8-2024

Efficient First-Order Methods for Some Smooth Nonlinear Optimization Problems

Yunheng Jiang
yunhenj@g.clemson.edu

Efficient first-order methods for some smooth nonlinear optimization problems

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mathematical Sciences

---

by
Yunheng Jiang
August 2024

Accepted by:
Dr. Yuyuan Ouyang, Committee Chair
Dr. Yibo Xu, Committee Co-Chair
Dr. Xinyi Li
Dr. Boshi Yang
Dr. Zhe Zhang

Abstract

The goal of this dissertation is to study algorithm design and complexity analysis of first-order methods for solving some smooth nonlinear optimization problems. It consists of three research projects. The first project is concerning the matrix-vector multiplication complexity for solving kernel projection problems and its application on decentralized consensus optimization. The second project focuses on optimal first-order algorithm design for gradient norm minimization. The third project is the study of conditional gradient type algorithms for functional constrained nonlinear optimization problems.

The first project is elaborated in Chapter 2. Specifically, we study the problem of projecting a vector to the kernel of a specified matrix. The complexity of algorithms for solving approximate solutions of such problem is evaluated by the number of matrix-vector multiplications needed for computing an approximate solution. We first study this problem from three perspectives: the control perspective, the optimization perspective, and the linear algebra perspective. From the control perspective, a first-order method has already been proposed in previous control literature. However, we find that such method converges slower than it is claimed, as its best linear convergence rate has an additional logarithmic term in the worst case. From the optimization perspective, motivated by the accelerated gradient method, we design a first-order method whose complexity is optimal in order. From the linear algebra perspective, there has been known study of exact oracle complexity for solving a linear equation provided with the same matrix-vector oracle. Here we use the term "exact complexity" to emphasize that not only the complexity is optimal in order, but also the constant appearing in the complexity bound is unimprovable. However, such exact complexity result in solving linear equation systems is not readily applicable to our problem of interest. Based on our observations from the three perspectives, under a linear-span assumption, we propose a novel iterative method which attains the exact oracle complexity for our problem of interest. In the realm

of general methods, we provide an exact lower complexity bound with the assumption that the dimension of our problem is sufficiently large.

In the second project, our focus is to design optimal gradient method for solving unconstrained optimization problems with convex and smooth objective function. This project is described in Chapter 3. Specifically, our goal is to compute an ϵ-approximate solution $x \in \mathbb{R}^n$ such that $\|\nabla f(x)\|^2 \leq \epsilon$ by one uniform algorithm. The algorithm we study is gradient extrapolation method (GEM), which was previously developed for function value difference minimization: the number of iterations required to obtain an approximate solution x_N such that $f(x_N) - f(x^*) \leq \varepsilon$ is bounded by $\mathcal{O}\left(\sqrt{L_f \|x_0 - x^*\|^2/\varepsilon}\right)$, where L_f is the smoothness constant of function f, x_0 is the initial iterate, and x^* is an optimal solution. In this dissertation, we show that with appropriately chosen parameters, GEM can compute an ϵ-solution $x \in \mathbb{R}^n$ satisfying $\|\nabla f(x)\|^2 \leq \epsilon$ with at most $\mathcal{O}(\sqrt{L_f(f(x_0) - f(x^*))/\epsilon})$ gradient evaluations. Consequently, if we first apply GEM with the parameters for minimizing function value difference, then apply GEM with the parameters for gradient norm minimization, we are able to compute an ε-approximate solution with optimal $\mathcal{O}(\sqrt{L_f \|x_0 - x^*\|}/\varepsilon^{1/4})$ complexity.

In the third project, we focus on designing projection-free algorithms for solving functional constrained optimization problems. This project is described in Chapter 4. We first provide two projection-free methods for solving convex functional constrained optimization problems. The constrained conditional gradient (CCG) method is proposed such that the gradient evaluation complexity and the linear objective optimization complexity are both of order $\mathcal{O}(1/\varepsilon)$. Then we incorporate a sliding procedure in CCG so that it could obtain a better $\mathcal{O}(1/\sqrt{\varepsilon})$ gradient evaluation complexity while maintaining the $\mathcal{O}(1/\varepsilon)$ linear objective optimization complexity. Moreover, we show that CCG with a linesearch strategy can also be adapted for solving nonconvex functional constrained optimization problems. If the objective function is nonconvex but the constraints are convex, our proposed method has an $\mathcal{O}(1/\varepsilon^2)$ complexity to compute an approximate stationary point. If both the objective and constraint functions are nonconvex, our proposed method has an $\mathcal{O}(1/\varepsilon^2)$ complexity to compute an approximate Fritz-John point.

Contents

Chapter 1

Preliminaries

In this dissertation, we focus on first-order methods for solving constrained and unconstrained smooth nonlinear optimization problems. In the most general form, all our problems of interest can be represented as follows:

$$\begin{aligned} &\min_{x \in X} f(x), \\ \text{s.t. } &g_i(x) \leq 0, \ i = 1, \ldots, m_g, \\ &h_i(x) = 0, \ i = 1, \ldots, m_h, \end{aligned}$$

where $X \subseteq \mathbb{R}^n$ is a closed convex set and f, g and h are continuously differentiable functions. In Chapter 2, we study a special case of the above problem in which the objective function $f(x) := (1/2)\|x - u\|_2^2$ for some u, $g_i \equiv 0$ for all i, and h_i's are affine constraint functions. In Chapter 3, the objective function f is convex, the set $X = \mathbb{R}^n$ and $g_i, h_i \equiv 0$ for all i. In Chapter 4, we only study functional inequality constraints, i.e., $h_i \equiv 0$ for all i. For all problems in this dissertation, we assume that the problem dimension n is high and first-order algorithms are preferred. Second and higher order algorithms (e.g., Newton methods) are out of the scope of this dissertation.

In this chapter, we describe some preliminary knowledge for first-order methods and nonlinear optimization. Such knowledge will be frequently used in the research projects described in the following chapters. Specifically, in Section 1.1 we describe some properties of smooth functions and convex smooth functions. In Section 1.2 we describe two fundamental methods commonly studied in first-order algorithm literature: the gradient descent and accelerated gradient descent methods.

