of $v^{(k)}$ in (30) into (31) yields (15). Also it is no coincidence that we used the same notation $v^{(k)}$ as the minimizer of $V_k(x)$: together with (31), the update of $v^{(k)}$ in (28) is equivalent to (30). To see this, we first check that with the choice of $\alpha_{-1} = 1$ and $x^{(-1)} = x^{(0)}$ in Algorithm 4, it holds that $y^{(0)} = v^{(0)} = x^{(0)}$. Then, with the choice of $x^{(k+1)} = T_{M_k}(y^{(k)})$ for $k \geq 0$, the expression of $v^{(k+1)}$ in (28) becomes

$$\begin{aligned} v^{(k+1)} &= (1-\alpha_k)v^{(k)} + \alpha_k y^{(k)} - \frac{1}{\alpha_k M_k} g_{M_k}(y^{(k)}) \\ &= (1-\alpha_k)v^{(k)} + \alpha_k y^{(k)} - \frac{1}{\alpha_k M_k} M_k(y^{(k)} - x^{(k+1)}) \\ &= (1-\alpha_k)v^{(k)} + \left(\frac{\alpha_k^2 - 1}{\alpha_k}\right) y^{(k)} + \frac{1}{\alpha_k} x^{(k+1)}. \end{aligned}$$

Now replacing $y^{(k)}$ in the above expression with the right-hand side of (31) yields

$$\begin{aligned} v^{(k+1)} &= (1-\alpha_k)v^{(k)} + \left(\frac{\alpha_k^2 - 1}{\alpha_k}\right) \frac{\alpha_k v^{(k)} + x^{(k)}}{\alpha_k + 1} + \frac{1}{\alpha_k} x^{(k+1)} \\ &= x^{(k+1)} + \frac{1-\alpha_k}{\alpha_k}(x^{(k+1)} - x^{(k)}), \end{aligned}$$

which is the same as (30). Therefore, the sequence y_k generated in Algorithm 3 is a convex combination of the current iterate $x^{(k)}$ and $v^{(k)}$, which is the minimizer of the function $V_k(x)$,

Finally, we are ready to prove that (23) holds for all $k \geq 0$. It holds for $k = 0$ simply by the definition of $V_0^\star$. Given that it holds for some k, i.e., $V_k^\star \geq \phi(x^{(k)})$, the expression of $V_{k+1}^\star$ in (29) implies

$$\begin{aligned} V_{k+1}^\star \geq\ & (1-\alpha_k)\phi(x^{(k)}) + \alpha_k \phi(x^{(k+1)}) - \frac{1-\alpha_k}{2M_k} \|g_{M_k}(y^{(k)})\|_2^2 \\ & + \alpha_k(1-\alpha_k)\left\langle g_{M_k}(y^{(k)}), v^{(k)} - y^{(k)} \right\rangle. \end{aligned} \tag{32}$$

According to Lemma 5, we have

$$\begin{aligned}\phi(x^{(k)}) &\geq \phi(x^{(k+1)}) + \left\langle g_{M_k}(y^{(k)}), x^{(k)} - y^{(k)} \right\rangle + \frac{1}{2M_k}\|g_{M_k}(y^{(k)})\|_2^2 + \frac{\mu}{2}\|x^{(k)} - y^{(k)}\|_2^2 \\ &\geq \phi(x^{(k+1)}) + \left\langle g_{M_k}(y^{(k)}), x^{(k)} - y^{(k)} \right\rangle + \frac{1}{2M_k}\|g_{M_k}(y^{(k)})\|_2^2.\end{aligned}$$

Applying this to $\phi(x^{(k)})$ in (32) yields

$$\begin{aligned}V_{k+1}^\star &\geq \phi(x^{(k+1)}) + (1-\alpha_k)\left\langle g_{M_k}(y^{(k)}), \alpha_k(v^{(k)} - y^{(k)}) + x^{(k)} - y^{(k)} \right\rangle \\ &= \phi(x^{(k+1)}) + (1-\alpha_k)\left\langle g_{M_k}(y^{(k)}), \left(\alpha_k v^{(k)} + x^{(k)}\right) - (\alpha_k + 1)y^{(k)} \right\rangle \\ &= \phi(x^{(k+1)}),\end{aligned}$$

where the last equality is due to (31). We have shown that (23) holds for all $k \geq 0$. Therefore, the first result (18) of Theorem 1 follows from Lemma 6 and the definition of $V_0(x)$.

It remains to prove (19). Using strong convexity of ϕ and (18), we have

$$\frac{\mu}{2}\|x^{(k)} - x^\star\|_2^2 \leq \phi(x^{(k)}) - \phi(x^\star) \leq \tau_k\left[\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right] = \tau_k\left(V_0(x^\star) - \phi(x^\star)\right).$$

According to (27),

$$\frac{\mu}{2}\|v^{(k)} - x^\star\|_2^2 = V_k(x^\star) - V_k^\star.$$

Since the relationship $\phi(x^{(k)}) \leq V_k^\star$ implies $\phi(x^\star) \leq V_k^\star$, we have

$$\begin{aligned}\frac{\mu}{2}\|v^{(k)} - x^\star\|_2^2 &\leq V_k(x^\star) - \phi(x^\star) \\ &\leq (1-\tau_k)\phi(x^\star) + \tau_k V_0(x^\star) - \phi(x^\star) \\ &= \tau_k\left(V_0(x^\star) - \phi(x^\star)\right),\end{aligned}$$

where in the second inequality we used the fact that $\{V_k(x)\}$ and $\{\tau_k\}$ is an estimate sequence of $\phi(x)$. Finally, by convexity of the function $\frac{\mu}{2}\|\cdot - x^\star\|_2^2$ and (31),

$$\begin{aligned}\frac{\mu}{2}\|y^{(k)} - x^\star\|_2^2 &\leq \frac{\alpha_k}{\alpha_k+1}\cdot\frac{\mu}{2}\|v^{(k)} - x^\star\|_2^2 + \frac{1}{\alpha_k+1}\cdot\frac{\mu}{2}\|x^{(k)} - x^\star\|_2^2 \\ &\leq \frac{\alpha_k}{\alpha_k+1}\tau_k\left(V_0(x^\star) - \phi(x^\star)\right) + \frac{1}{\alpha_k+1}\tau_k\left(V_0(x^\star) - \phi(x^\star)\right) \\ &= \tau_k\left(V_0(x^\star) - \phi(x^\star)\right) \\ &= \tau_k\left[\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right].\end{aligned}$$

This finishes the proof of Theorem 1.

3.2 The non-blowout property

The convergence results in Theorem 1 requires $\mu \leq \mu_f$. As we discussed in the introduction, such a lower bound on μ_f may be hard to obtain in practice. So we will develop an adaptive method that can automatically estimate μ_f in Section 4. Our estimation scheme involves repetitively calling the scAPG method with some μ without knowing if it satisfies $\mu \leq \mu_f$. Here we show that this will not cause instability or blowout of the algorithm. More precisely, we show that $\phi(x^{(k)}) \leq \phi(x^{(0)})$ for all $k \geq 1$ as long as $\mu \leq L_{\min}$, which can be easily enforced in the algorithm.

Lemma 9. *Suppose $0 < \mu \le L_{\min}$. Then Algorithm 4 guarantees that*

$$\phi(x^{(k+1)}) \le \phi(x^{(k)}) + \frac{M_{k-1}}{2}\left\|x^{(k)} - x^{(k-1)}\right\|_2^2 - \frac{M_k}{2}\left\|x^{(k+1)} - x^{(k)}\right\|_2^2. \tag{33}$$

Proof. According to the optimality of $x^{(k+1)} = T_{M_k}(y^{(k)})$ in minimizing the function $\psi(y^{(k)}, \cdot)$, there exists a $\xi \in \partial\Psi(x^{(k+1)})$ such that

$$\nabla f(y^{(k)}) + \xi + M_k(x^{(k+1)} - y^{(k)}) = 0. \tag{34}$$

Let $\beta_k = \frac{\alpha_k(1-\alpha_{k-1})}{\alpha_{k-1}(1+\alpha_k)}$. Using the assumed property of $f(x)$, we have

$$\begin{aligned}
\phi(x^{(k+1)}) &\le f(y^{(k)}) + \langle \nabla f(y^{(k)}), x^{(k+1)} - y^{(k)}\rangle + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 + \Psi(x^{(k+1)}) \\
&= f(y^{(k)}) + \langle \nabla f(y^{(k)}), x^{(k+1)} - x^{(k)}\rangle \\
&\quad + \langle \nabla f(y^{(k)}), x^{(k)} - y^{(k)}\rangle + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 + \Psi(x^{(k+1)}) \\
&= f(y^{(k)}) - \langle \xi + M_k(x^{(k+1)} - y^{(k)}), x^{(k+1)} - x^{(k)}\rangle \\
&\quad + \langle \nabla f(y^{(k)}), x^{(k)} - y^{(k)}\rangle + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 + \Psi(x^{(k+1)}) \\
&= f(y^{(k)}) + \langle \nabla f(y^{(k)}), x^{(k)} - y^{(k)}\rangle + \Psi(x^{(k+1)}) + \langle \xi, x^{(k)} - x^{(k+1)}\rangle \\
&\quad + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 - M_k\langle x^{(k+1)} - y^{(k)}, x^{(k+1)} - x^{(k)}\rangle \\
&\le f(x^{(k)}) - \frac{\mu_f}{2}\|x^{(k)} - y^{(k)}\|_2^2 + \Psi(x^{(k)}) \\
&\quad + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 - M_k\langle x^{(k+1)} - y^{(k)}, x^{(k+1)} - x^{(k)}\rangle.
\end{aligned}$$

Here, the first inequality is due to the stopping condition for searching M_k in algorithm 4. The first and third equalities are just reorganizing terms while the second one is due to (34). The last inequality are guaranteed by the strong convexity of $f(x)$ and the convexity of $\Psi(x)$. Given that $y^{(k)} = x^{(k)} + \beta_k(x^{(k)} - x^{(k-1)})$, the inequality above implies

$$\begin{aligned}
\phi(x^{(k+1)}) &\le \phi(x^{(k)}) - \frac{\mu_f\beta_k^2}{2}\|x^{(k)} - x^{(k-1)}\|_2^2 + \frac{M_k}{2}\|x^{(k+1)} - x^{(k)} - \beta_k(x^{(k)} - x^{(k-1)})\|_2^2 \\
&\quad - M_k\langle x^{(k+1)} - x^{(k)} - \beta_k(x^{(k)} - x^{(k-1)}), x^{(k+1)} - x^{(k)}\rangle \\
&= \phi(x^{(k)}) + \frac{(M_k - \mu_f)\beta_k^2}{2}\|x^{(k)} - x^{(k-1)}\|_2^2 - \frac{M_k}{2}\|x^{(k+1)} - x^{(k)}\|_2^2.
\end{aligned}$$

Using the fact $\alpha_k^2 M_k = \mu$, we can show that

$$\begin{aligned}
(M_k - \mu_f)\,\beta_k^2 &= (M_k - \mu_f)\,\frac{(1-\alpha_{k-1})^2\alpha_k^2}{(1+\alpha_k)^2\alpha_{k-1}^2} = (M_k - \mu_f)\,\frac{(1-\alpha_{k-1})^2 M_{k-1}}{(1+\alpha_k)^2 M_k} \\
&= \left(1 - \frac{\mu_f}{M_k}\right)\frac{(1-\alpha_{k-1})^2}{(1+\alpha_k)^2}M_{k-1} \le M_{k-1},
\end{aligned}$$

which implies our conclusion. □

Lemma 10. *Suppose $0 < \mu \le L_{\min}$. Then Algorithm 4 guarantees that*

$$\phi(x^{(k+1)}) \le \phi(x^{(0)}) - \frac{M_k}{2}\left\|x^{(k+1)} - x^{(k)}\right\|_2^2. \tag{35}$$

Proof. Applying inequality (33) recursively, we obtain

$$\begin{aligned}\phi(x^{(k+1)}) &\le \phi(x^{(0)}) + \frac{M_{-1}}{2}\|x^{(0)} - x^{(-1)}\|_2^2 - \frac{M_k}{2}\|x^{(k+1)} - x^{(k)}\|_2^2 \\ &= \phi_\lambda(x^{(0)}) - \frac{M_k}{2}\|x^{(k+1)} - x^{(k)}\|_2^2.\end{aligned}$$

Here the last equality holds because $x^{(0)} = x^{(-1)}$. □

The non-blowout property is also critical in our analysis of the homotopy method for solving the ℓ_1-LS problem (Section 5). In particular, it helps to show the sparsity of $x^{(k)}$ once $x^{(0)}$ is sparse.

