8-2022

Optimal First Order Methods for Reducing Gradient Norm in Unconstrained Convex Smooth Optimization

Yunheng Jiang
yunhenj@g.clemson.edu

Optimal first order methods for reducing gradient norm in unconstrained convex smooth optimization

---

A Master's Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Mathematical Sciences
Operations Research

---

by
Yunheng Jiang
August 2022

Accepted by:
Dr. Yuyuan Ouyang, Committee Chair
Dr. Boshi Yang
Dr. Cheng Guo

Abstract

In this thesis, we focus on convergence performance of first-order methods to compute an ϵ-approximate solution of minimizing convex smooth function f at the N-th iteration.

In our introduction of the above research question, we first introduce the gradient descent method with constant step size $h = 1/L$. The gradient descent method has a $\mathcal{O}(L^2\|x_0 - x^*\|^2/\epsilon)$ convergence with respect to $\|\nabla f(x_N)\|^2$. Next we introduce Nesterov's accelerated gradient method, which has an $\mathcal{O}(L\|x_0 - x^*\|\sqrt{1/\epsilon})$ complexity in terms of $\|\nabla f(x_N)\|^2$. The convergence performance of Nesterov's accelerated gradient method is much better than that of the gradient descent method but still not optimal. We also briefly introduce some other first order methods in the literature to compute an ϵ-approximate solution of minimizing convex smooth function f, including a monotone convergence accelerated gradient method and a perturbed gradient method in [8]. They have $\mathcal{O}(L^{2/3}\|x_0 - x^*\|^{2/3}/\epsilon^{1/3})$ and $\mathcal{O}((\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})\ln(1 + 2L\|x_0 - x^*\|/\sqrt{\epsilon}))$ complexities respectively. Those results are better than that of Nesterov's accelerated gradient method, but the convergence performance of first order methods can still be better.

Our main focus is to design a first order method for reducing the gradient norm of the objective function. Our research is closely related to [4], in which a first order method is proposed with complexity of order $\mathcal{O}(\sqrt{L(f(x_0) - f(x^*))}/\sqrt{\epsilon})$. This method is studied through the performance enhancement program (PEP) originated from [2]. In [9] it is pointed out that by combining the accelerated gradient method and the method in [4] into a two-phase optimal gradient method, one is actually able to obtain an optimal $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity.

Our new result in this thesis is a different set of parameters from [4] that also achieves the $\mathcal{O}(\sqrt{L(f(x_0) - f(x^*))}/\sqrt{\epsilon})$ convergence with respect to $\|\nabla f(x_N)\|^2$. Combining with Nesterov's accelerated gradient method, we are able to derive an $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity, which is optimal among first-order methods, by the two-phase optimal gradient method.

Contents

Chapter 1

Introduction

The problem of interest in this thesis is the following unconstrained optimization problem of convex smooth functions:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1.1}$$

where f is convex, continuously differentiable, and its gradient ∇f is Lipschitz continuous with constant L. We use notation $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ to denote such functions.[1] Our goal is to compute an approximate solution with accuracy threshold ϵ. Note that there are several different possible definitions of ϵ-approximate solutions. In our research, we define an ϵ-approximate solution as a solution x that satisfies

$$\|\nabla f(x)\|^2 \leq \epsilon.$$

Here and throughout this thesis, $\|\cdot\|$ denotes the Euclidean norm. To emphasize that the above definition requires that the gradient norm $\|\nabla f(x)\|$ is smaller than the threshold ϵ, in the sequel, we sometimes refer to the above definition as "ϵ-approximate solution with small gradient norm". Clearly, if $\epsilon = 0$, then x is a stationary point of problem (1.1) and hence an optimal solution due to the convexity of f.

There has been several possible methods for solving problem (1.1). Our focus is on first-order methods that relies on objective function value and gradient evaluations. Such methods are

[1] The notation $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$ follows from the book [8]. Such functions are also known as convex smooth functions. Here $\mathcal{F}$ stands for the set of convex functions; the subscript "$1,1$" denotes that f is continuously differentiable and its gradient is Lipschitz continuous; the subscript L denotes the Lipschitz constant of L.

commonly used when the accuracy threshold ϵ is modest to relative large and the dimension n is large. In such cases, higher order methods (Newton's method, etc.) requires more computational time per iteration and becomes less appealing.

Our chapter is organized as follows. In Section 1.1, we describe several key properties of convex smooth functions that we utilize throughout this thesis. In Sections 1.2 and 1.3 we describe two possible methods for solving problem (1.1) and discuss their convergence properties. In Section 1.4, we provide a literature review on other first-order methods for solving problem (1.1).

1.1 Properties of convex smooth functions

In this section, we describe several commonly known properties of convex smooth functions $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$. Some properties are utilized in many convergence analysis performed throughout this thesis. The proofs of all the results below are commonly known in convex optimization textbooks (see, e.g., [8]) and are skipped.

Lemma 1.1.1. *For any function $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and any $x, y \in \mathbb{R}^n$, we have*

$$\frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle.$$

Specially, with $y = x^*$ we have

$$\frac{1}{L}||\nabla f(x)||^2 \leq \langle \nabla f(x), x - x^* \rangle. \tag{1.2}$$

The lemma above states that the inner product of gradient difference $\nabla f(x) - \nabla f(y)$ and point difference $x - y$ is lower bounded by the squared gradient norm difference $\|\nabla f(x) - \nabla f(y)\|^2/L$. Note that the special case (1.2) of the above lemma has an important implication. Specifically, for any differentiable convex function f, the following property holds for its minimizer x^*:

$$0 \leq \langle \nabla f(x), x - x^* \rangle.$$

The above relationship is indeed an optimality condition for convex and differentiable functions. The property (1.2) states that for convex smooth functions, its optimality condition can be stronger

than the above equation with an extra gradient norm.

Lemma 1.1.2. *Let f be an arbitrary function s.t. $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$, $\forall x, y \in \mathbb{R}^n$,*

$$0 \leq f(x) - f(y) - \langle \nabla f(y), x - y \rangle \leq \frac{L}{2}||x - y||^2. \tag{1.3}$$

Especially, when we let $y = x^*$, since $\nabla f(x^*) = 0$, we have

$$f(x) - f(x^*) \leq \frac{L}{2}||x - x^*||^2. \tag{1.4}$$

Similar to Lemma 1.1.1, the above lemma states that convex smooth functions have a stronger condition than that of convex differentiable functions. Specifically, for convex differential function f we know that $0 \leq f(x) - f(y) - \langle \nabla f(y), x - y \rangle$, namely, $f(x)$ is lower bounded by a linear approximation at y. Lemma 1.1.2 reveals that f is not only lower bounded the aforementioned linear approximation, but also upper bounded by a quadratic function. Indeed, for convex smooth functions we can also further strengthen its lower bound from linear approximations, as stated in the lemma below.

Lemma 1.1.3. *Let f be an arbitrary function s.t. $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$, $\forall x, y \in \mathbb{R}^n$,*

$$f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq \frac{1}{2L}||\nabla f(x) - \nabla f(y)||^2.$$

Especially, when we let $y = x^*$, since $\nabla f(x^*) = 0$, we have

$$\frac{1}{2L}||\nabla f(x)||^2 \leq f(x) - f(x^*). \tag{1.5}$$

Lemmas 1.1.1, 1.1.2 and 1.1.3 are the fundamental properties of $f \in \mathcal{F}_L^{1,1}(X)$ that play important roles in convex smooth optimization analysis. Throughout this thesis we need them to derive convergence analysis of several first-order methods.

1.2 The gradient descent method

The most common first-order method for solving problem (1.1) is the gradient descent method. It is based on a straightforward observation that for any differentiable function, its negative gradient at a point is the direction along which the function decreases the fastest locally at such a point. We describe the gradient descent method and analyze its convergence performance in this section. The gradient descent algorithm is listed below.

Algorithm 1 The gradient descent method

Require: Initial point $x_0 \in \mathbb{R}^n$, $h > 0$

for $i = 0, 1, \ldots, N$ **do**

$$x_{i+1} = x_i - h\nabla f(x_i) \tag{1.6}$$

end for

For $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$, we first derive a convergence result in terms of $f(x_N) - f(x^*)$. The derivation is based on the analysis of the relationship between $f(x_i) - x^*$ and $f(x_{i-1}) - x^*$, as detailed in the proof of the following proposition.

Proposition 1.2.1. *Let f be a function in the function class $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and $\{x_i\}_{i=0}^N$ be the iterations of the gradient descent method applied to minimize f. If we have $0 < h < 1/2L$, then for any $N \geq 0$*

$$f(x_N) - f(x^*) \leq \frac{2(f(x_0) - f(x*))\|x_0 - x^*\|^2}{2\|x_0 - x^*\|^2 + Nh(2 - Lh)(f(x_0) - f(x*))}. \tag{1.7}$$

Proof. First, by the definition of x_i (1.6) in the description of Algorithm 1, the relationship (1.2) in Lemma 1.1.1, and noting that $\nabla f(x^*) = 0$, we have

$$\begin{aligned}
\|x_{i+1} - x^*\|^2 =& \|x_i - x^* - h\nabla f(x_i)\|^2 \\
=& \|x_i - x^*\|^2 - 2h\langle \nabla f(x_i), x_i - x^* \rangle + h^2 \|\nabla f(x_i)\|^2 \\
\leq& \|x_i - x^*\|^2 - h\left(\frac{2}{L} - h\right)\|\nabla f(x_i)\|^2.
\end{aligned}$$

Thus for any i, we have $\|x_{i+1} - x^*\| \leq \|x_i - x^*\|$. Consequently, we can observe that $\|x_i - x^*\| \leq \|x_0 - x^*\|$.