1.1 Properties of smooth functions

In this section, we describe several commonly known properties of smooth functions and convex smooth functions. Some properties are utilized in many convergence analysis performed throughout this dissertation. The proofs of all the results below are commonly known in convex and nonlinear optimization textbooks (see, e.g., [24, 11]) and are skipped.

We say that a function $f : \mathbb{R}^n \to \mathbb{R}$ is L_f-smooth with respect to norm $\|\cdot\|$ if it is continuously differentiable and its gradient ∇f is Lipschitz continuous with constant L_f, namely,

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L_f \|x - y\|, \ \forall x, y \in \mathbb{R}^n.$$

Here $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. When the context is clear, we may omit the description of the associated norm and simply state that f is L_f-smooth. For smooth functions, we have the following commonly used property:

Lemma 1.1.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an arbitrary L_f-smooth function with respect to norm $\|\cdot\|$. For all $x, y \in \mathbb{R}^n$, we have*

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

The above lemma shows that $f(x)$ is upper bounded by the sum of a linear approximation at y and the squared distance $\|x - y\|^2$. Note that a stronger result $|f(x) - f(y) - \langle \nabla f(y), x - y \rangle| \leq (L_f/2)\|x - y\|^2$ with an absolute value at the left-hand side also holds; however, the weaker version stated in the above lemma is already sufficient for all the analysis throughout our dissertation.

In addition to smoothness, if f is also convex, we can obtain several more properties. We use notation $\mathcal{F}^{1,1}_{L_f,\|\cdot\|}(\mathbb{R}^n)$ to denote the set of functions $f : \mathbb{R}^n \to \mathbb{R}$ that are both L_f-smooth (associated with norm $\|\cdot\|$) and convex.[1] When the associated norm is Euclidean, we will use a simpler notation $\mathcal{F}^{1,1}_{L_f}(\mathbb{R}^n)$. In the remainder of this section, the properties of convex smooth functions are described. Such properties will be utilized in the convergence analysis throughout this dissertation.

[1] The notation $\mathcal{F}^{1,1}_{L_f,\|\cdot\|}(\mathbb{R}^n)$ follows from the book [24]. Such functions are also known as convex smooth functions. Here $\mathcal{F}$ stands for the set of convex functions; the superscript "$1, 1$" denotes that f is continuously differentiable and its gradient is Lipschitz continuous; the subscript $L_f, \|\cdot\|$ denotes that the Lipschitz constant is L_f with respect to the norm $\|\cdot\|$.

Lemma 1.1.2. *For any function $f \in \mathcal{F}^{1,1}_{L_f,\|\cdot\|}(\mathbb{R}^n)$ and any $x, y \in \mathbb{R}^n$, we have*

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

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

$$\frac{1}{L_f}\|\nabla f(x)\|_*^2 \leq \langle \nabla f(x), x - x^*\rangle. \tag{1.1}$$

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 scaled squared gradient norm difference $\|\nabla f(x) - \nabla f(y)\|_*^2/L_f$. Note that the special case (1.1) 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 (1.1) states that for convex smooth functions, its optimality condition can be stronger than the above equation with an extra gradient norm.

Lemma 1.1.3. *For any function $f \in \mathcal{F}^{1,1}_{L_f,\|\cdot\|}(\mathbb{R}^n)$ and any $x, y \in \mathbb{R}^n$, we have*

$$0 \leq f(x) - f(y) - \langle \nabla f(y), x - y\rangle \leq \frac{L_f}{2}\|x - y\|^2. \tag{1.2}$$

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

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

The relation (1.2) is stronger than the result described in Lemma 1.1.1 since the difference between $f(x)$ and its linear approximation at y is now lower bounded by 0 due to convexity. Note that for such lower bound to hold, we only require that f is convex and differentiable. Indeed, for convex smooth functions such lower bound can be further strengthened, as stated in the lemma

below.

Lemma 1.1.4. *For any function $f \in \mathcal{F}^{1,1}_{L_f,\|\cdot\|}(\mathbb{R}^n)$ and any $x, y \in \mathbb{R}^n$, we have*

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

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

$$\frac{1}{2L_f} \|\nabla f(x)\|_*^2 \leq f(x) - f(x^*).$$

Lemmas 1.1.2, 1.1.3 and 1.1.4 are the fundamental properties of the function class $\mathcal{F}^{1,1}_{L_f}(\mathbb{R}^n)$ that play important roles in the analysis of convex smooth optimization problems. Throughout this dissertation we rely on them to perform convergence analysis of several first-order methods.

With the essential properties of smooth functions and convex smooth functions introduced, in the following section we introduce two commonly seen first-order methods that are related to this dissertation.

1.2 Acceleration of the gradient descent method

In this section, we introduce two commonly seen first-order methods for the following convex smooth optimization problem:

$$\min_{x \in X} f(x). \tag{1.3}$$

Here $X \subseteq \mathbb{R}^n$ is a closed convex set and $f \in \mathcal{F}^{1,1}_{L_f,\|\cdot\|}(\mathbb{R}^n)$. Our goal is to compute an approximate solution with accuracy threshold ε. Note that there are several different possible definitions of ε-approximate solutions. For instance, we will define an ε-approximate solution x as one that satisfies $\|x - x^*\| \leq \varepsilon$ in Chapter 2 or as one that satisfies $\|\nabla f(x)\|_*^2 \leq \varepsilon$ for gradient norm minimization in Chapter 3. In this subsection, our goal is to compute an approximate solution x such that $f(x) - f(x^*) \leq \varepsilon$, where x^* is an optimal solution to problem (1.3).

There have been several possible methods for solving problem (1.3). Our focus is on first-order methods; such methods are commonly used when the accuracy threshold ε is modest 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 favorable.

1.2.1 The gradient descent method

The most commonly used first-order method for solving problem (1.3) 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 point. We describe the gradient descent method and analyze its convergence performance in this subsection. For simplicity, we let the feasible set X be $\mathbb{R}^n$ and assume that f is L_f-smooth with respect to the Euclidean norm. The gradient descent algorithm is described below.