4 An Adaptive APG method with restart

When applied to strongly convex minimization problems, Nesterov's method in [Nes04, Algorithm (2.2.6)] need to use L_f and μ_f as input parameters. Thanks to the line-search technique, Algorithm 4 does not need to know L_f explicitly. However, it still need to know the convexity parameter μ_f or a lower bound of it in order to guarantee the geometric convergence rate given in Theorem 1.

Compared to searching for L_f, how to estimate μ_f on-the-fly is much more sophisticated. Nesterov [Nes13] suggested a restarting scheme to adjust the estimate of μ_f when it is unknown. This scheme does not require any lower bound of μ_f, and can be shown to have geometric convergence (up to a logarithmic factor). In this section, we adapt this restarting technique to Algorithm 4 and obtain an adaptive APG method. This method has the same convergence guarantees as Nesterov's scheme. However, there are two important differences, which we will elaborate on at the end of this section.

We first describe the basic idea of the restart scheme. If we know $\phi^\star$ (the minimum value of ϕ) and an upper bound of $\|x^{(0)} - x^\star\|_2$, then we can check numerically at each iteration to see if the inequality (18) holds. If this is not the case, then we must have $\mu > \mu_f$, and therefore need to reduce μ and restart the algorithm. However, it is rarely the case that $\phi^\star$ is known. Nevertheless, we can show that if $\mu \le \mu_f$, then the norm of the gradient mapping $g_{M_k}(y^{(k)})$ is also reducing at a geometric rate. Unlike the optimality gap $\phi(x^{(k)}) - \phi^\star$ (which we cannot compute in general), we can compute $\|g_{M_k}(y^{(k)}\|_2$ at each iteration and check explicitly if its expected reduction is achieved. If this is not the case, then we need to reduce μ and restart the algorithm.

The following lemma concerns the geometric decay of the norm of the gradient mapping.

Lemma 11. *Suppose $0 < \mu \le \mu_f$ and the initial point $x^{(0)}$ of Algorithm 4 is obtained by calling Algorithm 1, i.e.,*

$$\{x^{(0)}, M_{-1}, g^{(-1)}, S_{-1}\} \leftarrow \texttt{LineSearch}(x^{\text{ini}}, L_{\text{ini}})$$

with an arbitrary $x^{\text{ini}} \in \mathbb{R}^n$ and $L_{\text{ini}} \ge L_{\min}$. Then, for any $k \ge 0$ in Algorithm 4, we have

$$\left\|g_{M_k}(y^{(k)})\right\|_2 \le 2\sqrt{2\tau_k}\,\frac{M_k}{\mu}\left(1 + \frac{S_{-1}}{M_{-1}}\right)\left\|g^{(-1)}\right\|_2. \tag{36}$$

Proof. By definition of the gradient mapping,

$$\big\|g_{M_k}(y^{(k)})\big\|_2 = \big\|M_k\big(y^{(k)} - x^{(k+1)}\big)\big\|_2 \le M_k\Big(\big\|y^{(k)} - x^\star\big\|_2 + \big\|x^{(k+1)} - x^\star\big\|_2\Big),$$

where $x^\star$ is the unique minimizer of ϕ. By strong convexity of ϕ, we have

$$\frac{\mu}{2}\big\|x^{(k+1)} - x^\star\big\|_2 \le \phi(x^{(k+1)}) - \phi(x^\star).$$

Then using Theorem 1, we obtain

$$\begin{aligned}\|g_{M_k}(y^{(k)})\|_2 &\le M_k\left(\sqrt{2\tau_k} + \sqrt{2\tau_{k+1}}\right)\sqrt{\frac{1}{\mu}\left(\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right)} \\ &\le 2M_k\sqrt{2\tau_k}\sqrt{\frac{1}{\mu}\left(\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right)}. \end{aligned} \tag{37}$$

On the other hand, also by strong convexity of ϕ, we have

$$\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\big\|x^\star - x^{(0)}\big\|_2^2 \le -\left\langle \phi'(x^{(0)}), x^\star - x^{(0)}\right\rangle,$$

where $\phi'(x^{(0)})$ is a subgradient of ϕ at $x^{(0)}$. Since $x^{(0)} = T_{M_{-1}}(x^{\mathrm{ini}})$, $S_{-1} = S_{M_{-1}}(x^{\mathrm{ini}})$ and $g^{(-1)} = g_{M_{-1}}(x^{\mathrm{ini}})$, according to Lemma 2, we have

$$\left\langle \phi'(x^{(0)}), x^\star - x^{(0)}\right\rangle \ge -\left(1 + \frac{S_{-1}}{M_{-1}}\right)\big\|g^{(-1)}\big\|_2 \cdot \big\|x^{(0)} - x^\star\big\|_2.$$

Therefore,

$$\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\big\|x^{(0)} - x^\star\big\|_2^2 \le \left(1 + \frac{S_{-1}}{M_{-1}}\right)\big\|g^{(-1)}\big\|_2 \cdot \big\|x^{(0)} - x^\star\big\|_2.$$

Moreover, by Lemma 3,

$$\big\|x^{(0)} - x^\star\big\|_2 \le \frac{1}{\mu}\left(1 + \frac{S_{-1}}{M_{-1}}\right)\|g^{(-1)}\|_2.$$

The above two inequalities imply

$$\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\big\|x^{(0)} - x^\star\big\|_2^2 \le \frac{1}{\mu}\left(1 + \frac{S_{-1}}{M_{-1}}\right)^2\big\|g^{(-1)}\big\|_2^2.$$

Combining this with the inequality (37) gives the desired result. □

Now we are ready to explain the adaptive APG method presented in Algorithm 5. Similar to the restart method of Nesterov [Nes13, Section 5.3], we evaluate the quality of μ as an estimate of the convexity parameter by checking if the norm of the gradient mapping is reduced sufficiently. Let $\theta_{\mathrm{sc}} \in (0, 1)$ be a desired shrinking factor. we check the following two conditions at each iteration k:

- A: $\big\|g_{M_k}(y^{(k)})\big\|_2 \le \theta_{\mathrm{sc}}\big\|g^{(-1)}\big\|_2$.
- B: $2\sqrt{2\tau_k}\,\frac{M_k}{\mu}\left(1 + \frac{S_{-1}}{M_{-1}}\right) \le \theta_{\mathrm{sc}}$.

Algorithm 5: $\{\hat{x}, \hat{M}, \hat{\mu}\} \leftarrow \texttt{AdapAPG}\,(x^{\rm ini}, L_{\rm ini}, \mu_0, \hat{\epsilon})$

parameters: $L_{\min} \geq \mu_0,\ \ \gamma_{\rm dec} \geq 1,\ \ \gamma_{\rm sc} > 1,\ \ \theta_{\rm sc} \in (0,1)$
$\{x^{(0)}, M_{-1}, g^{(-1)}, S_{-1}\} \leftarrow \texttt{LineSearch}(x^{\rm ini}, L_{\rm ini})$
$x^{(-1)} \leftarrow x^{(0)},\ \ L_{-1} \leftarrow M_{-1},\ \ \mu \leftarrow \mu_0$
$\alpha_{-1} \leftarrow 1,\ \ \tau_0 \leftarrow 1$
$k \leftarrow 0$
repeat
  $\{x^{(k+1)}, M_k, \alpha_k, g^{(k)}, S_k\} \leftarrow \texttt{AccelLineSearch}(x^{(k)}, x^{(k-1)}, L_k, \mu, \alpha_{k-1})$
  $\tau_{k+1} \leftarrow \tau_k(1-\alpha_k)$
  if condition A holds, **then** `// restart from new` $x^{(0)}$ `with same` μ
    $x^{(0)} \leftarrow x^{(k+1)},\ \ x^{(-1)} \leftarrow x^{(k+1)},\ \ L_{-1} = M_k$
    $g^{(-1)} \leftarrow g^{(k)},\ \ M_{-1} \leftarrow M_k,\ \ S_{-1} \leftarrow S_k$
    $k \leftarrow 0,$
  else if condition B holds, **then** `// restart from old` $x^{(0)}$ `with reduced` μ
    $\mu \leftarrow \mu/\gamma_{\rm sc}$
    $k \leftarrow 0$
  else `// continue iteration without restart`
    $L_{k+1} \leftarrow \max\{L_{\min}, M_k/\gamma_{\rm dec}\}$
    $k \leftarrow k+1$
  end
until $\omega(x^{(k+1)}) \leq \hat{\epsilon}$
$\hat{x} \leftarrow x^{(k+1)},\ \ \hat{M} \leftarrow M_k,\ \ \hat{\mu} \leftarrow \mu$

If A is satisfied first, then we restart the iterations with $x^{(k+1)}$ as the new starting point, set $k = 0$, and update the three quantities $g^{(-1)}$, S_{-1} and M_{-1} accordingly (again use $\alpha_{-1} = 1$ and $\tau_0 = 1$). If A is not satisfied but B is satisfied first, it means the μ is larger than μ_f. In fact, if $\mu \leq \mu_f$, according to Lemma 11, we must have

$$\left\|g_{M_k}(y^{(k)})\right\|_2 \ \leq\ 2\sqrt{2\tau_k}\,\frac{M_k}{\mu}\left(1+\frac{S_{-1}}{M_{-1}}\right)\left\|g^{(-1)}\right\|_2 \ \leq\ \theta_{\rm sc}\left\|g^{(-1)}\right\|_2,$$

which implies A. Since A is not satisfied, this contradiction indicates that $\mu > \mu_f$, and we have to reduce μ, say by the factor $\gamma_{\rm sc} > 1$. In this case, we restart Algorithm 4 at the previous $x^{(0)}$ and keep $g^{(-1)}$, S_{-1} and M_{-1} unchanged. Note that, as k increases, at least one of A and B will eventually be satisfied because τ_k converges to zero and $M_k \leq \gamma_{\rm inc} L_f$. Based on this observation, we can analysis the convergence rate of Algorithm 5.