Next, by Lemma 1.1.2 and the description of x_i (1.6) in the gradient descent algorithm, we have

$$\begin{aligned} f(x_{i+1}) \leq & f(x_i) + \langle \nabla f(x_i), x_{i+1} - x_i \rangle + \frac{L}{2} ||x_{i+1} - x_i||^2 \\ = & f(x_i) - h\left(1 - \frac{L}{2}h\right) \|\nabla f(x_i)\|^2. \end{aligned} \tag{1.8}$$

Since the function f is convex differentiable, recalling our previous observation that $\|x_i - x^*\| \leq \|x_0 - x^*\|$, we know that

$$f(x_i) - f(x^*) \leq \langle \nabla f(x_i), x_i - x^* \rangle \leq \|\nabla f(x_i)\| \cdot \|x_i - x^*\| \leq \|\nabla f(x_i)\| \cdot \|x_0 - x^*\|.$$

Thus by (1.8), we obtain

$$f(x_{i+1}) \leq f(x_i) - h\left(1 - \frac{L}{2}h\right) \frac{(f(x_i) - f(x^*))^2}{\|x_0 - x^*\|^2},$$

i.e.

$$f(x_{i+1}) - f(x^*) \leq f(x_i) - f(x^*) - h\left(1 - \frac{L}{2}h\right) \frac{(f(x_i) - f(x^*))^2}{\|x_0 - x^*\|^2}.$$

Dividing $(f(x_i) - f(x^*))\,(f(x_{i+1}) - f(x^*))$ on both sides, we have

$$\begin{aligned} \frac{1}{f(x_{i+1}) - f(x^*)} &\geq \frac{1}{f(x_i) - f(x^*)} + \frac{h(1 - \frac{L}{2}h)}{\|x_0 - x^*\|^2} \cdot \frac{f(x_i) - f(x^*)}{f(x_{i+1}) - f(x^*)} \\ &\geq \frac{1}{f(x_i) - f(x^*)} + \frac{h(1 - \frac{L}{2}h)}{\|x_0 - x^*\|^2}. \end{aligned}$$

Summing the inequalities above from $k = 0, ..., N-1$, we have

$$\frac{1}{f(x_N) - f(x^*)} \geq \frac{1}{f(x_0) - f(x^*)} + N \cdot \frac{h(1 - \frac{L}{2}h)}{\|x_0 - x^*\|^2}.$$

The above result implies (1.7) immediately. □

In the convergence property above, the right-hand side of the result (1.7) is dependent on

the stepsize h. Theoretically, the best choice of stepsize h is the one such that $h(1-(L/2)h)$ in the denominator of the right-hand side is maximized, i.e., when $h = 1/L$. The convergence result when $h = 1/L$ is described in the theorem below.

Theorem 1.2.1. *Let f be a function in function class $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and $\{x_i\}_{i=0}^N$ be the iterations of the gradient descent method. If the stepsize is chosen to $h = 1/L$, then for any $N \geq 0$ we have*

$$f(x_N) - f(x^*) \leq \frac{2L||x_0 - x^*||^2}{N+4} \text{ and } \|\nabla f(x_N)\|^2 \leq \frac{4L^2\|x_0 - x^*\|^2}{N+4}. \tag{1.9}$$

Proof. The first result in (1.9) follows directly from Proposition 1.2.1 (with stepsize $h = 1/L$). Moreover, noting from the relationship (1.5) in Lemma 1.1.3 that $\|\nabla f(x)\|^2 \leq 2L(f(x) - f(x^*))$, we conclude the second result in (1.9). □

By the result (1.9) in the above theorem, in order to make sure that the iterate x_N is an ϵ-approximate solution to problem (1.1), i.e., $\|\nabla f(x_N)\|^2 \leq \epsilon$, it suffices to run N iterations of the gradient descent method with

$$N \geq \frac{4L^2\|x_0 - x^*\|^2}{\epsilon}. \tag{1.10}$$

Namely, the iteration complexity of the gradient descent method for computing an ϵ-approximate solution with small gradient norm is of order $\mathcal{O}(4L^2/\epsilon)$. Note that the above convergence rate can be further improved. Indeed, Nesterov introduced in [7] (see also [8]) an accelerated gradient method that has better convergence properties than that of the gradient descent method. In the following section we introduce Nesterov's method and analyze its convergence.

1.3 Nesterov's accelerated gradient method

The first version of Nesterov's accelerated gradient method appears in [7]. After the work in [7], there are several extensions and modifications proposed in the literature (see, e.g., [8, 6]). In this section our description of Nesterov's method in Algorithm 2 is based [5] under the Euclidean

setting. Note that [5] covers a more general treatment of Nesterov's method with non-Euclidean prox functions; however, this is out of the scope of this thesis.

Algorithm 2 Nesterov's accelerated gradient method

Require: Initial point $x_0 \in \mathbb{R}^n$, $q_i \in [0,1]$, $\gamma_i \geq 0$, $\alpha_i \in [0,1]$

Set $\overline{x}_0 = x_0$.

for $i = 1, \ldots, N$ **do**

Compute

$$\begin{aligned} \underline{x}_i &= (1-q_i)\overline{x}_{i-1} + q_i x_{i-1} \\ x_i &= \operatorname*{argmin}_{x \in \mathbb{R}} \{\gamma_i \langle \nabla f(\underline{x}_i), x \rangle + \|x_{i-1} - x\|_2^2\} \\ \overline{x}_i &= (1-\alpha_i)\overline{x}_{i-1} + \alpha_i x_i \end{aligned}$$

end for

Output approximate solution $\overline{x}_N$.

Here, $\{(\underline{x}_i, x_i, \overline{x}_i)\}_{i=0}^N \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ are the iterates generated by Nesterov's accelerated gradient method. Specially, the notation $\underline{x}_i$ denotes the iterates at which gradient of f is computed. Here the underline is since any gradient evaluation $\nabla f(\underline{x}_i)$ proves a linear approximation lower bound $f(\underline{x}_i) + \langle \nabla f(\underline{x}_i), x - \underline{x}_i \rangle$ of the function $f(x)$. The notation x_i denotes the iterates at which we perform gradient-descent-like updates. The notation $\overline{x}_i$ denotes the outputs of the approximate solutions of the algorithm. Here the overline is since $f(\overline{x}_i) \geq f(x^*)$ is an overestimate of the optimal objective function value. We can immediately observe that if $\alpha_i = 1$ and $\gamma_i = h$ for $i = 0, \ldots, N-1$, Algorithm 2 is identical to Algorithm 1. In the following proposition, we prove that the certain choices of algorithm parameters q_i, α_i and γ_i can lead us to a relationship between the i-th and $(i-1)$-th iterates of Algorithm 2.

Proposition 1.3.1. *Let f be a function such that $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and $\{(\underline{x}_i, x_i, \overline{x}_i)\}_{i=0}^N \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ be the iterates generated by Nesterov's accelerated gradient method in Algorithm 2 to minimize f. If parameters q_i, α_i and γ_i satisfy the following relationships:*

$$\alpha_i \geq q_i, \tag{1.11}$$

$$\frac{L(\alpha_i - q_i)}{1 - q_i} \leq 0, \tag{1.12}$$

$$\frac{L q_i (1 - \alpha_i)}{1 - q_i} \leq \frac{1}{\gamma_i}, \quad i = 1, ..., N, \tag{1.13}$$

then for any $x \in \mathbb{R}^n$, we obtain

$$f(\overline{x}_i) - f(x) + \frac{\alpha_i}{\gamma_i}\|x_i - x\|^2 \leq (1-\alpha_i)[f(\overline{x}_{i-1}) - f(x)] + \frac{\alpha_i}{\gamma_i}\|x_{i-1} - x\|^2. \tag{1.14}$$

Proof. First, by the definitions of $\overline{x}_i$, x_i and $\underline{x}_i$ in Algorithm 2, we have

$$\begin{aligned}
\overline{x}_i - \underline{x}_i =&(q_i - \alpha_i)\overline{x}_{i-1} + \alpha_i x_i - q_i x_{i-1} \\
=&\alpha_i \left[x_i - \frac{\alpha_i - q_i}{\alpha_i(1-q_i)}\underline{x}_i - \frac{q_i(1-\alpha_i)}{\alpha_i(1-q_i)} x_{i-1} \right] \\
=&\alpha_i \left[(\frac{\alpha_i - q_i}{\alpha_i(1-q_i)} + \frac{q_i(1-\alpha_i)}{\alpha_i(1-q_i)})x_i - \frac{\alpha_i - q_i}{\alpha_i(1-q_i)}\underline{x}_i - \frac{q_i(1-\alpha_i)}{\alpha_i(1-q_i)} x_{i-1} \right] \\
=&\alpha_i \left[\frac{\alpha_i - q_i}{\alpha_i(1-q_i)}(x_i - \underline{x}_i) + \frac{q_i(1-\alpha_i)}{\alpha_i(1-q_i)}(x_i - x_{i-1}) \right].
\end{aligned}$$

Thus by the relationship $\alpha_i \geq q_i$ of α_i and q_i in (1.11) and the convexity of norms, we obtain

$$\|\overline{x}_i - \underline{x}_i\|^2 \leq \alpha_i \left[\frac{\alpha_i - q_i}{1-q_i}\|x_i - \underline{x}_i\|^2 + \frac{q_i(1-\alpha_i)}{1-q_i}\|x_i - x_{i-1}\|^2 \right]. \tag{1.15}$$

Next, by property (1.3) in Lemma 1.1.2, the relationship $\overline{x}_i = (1-\alpha_i)\overline{x}_{i-1} + \alpha_i x_i$ described in Algorithm 2, the convexity of f, inequalities (1.15), (1.12) and (1.13) above, we are able to derive that

$$\begin{aligned}
f(\overline{x}_i) \leq& f(\underline{x}_i) + \langle \nabla f(\underline{x}_i), \overline{x}_i - \underline{x}_i \rangle + \frac{L}{2}\|\overline{x}_i - \underline{x}_i\|^2 \\
=&(1-\alpha_i)\left[f(\underline{x}_i) + \langle \nabla f(\underline{x}_i), \overline{x}_{i-1} - \underline{x}_i \rangle \right] + \alpha_i \left[f(\underline{x}_i) + \langle \nabla f(\underline{x}_i), x_i - \underline{x}_i \rangle \right] + \frac{L}{2}\|\overline{x}_i - \underline{x}_i\|^2 \\
\leq&(1-\alpha_i) f(\overline{x}_{i-1}) \\
&+ \alpha_i \left[f(\underline{x}_i) + \langle \nabla f(\underline{x}_i), x_i - \underline{x}_i \rangle + \frac{L(\alpha_i - q_i)}{2(1-q_i)}\|x_i - \underline{x}_i\|^2 + \frac{L q_i(1-\alpha_i)}{2(1-q_i)}\|x_i - x_{i-1}\|^2 \right] \\
\leq&(1-\alpha_i) f(\overline{x}_{i-1}) + \alpha_i \left[f(\underline{x}_i) + \langle \nabla f(\underline{x}_i), x_i - \underline{x}_i \rangle + \frac{1}{\gamma_i}\|x_i - x_{i-1}\|^2 \right].
\end{aligned}$$

In the above derivation, we first use (1.3) in the first inequality. Then we use the relationship that $\overline{x}_i = (1-\alpha_i)\overline{x}_{i-1} + \alpha_i x_i$ in the first equality. Next we use the convexity of f and relationship (1.15) in the second inequality. Finally we use (1.12) and (1.13) in the last inequality.