Algorithm 1.2.1 The gradient descent method

Require: Initial point $x_0 \in \mathbb{R}^n$, stepsize $h > 0$
for $t = 0, 1, \ldots, N$ **do**

$$x_{t+1} = x_t - h\nabla f(x_t). \tag{1.4}$$

end for

For $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$, we can derive the convergence result of the gradient method in terms of function value difference $f(x_N) - f(x^*)$. The derivation is based on the analysis of the relationship between $f(x_t) - x^*$ and $f(x_{t-1}) - x^*$, as detailed in the proof of the following proposition.

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

$$f(x_N) - f(x^*) \leq \frac{2(f(x_0) - f(x^*))\|x_0 - x^*\|^2}{2\|x_0 - x^*\|^2 + Nh(2 - L_f h)(f(x_0) - f(x^*))}. \tag{1.5}$$

Proof. First, by the definition of x_t (1.4) in the description of Algorithm 1.2.1, the relationship (1.1)

in Lemma 1.1.2, and noting that $\nabla f(x^*) = 0$, we have

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

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

Next, by Lemma 1.1.3 and the description of x_t (1.4) in the gradient descent algorithm, we have

$$\begin{aligned}f(x_{t+1}) \leq& f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t\rangle + \frac{L_f}{2}\|x_{t+1} - x_t\|^2 \\ =& f(x_t) - h\left(1 - \frac{L_f}{2}h\right)\|\nabla f(x_t)\|^2.\end{aligned} \tag{1.6}$$

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

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

Thus by (1.6), we obtain

$$f(x_{t+1}) \leq f(x_t) - h\left(1 - \frac{L_f}{2}h\right)\frac{(f(x_t) - f(x^*))^2}{\|x_0 - x^*\|^2},$$

i.e.

$$f(x_{t+1}) - f(x^*) \leq f(x_t) - f(x^*) - h\left(1 - \frac{L_f}{2}h\right)\frac{(f(x_t) - f(x^*))^2}{\|x_0 - x^*\|^2}.$$

Dividing $(f(x_t) - f(x^*))\,(f(x_{t+1}) - f(x^*))$ on both sides (note that the proof is trivial when $f(x_t) =$

$f(x^*)$ or $f(x_{t+1}) = f(x^*)$), we have

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

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

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

The above result implies (1.5) immediately. □

In the convergence property above, the right-hand side of the result in (1.5) is dependent on the stepsize h. Theoretically, the best choice of stepsize h is the one such that $h(1 - (L_f/2)h)$ in the denominator of the right-hand side is maximized, i.e., when $h = 1/L_f$. The convergence result when $h = 1/L_f$ is described in the theorem below.

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

$$f(x_N) - f(x^*) \leq \frac{2L_f\|x_0 - x^*\|^2}{N + 4}. \tag{1.7}$$

Proof. The convergence result in (1.7) follows directly from Proposition 1.2.1 (with stepsize $h = 1/L_f$). □

Remark from Theorem 1.2.1 that the gradient evaluation complexity of the gradient descent method is of order $\mathcal{O}(1/\varepsilon)$. In the following section, we show that the gradient evaluation complexity can be improved to $\mathcal{O}(1/\sqrt{\varepsilon})$ through the accelerated gradient descent method. The following section can be viewed as the acceleration technique for the classical gradient descent method.

1.2.2 Accelerated gradient descent method

In this subsection, we introduce the accelerated gradient descent method for solving the optimization problem (1.3). Unlike the previous section on gradient descent method, here we allow

general closed convex feasible set X and any norm $\|\cdot\|$. The first version of accelerated gradient descent method appears in [23]. After the work in [23], there have been several extensions and modifications proposed in the literature (see, e.g., [22, 24, 11] and the references within). It has already been shown that the accelerated gradient descent method is an optimal first-order method for minimizing function value difference in convex smooth optimization (see, e.g., [24, 20]). The concept of acceleration in this method is essential to several convergence analysis performed throughout this dissertation. In Subsection 2.2.2 of Chapter 2, we will apply the accelerated gradient descent method to obtain an approximate solution minimizing $\|x - x^*\|$ with $\mathcal{O}(\log(1/\varepsilon))$ matrix-vector multiplication complexity. When solving the problem of interest in Chapter 3, the methods we proposed can be considered as the dual version of accelerated gradient descent method.

Our description of the accelerated gradient method in Algorithm 1.2.2 and the analysis are based on [11]. Here we use a general prox function setting associated with arbitrary norm $\|\cdot\|$. Specifically, the prox function $V : X^o \times X \to \mathbb{R}_+$ is a proximity measure defined by

$$V(x, y) = v(y) - v(x) - \langle \nabla v(x), y - x \rangle. \tag{1.8}$$

The set X^o is defined as $X^o = \{x \in X : \text{ there exist } p \in \mathbb{R}^n \text{ such that } x \in \operatorname{argmin}_{u \in X}[p^\top u + v(u)]\}$, and the distance generating function $v : X \to \mathbb{R}$ is continuously differentiable and strongly convex with respect to the general norm $\|\cdot\|$. The accelerated gradient method is described in Algorithm 1.2.2.

Algorithm 1.2.2 Accelerated gradient descent method

Require: Initial point $x_0 \in X$, $q_t \in [0, 1]$, $\gamma_t \geq 0$, $\alpha_t \in [0, 1]$
  Set $\overline{x}_0 = x_0$.
  for $t = 1, \ldots, N$ **do**
    Compute

$$\begin{aligned} \underline{x}_t &= (1 - q_t)\overline{x}_{t-1} + q_t x_{t-1}, \\ x_t &= \operatorname*{argmin}_{x \in X}\{\gamma_t \langle \nabla f(\underline{x}_t), x \rangle + V(x_{t-1}, x)\}, \\ \overline{x}_t &= (1 - \alpha_t)\overline{x}_{t-1} + \alpha_t x_t. \end{aligned}$$

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

Here, $\{(\underline{x}_t, x_t, \overline{x}_t)\}_{t=0}^N$ are the iterates generated by the accelerated gradient descent method. Specifically, the notation $\underline{x}_t$ denotes the iterates at which gradients of f are computed. Here the

underline notation is since any gradient evaluation $\nabla f(\underline{x}_t)$ provides a linear approximation lower bound $f(\underline{x}_t) + \langle \nabla f(\underline{x}_t), x - \underline{x}_t \rangle$ of the function $f(x)$. The notation x_t denotes the iterates at which we perform gradient-descent-like updates. The notation $\overline{x}_t$ denotes the outputs of the approximate solutions of the algorithm. Here the overline is since $f(\overline{x}_t) \geq f(x^*)$ is an overestimate of the optimal objective function value. We can immediately observe that if we have the feasible set $X = \mathbb{R}^n$, the prox function $V(x, y) = (1/2)\|x - y\|_2^2$ be the half squared Euclidean norm, and the parameters $\alpha_t = 1$ and $\gamma_t = h$ for $t = 0, \ldots, N-1$, the accelerated gradient descent method in Algorithm 1.2.2 is identical to the gradient descent method in Algorithm 1.2.1. In the following proposition, we prove that the certain choices of algorithm parameters q_t, α_t and γ_t can lead us to a recursive relationship between the t-th and $(t-1)$-th iterates of Algorithm 1.2.2.