Theorem 2. *Assume $\mu_0 > \mu_f > 0$. Let $g^{\rm ini}$ denotes the first $g^{(-1)}$ computed by Algorithm 5, and $N_{\rm A}$ and $N_{\rm B}$ the number of times that conditions A and B are satisfied, respectively. Then*

$$\begin{aligned} N_{\rm A} &\leq \left\lceil \log_{1/\theta_{sc}}\left(\left(1+\frac{L_f}{L_{\min}}\right)\frac{\|g^{\rm ini}\|_2}{\hat{\epsilon}}\right)\right\rceil, \\ N_{\rm B} &\leq \left\lceil \log_{\gamma_{\rm sc}}\left(\frac{\mu_0}{\mu_f}\right)\right\rceil, \end{aligned}$$

and the total number of iterations of Algorithm 5 is at most

$$(N_{\rm A}+N_{\rm B})\sqrt{\frac{L_f\gamma_{\rm inc}\gamma_{sc}}{\mu_f}}\,\ln\left(8\left(\frac{L_f\gamma_{\rm inc}\gamma_{sc}}{\mu_f\theta_{sc}}\right)^2\left(1+\frac{L_f}{L_{\min}}\right)^2\right). \tag{38}$$

Proof. By Lemma 4 and the facts that $M_k \geq L_{\min}$ and $S_{M_k}(y^{(k)}) \leq L_f$, we have

$$\omega(x^{(k+1)}) \;\leq\; \left(1+\frac{L_f}{L_{\min}}\right)\left\|g_{M_k}(y^{(k)})\right\|_2.$$

According to the stopping criterion $\omega(x^{(k+1)}) \leq \hat{\epsilon}$, the algorithm stops after the condition A is satisfied N_A times if

$$\left(1+\frac{L_f}{L_{\min}}\right)\left\|g_{M_k}(y^{(k)})\right\|_2 \leq \left(1+\frac{L_f}{L_{\min}}\right)\theta_{sc}^{N_{\rm A}}\|g^{\rm ini}\|_2 \leq \hat{\epsilon}.$$

Therefore, N_A is at most $\left\lceil \log_{1/\theta_{sc}}\left(\left(1+\frac{L_f}{L_{\min}}\right)\frac{\|g^{\rm ini}\|_2}{\hat{\epsilon}}\right)\right\rceil$.

Note that condition B can be satisfied only when $\mu > \mu_f$. Once $\mu = \mu_0/\gamma_{sc}^{N_{\rm B}} \leq \mu_f$, it will no longer be satisfied. Therefore, $N_{\rm B}$ is at most $\left\lceil \log_{\gamma_{sc}}\left(\frac{\mu_0}{\mu_f}\right)\right\rceil$ and we always have $\mu \geq \mu_f/\gamma_{\rm sc}$.

Next we bound the number of iterations before either condition A or B must be satisfied. It suffices to find the bound for condition B. For this purpose, we first upper bound the squared left-hand side of condition B:

$$\begin{aligned}
8\tau_k\left(\frac{M_k}{\mu}\right)^2\left(1+\frac{S_{-1}}{M_{-1}}\right)^2 &\leq 8\left(1-\sqrt{\frac{\mu}{L_f\gamma_{\rm inc}}}\right)^k\left(\frac{L_f\gamma_{\rm inc}}{\mu}\right)^2\left(1+\frac{L_f}{L_{\min}}\right)^2 \\
&\leq 8\left(1-\sqrt{\frac{\mu_f/\gamma_{\rm sc}}{L_f\gamma_{\rm inc}}}\right)^k\left(\frac{L_f\gamma_{\rm inc}}{\mu_f/\gamma_{\rm sc}}\right)^2\left(1+\frac{L_f}{L_{\min}}\right)^2.
\end{aligned}$$

Setting the above upper bound be less than $\theta_{\rm sc}^2$, we find that either condition A or B must be satisfied after the following number of iterations:

$$\begin{aligned}
&\ln\left(8\left(\frac{L_f\gamma_{\rm inc}\gamma_{\rm sc}}{\mu_f\theta_{sc}}\right)^2\left(1+\frac{L_f}{L_{\min}}\right)^2\right)\Bigg/\ln\left(1\Big/\left(1-\sqrt{\frac{\mu_f}{L_f\gamma_{\rm inc}\gamma_{\rm sc}}}\right)\right) \\
\leq\;& \sqrt{\frac{L_f\gamma_{\rm inc}\gamma_{sc}}{\mu_f}}\,\ln\left(8\left(\frac{L_f\gamma_{\rm inc}\gamma_{\rm sc}}{\mu_f\theta_{sc}}\right)^2\left(1+\frac{L_f}{L_{\min}}\right)^2\right).
\end{aligned}$$

Hence, the total number iterations of Algorithm 5 is bounded by the above upper bound multiplied by $(N_{\rm A}+N_{\rm B})$. □

If $\mu_0 \leq \mu_f$, then Condition B is never satisfied, i.e., $N_{\rm B} = 0$. In this case, the total number of iterations of Algorithm 5 is bounded by

$$\left\lceil \log_{1/\theta_{sc}}\left(\left(1+\frac{L_f}{L_{\min}}\right)\frac{\|g^{\rm ini}\|_2}{\hat{\epsilon}}\right)\right\rceil\sqrt{\frac{L_f\gamma_{\rm inc}}{\mu_0}}\,\ln\left(8\left(\frac{L_f\gamma_{\rm inc}}{\mu_0\theta_{sc}}\right)^2\left(1+\frac{L_f}{L_{\min}}\right)^2\right),$$

where we have replaced $\mu_f/\gamma_{\rm sc}$ in (38) with μ_0, both of which are lower bound on μ_f.

The total number of iterations given in Theorem 2 is asymptotically

$$O\left(\kappa_f^{1/2}\log(\kappa_f)\log\left(\frac{\kappa_f}{\hat{\epsilon}}\right)\right)+O\left(\kappa_f^{1/2}\log(\kappa_f)\right),$$

where $\kappa_f = L_f/\mu_f$. This is the same complexity as for the restart scheme proposed by Nesterov for his accelerated dual gradient (ADG) method [Nes13, Section 5.3]. Despite using a similar restart scheme and having the same complexity bound, here we elaborate on some important differences between our method from Nesterov's.

- Nesterov's ADG method exploits strong convexity in Ψ instead of f. In order to use it under our assumption (that f is strongly convex), one needs to relocate a strong convexity term from f to Ψ, and this relocated term needs to be adjusted whenever the estimate μ is reduced.
- The restart scheme suggested in [Nes13, Section 5.3] uses two line-searches (Algorithm 1) at each iteration. The first one is to generate the next iterate $x^{(k+1)}$ using $y^{(k)}$. The second one is solely to compute the gradient mapping at $x^{(k+1)}$ in order to determine if any conditions (A or B) of restart are satisfied. Our method directly uses the gradient mapping at $y^{(k)}$ to check the restarting conditions, which does not require the second line-search and thus reduces the computational cost per iteration by half. This saving was made possible by our convergence analysis in Theorem 1, which shows that the auxiliary sequence $y^{(k)}$ also converges geometrically to the optimal solution $x^\star$.

5 Homotopy continuation for sparse optimization

In this section, we focus on the ℓ_1-regularized least-squares (ℓ_1-LS) problem (6), which is a special case of (1) with

$$f(x)=\frac{1}{2}\|Ax-b\|_2^2, \qquad \Psi=\lambda\|x\|_1,$$

where $A\in\mathbb{R}^{m\times n}$ and $b\in\mathbb{R}^m$ are problem data, and λ is a pre-specified regularization parameter. To emphasize its dependency on λ, we define

$$\phi_\lambda(x)\triangleq\frac{1}{2}\|Ax-b\|_2^2+\lambda\|x\|_1.$$

Correspondingly, some of the notations we introduced in the previous sections can be further parametrized by λ. More specifically,

$$\begin{aligned}
\psi_{\lambda,L}(y;x) &= f(y)+\nabla f(y)^T(x-y)+\frac{L}{2}\|x-y\|_2^2+\lambda\|x\|_1\\
T_{\lambda,L}(y) &= \arg\min_x\ \psi_{\lambda,L}(y;x)\\
g_{\lambda,L}(y) &= L\big(y-T_{\lambda,L}(y)\big)\\
\omega_\lambda(x) &= \min_{\xi\in\partial\|x\|_1}\|\nabla f(x)+\lambda\xi\|_\infty\\
S_{\lambda,L}(y) &= \frac{\|\nabla f(T_{\lambda,L}(y))-\nabla f(y)\|_2}{\|T_{\lambda,L}(y)-y\|_2}.
\end{aligned}$$

Algorithm 6: $\hat{x}^{(\mathrm{tgt})} \leftarrow \texttt{APGHomotopy}(A, b, \lambda_{\mathrm{tgt}}, \epsilon, L_0, \hat{\mu}_0)$

input: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^n$, $\lambda_{\mathrm{tgt}} > 0$, $\epsilon > 0$, $L_0 \geq \hat{\mu}_0 > 0$
parameters: $\eta \in (0,1)$, $\delta \in (0,1)$
initialize: $\lambda_0 \leftarrow \|A^T b\|_\infty$, $\hat{x}^{(0)} \leftarrow 0$, $\hat{M}_0 \leftarrow L_0$
$N \leftarrow \lfloor \ln(\lambda_0/\lambda_{\mathrm{tgt}}) \,/\, \ln(1/\eta) \rfloor$
for $K = 0, 1, 2, \ldots, N-1$ **do**
  $\lambda_{K+1} \leftarrow \eta\lambda_K$
  $\hat{\epsilon}_{K+1} \leftarrow \delta\lambda_{K+1}$
  $\{\hat{x}^{(K+1)}, \hat{M}_{K+1}, \hat{\mu}_{K+1}\} \leftarrow \texttt{AdapAPG}\big(\hat{x}^{(K)}, \hat{M}_K, \hat{\mu}_K, \hat{\epsilon}_{K+1}, \lambda_{K+1}\big)$
end
$\{\hat{x}^{(\mathrm{tgt})}, \hat{M}_{\mathrm{tgt}}\} \leftarrow \texttt{AdapAPG}\big(\hat{x}^{(N)}, \hat{M}_N, \hat{\mu}_N, \epsilon, \lambda_{\mathrm{tgt}}\big)$
return $\hat{x}^{(\mathrm{tgt})}$

Similarly, we use $\texttt{AdapAPG}(x^{\mathrm{ini}}, L_{\mathrm{ini}}, \mu_0, \hat{\epsilon}, \lambda)$ to represent applying Algorithm (5) to (6) whose regularization parameter is λ. Given the gradient $\nabla f(x)$, the optimality residue $\omega_\lambda(x)$ can be easily computed with $O(n)$ flops. For the ℓ_1-LS problem, the proximal gradient step, $T_{\lambda,L}(x)$, has the closed-form solution given as

$$T_{\lambda,L}(x) = \mathrm{shrink}\left(x - \frac{1}{L}\nabla f(x),\ \frac{\lambda}{L}\right), \tag{39}$$

where $\mathrm{shrink} : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^n$ is the well-known *shrinkage* or *soft-thresholding* operator, defined as

$$(\mathrm{shrink}(x,\alpha))_i = \mathrm{sgn}(x_i)\max\{|x_i| - \alpha,\ 0\}, \quad i = 1, \ldots, n. \tag{40}$$

We are mainly interested in solving the ℓ_1-LS problem in the context of high-dimensional sparse optimization. In particular, we focus on the case when $m < n$ and the solution $x^\star(\lambda)$ is sparse (which requires the parameter λ to be sufficiently large). As discussed in the introduction, in such a context, the function $f(x) = (1/2)\|Ax - b\|_2^2$ is not strongly convex. Therefore, we only expect a sublinear convergence rate (at least globally) when using first-order optimization methods. For example, we have the iteration complexity $O(L_f/\epsilon)$ when using the PG method, and $O(\sqrt{L_f/\epsilon})$ for the APG methods.

Nevertheless, as explained in Section 1.2, we can use a homotopy continuation strategy to obtain much faster convergence. The key idea is to solve the ℓ_1-LS problem with a large regularization parameter λ_0 first, and then gradually decreases the value of λ until the target regularization is reached. In [XZ12], the PG method (Algorithm 2) is employed to solve the ℓ_1-LS problem for a fixed λ up to an adequate precision, then the solution is used to warm start the next stage. It is shown in [XZ12] that under a restricted eigenvalue condition on A, such a homotopy scheme guarantees that all iterates along the solution path are sufficiently sparse, which implies restricted strong convexity. As a result, a geometric rate of convergence can be established for each homotopy stage, and the overall complexity of the method is $\widetilde{O}(\kappa_s \log(1/\epsilon))$ for an appropriate sparsity level s, where κ_s is a restricted condition number defined in (8), and the notation $\widetilde{O}(\cdot)$ hides additional $\log(\kappa_s)$ factors.