Finally, by the optimality condition of x_i in its definition in Algorithm 2, we have that for

all $x \in \mathbb{R}^n$,

$$\gamma_i \langle \nabla f(\underline{x}_i), x_i \rangle + \|x_{i-1} - x_i\|_2^2 \leq \gamma_i \langle \nabla f(\underline{x}_i), x \rangle + \|x_{i-1} - x\|_2^2.$$

Hence combining with the fact that $\|x_{i-1} - x_i\|^2 \geq \|x_{i-1} - x\|^2 - \|x_i - x\|^2$ and the convexity of f, we conclude that

$$\begin{aligned} f(\overline{x}_i) \leq & (1-\alpha_i) f(\overline{x}_{i-1}) + \alpha_i \left[f(\underline{x}_i) + \langle \nabla f(\underline{x}_i), x - \underline{x}_i \rangle \right] + \frac{\alpha_i}{\gamma_i} \|x_{i-1} - x\|^2 - \frac{\alpha_i}{\gamma_t} \|x_i - x\|^2 \\ \leq & (1-\alpha_i) f(\overline{x}_{i-1}) + \alpha_i f(x) + \frac{\alpha_i}{\gamma_i} \|x_{i-1} - x\|^2 - \frac{\alpha_i}{\gamma_i} \|x_i - x\|^2. \end{aligned}$$

□

In the above proposition, we prove that when the parameters satisfy (1.11), (1.12) and (1.13), we obtain a recursive relationship between the i-th and $(i-1)$-th iterates of Nesterov's gradient method (1.14). Consequently, we show in the following proposition that by induction it is now possible to expand the aforementioned relationship to one that is between the N-th and the initial iterates.

Proposition 1.3.2. *Let f be a function such that $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and $\{(\underline{x}_i, x_i, \overline{x}_i)\}_{i=0}^N \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ be the iterates generated by Nesterov's accelerated gradient method to minimize f. If $\alpha_i = q_i$, $L\alpha_i \leq 1/\gamma_i$ and $\gamma_i(1-\alpha_i)/\alpha_i \leq \gamma_{i-1}/\alpha_{i-1}$ for all $i = 1, ..., N$, then it holds that*

$$f(\overline{x}_N) - f(x^*) + \frac{\alpha_N}{\gamma_N} \|x_N - x^*\|_2^2 \leq \frac{\alpha_N(1-\alpha_1)\gamma_1}{\gamma_N \alpha_1} [f(\overline{x}_0) - f(x^*)] + \frac{\alpha_N}{\gamma_N} \|x_0 - x^*\|_2^2. \quad (1.16)$$

Proof. It is straightforward to verify that assumptions (1.11)–(1.13) holds and hence we can use Proposition 1.3.1 to conclude that

$$f(\overline{x}_i) - f(x^*) + \frac{\alpha_i}{\gamma_i} \|x_i - x^*\|_2^2 \leq (1-\alpha_i)[f(\overline{x}_{i-1}) - f(x^*)] + \frac{\alpha_i}{\gamma_i} \|x_{i-1} - x^*\|_2^2.$$

Applying relationship $\gamma_i(1-\alpha_i)/\alpha_i \leq \gamma_{i-1}/\alpha_{i-1}$ to the above result, we have

$$\begin{aligned}\frac{\gamma_i}{\alpha_i}[f(\overline{x}_i)-f(x^*)]+\|x_i-x^*\|_2^2 \leq & \frac{(1-\alpha_i)\gamma_i}{\alpha_i}[f(\overline{x}_{i-1})-f(x^*)]+\|x_{i-1}-x^*\|_2^2 \\ \leq & \frac{\gamma_{i-1}}{\alpha_{i-1}}[f(\overline{x}_{i-1})-f(x^*)]+\|x_{i-1}-x^*\|_2^2.\end{aligned}$$

Repeating the above relationship inductively for N times, we are able to derive that

$$\frac{\gamma_N}{\alpha_N}[f(\overline{x}_N)-f(x^*)]+\|x_N-x^*\|_2^2 \leq \frac{(1-\alpha_1)\gamma_1}{\alpha_1}[f(\overline{x}_0)-f(x^*)]+\|x_0-x^*\|_2^2$$

and conclude (1.16). □

With help from the above result, we are now ready to analyze the convergence properties of Algorithm 2.

Theorem 1.3.1. *Let f be a function that belongs to function class $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$, and $\{(\underline{x}_i, x_i, \overline{x}_i)\}_{i=0}^N \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ be the iterates generated by Nesterov's accelerated gradient method with parameters $\alpha_i = q_i = 2/(i+1)$, $\gamma_i = i/(2L)$, we obtain*

$$f(\overline{x}_N)-f(x^*) \leq \frac{4L}{N(N+1)}\|x_0-x^*\|_2^2 \text{ and } \|\nabla f(\overline{x}_N)\|^2 \leq \frac{8L^2}{N(N+1)}\|x_0-x^*\|_2^2. \tag{1.17}$$

Proof. By the choices of α_i, q_i, and γ_i we have that $\alpha_i/\gamma_i = 4L/(i(i+1))$ and it is easy to verify that assumptions of Proposition 1.3.2 hold. Thus by Proposition 1.3.2, we obtain

$$f(\overline{x}_N)-f(x^*) \leq \frac{4L}{N(N+1)}(\|x_0-x^*\|_2^2-\|x_N-x^*\|_2^2) \leq \frac{4L}{N(N+1)}\|x_0-x^*\|_2^2.$$

In addition, noting from the relationship (1.5) in Lemma 1.1.3 that $\|\nabla f(x)\|^2 \leq 2L(f(x)-f(x^*))$, we conclude the second result in (1.17). □

According to the results (1.17) in the above theorem, in order to make sure that the iterate $\overline{x}_N$ of Algorithm 2 is an ϵ-approximate solution to problem (1.1), i.e., $\|\nabla f(\overline{x}_N)\|^2 \leq \epsilon$, it suffice to

set the total number of iterations N to

$$N \geq \sqrt{\frac{8L^2\|x_0 - x^*\|^2}{\epsilon}}. \tag{1.18}$$

Thus the iteration complexity of the accelerated gradient method for computing an ϵ-approximate solution with small gradient norm is of order $\mathcal{O}(L/\sqrt{\epsilon})$. Comparing the above complexity to that of the gradient descent method in (1.10), we can observe that Nesterov's method significantly improves the convergence properties comparing to that of the gradient descent method.

1.4 Other methods in the literature

In addition to the gradient descent method in Algorithm 1 and Nesterov's accelerated gradient method in Algorithm 2, there also exists other first order methods in the literature that are able to compute an ϵ-approximate solution to problem (1.1) with small gradient norm. Some of the existing methods have better convergence properties than that of Algorithms 1 and 2.

In [8], a monotone convergence accelerated gradient method is proposed that is able to compute an ϵ-solution with at most $\mathcal{O}(L^{2/3}\|x_0 - x^*\|^{2/3}/\epsilon^{1/3})$ iterations. However, the convergence property of such method is different from that of Algorithms 1 and 2, since its convergence result is with respect to the best possible iterate, i.e.,

$$\min_{i=0,\ldots,N} \|\nabla f(x_i)\|^2 \leq \mathcal{O}\left(\frac{L^2\|x_0 - x^*\|^2}{N^3}\right), \tag{1.19}$$

where $\{x_i\}_{i=0}^N$ is the sequence of iterates produced by the algorithm. The above convergence is different from that of Algorithms 1 and 2 which are with respect to last iterate or weighted average of iterates. A similar convergence result is also discovered in [3].

By perturbing the objective function of problem (1.1) to $f_\delta(x) := f(x) + (\delta/2)\|x - x_0\|^2$ and minimizing the perturbed function f_δ instead, in [8] one other first order method is described that computes an ϵ-solution of problem (1.1) with at most

$$\mathcal{O}\left(\frac{\sqrt{L\|x_0 - x^*\|}}{\epsilon^{1/4}} \ln(1 + 2L\|x_0 - x^*\|/\sqrt{\epsilon})\right) \tag{1.20}$$

iterations. Similar to the discussion above, the complexity is also concerning the best possible

iterate. So far, there has not yet been any proposed modification of such method that eliminates the $\ln(1 + 2L\|x_0 - x^*\|/\sqrt{\epsilon})$ term in the complexity.