Proposition 1.2.2. *Let f be a function such that $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$ and $\{(\underline{x}_t, x_t, \overline{x}_t)\}_{t=0}^N$ be the iterates generated by accelerated gradient descent method in Algorithm 1.2.2 to minimize f. If parameters q_t, α_t and γ_t satisfy the following relationships:*

$$\alpha_t \geq q_t, \tag{1.9}$$

$$\frac{L_f(\alpha_t - q_t)}{1 - q_t} \leq 0, \tag{1.10}$$

$$\frac{L_f q_t(1 - \alpha_t)}{1 - q_t} \leq \frac{1}{\gamma_t}, \quad t = 1, ..., N, \tag{1.11}$$

then for any $x \in X$, we have

$$f(\overline{x}_t) - f(x) + \frac{\alpha_t}{\gamma_t} V(x_t, x) \leq (1 - \alpha_t)[f(\overline{x}_{t-1}) - f(x)] + \frac{\alpha_t}{\gamma_t} V(x_{t-1}, x). \tag{1.12}$$

Proof. First, by the definitions of $\overline{x}_t$, x_t and $\underline{x}_t$ in Algorithm 1.2.2, we have

$$\begin{aligned}
\overline{x}_t - \underline{x}_t =& (q_t - \alpha_t)\overline{x}_{t-1} + \alpha_t x_t - q_t x_{t-1} \\
=& \alpha_t \left[x_t - \frac{\alpha_t - q_t}{\alpha_t(1 - q_t)} \underline{x}_t - \frac{q_t(1 - \alpha_t)}{\alpha_t(1 - q_t)} x_{t-1} \right] \\
=& \alpha_t \left[(\frac{\alpha_t - q_t}{\alpha_t(1 - q_t)} + \frac{q_t(1 - \alpha_t)}{\alpha_t(1 - q_t)}) x_t - \frac{\alpha_t - q_t}{\alpha_t(1 - q_t)} \underline{x}_t - \frac{q_t(1 - \alpha_t)}{\alpha_t(1 - q_t)} x_{t-1} \right] \\
=& \alpha_t \left[\frac{\alpha_t - q_t}{\alpha_t(1 - q_t)} (x_t - \underline{x}_t) + \frac{q_t(1 - \alpha_t)}{\alpha_t(1 - q_t)} (x_t - x_{t-1}) \right].
\end{aligned}$$

Thus by the relationship $\alpha_t \geq q_t$ of α_t and q_t in (1.9) and the convexity of norms, we obtain

$$\|\overline{x}_t - \underline{x}_t\|^2 \leq \alpha_t \left[\frac{\alpha_t - q_t}{1 - q_t} \|x_t - \underline{x}_t\|^2 + \frac{q_t(1-\alpha_t)}{1-q_t} \|x_t - x_{t-1}\|^2 \right]. \tag{1.13}$$

Next, by property (1.2) in Lemma 1.1.3, the relationship $\overline{x}_t = (1-\alpha_t)\overline{x}_{t-1} + \alpha_t x_t$ described in Algorithm 1.2.2, the convexity of f, and inequalities (1.13), (1.10) and (1.11) above, we are able to derive that

$$\begin{aligned}
f(\overline{x}_t) \leq & f(\underline{x}_t) + \langle \nabla f(\underline{x}_t), \overline{x}_t - \underline{x}_t \rangle + \frac{L_f}{2} \|\overline{x}_t - \underline{x}_t\|^2 \\
= & (1-\alpha_t) \left[f(\underline{x}_t) + \langle \nabla f(\underline{x}_t), \overline{x}_{t-1} - \underline{x}_t \rangle \right] + \alpha_t \left[f(\underline{x}_t) + \langle \nabla f(\underline{x}_t), x_t - \underline{x}_t \rangle \right] + \frac{L_f}{2} \|\overline{x}_t - \underline{x}_t\|^2 \\
\leq & (1-\alpha_t) f(\overline{x}_{t-1}) \\
& + \alpha_t \left[f(\underline{x}_t) + \langle \nabla f(\underline{x}_t), x_t - \underline{x}_t \rangle + \frac{L_f(\alpha_t - q_t)}{2(1-q_t)} \|x_t - \underline{x}_t\|^2 + \frac{L_f q_t (1-\alpha_t)}{2(1-q_t)} \|x_t - x_{t-1}\|^2 \right] \\
\leq & (1-\alpha_t) f(\overline{x}_{t-1}) + \alpha_t \left[f(\underline{x}_t) + \langle \nabla f(\underline{x}_t), x_t - \underline{x}_t \rangle + \frac{1}{\gamma_t} V(x_t, x_{t-1})^2 \right].
\end{aligned}$$

Finally, by the optimality condition of x_t in its definition in Algorithm 1.2.2, we have that for all $x \in X$,

$$\gamma_t \langle \nabla f(\underline{x}_t), x_t \rangle + V(x_{t-1}, x_t) \leq \gamma_t \langle \nabla f(\underline{x}_t), x \rangle + V(x_{t-1}, x).$$