In this section, we show that, by combining the AdapAPG method (Algorithm 5) with the same homotopy continuation scheme, the iteration complexity for solving the ℓ_1-LS problem can be

improved to $\widetilde{O}\big(\sqrt{\kappa_{s'}}\log(1/\epsilon)\big)$, with a slightly larger sparsity level s'. The APG homotopy method is presented in Algorithm 6. To avoid confusion over the notations, we use $\lambda_{\rm tgt}$ to denote the target regularization parameter. The method starts with

$$\lambda_0 = \|A^T b\|_\infty,$$

since this is the smallest value for λ such that the ℓ_1-LS problem has the trivial solution 0 (by examining the optimality condition). Our method has two parameters $\eta \in (0,1)$ and $\delta \in (0,1)$. They control the algorithm as follows:

- The sequence of values for the regularization parameter is determined as $\lambda_k = \eta^k \lambda_0$ for $k = 1, 2, \ldots$, until the target value $\lambda_{\rm tgt}$ is reached.
- For each λ_k except $\lambda_{\rm tgt}$, we solve problem (6) with a proportional precision $\delta\lambda_k$. For the last stage with $\lambda_{\rm tgt}$, we solve to the absolute precision ϵ.

Our convergence analysis of the APG homotopy method is based on the following assumption, which involves the restricted eigenvalues defined in (7).

Assumption 1. *Suppose $b = A\bar{x} + z$. Let $\bar{S} = \mathrm{supp}(\bar{x})$ and $\bar{s} = |\bar{S}|$. There exist $\gamma > 0$ and $\delta' \in (0, 0.2]$ such that $\gamma > (1+\delta')/(1-\delta')$ and*

$$\lambda_{\rm tgt} \;\geq\; \max\left\{2,\; \frac{\gamma+1}{(1-\delta')\gamma - (1+\delta')}\right\} 4\,\|A^T z\|_\infty. \tag{41}$$

Moreover, there exists an integer $\tilde{s}$ such that $\rho_-(A, \bar{s}+3\tilde{s}) > 0$ and

$$\tilde{s} \;>\; \frac{24\big(\gamma_{\rm inc}\rho_+(A, \bar{s}+3\tilde{s}) + 3\rho_+(A,\tilde{s})\big)}{\rho_-(A, \bar{s}+\tilde{s})}(1+\gamma)\bar{s}. \tag{42}$$

We also assume that $L_{\min} \leq \gamma_{\rm inc}\rho_+(A, \bar{s}+3\tilde{s})$.

As we will see later, the quantity δ' in the above assumption is related to the parameter δ in Algorithm 6, and γ defines a conic condition on $x - \bar{x}$, i.e.,

$$\|(x-\bar{x})_{\bar{S}^c}\|_1 \leq \gamma \|(x-\bar{x})_{\bar{S}}\|_1,$$

which holds whenever $\omega_\lambda(x) \leq \delta'\lambda$. According to [ZH08], the above assumption implies that the solution $x^\star(\lambda)$ is sparse whenever $\lambda \geq \lambda_{\rm tgt}$; more specifically, $\|x^\star(\lambda)_{\bar{S}^c}\|_0 \leq \tilde{s}$ (here $\bar{S}^c$ denotes the complement of the support set $\bar{S}$). We will show that by choosing the parameters η and δ in Algorithm 6 appropriately, these conditions also imply that all iterates along the solution path are sparse. We note that Assumption 1 is very similar to Assumption 1 in [XZ12], and the interpretations and remarks made there also apply here. We repeats some important points here:

- The existence of $\tilde{s}$ satisfying the conditions like (42) is necessary and standard in sparse recovery analysis. This is closely related to the restricted isometry property (RIP) of [CT05] which assumes that there exist some $s > 0$, and $\nu \in (0,1)$ such that $\kappa(A,s) < (1+\nu)/(1-\nu)$. See [XZ12, Section 3] for an example of sufficient RIP conditions.

- Our RIP-like condition (42) can be much stronger than the corresponding conditions established in the sparse recovery literature (see, e.g., [LM11] and references therein), which are only concerned about the recovery property of the optimal solution $x^\star$. In contrast, our condition needs to guarantee sparsity for all iterates along the solution path, thus is "dynamic" in nature. In particular, in addition to the matrix A, our RIP-like condition (42) also depends on algorithmic parameters γ_{inc}, η and δ (Theorem 4 will relate η to δ and δ'). For example, if we allow $\delta' \in (0, 1)$, then we need to increase the constant in (42) from 24 to 48 for the convergence results in this section to hold.

- If $L_{\min} > \gamma_{\text{inc}}\rho_+(A, \bar{s} + 3\tilde{s})$, then we may simply replace $\gamma_{\text{inc}}\rho_+(A, \bar{s} + 3\tilde{s})$ by $L_{\min}$ in the assumption, and all theorem statements hold with $\gamma_{\text{inc}}\rho_+(A, \bar{s} + 3\tilde{s})$ replaced by $L_{\min}$. Nevertheless in practice, it is natural to simply pick

$$L_{\min} = \rho_+(A, 1) = \max_{i \in \{1,\ldots,n\}} \|A_i\|_2^2,$$

  where A_i is the i-th column of A. It automatically satisfies the condition $L_{\min} \leq \rho_+(A, \bar{s}+3\tilde{s})$.

Our first result below concerns the local geometric convergence of Algorithm 5 when applied to solve the ℓ_1-LS problem. Recall that we have $\mu_f = 0$ under the assumption $m < n$, therefore global geometric convergence starting from an arbitrary point cannot be established. Nevertheless, if the starting point $x^{(0)}$ is sparse and the optimality condition is satisfied with adequate precision, then all iterates along the solution path are sparse. This implies that restricted strong convexity holds and Algorithm 5 actually has geometric convergence.

Theorem 3. *Suppose Assumption 1 holds. If the initial point x^{ini} in Algorithm 5 satisfies*

$$\left\|x^{\text{ini}}_{\bar{S}^c}\right\|_0 \leq \tilde{s}, \qquad \omega_\lambda(x^{\text{ini}}) \leq \delta'\lambda, \tag{43}$$

then for all $k \geq 0$, we have $\left\|x^{(k)}_{\bar{S}^c}\right\|_0 \leq \tilde{s}$. Moreover, all the three conclusions of Theorem 2 holds by replacing L_f and μ_f with $\rho_+(A, \bar{s} + 3\tilde{s})$ and $\rho_-(A, \bar{s} + 3\tilde{s})$, respectively.

Our next result gives the overall iteration complexity of the APG homotopy method in Algorithm 6. To simplify presentation, we let $s' = \bar{s} + 3\tilde{s}$, and use the following notations:

$$\begin{aligned} \rho_+(s') &= \rho_+(A, \bar{s} + 3\tilde{s}), \\ \rho_-(s') &= \rho_-(A, \bar{s} + 3\tilde{s}), \\ \kappa_{s'} &= \kappa(A, \bar{s} + 3\tilde{s}) = \frac{\rho_+(A, \bar{s} + 3\tilde{s})}{\rho_-(A, \bar{s} + 3\tilde{s})}. \end{aligned}$$

Roughly speaking, if the parameters δ and η are chosen appropriately, then the total number of proximal-gradient steps for finding an ϵ-optimal solution is $\widetilde{O}(\sqrt{\kappa_{s'}}\ln(1/\epsilon))$, which has a weaker dependence on the restricted condition number than the PGH method.

Theorem 4. *Suppose Assumption 1 holds for some δ', γ and $\tilde{s}$, and the parameters δ and η in Algorithm 6 are chosen such that*

$$\frac{1+\delta}{1+\delta'} \leq \eta < 1.$$

Let $N = \left\lfloor \ln(\lambda_0/\lambda_{\text{tgt}}) / \ln \eta^{-1} \right\rfloor$ as in the algorithm. Then:

1. *The condition (43) holds for each call of Algorithm 5. For $K = 0, \ldots, N-1$, the number of proximal-gradient steps in each call of Algorithm 5 is no more than*

$$\left(\log_{\frac{1}{\theta_{sc}}}\left(\frac{C}{\delta}\right) + D\right)\sqrt{\kappa_{s'}\gamma_{\text{inc}}\gamma_{sc}}\ \ln\left(8\left(\frac{\kappa_{s'}\gamma_{\text{inc}}\gamma_{sc}}{\theta_{sc}}\right)^2\left(1+\frac{\rho_+(s')}{L_{\min}}\right)^2\right),$$

where

$$C = \left(1+\frac{\rho_+(s')}{L_{\min}}\right)\sqrt{8\gamma_{\text{inc}}\kappa_{s'}(1+\gamma)\bar{s}}, \qquad D = \left\lceil \log_{\gamma_{sc}}\left(\frac{\hat{\mu}_0}{\rho_-(s')}\right)\right\rceil + 1.$$

Note that this bound is independent of λ_K.

2. *For $K = 0, \ldots, N-1$, the outer-loop iterates $\hat{x}^{(K)}$ satisfies*

$$\phi_{\lambda_{\text{tgt}}}(\hat{x}^{(K)}) - \phi^\star_{\lambda_{\text{tgt}}} \le \eta^{2(K+1)}\,\frac{4.5(1+\gamma)\lambda_0^2\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}, \tag{44}$$

and the following bound on sparse recovery performance holds

$$\|\hat{x}^{(K)} - \bar{x}\|_2 \le \eta^{K+1}\,\frac{2\lambda_0\sqrt{\bar{s}}}{\rho_-(A, \bar{s}+\tilde{s})}.$$

3. *When Algorithm 6 terminates, the total number of proximal-gradient steps is $\widetilde{O}\left(\sqrt{\kappa_{s'}}\ln(1/\epsilon)\right)$, where the notation $\widetilde{O}(\cdot)$ hides additional $\ln(\kappa_{s'})$ factors. Moreover, the output $\hat{x}^{(\text{tgt})}$ satisfies*

$$\phi_{\lambda_{\text{tgt}}}(\hat{x}^{(\text{tgt})}) - \phi^\star_{\lambda_{\text{tgt}}} \le \frac{4(1+\gamma)\lambda_{\text{tgt}}\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}\,\epsilon.$$

We note that even if we solve each homotopy stage to the same high precision as the final stage, i.e., setting $\hat{\epsilon}_{K+1} = \min(\epsilon, \delta\lambda_{K+1})$, the global convergence rate is still near geometric, and the total number of proximal-gradient steps is no more than $\widetilde{O}\left(\sqrt{\kappa_{s'}}\,(\ln(1/\epsilon))^2\right)$.

The rest of this section is devoted to the proofs of the above convergence results. In Section 5.1 we recall and adapt several lemmas from [XZ12] that are necessary for us to show that all the iterates along the solution path of the homotopy method is sparse. Then Sections 5.2 and 5.3 contain the proofs for Theorems 3 and 4, respectively.

5.1 Sparsity along the solution path

First, we list some useful inequalities that are direct consequences of (41) and $\delta' \in (0, 0.2]$:

$$(1-\delta')\lambda - 4\|A^T z\|_\infty > 0 \tag{45}$$

$$(1+\delta')\lambda + \|A^T z\|_\infty \le 1.4\lambda \tag{46}$$

$$\lambda + \|A^T z\|_\infty \le (1.4-\delta')\lambda \tag{47}$$

$$\frac{(1+\delta')\lambda + \|A^T z\|_\infty}{(1-\delta')\lambda - \|A^T z\|_\infty} \le \gamma. \tag{48}$$

The following result means that if x is sparse, and it satisfies an approximate optimality condition for minimizing ϕ_λ, then $\phi_\lambda(x)$ is not much larger than $\phi_\lambda(\bar{x})$.