In [4], a new first order method is proposed which computes an ϵ-solution of problem (1.1) with at most $\mathcal{O}(\sqrt{L(f(x_0) - f(x^*))/\epsilon})$ iterations. Note that such convergence result contains a $(f(x_0) - f(x^*))$ term rather than a $\|x_0 - x^*\|^2$ term as seen in all the previously discussed complexity results. Recalling the relation $f(x) - f(x^*) \leq (L/2)\|x - x^*\|^2$ in (1.4) of Lemma 1.1.1, we observe that an $\mathcal{O}(\sqrt{L(f(x_0) - f(x^*))/\epsilon})$ complexity implies an $\mathcal{O}(\sqrt{L^2\|x_0 - x^*\|^2/\epsilon})$ complexity, although the converse is not necessarily true. It should be pointed out that the analysis of [4] is based on a semidefinite programming analysis framework originated from [2], known as the performance enhancement program (PEP). The PEP-type analysis is significantly different from our previously discussed analysis of gradient descent and accelerated gradient methods. A simplified analysis that shares some analogy with that of gradient descent and accelerated gradient methods is developed in [1]. It should also be noted that although the $\mathcal{O}(\sqrt{L(f(x_0) - f(x^*))/\epsilon})$ complexity of [4] seems to be in the same $\mathcal{O}(1/\sqrt{\epsilon})$ order as the accelerated gradient method complexity (1.18) and worse than the previously mentioned results (1.19) and (1.20), in [9] it is pointed out that by combining the accelerated gradient method and the method in [4], one is actually able to obtain a best $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity.

The understanding of the aforementioned new first order method and its convergence analysis [4, 1, 9] is the main topic of this thesis, as detailed in the following chapter. Specifically, we review and summarize the results in [4, 1, 9]. We also propose a new set of parameters that achieves the same $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity as the results in [4].

Chapter 2

Optimal gradient methods for minimizing gradient norm

In this chapter, we use the performance enhancement program (PEP) framework described in [4] to design gradient method for solving unconstrained optimization problems of form (1.1), namely,

$$\min_{x\in\mathbb{R}^n} f(x). \tag{2.1}$$

Recall that we assume that f is convex and smooth, i.e., $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$. Our goal is to compute an ϵ-approximate solution $x \in \mathbb{R}^n$ such that $\|\nabla f(x)\|^2 \leq \epsilon$. We first describe in Section 2.1 the general algorithm and its convergence analysis. Then in Section 2.2 we describe the algorithm proposed in [4] and its convergence analysis, and we propose a new set of parameters that is different from [4] with the same gradient evaluation complexity. Finally, in Section 2.3, we show that by combining our analysis and the comment made in [9], it is possible to compute an approximate solution x such that $\|\nabla f(x)\|^2 \leq \epsilon$ within $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ number of iterations starting from initial point x_0.

2.1 First-order method convergence analysis through PEP

Performance estimation problem (PEP) is an efficient convergence analyses method for optimization problems. Given a certain class of objective function, such as convex smooth functions, PEP guarantees convergence to even the worst case complexity by adjusting the parameter of an algorithm. The algorithm in [4] is built on the PEP framework originally developed in [2]. For any unconstrained optimization problem of form (2.1), the PEP framework starts by assuming that each iteration x_i is built within the linear span of gradients of previous iterations, namely,

$$x_i \in x_0 + \text{span}\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_i)\}. \tag{2.2}$$

The above linear span construction covers many existing first-order algorithms, e.g., the gradient descent and accelerated gradient method, described in the previous chapter in Algorithms 1 and 2 respectively. Based on the above linear span description, the generic form of an algorithm studied in the PEP framework can be described in Algorithm 3.

Algorithm 3 Generic algorithm description in the PEP framework for solving unconstrained optimization problem (2.1)

Require: Initial point $x_0 \in \mathbb{R}^n$, maximum number of iterations N

for $i = 1, \ldots, N$ **do**

Compute

$$\begin{aligned} g_{i-1} &= \nabla f(x_{i-1}) \\ x_i &= x_{i-1} - \frac{1}{L}\sum_{k=0}^{i-1} h_{i,k} g_k \end{aligned} \tag{2.3}$$

end for

Output approximate solution x_N.

A few remarks are in place for the above algorithm. First, by induction on equation (2.3) we can easily observe that x_i is in the linear span of gradients of previous iterations, i.e., (2.2) holds. Conversely, we can also observe that for any algorithm whose iterates satisfy the linear span relationship (2.2), it can always be described in the form of Algorithm 3. Second, for different iteration number i, the coefficients $h_{i,k}$ for gradients g_k are different. Finally, gradient descent method with $h = 1/L$ described in Algorithm 1 is a special case of the PEP description above with $h_{i,k} = 0$ for all $k = 0, ..., i-2$, and $h_{i,i-1} = 1$. Note that the accelerated gradient method in Algorithm 2 is also a special case of the above PEP description, although the coefficients $h_{i,k}$ are

more complicated than that of the gradient descent method.

In order to perform convergence analysis of the generic form algorithm in Algorithm 3, we apply several properties of convex smooth functions at the algorithm's iterates. Specifically, applying Lemma 1.1.3 we have the following relationships:

$$f(x_i) - f(x_j) - \langle \nabla f(x_j), x_i - x_j \rangle \geq \frac{1}{2L} ||\nabla f(x_i) - \nabla f(x_j)||^2, i, j = 1, ..., N \tag{2.4}$$

$$f(x_i) - f(x^*) \geq \frac{1}{2L} ||\nabla f(x_i)||^2, i = 0, ..., N \tag{2.5}$$

Note that in (2.5) we use the fact that $\nabla f(x^*) = 0$. All the analysis performed throughout this chapter is based on the above three relationships. We demonstrate that by combining the above inequalities (2.4) and (2.5), we above to derive a bound of form

$$\|\nabla f(x_N)\|^2 \leq \Gamma_N \|x_0 - x^*\|^2,$$

where Γ_N is a constant that depends on the maximum number of iterations N. The convergence property of Algorithm 3 is described explicitly through the estimate of Γ_N.

Throughout the convergence analysis, we frequently utilize the following technical lemma on changing the order of double summands.

Lemma 2.1.1. *For any nonnegative integer m, n, α such that $m \leq n$, we have*

$$\sum_{i=m}^{n} \sum_{j=i+\alpha}^{n+\alpha} b_{i,j} = \sum_{j=m+\alpha}^{n+\alpha} \sum_{i=m}^{j-\alpha} b_{i,j}.$$

Here $b_{i,j}$'s are any numbers indexed by i and j.

Proof.

$$\sum_{i=m}^{n} \sum_{j=i+\alpha}^{n+\alpha} b_{i,j} = \sum_{i=m}^{n} \sum_{j=m+\alpha}^{n+\alpha} \mathbb{1}_{m \leq i \leq j-\alpha \leq n} b_{i,j} = \sum_{j=m+\alpha}^{n+\alpha} \sum_{i=m}^{j-\alpha} b_{i,j}$$

□

For Algorithm 3, after combining inequalities (2.4) and (2.5) described and substituting the

description of iterates, we obtain the following lemma concerning Algorithm 3.

Lemma 2.1.2. *For any nonnegative constants denoted by $A = (a_{ij}) \in \mathbb{R}_+^{(N+1)\times(N+1)}$, $c \in \mathbb{R}_+^{N+1}$, $r \in \mathbb{R}_+$, we have the following property concerning the iterates of Algorithm 3:*

$$Q(G) + r||g_N||^2 \leq \sum_{i=0}^{N} \left(\sum_{j=0}^{N} (a_{ij} - a_{ji}) + c_i \right) (f(x_i) - f(x^*)). \tag{2.6}$$

Here $Q(G)$ is a quadratic form of gradients $G := (g_0, g_1, \ldots, g_N) \in \mathbb{R}^{n\times(N+1)}$.

Proof. Combine inequalities (2.4) and (2.5), we have

$$\begin{aligned} 0 \leq & \sum_{i=0}^{N}\sum_{j=0}^{N} a_{ij} \left(f(x_i) - f(x_j) - \langle g_j, x_i - x_j \rangle - \frac{1}{2L}\|g_i - g_j\|^2 \right) \\ & + \sum_{i=0}^{N} c_i \left(f(x_i) - f(x^*) - \frac{1}{2L}\|g_i\|^2 \right). \end{aligned}$$

By combining terms in the above relationship, we obtain

$$\begin{aligned} & \sum_{i=0}^{N}\sum_{j=0}^{N} a_{ij} \left(\langle g_j, x_i - x_j \rangle + \frac{1}{2L}\|g_i - g_j\|^2 \right) + \frac{1}{2L}\sum_{i=0}^{N} c_i \|g_i\|^2 \\ \leq & \sum_{j=0}^{N}\sum_{i=0}^{N} a_{ij} f(x_i) - \sum_{i=0}^{N}\sum_{j=0}^{N} a_{ij} f(x_j) - \sum_{i=0}^{N} c_i f(x^*) + \sum_{i=0}^{N} c_i f(x_i) \\ = & \sum_{i=0}^{N} \left(\sum_{j=0}^{N} (a_{ij} - a_{ji}) + c_i \right) (f(x_i) - f(x^*)). \end{aligned} \tag{2.7}$$

We make some observations concerning iterates x_i and x_j in the above result. For all $j \geq i$, summing equation (2.3) from $i, i+1, \ldots$ to j we have

$$\begin{aligned} x_j &= x_{i-1} - \frac{1}{L}\sum_{l=i}^{j}\sum_{k=0}^{l-1} h_{l,k} g_k = x_{i-1} - \frac{1}{L}\sum_{k=0}^{j-1}\sum_{l=1}^{j} h_{l,k} g_k \mathbb{1}_{1\leq i\leq l\leq j} \mathbb{1}_{0\leq k\leq l-1\leq j-1} \\ &= x_{i-1} - \frac{1}{L}\sum_{k=0}^{j-1}\sum_{l=max\{i,k+1\}}^{j} h_{l,k} g_k. \end{aligned}$$