Hence combining with the fact that $V(x_{t-1}, x_t) \geq V(x_{t-1}, x) - V(x_t, x)$ and the convexity of f, we conclude that

$$\begin{aligned}
f(\overline{x}_t) \leq & (1-\alpha_t) f(\overline{x}_{t-1}) + \alpha_t \left[f(\underline{x}_t) + \langle \nabla f(\underline{x}_t), x - \underline{x}_t \rangle \right] + \frac{\alpha_t}{\gamma_t} V(x_{t-1}, x) - \frac{\alpha_t}{\gamma_t} V(x_t, x) \\
\leq & (1-\alpha_t) f(\overline{x}_{t-1}) + \alpha_t f(x) + \frac{\alpha_t}{\gamma_t} V(x_{t-1}, x) - \frac{\alpha_t}{\gamma_t} V(x_t, x).
\end{aligned}$$

□

In the above proposition, we prove that when the parameters satisfy (1.9), (1.10) and (1.11), we can obtain a recursive relationship between the t-th and $(t-1)$-th iterates of Nesterov's gradient method (1.12). 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 1.2.3. *Let f be a function such that $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$ and $\{(\underline{x}_t, x_t, \overline{x}_t)\}_{t=0}^N$ be the iterates generated by accelerated gradient descent method to minimize f. If $\alpha_t = q_t$, $L_f\alpha_t \leq 1/\gamma_t$ and $\gamma_t(1-\alpha_t)/\alpha_t \leq \gamma_{t-1}/\alpha_{t-1}$ for all $t = 1, ..., N$, then it holds that*

$$f(\overline{x}_N) - f(x^*) + \frac{\alpha_N}{\gamma_N} V(x_N, x^*) \leq \frac{\alpha_N(1-\alpha_1)\gamma_1}{\gamma_N \alpha_1}[f(\overline{x}_0) - f(x^*)] + \frac{\alpha_N}{\gamma_N} V(x_0, x^*). \tag{1.14}$$

Proof. It is straightforward to verify that assumptions (1.9)–(1.11) holds and hence we can use Proposition 1.2.2 to conclude that

$$f(\overline{x}_t) - f(x^*) + \frac{\alpha_t}{\gamma_t} V(x_t, x^*) \leq (1-\alpha_t)[f(\overline{x}_{t-1}) - f(x^*)] + \frac{\alpha_t}{\gamma_t} V(x_{t-1}, x^*).$$

Applying relationship $\gamma_t(1-\alpha_t)/\alpha_t \leq \gamma_{t-1}/\alpha_{t-1}$ to the above result, we have

$$\begin{aligned} \frac{\gamma_t}{\alpha_t}[f(\overline{x}_t) - f(x^*)] + V(x_t, x^*) \leq & \frac{(1-\alpha_t)\gamma_t}{\alpha_t}[f(\overline{x}_{t-1}) - f(x^*)] + V(x_{t-1}, x^*) \\ \leq & \frac{\gamma_{t-1}}{\alpha_{t-1}}[f(\overline{x}_{t-1}) - f(x^*)] + V(x_{t-1}, x^*). \end{aligned}$$

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

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

and conclude (1.14). □

With help from the above result, we are now ready to analyze the convergence properties of Algorithm 1.2.2.

Theorem 1.2.2. *Suppose that f is a function that belongs to function class $\mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$, and that $\{(\underline{x}_t, x_t, \overline{x}_t)\}_{t=0}^N$ are the iterates generated by accelerated gradient descent method with parameters $\alpha_t = q_t = 2/(t+1)$, $\gamma_t = t/(2L_f)$. Then we have*

$$f(\overline{x}_N) - f(x^*) \leq \frac{4L_f}{N(N+1)} V(x_0, x^*).$$

Proof. By the choices of α_t, q_t, and γ_t, we have that $\alpha_t/\gamma_t = 4L_f/(t(t+1))$ and it is easy to verify that the assumptions of Proposition 1.2.3 hold. Thus by Proposition 1.2.3, we obtain

$$f(\overline{x}_N) - f(x^*) \leq \frac{4L_f}{N(N+1)}(V(x_0, x^*) - V(x_N, x^*)) \leq \frac{4L_f}{N(N+1)}V(x_0, x^*).$$

□

Remark from Theorem 1.2.1 that the gradient evaluation complexity of accelerated gradient descent method for computing an approximate solution of problem (1.3) is $\mathcal{O}(1/\sqrt{\varepsilon})$, which is better than that of the gradient descent method and is optimal for function value difference minimization (see, e.g., the lower complexity bound analysis in [20, 24]). We will incorporate such acceleration strategy in our algorithm design later in Chapter 2. However, note that the gradient evaluation is performed at a different point than the output approximate solution in each iteration of the accelerated gradient descent method, which is less convenient for gradient norm minimization. We will elaborate on algorithm design and complexity analysis for gradient norm minimization later in Chapter 3.

Chapter 2

Exact matrix-vector multiplication complexity for kernel projection and its application on distributed consensus optimization

Motivated by the consensus problem in distributed optimization, in this chapter we study the problem of computing a matrix kernel projection of a vector, when the matrix is not known, but its matrix-vector oracle is accessible. We first study this problem from three perspectives: the control perspective, the optimization perspective, and the linear algebra perspective. From the control perspective, we find that a previous method in control literature converges slower than it is claimed, as its best linear convergence rate has an additional logarithmic term in the worst case. From the optimization perspective, we derive a novel first-order method based on the accelerated gradient method, which has a linear convergence rate that is optimal in the order. From the linear algebra perspective, there has been known study of exact oracle complexity for solving a linear equation provided with the same matrix-vector oracle, but such result is not readily applicable to our problem of interest. Here we use the term "exact complexity" to emphasize that not only the complexity is optimal in order, but also the constant appearing in the complexity bound is unimprovable. Based

on our observations from the three perspectives, under a linear span assumption, we propose a novel iterative method which attains the exact oracle complexity for our problem of interest. In the realm of general methods, we provide an exact lower complexity bound under the assumption that the dimension of our problem is sufficiently large.