Lemma 12 (Lemma 4 in [XZ12]). *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$, and $\lambda \geq \lambda_{\mathrm{tgt}}$. If x is sparse, i.e., $\|x_{\bar{S}^c}\|_0 \leq \tilde{s}$, and it satisfies the approximate optimality condition*

$$\min_{\xi \in \partial \|x\|_1} \left\| A^T(Ax - b) + \lambda \xi \right\|_\infty \leq \delta' \lambda, \tag{49}$$

then we have

$$\|(x - \bar{x})_{\bar{S}^c}\|_1 \leq \gamma \|(x - \bar{x})_{\bar{S}}\|_1 \tag{50}$$

and

$$\|x - \bar{x}\|_2 \leq \frac{1.4\,\lambda\sqrt{\bar{s}}}{\rho_-(A, \bar{s} + \tilde{s})} \tag{51}$$

and

$$\phi_\lambda(x) \leq \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2 \bar{s}}{\rho_-(A, \bar{s} + \tilde{s})}. \tag{52}$$

The next lemma means that if x is sparse, and $\phi_\lambda(x)$ is not much larger than $\phi_\lambda(\bar{x})$, then both $\|x - \bar{x}\|_2$ and $\|x - \bar{x}\|_1$ are small.

Lemma 13 (Lemma 5 in [XZ12]). *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$, and $\lambda \geq \lambda_{\mathrm{tgt}}$. Consider x such that*

$$\|x_{\bar{S}^c}\|_0 \leq \tilde{s}, \qquad \phi_\lambda(x) \leq \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2 \bar{s}}{\rho_-(A, \bar{s} + \tilde{s})},$$

then

$$\max\left\{ \frac{1}{2.8\,\lambda} \|A(x - \bar{x})\|_2^2,\ \|x - \bar{x}\|_1 \right\} \leq \frac{1.4\,(1+\gamma)\lambda \bar{s}}{\rho_-(A, \bar{s} + \tilde{s})}.$$

The next lemma implies that if both $x^{(k)}$ and $x^{(k-1)}$ are sparse and their objective values are not much larger than $\phi_\lambda(\bar{x})$, then the next iterate $x^{(k+1)}$ generated by the accelerated line search procedure (Algorithm 3) is also sparse. Its proof uses similar arguments as in [XZ12, Lemma 6].

Lemma 14. *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$, and $\lambda \geq \lambda_{\mathrm{tgt}}$. Suppose x and x' satisfies*

$$\begin{aligned} &\|x_{\bar{S}^c}\|_0 \leq \tilde{s}, \qquad \phi_\lambda(x) \leq \phi_\lambda(\bar{x}) + \tfrac{2\delta'(1+\gamma)\lambda^2 \bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}, \\ &\|x'_{\bar{S}^c}\|_0 \leq \tilde{s}, \qquad \phi_\lambda(x') \leq \phi_\lambda(\bar{x}) + \tfrac{2\delta'(1+\gamma)\lambda^2 \bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}, \end{aligned} \tag{53}$$

and $y = x + \beta(x - x')$ with $0 \leq \beta \leq 1$. Then for any $L < \gamma_{\mathrm{inc}} \rho_+(A, \bar{s} + 3\tilde{s})$, we have

$$\left\| \left(T_{\lambda, L}(y)\right)_{\bar{S}^c} \right\|_0 < \tilde{s}.$$

Proof. Recall that $T_{\lambda,L}$ can be computed by the soft-thresholding operator as in (39). That is,

$$(T_L(y))_i = \mathrm{sgn}(\tilde{y}_i) \max\left\{ |\tilde{y}_i| - \frac{\lambda}{L},\ 0 \right\}, \qquad i = 1, \ldots, n,$$

where

$$\tilde{y} = y - \frac{1}{L} A^T(Ay - b) = y - \frac{1}{L} A^T A(y - \bar{x}) + \frac{1}{L} A^T z.$$

In order to upper bound the number of nonzero elements in $(T_L(y))_{\bar{S}^c}$, we split the truncation threshold λ/L on elements of $\tilde{y}_{\bar{S}^c}$ into three parts:

- $0.175\,\lambda/L$ on elements of $y_{\bar{S}^c}$,
- $0.125\,\lambda/L$ on elements of $(1/L)A^Tz$, and
- $0.7\,\lambda/L$ on elements of $(1/L)A^TA(y-\bar{x})$.

Since by assumption $\|A^Tz\|_\infty \le \lambda/8$, we have $\big|\{j : ((1/L)A^Tz)_j > 0.125\,\lambda/L\}\big| = 0$. Therefore,

$$\big\|\big(T_{\lambda,L}(y)\big)_{\bar{S}^c}\big\|_0 \;\le\; \big|\{j \in \bar{S}^c : |y_j| > 0.175\,\lambda/L\}\big| + \big|\{j : \big|\big(A^TA(y-\bar{x})\big)_j\big| \ge 0.7\,\lambda\}\big|.$$

Note that

$$\begin{aligned}
\big|\{j \in \bar{S}^c : |y_j| \ge 0.175\,\lambda/L\}\big| &= \big|\{j \in \bar{S}^c : |(y-\bar{x})_j| \ge 0.175\,\lambda/L\}\big| \\
&\le \big|\{j : |(y-\bar{x})_j| \ge 0.175\,\lambda/L\}\big| \\
&\le L(0.175\,\lambda)^{-1}\|y-\bar{x}\|_1 \\
&\le L(0.175\,\lambda)^{-1}((1+\beta)\|x-\bar{x}\|_1 + \beta\|x'-\bar{x}\|_1) \\
&\le \frac{1.4\,L(1+2\beta)(1+\gamma)\lambda\bar{s}}{0.175\,\lambda\rho_-(A,\bar{s}+\tilde{s})} \qquad (54)\\
&\le \frac{24\,L(1+\gamma)\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}, \qquad (55)
\end{aligned}$$

where the second-to-the-last inequality follows from Lemma 13, and the last one used $\beta \in [0,1]$.

For the last part, consider S with maximum size $s = |S| \le \tilde{s}$ such that

$$S \subset \{j : |(A^TA(y-\bar{x}))_j| \ge 0.7\,\lambda\}.$$

Then there exists u such that $\|u\|_\infty = 1$ and $\|u\|_0 = s$, and $0.7\,s\lambda \le u^TA^TA(y-\bar{x})$. Moreover,

$$0.7\,s\lambda \;\le\; u^TA^TA(y-\bar{x}) \;\le\; \|Au\|_2\|A(y-\bar{x})\|_2 \;\le\; \sqrt{\rho_+(A,s)}\sqrt{s}(1+2\beta)\sqrt{\frac{2\cdot 1.4^2\,(1+\gamma)\lambda^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}},$$

where the last inequality again follows from Lemma 13. Taking squares of both sides of the above inequality gives

$$s \le \frac{8\,\rho_+(A,s)(1+\gamma)\bar{s}(1+2\beta)^2}{\rho_-(A,\bar{s}+\tilde{s})} \le \frac{72\,\rho_+(A,\tilde{s})(1+\gamma)\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})} < \tilde{s},$$

where the last inequality is due to (42). Since $s = |S|$ achieves the maximum possible value such that $s \le \tilde{s}$ for any subset S of $\{j : |(A^TA(y-\bar{x}))_j| \ge 0.7\,\lambda\}$, and the above inequality shows that $s < \tilde{s}$, we must have

$$S = \{j : |(A^TA(y-\bar{x}))_j| \ge 0.7\,\lambda\},$$

and thus

$$s = \big|\{j : |(A^TA(y-\bar{x}))_j| \ge 0.7\,\lambda\}\big| \le \left\lfloor \frac{72\,\rho_+(A,\tilde{s})(1+\gamma)\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})} \right\rfloor.$$

Finally, combining the above bound with the bound in (55) gives

$$\big\|\big(T_{\lambda,L}(x)\big)_{\bar{S}^c}\big\|_0 \;\le\; \frac{24\,(L+3\rho_+(A,\tilde{s}))}{\rho_-(A,\bar{s}+\tilde{s})}(1+\gamma)\bar{s}.$$

Under the assumption $L < \gamma_{\text{inc}}\rho_+(A,\bar{s}+3\tilde{s})$ and (42), the right-hand side of the above inequality is less than $\tilde{s}$. This proves the desired result. □

5.2 Proof of Theorem 3

According to (14), the PG method keeps the value of objective function decreasing monotonically. This is the key property for the PGH method in [XZ12] to enforce all the iterates along the solution path to be sufficiently sparse. Unfortunately, the scAPG and AdapAPG methods do not have such a monotone decreasing property. As an alternative, we proved that they have a non-blowout property (Lemma 10); that is, the objective value at any intermediate step will not exceed the initial objective value. This is the key in showing that all the iterates along the solution path are sufficiently sparse for the AdapAPG method, provided that the initial point is sparse and not far from optimality.

Lemma 15. *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$. In addition, assume $\lambda \geq \lambda_{\mathrm{tgt}}$ and $\mu \leq L_{\min}$. If the initial point x^{ini} in Algorithm 5 satisfies*

$$\left\|x^{\mathrm{ini}}_{\bar{S}^c}\right\|_0 \leq \tilde{s}, \qquad \omega_\lambda(x^{\mathrm{ini}}) \leq \delta'\lambda,$$

then for all $k \geq 0$, we have

$$\left\|x^{(k)}_{\bar{S}^c}\right\|_0 \leq \tilde{s}, \qquad \left\|y^{(k)}_{\bar{S}^c}\right\|_0 \leq 2\tilde{s}.$$

Proof. According to Lemma 12, the assumptions on x^{ini} implies

$$\phi_\lambda(x^{\mathrm{ini}}) \leq \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}.$$

Because $x^{(0)} = T_{\lambda,M}(x^{\mathrm{ini}})$, we have $\phi_\lambda(x^{(0)}) \leq \phi_\lambda(x^{\mathrm{ini}})$ so that

$$\phi_\lambda(x^{(0)}) \leq \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}.$$

Although Algorithm 5 is not monotone decreasing, the non-blowout property in Lemma 10 guarantees that, for all $k \geq 0$,

$$\phi_\lambda(x^{(k+1)}) \leq \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}.$$

Because $\left\|x^{\mathrm{ini}}_{\bar{S}^c}\right\|_0 \leq \tilde{s}$, we have $\left\|x^{(-1)}_{\bar{S}^c}\right\|_0 = \left\|x^{(0)}_{\bar{S}^c}\right\|_0 \leq \tilde{s}$ according to Lemma 14. Suppose $\left\|x^{(k)}_{\bar{S}^c}\right\|_0 \leq \tilde{s}$ and $\left\|x^{(k-1)}_{\bar{S}^c}\right\|_0 \leq \tilde{s}$. Since $y^{(k+1)} = x^{(k)} + \beta_k(x^{(k)} - x^{(k-1)})$ and $x^{(k+1)} = T_{M_k}(y^{(k)})$, Lemma 14 again implies $\left\|x^{(k+1)}_{\bar{S}^c}\right\|_0 \leq \tilde{s}$. By induction, we have $\left\|x^{(k)}_{\bar{S}^c}\right\|_0 \leq \tilde{s}$ holds for all k, which further implies $\left\|y^{(k)}_{\bar{S}^c}\right\|_0 \leq 2\tilde{s}$ for all k. □

According to Lemma 12, under the condition (43), Algorithm 5 essentially operates only on vectors with at most either $\tilde{s}$ or $2\tilde{s}$ nonzero components. Therefore, we are solving the ℓ_1-LS problem restricted in a sparse subspace, where the restricted smoothness and restricted strong convexity are available, that is,

$$f(x) \;\geq\; f(y) + \langle \nabla f(y), x-y\rangle + \frac{\rho_-(A, \bar{s}+3\tilde{s})}{2}\|x-y\|_2^2,$$

$$f(x) \;\leq\; f(y) + \langle \nabla f(y), x-y\rangle + \frac{\rho_+(A, \bar{s}+3\tilde{s})}{2}\|x-y\|_2^2.$$

Here, the effective sparse level is $s' = \bar{s} + 3\tilde{s}$ because when the above two inequalities are used in Section 3 and Section 4, they are always applied to x and y with $\left\|x_{\bar{S}^c}\right\|_0 \leq \tilde{s}$ and $\left\|y_{\bar{S}^c}\right\|_0 \leq 2\tilde{s}$. To show Theorem 3, we just need to repeat the proof of Theorem 2 by replacing L_f and μ_f with $\rho_+(A, \bar{s}+3\tilde{s})$ and $\rho_-(A, \bar{s}+3\tilde{s})$, respectively.