Therefore,

$$x_j = x_i - \frac{1}{L}\sum_{k=0}^{j-1}\sum_{l=max\{i+1,k+1\}}^{j} h_{l,k}g_k,\ \forall j \geq i, \tag{2.8}$$

$$x_i = x_j - \frac{1}{L}\sum_{k=0}^{i-1}\sum_{l=max\{j+1,k+1\}}^{i} h_{l,k}g_k,\ \forall i \geq j, \tag{2.9}$$

$$x_i = x_0 - \frac{1}{L}\sum_{k=0}^{i-1}\sum_{l=k+1}^{i} h_{l,k}g_k, \forall i = 1, ..., N. \tag{2.10}$$

Applying observations (2.8), (2.9), and (2.10) above to (2.7) and combining terms, we conclude (2.6). □

A few remarks are in place of the above lemma. First, in the result (2.6) we define a quadratic form $Q(G)$. This is a quadratic form concerning all the gradients $g_0, \ldots, g_N$ computed by Algorithm 3. Second, in the statement of the above lemma we did not specify the exact expression of the quadratic form $Q(G)$. Indeed, by applying (2.8) through (2.10) to (2.7) and isolating all the quadratic forms concerning gradients $g_0, \ldots, g_N$, we can observe that $Q(G)$ has the following form:

$$\begin{aligned} Q(G) :=& \frac{1}{L}\sum_{i=0}^{N-1}\sum_{j=i+1}^{N} a_{ij}\left\langle g_j, \sum_{k=0}^{j-1}\sum_{l=max\{i+1,k+1\}}^{j} h_{l,k}g_k \right\rangle \\ &- \frac{1}{L}\sum_{i=1}^{N}\sum_{j=0}^{i-1} a_{ij}\left\langle g_j, \sum_{k=0}^{i-1}\sum_{l=max\{j+1,k+1\}}^{i} h_{l,k}g_k \right\rangle \\ &+ \sum_{i=0}^{N}\sum_{j=0}^{N} a_{ij}\left(\frac{1}{2L}||g_i||^2 - \frac{1}{L}\langle g_i, g_j\rangle + \frac{1}{2L}||g_j||^2\right) + \frac{1}{2L}\sum_{i=0}^{N} c_i||g_i||^2 - r||g_N||^2. \end{aligned} \tag{2.11}$$

Third, observe that if $Q(G)$ is a positive-semidefinite quadratic form with respect to any vectors $g_0, \ldots, g_N$, then by (2.6) we have immediately that

$$r||g_N||^2 \leq \sum_{i=0}^{N}\left(\sum_{j=0}^{N}(a_{ij} - a_{ji}) + c_i\right)(f(x_i) - f(x^*)). \tag{2.12}$$

The above relationship leads directly to our convergence property of interest. For example, we have the following immediate corollary:

Corollary 2.1.1. *In the result of Lemma 2.1.2, if $Q(G)$ is a positive-semidefinite quadratic form*

with respect to any vectors $g_0, \ldots, g_N$ and the constants A and c satisfy

$$\sum_{j=0}^{N}(a_{ij} - a_{ji}) + c_i = 0, \quad \forall i = 1, ..., N, \tag{2.13}$$

then we have the following convergence property for Algorithm 3:

$$r||g_N||^2 \leq \Gamma_N(f(x_0) - f(x^*)), \text{ where } \Gamma_N := \sum_{j=0}^{N}(a_{0j} - a_{j0}) + c_0.$$

Proof. By (2.12) and $\sum_{j=0}^{N}(a_{ij} - a_{ji}) + c_i = 0$, for $i = 1, ..., N$, we obtain

$$r||g_N||^2 \leq \left(\sum_{j=0}^{N}(a_{0j} - a_{j0}) + c_0\right)(f(x_0) - f(x^*)).$$

□

By the above corollary, if the parameters $h_{i,k}$'s and the constants A, c and r are chosen properly, then the convergence of Algorithm 3 is dependent on Γ_N/r. As long as we can estimate the rate of Γ_N/r, we obtain the convergence property of Algorithm 3.

So far in all the statements of results we assume that $Q(G)$ is a positive-semidefinite quadratic form. Indeed, since any quadratic form can be described by $Q(G) = \mathrm{Tr}(SGG^\top)$ where $S \in \mathbb{R}^{(N+1)\times(N+1)}$ is a symmetric matrix, to prove positive-semidefiniteness of $Q(G)$ it suffices to prove that S is a positive-semidefinite matrix. In the following proposition we describe the entries of S explicitly.

Proposition 2.1.1. *In the statement of Lemma 2.1.2, the quadratic form $Q(G)$ can be described by*

$$Q(G) = \mathrm{Tr}(SGG^\top) = \sum_{i=0}^{N} S_{ii}||g_i||^2 + \sum_{j=1}^{N}\sum_{k=0}^{j-1} S_{j,k}\langle g_j, g_k\rangle,$$

where the entries of the symmetric matrix $S \in \mathbb{R}^{(N+1)\times(N+1)}$ are the following:

$$
\begin{aligned}
S_{ii} =& \frac{1}{2L}\left[c_i + \sum_{\substack{j=0\\ j\neq i}}^{N}(a_{ij}+a_{ji}) - 2\sum_{j=i+1}^{N} a_{ji}(\sum_{l=i+1}^{j} h_{l,i})\right],\ \forall i = 1,\dots,N \\
S_{NN} =& \frac{1}{2L}\left[c_N + \sum_{j=0}^{N-1}(a_{Nj}+a_{jN})\right] - r \\
S_{j,k} =& \frac{1}{2L}\left[-a_{jk} - a_{kj} + \sum_{i=0}^{j-1} a_{ij} \sum_{l=max\{i+1,k+1\}}^{j} h_{l,k} - (\sum_{i=j+1}^{N}(a_{ij}(\sum_{l=j+1}^{i} h_{l,k}) + a_{ik}(\sum_{l=j+1}^{i} h_{l,j})))\right], \\
& \forall j = 1,...,N, k = 0,...,j-1
\end{aligned}
\tag{2.14}
$$

Proof. Applying Lemma 2.1.1 to (2.11) we have two results. First,

$$
\begin{aligned}
\frac{1}{L}\sum_{i=0}^{N-1}\sum_{j=i+1}^{N} a_{ij}\left\langle g_j, \sum_{k=0}^{j-1}\sum_{l=max\{i+1,k+1\}}^{j} h_{l,k} g_k\right\rangle =& \frac{1}{L}\sum_{i=0}^{N-1}\sum_{j=i+1}^{N}\sum_{k=0}^{j-1}\sum_{l=max\{i+1,k+1\}}^{j} a_{ij}h_{l,k}\langle g_j, g_k\rangle \\
=& \frac{1}{L}\sum_{j=1}^{N}\sum_{i=0}^{j-1}\sum_{k=0}^{j-1}\sum_{l=max\{i+1,k+1\}}^{j} a_{ij}h_{l,k}\langle g_j, g_k\rangle \\
=& \frac{1}{L}\sum_{j=1}^{N}\sum_{k=0}^{j-1}\sum_{i=0}^{j-1} a_{ij}\sum_{l=max\{i+1,k+1\}}^{j} h_{l,k}\langle g_j, g_k\rangle.
\end{aligned}
\tag{2.15}
$$

In the above derivation we apply Lemma 2.1.1 three consecutive times. Second, by a similar consecutive application of Lemma 2.1.1, we also have

$$
\begin{aligned}
& -\frac{1}{L}\sum_{i=1}^{N}\sum_{j=0}^{i-1} a_{ij}\left\langle g_j, \sum_{k=0}^{i-1}\sum_{l=max\{j+1,k+1\}}^{i} h_{l,k} g_k\right\rangle \\
=& -\frac{1}{L}\sum_{i=2}^{N}\sum_{j=i}^{i-1}\sum_{k=0}^{j-1}\sum_{l=j+1}^{i} a_{ij}h_{l,k}\langle g_j, g_k\rangle - \frac{1}{L}\sum_{i=1}^{N}\sum_{j=0}^{i-2}\sum_{k=j+1}^{i-1}\sum_{l=k+1}^{i} a_{ij}h_{l,k}\langle g_j, g_k\rangle \\
& -\frac{1}{L}\sum_{i=1}^{N}\sum_{j=0}^{i-1}\sum_{l=j+1}^{i} a_{ij}h_{l,j}||g_j||^2 \\
=& -\frac{1}{L}\sum_{j=1}^{N-1}\sum_{k=0}^{j-1}\sum_{i=j+1}^{N} a_{ij}\sum_{l=j+1}^{i} h_{l,k}\langle g_j, g_k\rangle - \frac{1}{L}\sum_{j=1}^{N-1}\sum_{k=0}^{j-1}\sum_{i=j+1}^{N}\sum_{l=j+1}^{i} a_{ik}h_{l,j}\langle g_k, g_j\rangle \\
& -\frac{1}{L}\sum_{i=0}^{N-1}\sum_{j=i+1}^{N} a_{ji}\sum_{l=i+1}^{j} h_{l,i}||g_i||^2
\end{aligned}
\tag{2.16}
$$

Reading coefficients of $\|g_i\|^2$ and $\langle g_j, g_k\rangle$ from (2.11) and noting the above results (2.15) and (2.16), we conclude the description of S in (2.14) immediately. □

Our results in Lemma 2.1.2, Corollary 2.1.1, and Proposition 2.1.1 allows us to study the convergence of Algorithm 3 by finding nonnegative constants A, b, c, d, and r such that the matrix S defined in (2.14) is positive-semidefinite. For bookkeeping purpose, we summarize the convergence analysis concept in the following theorem.

Theorem 2.1.1. *Suppose that nonnegative constants denoted by $A = (a_{ij}) \in \mathbb{R}_+^{(N+1)\times(N+1)}$, $c \in \mathbb{R}_+^{N+1}$, and $r \in \mathbb{R}_+$ are chosen such that* (2.13) *holds. Then we have the following property concerning the iterates of Algorithm 3:*

$$\mathrm{Tr}(SGG^\top) + r||g_N||^2 \leq \Gamma_N(f(x_0) - f(x^*)), \text{ where } \Gamma_N := \sum_{j=0}^{N}(a_{0j} - a_{j0}) + c_0.$$

where S is defined in (2.14) *in Proposition 2.1.1.*

Proof. Immediate from Lemma 2.1.2, Corollary 2.1.1, and Proposition 2.1.1. □

By Theorem 2.1.1 above, in order to design efficient algorithms in the form described by Algorithm 3, it suffices to find algorithm parameters $h_{i,k}$ and constants A, b, c, d, and r such that the matrix S described in (2.14) is positive-semidefinite. In the following section, we describe a few choices of algorithm parameters $h_{i,k}$ and constants A, c, and r that yield efficient algorithms for computing approximate solutions of problem (2.1) with small gradient norms.