2.1 Introduction

The problem of interest in this chapter is the following kernel projection problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x-u\|^2 \text{ subject to } Ax = 0. \tag{2.1}$$

Here we assume that A is symmetric and positive semi-definite and our goal is to compute an Euclidean projection of vector u to the kernel space of A, denoted by $\ker A$. The problem above clearly has a unique optimum, which we denote as x^*, the orthogonal projection of u onto $\ker A$. Our goal is to compute an ε-solution x such that $\|x-x^*\| \leq \varepsilon$. The above problem is motivated by the distributed consensus problem widely studied in optimization and control literature. Specifically, we have m-agents with their respective $u^{(i)}$'s hoping to reach a consensus by solving problem

$$\min_{y\in\mathbb{R}^d} \frac{1}{2}\sum_{i=1}^{m} \|y-u^{(i)}\|^2. \tag{2.2}$$

To solve the above problem, the agents are only able to communicate with each other through a communication network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. At each time-step, an agent is only able to communicate with its neighboring agents. In order to solve problem (2.2), it is a common approach to reformulate it to

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x-u\|^2 \text{ subject to } (W\otimes I_d)x = x, \tag{2.3}$$

where $n = md$, $u = ((u^{(1)})^\top, \ldots, (u^{(m)})^\top)^\top \in \mathbb{R}^n$, I_d is the $d\times d$ identity matrix, $\otimes$ denotes the Kronecker product, and $W \in \mathbb{R}^m \times \mathbb{R}^m$ is a symmetric positive semi-definite matrix that is determined by the network topology. The assumption on W is that its largest eigenvalue is 1 with multiplicity 1 and associated eigenvector $1_m := (1,\ldots,1)^\top \in \mathbb{R}^m$. With such assumption of W, if we denote $x = ((x^{(1)})^\top, \ldots, (x^{(m)})^\top) \in \mathbb{R}^n$ where $x^{(i)} \in \mathbb{R}^d$ for all i, then the constraint

$(W \otimes I_d)x = x$ enforces that all components $x^{(i)}$'s are the same, i.e., that the agents have consensus $x^{(1)} = \ldots = x^{(m)}$. Indeed, any symmetric positive semi-definite W that satisfies the aforementioned eigenvalue and eigenvector assumption can be used for enforcing the consensus constraint; any W that satisfies such assumptions are usually referred to as a *mixing matrix*. For some examples of mixing matrices, see, e.g., [27]. The formulation (2.3) is clearly a special case of our problem of interest (2.1) with $A := I_n - W \otimes I_d$.

2.1.1 Notations

We denote the kernel and image space for a matrix $A \in \mathbb{R}^{n\times n}$ as $\ker A \coloneqq \{x \in \mathbb{R}^n | Ax = 0\}$ and $\operatorname{im} A \coloneqq \{y \in \mathbb{R}^n | y = Ax,\, x \in \mathbb{R}^n\}$, respectively. For a symmetric matrix A with eigenvalue decomposition $A = U\Lambda U^\top$ where U is an orthogonal matrix and Λ is a diagonal matrix, we denote $|A| \coloneqq U|\Lambda|U^\top$, where $|\Lambda|$ is a diagonal matrix whose entries are absolute values of that of Λ. If A is in addition positive semi-definite A, we denote $A^\alpha \coloneqq U\Lambda^\alpha U^\top$, where the power function $(\cdot)^\alpha$ acts element-wisely on the entries of Λ. We assume that the largest and smallest nonzero eigenvalues of A are ℓ and σ, respectively. We use $\|\cdot\|$ to denote the Euclidean norm on vectors or the spectral norm for matrices. For any positive integer n, we use $I_n \in \mathbb{R}^{n\times n}$ to denote the identity matrix. The i-th standard basis vector is denoted by $e_i := (0, \ldots, 0, 1, 0, \ldots, 0)^\top$.

2.2 Three perspectives associated with the problem

In this section, we provide three perspectives on studying the kernel projection problem (2.1). First, we study the problem from the control perspective, following the result on linear iterations for distributed averaging [16]. Second, we study the problem from the optimization perspective by reformulating it as a least squares problem and solving it using a first-order method in [24]. Third, we study the problem from the linear algebra perspective, following the exact complexity analysis for solving linear systems in [21]. We will point out the possibility for small improvements in each perspective. With the small improvements, we will show in the next sections that by modifying and associating the results in the three perspectives together, we are able to develop the exact matrix-vector complexity for problem (2.1) and obtain a new result on distributed consensus optimization.

2.2.1 The control perspective

In this subsection, we discuss our problem of interest (2.1) from the control perspective, following the analysis on distributed consensus optimization in [16]. The goal of [16] is to design a weighted averaging type iterative algorithm for solving the consensus problem (2.2). In each iteration, it is assumed that each agent will use the mixing matrix W to update its local variable $x^{(i)}$. For example, the simpliest consensus update scheme is $x_{t+1}^{(i)} = W x_t^{(i)}$, where $x_{t+1}^{(i)}$ and $x_t^{(i)}$ are the local variable $x^{(i)}$ in iterations $t+1$ and t, respectively. The authors in [16] propose to study the weighted averaging type algorithm of form $x_{t+1}^{(i)} = (g+1)W x_t^{(i)} - g x_{t-1}^{(i)}$ and study the best weights $g \in (-1, 1)$ for achieving the fastest convergence to consensus. Note that using our notation of $A = I_n - W \otimes I_d$, such weighted averaging type algorithm becomes

$$x_{t+1} = (g+1)(I_n - A)x_t - g x_{t-1}, \quad t \geq 1. \tag{2.4}$$

The core concept behind the analysis of [16] is to study the fix point iteration properties of algorithm in (2.4) and the best choice of parameter g. In [16] the authors first proved that this algorithm converges to the optimal solution when $g \in (-1, 1)$:

Proposition 2.2.1. *When applying iteration* (2.4) *to solve problem* (2.2) *with initial iterates* $x_1 = x_0 = u$*, the iterates* x_t *of converges asymptotically to the optimal solution of problem* (2.2) *if and only if* $g \in (-1, 1)$*.*

Our focus throughout this subsection is to review the analysis of [16] on the convergence of algorithm in (2.4) and establish an explicit matrix-vector multiplication complexity.

Clearly the system (2.4) will give an $x^\star \in \ker A$ if it has a fixed point, e.g., $x^\star = (g+1)(I_n - A)x^\star - g x^\star$ and hence $Ax^\star = 0$.