5.3 Proof of Theorem 4

In Algorithm 6, $\hat{x}^{(K)}$ denotes an approximate solution for minimizing the function ϕ_{λ_K}. A key idea of the APG homotopy method is to use $\hat{x}^{(K)}$ as the starting point in the AdapAPG method for minimizing the next function $\phi_{\lambda_{K+1}}$. The following lemma shows that if we choose the parameters δ and η appropriately, then $\hat{x}^{(K)}$ satisfies the approximate optimality condition for λ_{K+1} that guarantees local geometric convergence.

Lemma 16 (Lemma 7 in [XZ12]). *Suppose $\hat{x}^{(K)}$ satisfies the approximate optimality condition*

$$\omega_{\lambda_K}(\hat{x}^{(K)}) \leq \delta\lambda_K$$

for some $\delta < \delta'$. Let $\lambda_{K+1} = \eta\lambda_K$ for some η that satisfies

$$\frac{1+\delta}{1+\delta'} \leq \eta < 1. \tag{56}$$

Then we have

$$\omega_{\lambda_{K+1}}(\hat{x}^{(K)}) \leq \delta'\lambda_{K+1}.$$

Lemma 17 (Lemma 8 in [XZ12]). *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$, and $\lambda \geq \lambda_{\mathrm{tgt}}$. If x satisfies*

$$\omega_\lambda(x) \leq \delta'\lambda,$$

then for all $\lambda' \in [\lambda_{\mathrm{tgt}}, \lambda]$, we have

$$\phi_{\lambda'}(x) - \phi_{\lambda'}(x^\star(\lambda')) \leq \frac{2(1+\gamma)(\lambda+\lambda')(\omega_\lambda(x)+\lambda-\lambda')\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}.$$

Now we are ready to give an estimate of the overall complexity of the APG homotopy method (Algorithm 6). First, we need to bound the number of iterations within each call of Algorithm 5. According to Theorem 3 and Theorem 2, the total number of iterations in each call of $\texttt{AdapAPG}\big(\hat{x}^{(K)}, \hat{M}_K, \hat{\mu}_K, \hat{\epsilon}_{K+1}, \lambda_{K+1}\big)$ is no more than

$$(N_A + N_B)\sqrt{\kappa_{s'}\gamma_{\mathrm{inc}}\gamma_{sc}}\ \ln\left(8\left(\frac{\kappa_{s'}\gamma_{\mathrm{inc}}\gamma_{sc}}{\theta_{sc}}\right)^2\left(1+\frac{\rho_+(s')}{L_{\min}}\right)^2\right), \tag{57}$$

where N_A is the number of times that condition A is satisfied first, which is bounded as

$$N_A \leq \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\|g_{\lambda_{K+1},M}(\hat{x}^{(K)})\|_2}{\hat{\epsilon}}\right)\right\rceil$$

with M generated from $\{x^{(0)}, M, g^{(-1)}, S_{-1}\} \leftarrow \texttt{LineSearch}(\hat{x}^{(K)}, \hat{M}^{(K)})$, and N_B is the number of times that condition B is satisfied first, which is bounded as

$$N_B \ \leq\ \left\lceil \log_{\gamma_{sc}}\left(\frac{\hat{\mu}_K}{\rho_-(s')}\right)\right\rceil \ \leq\ \left\lceil \log_{\gamma_{sc}}\left(\frac{\hat{\mu}_0}{\rho_-(s')}\right)\right\rceil.$$

The bound on N_A depends on $\|g_{\lambda_{K+1},M}(\hat{x}^{(K)})\|_2$, which we can further bound using Lemma 1 to obtain

$$\begin{aligned}\left\|g_{\lambda_{K+1},M}(\hat{x}^{(K)})\right\|_2^2 &\leq 2M\left(\phi_{\lambda_{K+1}}(\hat{x}^{(K)}) - \phi^\star_{\lambda_{K+1}}\right)\\ &\leq 2\gamma_{\text{inc}}\rho_+(s')\left(\phi_{\lambda_{K+1}}(\hat{x}^{(K)}) - \phi^\star_{\lambda_{K+1}}\right),\end{aligned}$$

where $\phi^\star_{\lambda_{K+1}} = \min_x \phi_{\lambda_{K+1}}(x)$. We still need to bound the gap $\phi_{\lambda_{K+1}}(\hat{x}^{(K)}) - \phi^\star_{\lambda_{K+1}}$. Since Lemma 16 implies that $\omega_{\lambda_{K+1}}(\hat{x}^{(K)}) \leq \delta'\lambda_{K+1}$, we can obtain directly from Lemma 17 the following inequality by setting $\lambda' = \lambda = \lambda_{K+1}$ and $x = \hat{x}^{(K)}$:

$$\phi_{\lambda_{K+1}}(\hat{x}^{(K)}) - \phi^\star_{\lambda_{K+1}} \;\leq\; \frac{4(1+\gamma)\lambda_{K+1}^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})} \leq \frac{4(1+\gamma)\lambda_{K+1}^2\bar{s}}{\rho_-(s')}.$$

Therefore, the bound on N_A can be relaxed as

$$\begin{aligned}N_A &\leq \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\|g_{\lambda_{K+1},M}(\hat{x}^{(K)})\|_2}{\delta\lambda_{K+1}}\right)\right\rceil\\ &\leq \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\sqrt{2\gamma_{\text{inc}}\rho_+(s')\left(\phi_{\lambda_{K+1}}(\hat{x}^{(K)}) - \phi^\star_{\lambda_{K+1}}\right)}}{\delta\lambda_{K+1}}\right)\right\rceil\\ &\leq \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\sqrt{8\gamma_{\text{inc}}\rho_+(s')(1+\gamma)\lambda_{K+1}^2\bar{s}}}{\delta\lambda_{K+1}\sqrt{\rho_-(s')}}\right)\right\rceil\\ &= \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\sqrt{8\gamma_{\text{inc}}\kappa_{s'}(1+\gamma)\bar{s}}}{\delta}\right)\right\rceil.\end{aligned}$$

Combining the above bounds on N_A and N_B with (57) yields Part 1 of Theorem 4. We note that this bound is independent of λ_{K+1}.

In the homotopy method (Algorithm 6), after K outer iterations for $K \leq N-1$, we have from Lemma 16 that $\omega_{\lambda_{K+1}}(\hat{x}^{(K)}) \leq \delta'\lambda_{K+1}$. The sparse recovery performance bound

$$\|\hat{x}^{(K)} - \bar{x}\|_2 \leq 2\eta^{K+1}\lambda_0\sqrt{\bar{s}}/\rho_-(A,\bar{s}+\tilde{s})$$

follows directly from Lemma 12 and $\lambda_{K+1} = \eta^{K+1}\lambda_0$. Moreover, from Lemma 17 with $\lambda' = \lambda_{\text{tgt}}$, $\lambda = \lambda_{K+1}$, and $x = \hat{x}^{(K)}$, we obtain

$$\phi_{\lambda_{\text{tgt}}}(\hat{x}^{(K)}) - \phi^\star_{\lambda_{\text{tgt}}} \leq \frac{4.5(1+\gamma)\lambda_{K+1}^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})} = \eta^{2(K+1)}\frac{4.5(1+\gamma)\lambda_0^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}.$$

This proves Part 2 of Theorem 4.

In Algorithm 6, the number of homotopy stages, excluding the last one for λ_{tgt}, is

$$N = \left\lfloor \frac{\ln(\lambda_0/\lambda_{\text{tgt}})}{\ln(1/\eta)} \right\rfloor.$$

The last iteration for λ_{tgt} uses an absolute precision ϵ instead of the relative precision $\delta\lambda_{\mathrm{tgt}}$. Therefore, the overall complexity is bounded by

$$\left(\frac{\ln(\lambda_0/\lambda_{\mathrm{tgt}})}{\ln(1/\eta)}\left(\log_{\frac{1}{\theta_{sc}}}\left(\frac{C}{\delta}\right)+D\right)+\log_{\gamma_{sc}}\max\left(1,\frac{\lambda_{\mathrm{tgt}}C}{\epsilon}\right)+D\right)\sqrt{\kappa_{s'}\gamma_{\mathrm{inc}}\gamma_{sc}}\,\ln\left(8\left(\frac{\kappa_{s'}\gamma_{\mathrm{inc}}\gamma_{sc}}{\theta_{sc}}\right)^2\left(1+\frac{\rho_+(s')}{L_{\min}}\right)^2\right),$$

which is $\widetilde{O}\left(\sqrt{\kappa_{s'}}\ln(1/\epsilon)\right)$. Finally, when Algorithm 6 terminates, we have $\omega_{\lambda_{\mathrm{tgt}}}(\hat{x}^{(\mathrm{tgt})})\le\epsilon$. Therefore we can apply Lemma 17 with $\lambda=\lambda'=\lambda_{\mathrm{tgt}}$ and $x=\hat{x}^{(\mathrm{tgt})}$ to obtain the last desired bound in Part 3 of Theorem 4.

6 Numerical experiments

In this section, we present preliminary numerical experiments to support our theoretical analysis. Here our focus is on illustrating the convergence properties of different algorithms on a few representative examples, in order to visualize and better understand the essential message of the developed theory.

In addition to the PG method (Algorithm 2) and FISTA [BT09], we also compare with a simple restart scheme for dealing with unknown convexity parameters suggested by O'Donoghue and Candès [OC12]. In the context of minimizing the composite objective in (1), they suggested to restart FISTA whenever it exhibits nonmonotone behaviors (i.e., when the objective value increases). More specifically, the following two schemes were suggested in [OC12]:

- *Function scheme:* restart FISTA whenever $\phi(x^{(k)})>\phi(x^{(k-1)})$.
- *Gradient scheme:* restart FISTA whenever $g_{L_f}(y^{(k-1)})^T(x^{(k)}-x^{(k-1)})>0$ or equivalently

$$(y^{(k-1)}-x^{(k)})^T(x^{(k)}-x^{(k-1)})>0. \tag{58}$$

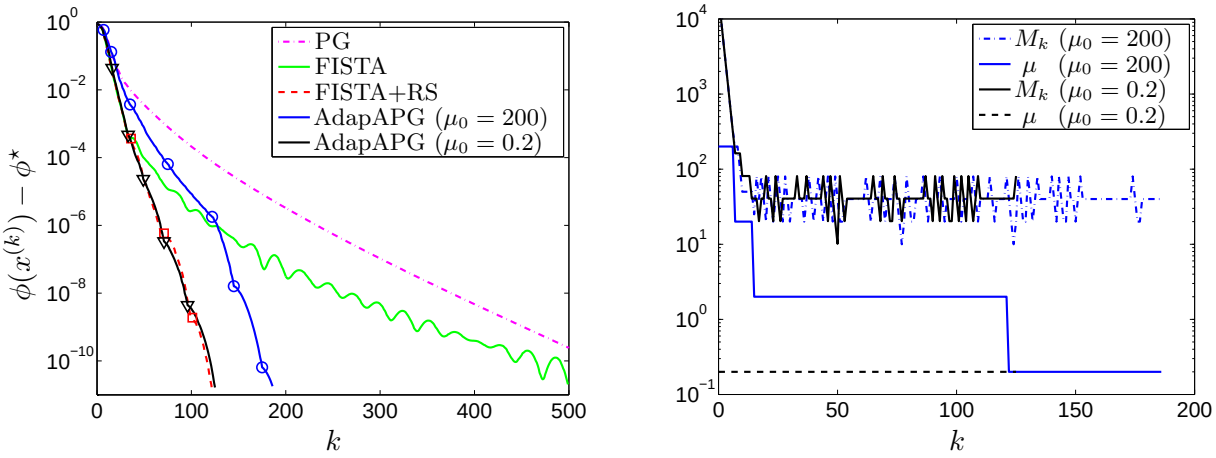
Note that FISTA can be considered as a variant of Nesterov's method [Nes04] that always use $\mu=0$. The analysis in [OC12] reveals that when such a method is applied to minimize strongly convex quadratic functions, it exhibits oscillatory (nonmonotone) regimes, and the period of the dominant mode is proportional to $\sqrt{L_f/\mu_f}$. Then following the arguments in [Nes13], it can be shown that restarting FISTA with such a period will leads to the optimal complexity of $O\left(\sqrt{L_f/\mu_f}\ln(1/\epsilon)\right)$. However, their analysis does not hold in general for non-quadratic functions.