2.2 First-order methods based on special cases of PEP

We start with the description of parameter and constant choice in [4] that achieves an $\mathcal{O}(\sqrt{(f(x_0) - f(x^*))/\varepsilon})$ complexity for computing an ϵ-solution with small gradient norm. Note that in [4] there is no discussion on the rationale behind the choice of parameters and constants. However, we show later through the proof of Theorem 2.2.1 that their parameters are chosen so that the all the entries of S except S_{NN} are all 0. Motivated by their parameter and constant choice, we propose later in this section a new choice of parameters with the same $\mathcal{O}(\sqrt{(f(x_0) - f(x^*))/\varepsilon})$ complexity.

The analysis performed in this section is relatively technical and detail oriented. Therefore, before delving deeply into the technical details, here we briefly describe the main ingredients behind the analysis throughout this section. Note that according to Theorem 2.1.1, our goal becomes finding proper parameters $\{h_{i,k}\}$ and constants A and c to obtain an upper bound of $\|\nabla f(x_N)\|^2$ of the form $\|\nabla f(x_N)\|^2 \leq \Gamma_N(f(x_0) - f(x^*))$. Such goal is equivalent to finding $\{h_{i,k}\}$, A and c such that S is positive semi-definite, and that $\sum_{j=0}^{N}(a_{ij} - a_{ji}) + c_i = 0$ for any $i = 1, \ldots, N$. While determining positive semi-definiteness is not necessarily straightforward, in the special case when S is diagonal it becomes trivial to determine its positive semi-definiteness. As we show in the sequel, the result in [4] yields a simple diagonal S with only one nonzero entry. Following such concept of constructing simple diagonal S, we are able to derive a new set of parameters that enjoys the same order of complexity as the method in [4].

Inspired by the analysis in [4], we first introduce a specialized setup of parameters $\{h_{i,k}\}$ and constants A and c of Algorithm 3.

Proposition 2.2.1. *Let f be a function such that $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and $\{x_i\}_{i=0}^N$ be the iterations of Algorithm 3 applied to minimize f. Suppose that a_i', α_i and β_i are positive constants such that $a_i' \geq a_{i-1}'$ for all $i = 1, \ldots, N$, and that the parameters $\{h_{i,k}\}$ in Algorithm 3 are set to*

$$h_{ik} = \begin{cases} 1 + (\alpha_i - \alpha_{i+1})(\beta_{k+1} - \beta_k), & i = 1, ..., N, k = i - 1 \\ (\alpha_i - \alpha_{i+1})(\beta_{k+1} - \beta_k), & i = 2, ..., N, k = 0, ..., i - 2. \end{cases} \tag{2.17}$$

Let $G = (g_0, ...g_N) \in \mathbb{R}^{(n)\times(N+1)}$ be the matrix consisting of all gradients involved in Algorithm 3, then $r = (1/L)a_N'$ derives the fastest convergence and we have the following convergence property of Algorithm 3:

$$\mathrm{Tr}(SG^\top G) + \frac{1}{L}a_N'||g_N||^2 \leq 2a_0'(f(x_0) - f(x^*)), \tag{2.18}$$

where $S = \{S_{i,k}\} \in \mathbb{R}^{(N+1)\times(N+1)}$ *is a symmetric matrix whose entries are*

$$\begin{aligned}
S_{ii} &= \frac{1}{2L}(2a'_i - 2(a'_{i+1} - 2a'_i)(\beta_{i+1} - \beta_i)(\alpha_{i+1} - \alpha_{N+1})), \quad i = 0, ..., N-1; \\
S_{NN} &= 0; \\
S_{jk} &= \frac{1}{2L}(a'_j(\alpha_j - \alpha_{j+1})(\beta_{k+1} - \beta_k) - (a'_{j+1} - a'_j)(\alpha_{j+1} - \alpha_{N+1})(\beta_{k+1} - \beta_k) \\
&\quad - (a'_{k+1} - a'_k)(\alpha_{j+1} - \alpha_{N+1})(\beta_{j+1} - \beta_j) - (a'_{k+1} - a'_k)), \quad j = 1, ..., N, k = 0, ..., j-1.
\end{aligned} \tag{2.19}$$

Proof. Let us set constants A and c to be the following:

$$c_i = \begin{cases} 0, & i = 1, \ldots, N-1 \\ a'_0, & i = 0 \text{ or } N, \end{cases} \qquad a_{jk} = \begin{cases} a'_i, & j = i-1, k = i, i = 1, ..., N \\ a'_i - a'_{i-1}, & j = N, k = i-1, i = 1, ..., N \\ 0 & \text{otherwise} \end{cases} \tag{2.20}$$

It is easy to verify that the conditions for Theorem 2.1.1 holds. Moreover, applying the choices of $h_{i,k}$ in (2.17) and constants A and c in (2.20) to the entries of S described in (2.14), we obtain S_{ii} for $i = 0, ..., N-1$ and S_{jk} for $j = 1, ..., N$, $k = 0, ..., j-1$ in (2.19), and $S_{NN} = (1/L)a'_N - r$. Also, since we want our algorithm to converge as fast as possible, we would like the constant r to be as large as possible. Thus we define $r = (1/L)a'_N$, so that $S_{N,N} = 0$. And hence (2.18) and (2.19) are derived.

□

With the help of the above proposition, we are now ready to describe the convergence properties of the first-order method described in [4]. We start with the following proposition.

Proposition 2.2.2. *Let* f *be a function such that* $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ *and* $\{x_i\}_{i=0}^N$ *be the iterations of Algorithm 3 applied to minimize* f*. Suppose that the parameters* $\{h_{i,k}\}$ *in Algorithm 3 are set to*

$$h_{ik} = \begin{cases} 1 + (\theta_i^4 - \theta_{i+1}^4)(1/\theta_{k+1}^2 - 1/\theta_k^2), & i = 1, ..., N, k = i-1; \\ (\theta_i^4 - \theta_{i+1}^4)(1/\theta_{k+1}^2 - 1/\theta_k^2), & i = 2, ..., N, k = 0, ..., i-2, \end{cases} \tag{2.21}$$

where $\theta_{N+1} = 0$ *and* $\theta_i = (1 + \sqrt{1 + 4\theta_{i+1}^2})/2$ *for* $i = 0, ..., N$*, we have the following convergence*

property of Algorithm 3:

$$\frac{1}{L}a'_N||g_N||^2 \leq 2a'_0(f(x_0) - f(x^*)). \tag{2.22}$$

Proof. In Proposition 2.2.1, let us set $r = (1/L)a'_N$ and

$$\begin{aligned}
&\alpha_i := \theta_i^4, \quad i = 1, ..., N+1;\\
&\beta_i := \frac{1}{\theta_i^2}, \quad i = 0, ..., N;\\
&a'_i := \frac{1}{\theta_i^2}, \quad i = 0, ..., N, \text{ where}\\
&\theta_i := \begin{cases} 0, & i = N+1;\\ \frac{1+\sqrt{1+4\theta_{i+1}^2}}{2}, & i = 0, ..., N. \end{cases}
\end{aligned} \tag{2.23}$$

Here we observe that the sequence $\{\theta_i\}_{i=0}^{N+1}$ defined in (2.23) satisfies the following relationships:

$$\begin{aligned}
&\theta_i^2 = \theta_{i+1}^2 + \theta_i, i = 0, ..., N-1,\\
&\theta_0^2 = \theta_N^2 + \sum_{i=0}^{N} \theta_i.
\end{aligned} \tag{2.24}$$

By the parameter and constant settings in (2.23) and applying relationships (2.24), the entries of S in (2.19) are simplified as follows:

$$S_{ii} = \frac{1}{2L}\left(2\frac{1}{\theta_i^2} - 2\left(\frac{1}{\theta_{i+1}^2} - \frac{1}{\theta_i^2}\right)^2(\theta_{i+1}^4 - \theta_{N+1}^4)\right) = 0, \quad i = 0, ..., N-1;$$

$$S_{NN} = 0;$$

$$S_{jk} = \frac{1}{2L}\left(\frac{1}{\theta_j^2}(\theta_j^4 - \theta_{j+1}^4)\left(\frac{1}{\theta_{k+1}^2} - \frac{1}{\theta_k^2}\right) - \left(\frac{1}{\theta_{j+1}^2} - \frac{1}{\theta_j^2}\right)(\theta_{j+1}^4 - 0)\left(\frac{1}{\theta_{k+1}^2} - \frac{1}{\theta_k^2}\right)\right) = 0,$$
$$\forall j = 1, ..., N, k = 0, ..., j-1.$$

In summary, we have $S = 0$, and hence $\mathrm{Tr}(SG^\top G) = 0$. Thus by (2.18), we obtain (2.22). □

By the above proposition, it suffices to estimate bounds of a_N and a_0 to derive the convergence bound of the method proposed in [4]. We describe the convergence property in the following

theorem.