In order to study the fix point iteration property of the algorithm in (2.4), the authors of [16] reformulate it as a linear system of form $z_{t+1} = B z_t$ and study its eigenvalue and eigenvector properties:

$$z_{t+1} \coloneqq \begin{bmatrix} x_{t+1} \\ x_t \end{bmatrix} = B z_t := \begin{bmatrix} (g+1)(I_n - A) & -g I_n \\ I_n & 0 \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix}. \tag{2.5}$$

Clearly, fixed points of the above system is $z^\star = \begin{bmatrix} x^\star \\ x^\star \end{bmatrix}$ where $x^\star$ is a fixed point of system (2.4). The matrices B and A are related in the following way:

Proposition 2.2.2. *The characteristic polynomials of B and A satisfy that $\det(\lambda I_{2n} - B) = \det[(\lambda^2 - g\lambda - \lambda + g)I_n + \lambda(g+1)A]$. Moreover, for each multiplicity of eigenvalue $\mu \in [0,1]$ of A, the complex pair λ_+ and λ_- of form*

$$\lambda_\pm = \frac{(g+1)(1-\mu) \pm \sqrt{-1}\sqrt{4g - (g+1)^2(1-\mu)^2}}{2}$$

contributes to one pair of eigenvalues of B. Specifically, $\lambda_\pm$ are solutions to the quadratic equation

$$-(\lambda^2 - g\lambda - \lambda + g) = \mu(g+1)\lambda \tag{2.6}$$

with respect to λ. As a consequence, the second largest absolute value of the eigenvalues of B is

$$\rho(g) = \begin{cases} \frac{(g+1)(1-\sigma)+\sqrt{(g+1)^2(1-\sigma)^2-4g}}{2} & \text{if } -1 < g \le g^\star \\ \sqrt{g} & \text{if } g^\star \le g < 1. \end{cases}$$

Here $g^\star := \left(\frac{1-\sqrt{2\sigma-\sigma^2}}{1-\sigma}\right)^2$.

Proof. We study the relationship between the eigenspaces of B and A by relating the matrix polynomial below,

$$\begin{aligned} \det(\lambda I_{2n} - B) &= \det \begin{bmatrix} \lambda I_n - (g+1)(I_n - A) & gI_n \\ -I_n & \lambda I_n \end{bmatrix} \\ &= \det(\lambda I_n)\det[\lambda I_n - (g+1)(I_n - A) - gI_n(\lambda I_n)^{-1}(-I_n)] \\ &= \det(\lambda I_n)\det[(\lambda - g - 1 + g/\lambda)I_n + (g+1)A] \\ &= \lambda^n \det[(\lambda - g - 1 + g/\lambda)I_n + (g+1)A] \\ &= \det[(\lambda^2 - g\lambda - \lambda + g)I_n + \lambda(g+1)A]. \end{aligned}$$

Here the second equality follows from the Schur complement of the lower right block, and the rest of the equalities follow from properties of matrix determinant. Notice that the behavior of system

(2.5) relies completely on the spectral radius of B, or more precisely, the maximum absolute value of eigenvalues of B which are not 1, and the corresponding eigenspace.

Clearly, for any eigenvalue $\mu \in [0,1]$ of A, the solutions

$$\lambda = \frac{(g+1)(1-\mu) \pm \sqrt{(g+1)^2(1-\mu)^2 - 4g}}{2} = \frac{(g+1)(1-\mu) \pm \sqrt{-1}\sqrt{4g-(g+1)^2(1-\mu)^2}}{2}$$

of quadratic equation $-\frac{\lambda^2 - g\lambda - \lambda + g}{\lambda(g+1)} = \mu$ with respect to (w.r.t.) λ give a pair of eigenvalues of B; e.g., when $\mu = 0$, we have $\lambda \in \{1, g\}$, where $\lambda = 1$ acts as an identity map on its eigenspace of system (2.5), and $\lambda = g$ acts as a contraction on its eigenspace of system (2.5). Due to the fact that these solutions are both positive or both complex, and hence the monotonicity of $|\lambda|$ w.r.t. μ, the maximum absolute value of eigenvalues of B which do not equal 1 is

$$\begin{aligned}\rho(g) &= \begin{cases} \frac{(g+1)(1-\sigma)+\sqrt{(g+1)^2(1-\sigma)^2-4g}}{2} & \text{if } (g+1)^2(1-\sigma)^2 \geq 4g \\ \sqrt{g} & \text{if } \sigma \geq 1 - \frac{2\sqrt{g}}{1+g} \end{cases} \\ &= \begin{cases} \frac{(g+1)(1-\sigma)+\sqrt{(g+1)^2(1-\sigma)^2-4g}}{2} & \text{if } -1 < g \leq g^\star \\ \sqrt{g} & \text{if } g^\star \leq g < 1 \end{cases}\end{aligned}$$

because $|\lambda|$ is monotonic w.r.t. μ. □

The above proposition is a more detailed description of the results in Lemmas 1, 4, 5 and Proposition 3 in [16]. Clearly, the value of $\rho(g)$ is the smallest with $\rho(g^\star) = \sqrt{g^\star} = \frac{1-\sqrt{2\sigma-\sigma^2}}{1-\sigma}$ when $g = g^\star$. Based on the observation of the g value that yields the smallest $\rho(g)$, it is concluded in [16] that such a choice of g is the best choice for the consensus algorithm in (2.4).

The convergence rate of algorithm in (2.4) is not discussed in [16]. Below we extend their results by providing the convergence in terms of $\|x_{t+1} - x^\star\|$:

Theorem 2.2.1. *The algorithm in* (2.4) *satiesfies that*

$$\|x_{t+1} - x^\star\| = \begin{cases} t\rho(g)^t O(1)\|u - x^\star\| & \text{if } g = g^\star \\ \rho(g)^t O(1)\|u - x^\star\| & \text{if } g \neq g^\star,\ -1 < g < 1. \end{cases}$$

Proof. It is important to note that, even if A is symmetric and admits an orthogonal diagonalization,

it is not the case for B. In fact, Liu and Morse showed Lemma 1 in [16] that, if (2.6) holds, (μ, a) is an eigenvalue-eigenvector pair of A if and only if $(\lambda, [a^\top, a^\top/\lambda]^\top)$ is an eigenvalue-eigenvector pair of B.