The empirical study in [OC12] show that these two simple restart schemes perform similarly well. But the gradient scheme has the advantages of being more numerically stable near the optimum and not requiring extra computation. So in our experiments we only show comparisons against the gradient scheme, which we refer to as FISTA+RS (meaning FISTA with ReStart).

For our AdapAPG method (Algorithm 5) and APG homotopy method (Algorithm 6), we use the following values of the parameters unless otherwise stated:

parameters	γ_{inc}	γ_{dec}	θ_{sc}	γ_{sc}	η	δ
values	2	2	0.1	10	0.8	0.2

We also take advantage of the non-blowout property shown in Section 3.2, and always restart with the most recent iterate even if condition B is satisfied first in Algorithm 5.



(a) Objective gap vs. number of iterations. (b) AdapAPG: automatic tuning of M_k and μ.

Figure 1: Minimizing a random instance of the log-sum-exp function.

6.1 Experiments on the AdapAPG method

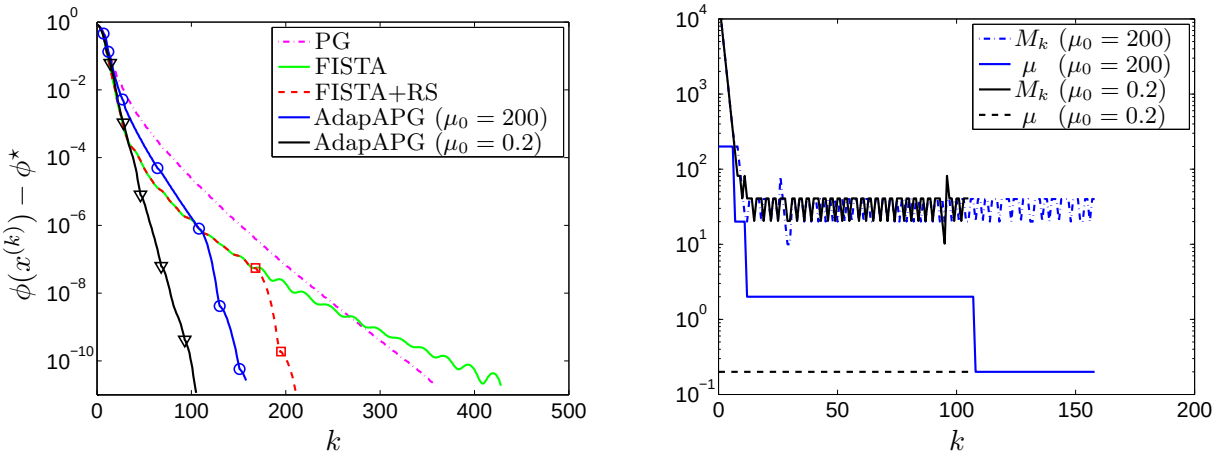
We consider the problem of minimizing the *log-sum-exp* function, i.e.,

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \triangleq \rho \log\left(\sum_{i=1}^{m} \exp\left(\frac{1}{\rho}(a_i^T x - b_i)\right)\right)$$

where all $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$, for $i = 1, \ldots, m$. This corresponds to problem (1) with $\Psi(x) = 0$. In our experiments we took $n = 200$ and $m = 10000$, and generated the a_i's and b_i's randomly with independent, standard normal distribution. Note that this is not really a strongly convex function, since it grows linearly asymptotically. However it is smooth and the region around the optimum may be well approximated by a strongly convex quadratic function. The parameter ρ controls the smoothness of f and is set to $\rho = 0.1$.

Figure 1 shows the convergence characteristics of fifth different methods on a random instance, and the AdapAPG method was initialized with two different values of μ_0. All methods are equipped with a line search procedure on the Lipschitz constant with the initial value $L_0 = 10000$. We see that the PG method converged with a slow linear rate. FISTA was much faster than PG in the beginning but slowed down eventually due to its lack of capability of exploiting strong convexity; it also demonstrated nonmotone ripples or bumps in the objective value. FISTA+RS converged fast with a linear rate. For the first run of the AdapAPG method, we intentionally chose a large initial value $\mu_0 = 200$ to test its automatic tuning capability. In fact this initial value is even larger than the restricted Lipschitz constant M_k in later iterations found by the line search procedure; see Figure 1(b). For the second run, we set μ_0 to the final estimate of μ by the first run.

In Figure 1(a), each marker on the curves indicates a restart of the corresponding algorithm. We see that FISTA+RS had three restarts, which was activated by the condition (58). Out of the seven restarts of the AdapAPG method with $\mu_0 = 200$, four of them was due to condition A, and three of them was due to condition B (see Algorithm 5). Correspondingly, Figure 1(b) shows that the estimate of the convexity parameter μ was reduced three times, each by a factor of 10, and the final estimate was 0.2. After the last reduction of μ (around $k = 120$), AdapAPG converged



(a) Objective gap vs. number of iterations. (b) AdapAPG: automatic tuning of M_k and μ.

Figure 2: Minimizing another random instance of the log-sum-exp function.

fast with a linear rate that is similar to FISTA+RS. For the second run of the AdapAPG method, we used the initial estimate $\mu_0 = 0.2$ directly. As a consequence, all of the five restarts in this case was due to condition A, and the value of μ stayed at the constant 0.2. Without the need for tuning μ, the second run of the AdapAPG converged as fast as FISTA+RS.

From the above comparison, it looks that FISTA+RS is the best method for this particular problem instance, since it demonstrated the fastest convergence without explicit tuning of the convexity parameter. AdapAPG may achieve the same convergence speed, but needs to be initialized with a good estimate of μ to avoid the extra effort involved in tuning it. In general, the procedure of tuning μ costs extra number of iterations, but with a quite modest degradation of performance. For example, Figure 1 showed that AdapAPG with $\mu_0 = 200$ needed an extra 50% iterations while reducing μ by three orders of magnitude.

However, the performance of FISTA+RS vary substantially even on the same class of log-sum-exp functions. Figure 2 illustrates the situation with another random instance in this problem class, in which we simply changed the random seed for generating the problem with the same size. For this instance, the non-monotone behaviour of FISTA appeared quite late, so the first restart of FISTA+RS occurred after $k = 170$. By that time both runs of the AdapAPG method had already finished with high precision (even for the first run which needed to reduce μ three times by a total factor of 1000). Therefore, the AdapAPG method often has a more robust performance guarantee, which is backed by our convergence analysis for general convex functions. In contrast, the FISTA+RS scheme is motivated by the analysis on the quadratic functions, and its behavior on non-quadratics can be hard to predict.

6.2 Experiments on homotopy continuation

We demonstrate the effectiveness of combining the AdapAPG method with homotopy continuation on the ℓ_1-LS problem (6). To make the comparison clear, we generate an ill-conditioned random matrix A following the experimental setup in [ANW12]:

- Generate a random matrix $B \in \mathbb{R}^{m\times n}$ with B_{ij} following i.i.d. standard normal distribution.

- Choose $\omega \in [0, 1)$, and for $i = 1, \ldots, m$, generate each row $A_{i,:}$ as follows:

$$\begin{aligned} A_{i,1} &= B_{i,1}/\sqrt{1-\omega^2} \\ A_{i,j+1} &= \omega A_{i,j} + B_{i,j}, \quad j = 2, \ldots, n \end{aligned}$$

It can be shown that the eigenvalues of $\mathbf{E}[A^T A]$, the covariance matrix of the row vectors, lie within the interval $\left[\frac{1}{(1+\omega)^2}, \frac{2}{(1-\omega)^2(1+\omega)}\right]$. If $\omega = 0$, then $A = B$ and the covariance matrix is well conditioned. As $\omega \to 1$, the covariance matrix becomes progressively more ill-conditioned. In the following experiments, the matrix A is generated with $m = 1000$, $n = 5000$, and $\omega = 0.9$.

Figure 3 shows the computational results of the four different methods: PG, FISTA, FISTA+RS, AdapAPG, and their homotopy continuation variants (denoted by "+H"). As we mentioned before, the homotopy continuation parameters are set to $\eta = 0.8$ and $\delta = 0.2$. For each method, we initialize the Lipschitz constant by

$$L_0 = \max_{j \in \{1,\ldots,n\}} \|A_{:,j}\|_2^2.$$

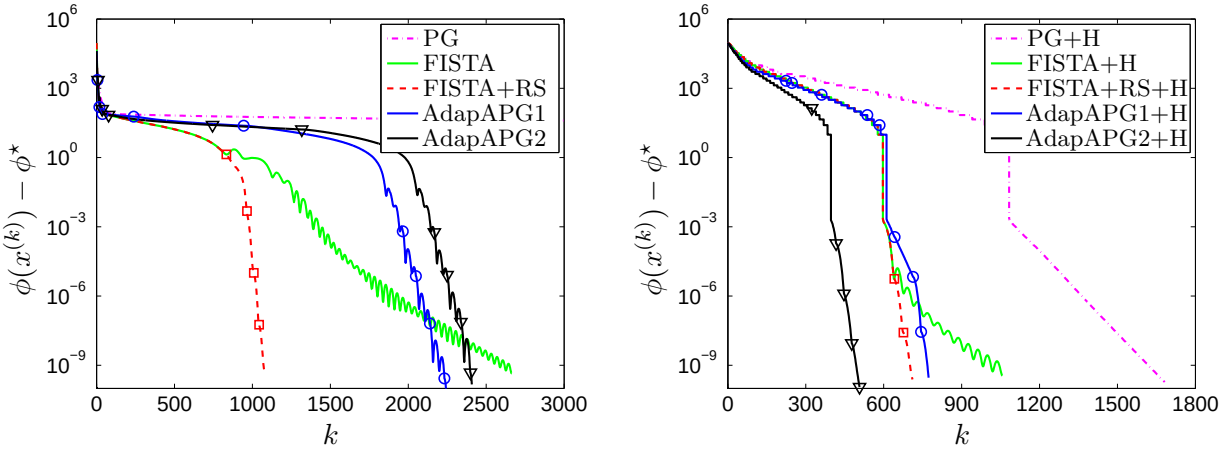
For the AdapAPG method, we initialize the estimate of convexity parameter with two different values, $\mu_0 = L_0/10$ and $\mu_0 = L_0/100$, and denote their results by AdapAPG1 and AdapAPG2, respectively.