Theorem 2.2.1. *Let f be a function such that $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and $\{x_i\}_{i=0}^N$ be the iterations of Algorithm 3 to minimize f. The algorithm proposed in [4] has a $\mathcal{O}\left(\sqrt{(f(x_0)-f(x^*))/\epsilon}\right)$ convergence with parameters $\{h_{i,k}\}$ defined by (2.21) in Proposition 2.2.2. Specifically, we have*

$$||g_N||^2 \leq \frac{4\sqrt{5}L}{(N+1)(N+2\sqrt{5})}(f(x_0)-f(x^*)) \tag{2.25}$$

Proof. Since $\theta_i = (1+\sqrt{1+4\theta_{i+1}^2})/2$, we observe immediately that $\theta_i \geq \theta_{i+1}$ and $\theta_i \geq 1$. Using such observations and applying (2.24) we have that for all $i = 0, ..., N-2$,

$$\begin{aligned}\theta_i - \theta_{i+1} =& \frac{1}{2}(\sqrt{1+4\theta_{i+1}^2} - \sqrt{1+4\theta_{i+2}^2}) \\ =& \frac{2\theta_{i+1}}{\sqrt{1+4\theta_{i+1}^2}+\sqrt{1+4\theta_{i+2}^2}} \geq \frac{2\theta_{i+1}}{2\sqrt{1+4\theta_{i+1}^2}} \geq \frac{\theta_{i+1}}{\sqrt{5\theta_{i+1}^2}} = \frac{1}{\sqrt{5}}.\end{aligned}$$

Thus we have $\theta_i \geq \theta_N + (N-i)/\sqrt{5}$. Here by the definition of θ_i we have $\theta_N = 1$, and hence

$$\begin{aligned}&\sum_{i=0}^N \theta_i \geq \sum_{i=0}^N \left(\theta_N + \frac{(N-i)}{\sqrt{5}}\right) = N + \frac{N(N+1)}{2\sqrt{5}} \textbf{ and} \\ &\theta_0^2 = \theta_N^2 + \sum_{i=0}^N \theta_i \geq 1 + N + \frac{N(N+1)}{2\sqrt{5}} = (N+1)\left(1+\frac{N}{2\sqrt{5}}\right).\end{aligned}$$

Recalling the definitions of $\{a_i'\}_{i=0}^N$ in (2.23) we have an upper bound of a_0':

$$a_0' = \frac{1}{\theta_0^2} \leq \frac{1}{(N+1)(1+\frac{N}{2\sqrt{5}})} = \frac{2\sqrt{5}}{(N+1)(N+2\sqrt{5})}.$$

Also note that $a_N' = 1/\theta_N^2 = 1$. Thus by (2.22), we conclude (2.25). □

It should be noted that in Theorem 2.2.1, the convergence result has constant $f(x_0)-f(x^*)$ at the right hand side rather than $\|x_0 - x^*\|^2$ as seen previously in Sections 1.2 and 1.3. Clearly, we can simply use the property of convex smooth function (1.4), i.e., $f(x) - f(x^*) \leq (L/2)||x-x^*||^2$,

to obtain that

$$||g_N||^2 \leq \frac{2\sqrt{5}L^2}{(N+1)(N+2\sqrt{5})}\|x_0 - x^*\|^2.$$

In other words, we obtain an $\mathcal{O}\left(L\|x_0 - x^*\|\sqrt{1/\epsilon}\right)$ complexity for solving problem (2.1) with small gradient norm square $\|g_N\|^2$.

Comparing the above complexity result with that discussed previously in Sections 1.2, 1.3, 1.4, we observe that the above complexity result matches that of the accelerated gradient method in (1.18), but is worse than that of (1.19) and (1.20). However, we show later in the following section that the above analysis actually leads to the optimal complexity in the order of $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$.

However, before moving to the optimal complexity bound analysis, we would like to conclude this section by addressing a small issue in the above result. Here, the parameter choice $h_{i,k}$ depends on a sequence $\{\theta_i\}_{i=0}^{N+1}$, but the exact dependence of $h_{i,k}$ with respect to iteration count i and maximum number of iteration N is not very clear. This is because the sequence $\{\theta_i\}_{i=0}^{N+1}$ is only defined recursively backwards from $\theta_{N+1} = 0$. It would be more preferable if we are able to develop parameter choice $h_{i,k}$ that has explicit dependence in i and N.

In order to develop a different set of parameters $h_{i,k}$ from [4], we make the following observation on the choice of matrix S in the above analysis. Note that the parameters are chosen intentionally to make sure that $S = 0$. However, note that the crucial idea behind the proof is to make sure that S is positive-semidefinite. Therefore, we do not necessarily need to enforce that S is a zero matrix. Rather, as long as S is diagonal, it is already easy to determine its positive-semidefiniteness. In the following proposition, we study possible necessary condition for S to be diagonal and positive-semidefiniteness.

Proposition 2.2.3. *For constants a'_i, α_i and β_i in Proposition 2.2.1, if they satisfy the following conditions:*

1. *$\{a'_i\}_{i=0}^{N}$ is a non-negative monotone increasing sequence;*
2. *There exist some $\xi \in \mathbb{R}_+$, such that*

$$\beta_{k+1} - \beta_k = \xi(a'_{k+1} - a'_k) \text{ and } a'_{i+1} - a'_i = \frac{1}{2\alpha_{i+1}}\left[a'_i(\alpha_i - \alpha_{i+1}) - \frac{1}{\xi}\right]. \tag{2.26}$$

3. There exists non-negative monotone decreasing sequence $\{\alpha_i\}_{i=1}^{N+1}$ *with* $\alpha_{N+1} = 0$ *such that*

$$\frac{1}{\xi(\sqrt{\alpha_i} + \sqrt{\alpha_{i+1}})^2} \leq a'_i \leq \frac{1}{\xi(\sqrt{\alpha_i} - \sqrt{\alpha_{i+1}})^2}, \quad i = 0, .., N-1. \tag{2.27}$$

Then setting $r = \frac{1}{L}a'_N$ *as in Proposition 2.2.1, the matrix* S *in the result* (2.18) *of Proposition 2.2.1 is positive semi-definite.*

Proof. As discussed before, we would like S to a diagonal matrix. Hence, we need to prove that all off-diagonal entries of S are 0. In order for this to happen, by the description of entries of S in (2.19), the relationship that $S_{jk} = 0$ for all $j = 1, ..., N$ and $k = 0, ..., j-1$ is equivalent to

$$\frac{\beta_{k+1} - \beta_k}{a'_{k+1} - a'_k} = \frac{1 + (\alpha_{j+1} - \alpha_{N+1})(\beta_{j+1} - \beta_j)}{a'_j(\alpha_j - \alpha_{j+1}) - (a'_{j+1} - a'_j)(\alpha_{j+1} - \alpha_{N+1})}, \quad k = 0, ..., j-1, \forall j.$$

Noting that $\alpha_{N+1} = 0$, the above is equivalent to

$$\beta_{k+1} - \beta_k = \xi(a'_{k+1} - a'_k) \text{ where } \xi \text{ satisfies } \xi = \frac{1 + \alpha_{j+1}\xi(a'_{j+1} - a'_j)}{a'_j(\alpha_j - \alpha_{j+1}) - (a'_{j+1} - a'_j)\alpha_{j+1}}, \quad \forall j.$$

The second relationship concerning ξ is equivalent to

$$a'_{i+1} - a'_i = \frac{1}{2\alpha_{i+1}}\left[a'_i(\alpha_i - \alpha_{i+1}) - \frac{1}{\xi}\right].$$

Therefore, the condition (2.26) guarantees that S is zero off-diagonal entries.

Let us now move our focus to the diagonal entries of S. By (2.19) and (2.26), we have that $S_{NN} = 0$ and also that for $i = 0, ..., N-1$

$$\begin{aligned} S_{ii} =& \frac{1}{2L}(2a'_i - 2\xi(a'_{i+1} - 2a'_i)^2(\alpha_{i+1} - \alpha_{N+1})) \\ =& -\frac{\xi}{2\alpha_{i+1}}\left[a'^2_i(\alpha_i - \alpha_{i+1})^2 - \frac{2}{\xi}a'_i(\alpha_i + \alpha_{i+1}) + \frac{1}{\xi^2}\right]. \end{aligned}$$

In order to have $S_{ii} \geq 0$, the quadratic term within the brackets should be non-positive, which is equivalent to (2.27). □

The above proposition describes possible conditions for S to be a diagonal and positive-semidefinite matrix. Therefore, it suffices to find proper parameters ξ, a_i and α_i that satisfies the

conditions of the above proposition. Actually, for ξ simply set to 1, we can find a specific set of parameters that meets the requirement of Proposition 2.2.3 and we introduce them in the following theorem.

Theorem 2.2.2. *Let f be a function such that $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and $\{x_i\}_{i=0}^N$ be the iterations of Algorithm 3 applied to minimize f. Suppose that the parameters $\{h_{i,k}\}$ in Algorithm 3 are set to*

$$h_{ik} = \begin{cases} 1 + \dfrac{2(N-i+1)(N-i+2)(N-i+3)}{(N-k+1)(N-k+2)(N-k+3)}, \ i = 1, ..., N, k = i-1; \\ \dfrac{2(N-i+1)(N-i+2)(N-i+3)}{(N-k+1)(N-k+2)(N-k+3)}, \ i = 2, ..., N, k = 0, ..., i-2, \end{cases} \tag{2.28}$$

then the algorithm has a $\mathcal{O}\left(\sqrt{L(f(x_0) - f(x^*))/\epsilon}\right)$ complexity. Specifically, we have

$$||g_N||^2 \leq \frac{6L}{(N+2)(N+3)}(f(x_0) - f(x^*)). \tag{2.29}$$

Proof. Let us set constants to $\xi = 1$, $r = (1/L)a'_N$, $\alpha_{N+1} = 0$, $a'_N = 1$,

$$\begin{aligned} a'_i &= \frac{6}{(N-i+2)(N-i+3)}, \text{ and} \\ \alpha_i &= \frac{1}{24}(N-i+1)(N-i+2)(N-i+3)(N-i+4). \end{aligned} \tag{2.30}$$

We can verify that the above constants satisfy the conditions of Proposition 2.2.3. According to (2.30), we can immediately see that $\{\alpha_i\}_{i=1}^{N+1}$ and $\{a'_i\}_{i=0}^N$ are non-negative monotone decreasing and increasing respectively. Also, for all $N \geq 1$ and $i = 0, ..., N-1$, (2.27) is satisfied too. Thus by Proposition 2.2.3, S is positive semi-definite. Now by (2.17) in Proposition 2.2.1, we obtain

$$\frac{1}{L}a'_N||g_N||^2 \leq 2a'_0(f(x_0) - f(x^*)). \tag{2.31}$$

Applying the values of α'_N and α'_0 in (2.30) to (2.31), we directly obtain the convergence result (2.29). Consequently, Algorithm 3 has a $\mathcal{O}\left(\sqrt{L(f(x_0) - f(x^*))/\epsilon}\right)$ complexity. □

We have several remarks regarding the convergence result above. First, our method with parameters h_{ik} defined in (2.28) has the same $\mathcal{O}\left(\sqrt{L(f(x_0) - f(x^*))/\epsilon}\right)$ complexity as that of [4] described in Theorem 2.2.1. The difference is that our method has h_{ik} explicitly expressed by

i, k, and N while the method in [4] relies on a recursively defined sequence $\{\theta_k\}_{k=0}^{N+1}$. Second, although it seems that our method with parameters h_{ik} defined in (2.28), method presented in [4] and Nesterov's accelerated gradient method all have an $\mathcal{O}\left(\sqrt{1/\epsilon}\right)$ complexity, our result above and the result in [4] depends on $(f(x_0) - f(x^*))$. As shown in the following section, a result in [9] states that combining either of these two methods with Nesterov's accelerated gradient method, we have an optimal $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity for solving problem (2.1) with small gradient norm.

2.3 Two-phase optimal gradient methods for minimizing gradient norm

In this section, we demonstrate that combining either method described in the previous section with Nesterov's accelerated gradient method, one could derive an optimal $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity. Such result of combination of method in [4] and Nesterov's accelerated gradient method is previously commented in a remark in [9], although such comment is not the focus of [9]. We apply the same technique to the gradient method with our new parameters in (2.28). Specifically, Consider the an algorithm with two phases. In phase one we perform $N/2$ iterations of Nesterov's accelerated gradient method with parameters defined in Proposition 1.3.2. In phase two, we continue with $N/2$ iterations of Algorithm 3. The details of the two-phase algorithm is described in Algorithm 4.

Note that with parameters h_{ik} defined be (2.21), Algorithm 4 is the procedure mentioned in [9]. In the following theorem, we introduce the convergence analysis of Algorithm 4 with our parameters h_{ik} defined in (2.28).

Theorem 2.3.1. *Suppose that the maximum number of iterations N is a pre-specified even number. Let f be a convex smooth function such that $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^n)$ and $\{\overline{x}_i\}_{i=0}^{N/2}$ and $\{x_i\}_{i=N/2+1}^{N}$ be iterates of Algorithm 4 with parameters h_{ik} defined in* (2.28)*. Then Algorithm 4 achieves an $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity. Specifically,*

$$\|g_N\|^2 \leq \frac{96L^2}{N(N+2)^2(N+3)}\|x_0 - x^*\|_2^2. \tag{2.32}$$

Proof. In phase one, we iterate $N/2$ steps of Nesterov's accelerated gradient method. By complexity

Algorithm 4 Two-phase optimal gradient methods

Require: Initial point $x_0 \in \mathbb{R}^n$, $\gamma_i \geq 0$, $\alpha_i \in [0,1]$, N a even number,

Set $\overline{x}_0 = x_0$.

for $i = 1, \ldots, \frac{N}{2}$ **do**

Compute

$$\begin{aligned} \underline{x}_i =& \frac{i-1}{i+1}\overline{x}_{i-1} + \frac{2}{i+1}x_{i-1} \\ x_i =& \operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{\frac{i}{2L}\langle\nabla f(\underline{x}_i), x\rangle + \|x_{i-1} - x\|_2^2\right\} \\ \overline{x}_i =& \frac{i-1}{i+1}\overline{x}_{i-1} + \frac{2}{i+1}x_i \end{aligned}$$

end for

Set $x_{N/2} = \overline{x}_{N/2}$.

for $i = N/2+1, \ldots, N$ **do**

Compute

$$\begin{aligned} g_{i-1} &= \nabla f(x_{i-1}) \\ x_i &= x_{i-1} - \frac{1}{L}\sum_{k=0}^{i-1} h_{i,k} g_k \end{aligned}$$

end for

Output approximate solution x_N.

bound (1.17),

$$f(\overline{x}_{N/2}) - f(x^*) \leq \frac{16L}{N(N+2)}\|x_0 - x^*\|_2^2. \tag{2.33}$$

Then in phase two we start from $\overline{x}_{N/2}$ and iterate $N/2$ steps of Algorithm 3 with h_{ik} defined in (2.28). By complexity bound (2.29), we have

$$||g_N||^2 \leq \frac{6L}{(N+2)(N+3)}(f(\overline{x}_{N/2}) - f(x^*)). \tag{2.34}$$

Combining (2.33) and (2.34), we conclude that

$$||g_N||^2 \leq \frac{96L^2}{N(N+2)^2(N+3)}\|x_0 - x^*\|_2^2.$$

In other words, an $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity is obtained.

□

According to the result (2.32) in the above theorem, in order to make sure that the iterate x_N of the combined algorithm is an ϵ-approximate solution to problem (1.1), i.e., $\|\nabla f(\overline{x}_N)\|^2 \leq \epsilon$, it suffice to set the total number of iterations N to

$$N \geq \mathcal{O}\left(\frac{\sqrt{L\|x_0 - x^*\|}}{\epsilon^{1/4}}\right). \tag{2.35}$$

Indeed, the complexity result above could not be improved to any better order, for the following reason. By [8], for any first-order method for minimizing convex smooth functions in the class $\mathcal{F}_L^{1,1}(\mathbb{R}^n)$ that produces iterates $\{x_i\}_{i=1}^N \in \mathbb{R}^n$ (where $N \leq (n-2)/2$), it is known that there is a lower complexity bound with respect to function value difference $f(x_N) - f(x^*)$. Specifically, we have

$$f(x_N) - f(x^*) \geq \frac{3L\|x_0 - x^*\|^2}{32(N+1)^2}. \tag{2.36}$$

Since f is convex smooth, we have $f(x_N) - f(x^*) \leq \langle \nabla f(x_N), x_N - x^* \rangle$ and hence

$$(f(x_N) - f(x^*))^2 \leq \|\nabla f(x_N)\|^2 \cdot \|x_N - x^*\|^2. \tag{2.37}$$

Combining (2.36) and (2.37), we can immediately observe that if there exists any first-order method whose complexity for computing approximate solutions with small gradient norm is better than (2.35) (with respect to the order of ϵ), then such algorithm produces approximate solutions with small objective function difference $f(x_N) - f(x^*)$ better than the lower complexity bound, leading to a contradiction.

2.4 Conclusion

We discuss convergence performance of first-order methods to compute an ϵ-approximate solution for minimizing convex smooth function f at the N-th iteration with small gradient norm.

As background introduction and literature review, we first show that the gradient descent method with constant step size $h = 1/L$ has a $\mathcal{O}(L^2\|x_0 - x^*\|^2/\epsilon)$ convergence with respect to $\|\nabla f(x_N)\|^2$. The convergence performance of the gradient descent method is not good enough. Next we introduc Nesterov's accelerated gradient method, which gives a $\mathcal{O}(L\|x_0 - x^*\|\sqrt{1/\epsilon})$ convergence

with respect to $\|\nabla f(x_N)\|^2$. The convergence performance of Nesterov's accelerated gradient method is much better than gradient descent method, but it can still be better. And then we briefly introduce some other first-order methods in literature such as a monotone convergence accelerated gradient method with $\mathcal{O}(L^{2/3}\|x_0 - x^*\|^{2/3}/\epsilon^{1/3})$ complexity and a perturbed gradient method with a $\mathcal{O}((\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})\ln(1 + 2L\|x_0 - x^*\|/\sqrt{\epsilon}))$ complexity. Those convergence performance is better than Nesterov's accelerated gradient method, but still not the best.

Our focus is on optimal first-order method for computing solutions with small gradient norm. In [4], a first-order method with a $\mathcal{O}(\sqrt{L(f(x_0) - f(x^*))}/\sqrt{\epsilon})$ convergence is proposed. An extension is suggested in [9]: combining the accelerated gradient method and the method in [4], we obtain the best $\mathcal{O}(\sqrt{L}/\epsilon^{1/4})$ complexity. Inspired by the above procedure, we propose an optimal gradient method for minimizing gradient norm. We find a set of parameters that allow an $\mathcal{O}(\sqrt{L(f(x_0) - f(x^*))}/\sqrt{\epsilon})$ convergence in terms of $\|\nabla f(x_N)\|^2$. Hence, combining with Nesterov's accelerated gradient method, we are able to obtain the optimal $\mathcal{O}(\sqrt{L\|x_0 - x^*\|}/\epsilon^{1/4})$ complexity.

Note that Algorithm 2.32 requires a pre-specified maximum number of iteration N. An interesting future direction is to design a method that relaxes such requirement.

Bibliography

[1] Jelena Diakonikolas and Puqian Wang. Potential function-based framework for making the gradients small in convex and min-max optimization. *arXiv preprint arXiv:2101.12101*, 2021.

[2] Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1):451–482, 2014.

[3] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[4] Donghwan Kim and Jeffrey A Fessler. Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions. *Journal of optimization theory and applications*, 188(1):192–219, 2021.

[5] Guanghui Lan. *First-order and stochastic optimization methods for machine learning*. Springer, 2020.

[6] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[7] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $\mathcal{O}(1/k^2)$. In *Doklady an ussr*, volume 269, pages 543–547, 1983.

[8] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

[9] Yurii Nesterov, Alexander Gasnikov, Sergey Guminov, and Pavel Dvurechensky. Primal–dual accelerated gradient methods with small-dimensional relaxation oracle. *Optimization Methods and Software*, 36(4):773–810, 2021.