In Figure 3(a), we observe that PG, FISTA+RS and the two runs of AdapAPG all go through a slow plateau before reaching fast local linear convergence. (PG enters the fast convergence zone well after 6000 iterations, which is outside of the plot range here.) FISTA without restart does not exploit the strong convexity and is the slowest asymptotically. Their homotopy continuation variants shown in the right plot are much faster. Each vertical jump on the curves indicates a change in the value of λ in the homotopy continuation scheme. In particular, it is clear that all except FISTA enter the final homotopy stage with fast linear convergence. In the final stage, the PGH method has a rather flat slope due to ill-conditioning of the A matrix; in contrast, FISTA+RS and AdapAPG2 have much steeper slopes due to their accelerated schemes. AdapAPG1 started with a modest slope, and then detected that the μ value was too big and reduced it by a factor of 10, which resulted in the same fast convergence rate as AdapAPG1 after that.

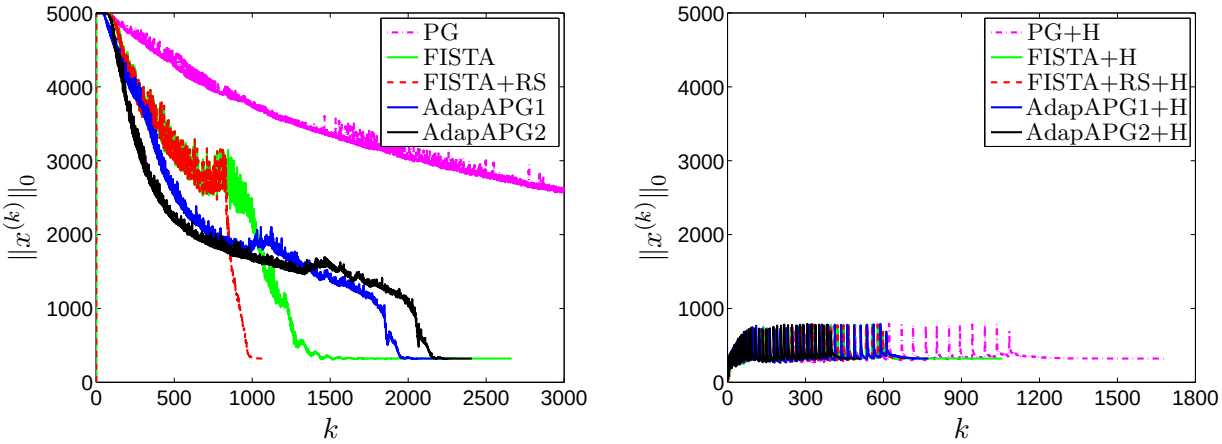
As we analyzed in Section 5, in addition to the acceleration scheme used, the differences in the convergence rates of these methods can be explained by whether or not they exploit the restricted strong convexity. Figure 3(b) shows the sparsity of each iterates along the solution paths of these methods. We observe that FISTA+RS and the two runs of AdapAPG entered fast local convergence precisely when their iterates became sufficiently sparse, i.e., when $\|x^{(k)}\|_0$ became close to that of the final solution. In contrast, the homotopy variants of these algorithms kept all iterates sparse by using the warm start from previous stages. Therefore, restricted strong convexity hold along the solution path and fast linear convergence was maintained at each stage.

Figure 3(c) shows the automatic tuning of the local Lipschitz constant M_k and the restricted convexity parameter μ. We see that the homotopy methods (right plot) have relatively smaller M_k and larger μ than the ones without using homotopy continuation (right plot), which means much better conditioning along the iterates. In particular, the homotopy AdapAPG method used fewer number of reductions of μ, for both initializations $\mu_0 = L_0/10$ and $\mu_0 = L_0/100$.

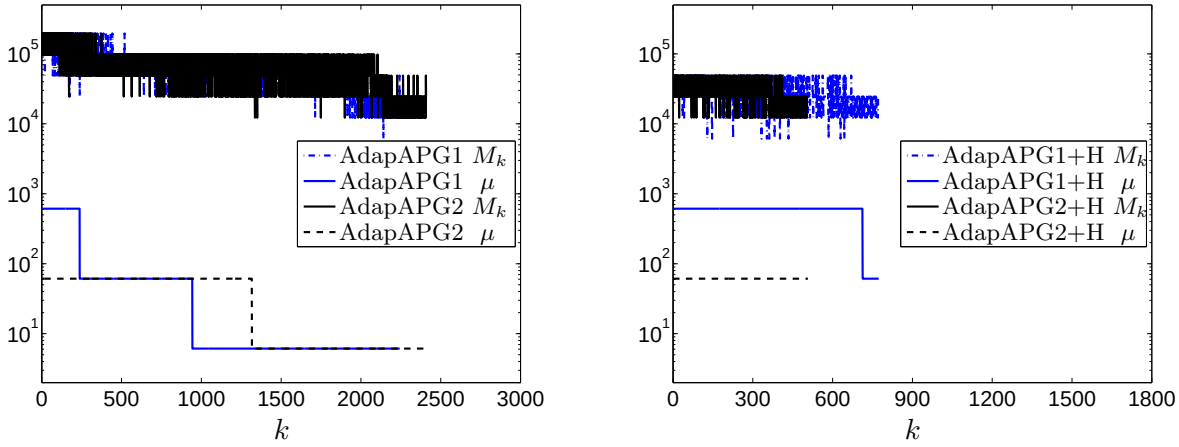
Overall, we observe that for the sparse least-squares problem, the homotopy continuation scheme is very effective in speeding up different methods. Even with the overhead of estimating and tuning μ, the AdapAPG+H method is close in efficiency compared with the FISTA+RS+H method.



(a) Objective gap vs. number of iterations, without homotopy (left) and with homotopy (right).



(b) Number of nonzero elements, without homotopy (left) and with homotopy (right).



(c) Automatic tuning of M_k and μ, without homotopy (left) and with homotopy (right)

Figure 3: Solving an ill-conditioned ℓ_1-LS problem. AdapAPG1 starts with $\mu_0 = L_0/10$, and AdapAPG2 starts with $\mu_0 = L_0/100$.

If μ is initialized with the right value, then AdapAPG+H gives the best performance. From the right plot in Figure 3(a), we see that FISTA+RS+H and the two runs of AdapAPG+H have roughly the same linear rate of convergence in the final stage. However, AdapAPG2+H used much less number of iterations before reaching the final stage, which again indicates the importance of exploiting the right amount of restricted strong convexity.

7 Conclusion and discussions

We first proposed an accelerated proximal gradient (APG) method for minimizing composite objective functions where the smooth part is also strongly convex. Our method employs a line-search routine in each iteration to search for the local Lipschitz constant, which corresponds to a larger step size and faster convergence in practice. When the strong convexity parameter of the problem is unknown, we developed an adaptive APG method which incorporates a restart scheme for estimating and tuning the convexity parameter. We show that this restart scheme only affects the complexity of the algorithm with an additional logarithmic factor. This scheme is inspired by a similar one proposed by Nesterov [Nes13] for his accelerated dual gradient method, and they share the same complexity estimate. However, our method avoids the relocation of strong convexity in the objective function required in the dual gradient method, also avoids the extra line search required by Nesterov's scheme, thus is computationally more efficient.

Then we focused on the special case of solving the ℓ_1-regularized least-squares (ℓ_1-LS) problem in the high-dimensional setting. In such a context, the smooth part of the objective (least-squares) is not strongly convex over the entire domain. Nevertheless, we exploit its restricted strong convexity over sparse vectors using the adaptive APG method combined with a homotopy continuation scheme. Under a suitable restricted eigenvalue condition, this homotopy method generates a path of solutions within a sparse subspace, and a global geometric rate of convergence can be established. Compared to previous analysis of the homotopy proximal gradient method [XZ12], the complexity of our accelerated algorithm has a weaker dependence on the restricted condition number of the least-squares problem (roughly proportional to its square root, instead of the restricted conditional number itself). Our theoretical analysis are supported with preliminary numerical experiments.

Similar to the discussions in [XZ12], the conditions that guarantee the geometric convergence of the APG homotopy method (Assumption 1) are rather strong, especially when compared with recovery conditions established in the compressed sensing literature (e.g., [LM11] and references therein). This can be expected, since our analysis is based on keeping all the intermediate iterates sparse, rather than only for the optimal solution. In fact, our conditions depend on not only the measurement matrix A, but also the algorithmic parameters (η and δ) that control how fast the regularization parameter is reduced and how accurate each stage needs to be solved. Nevertheless, our numerical experiences indicate that the performance of the APG homotopy method is not very sensitive to the algorithmic parameters, and geometric convergence may happen even in cases where our assumption does not hold. This suggests that it is possible to develop less restrictive conditions to guarantee a fast global convergence rate.

Acknowledgments

We thank Professor Tong Zhang for helpful discussions on the APG homotopy method.

References

[ANW12] A. Agarwal, S. N. Negahban, and M. J. Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

[BDE09] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[BL08] K. Bredies and D. A. Lorenz. Linear convergence of iterative soft-thresholding. *Journal of Fourier Analysis and Applications*, 14:813–837, 2008.

[BT09] A. Beck and M. Teboulle. A fast iterative shrinkage-threshold algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[CDS98] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[CRT06] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[CT05] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005.

[CT06] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, December 2006.

[Don06] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[GK13] C. C. Gonzaga and E. W. Karas. Fine tuning Nesterov's steepest descent algorithm for differentiable convex programming. *Mathematical Programming, Series A*, 138:141–166, 2013.

[GLW13] M. Gu, L.-H. Lim, and C. J. Wu. ParNes: A rapidly convergent algorithm for accurate recovery of sparse and approximately sparse signals. *Numerical Algorithms*, 64:321–347, 2013.

[HYZ08] E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for ℓ_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

[LM11] S. Li and Q. Mo. New bounds on the restricted isometry constant δ_{2k}. *Applied and Computational Harmonic Analysis*, 31(3):460–468, 2011.

[LT92] Z.-Q. Luo and P. Tseng. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2):408–425, 1992.

[MOS12] Renato D.C. Monteiro, Camilo Ortiz, and Benar F. Svaiter. An adaptive accelerated first-order method for convex optimization. Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 2012.

[Nes04] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, 2004.

[Nes05] Y. Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[Nes08] Y. Nesterov. How to advance in Structural Convex Optimization. *OPTIMA: Mathematical Programming Society Newsletter*, 78:2–5, November 2008.

[Nes13] Y. Nesterov. Gradient methods for minimizing composite objective function. *Mathematical Programming, Series B*, 140(2007/76):125–161, September 2013.

[NN94] Y. Nesterov and A. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics. SIAM, 1994.

[NY83] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. J. Wiley & Sons, New York, 1983.

[OC12] B. O'Donoghue and E. J. Candès. Adaptive restart for accelerated gradient schemes. Manuscript, April 2012. To appear in *Foundations of Computational Mathematics*.

[Roc70] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58:267–288, 1996.

[Tse08] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Manuscript, 2008.

[WNF09] S. J. Wright, R. D. Nowad, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, July 2009.

[Wri12] S. J. Wright. Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization*, 22(1):159–186, 2012.

[WYGZ10] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation. *SIAM Journal on Scientific Computing*, 32(4):1832–1857, 2010.

[XZ12] L. Xiao and T. Zhang. A proximal-gradient homotopy method for the sparse least-squares problem. Technical Report MSR-TR-2012-36, Microsoft Research, March 2012. To appear in *SIAM Journal on Optimization*.

[ZH08] C.-H. Zhang and J. Huang. The sparsity and bias of the lasso selection in high–dimensional linear regression. *Annals of Statistics*, 36:1567–1594, 2008.