Indeed, compute

$$(\lambda I_{2n} - B)\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \lambda I_n - (g+1)(I_n - A) & \lambda I_n \\ -I_n & \lambda I_n \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = 0$$

we must have that $a = \lambda b$ from the second row, and substitute into the first row and obtain $[(\lambda - g - 1 + g/\lambda)I_n + (g+1)A]a = 0$. Note that (2.6) holds, the above equation becomes $(g+1)(-\mu I_n + A)a = 0$, i.e., $(\mu I_n - A)a = 0$, (μ, a) must be an eigenvalue-eigenvector pair of A. Hence $\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ a/\lambda \end{bmatrix}$ is the only eigenvector which associates to (μ, a) of A. This means (i), if (2.6) has no repeated solution for all eigenvalues μ of A, then B is diagonalizable; and (ii), if (2.6) has repeated solutions $\lambda_+ = \lambda_-$ for some eigenvalue μ of A, then B is NOT diagonalizable, as it has a 2×2 Jordan block of the form $\begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}$. It is also important to note that in the case $g = g^\star$, $\lambda_+ = \lambda_-$ give a pair of repeated root for each multiplicity of $\mu = \sigma$, which results in a Jordan block of the form $\begin{bmatrix} \sqrt{g^\star} & 1 \\ 0 & \sqrt{g^\star} \end{bmatrix}$.

Hence we have two kinds of vectors in the eigenspace of B for a pair of $\lambda_\pm$, (i) an eigenvector v such that $Bv = \lambda v$ for $\lambda \in \{\lambda_\pm\}$, or (ii) a vector w such that $(B - \lambda I_{2n})w = v$ for some eigenvector v which satisfies that $Bv = \lambda v$. Then for case (ii), clearly $B^t w = \lambda^t w + t\lambda^{t-1}v = \lambda^t w + t\lambda^{t-1}(Bw - \lambda w)$ for all $t \geq 1$; here it is expressed in terms of w for a simplified notation by omitting v.

Let us introduce the Jordan decomposition B in $B\hat{u} = VJV^{-1}\hat{u}$, where V is an invertible but not necessary orthogonal matrix, J is the block diagonal matrix which consists of either 1×1 or 2×2 Jordan blocks, and $\hat{u} \coloneqq \begin{bmatrix} u \\ u \end{bmatrix}$. Furthermore, denote $y \coloneqq V^{-1}\hat{u} \in \mathbb{R}^{2n}$, i.e., $\hat{u} = Vy = \sum_{j=1}^{2n} y_j v_j$,

where v_i is the i-th column of V. Hence for the algorithm in (2.4),

$$\begin{aligned}
z_{t+1} =& B^t z_1 = B^t \hat{u} = \sum_{j=1}^{2n} y_j B^t v_j \\
=& \sum_{j:v_j \text{ eigenvector}} y_j B^t v_j + \sum_{j:v_j \text{ non-eigenvector}} y_j B^t v_j \\
=& \sum_{j:v_j \text{ eigenvector}} y_j \lambda_j^t v_j + \sum_{j:v_j \text{ non-eigenvector}} y_j [\lambda_j^t v_j + t\lambda_j^{t-1}(Bv_j - \lambda_j v_j)] \\
=& \sum_{j=1}^{2n} y_j \lambda_j^t v_j + t \sum_{j:v_j \text{ non-eigenvector}} y_j \lambda_j^{t-1}(Bv_j - \lambda_j v_j)
\end{aligned}$$

The convergence of z_{t+1} relies on the spectral radius B, thus the limit $z^\star$ of the z-trajectory is exactly $y_j v_j$ for v_j being the eigenvector associated with $\lambda_j = 1$. By computing $z_{t+1} - z^\star$, all remaining eigenvalues have absolute values strictly less than 1, therefore

$$\|z_1 - z^\star\| = \left\| \sum_{j:\lambda_j \neq 1} y_j v_j \right\|$$

and, more generally,

$$\begin{aligned}
\|z_{t+1} - z^\star\| =& \left\| \sum_{j:\lambda_j \neq 1} y_j \lambda_j^t v_j + t \sum_{j:v_j \text{ non-eigenvector}} y_j \lambda_j^{t-1}(Bv_j - \lambda_j v_j) \right\| \\
\leq& \sum_{j:\lambda_j \neq 1} \|y_j \lambda_j^t v_j\| + t \sum_{j:v_j \text{ non-eigenvector}} \|y_j \lambda_j^{t-1}(Bv_j - \lambda_j v_j)\| \\
\leq& \rho(g)^t \sum_{j:\lambda_j \neq 1} \|y_j v_j\| + t\rho(g)^{t-1} \sum_{j:v_j \text{ non-eigenvector}} \|y_j (Bv_j - \lambda_j v_j)\| \\
=& t\rho(g)^t O(1) \|z_1 - z^\star\|.
\end{aligned}$$

Here the first inequality is triangle inequality for vector norm, the second inequality follows from the definition of $\rho(g)$. For the last equality, note that if B is diagonalizable for a particular choice of g, then the second summation involving non-eigenvectors will not exist, then the bound reduces to

$$\|z_{t+1} - z^\star\| = \rho(g)^t O(1) \|z_1 - z^\star\|.$$

Finally, by definition, $\|x_{t+1} - x^\star\| \leq \|z_{t+1} - z^\star\|$, and $\|z_1 - z^\star\| = \sqrt{2}\|u - x^\star\|$, we can translate the results in terms of x. □

From Theorem 2.2.1 we can observe that the number of matrix-vector multiplications in order to obtain an ε-solution is of order $\Omega(\sqrt{\frac{\ell}{\sigma}}\log(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}))$, which is optimal in order but converges slower with a double logarithm.

2.2.2 The optimization perspective

In this subsection, we study our problem of interest (2.1) from an optimization perspective. Specifically, we show that the Lagrangian dual problem of (2.1) can be formulated as a convex quadratic program and solved by gradient based methods, e.g., the accelerated gradient method [23] (see also [24, 11]). We show that linear convergence can be established by observing the strong convexity of the dual problem in $\operatorname{im} A$. Our analysis follows that of [11] and adopting the analysis in [24].

It should be noted that the kernel projection problem is equivalent to

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x-u\|^2 \text{ subject to } A^{1/2}x = 0, \tag{2.7}$$

The equivalence of problems (2.1) and (2.7) is straightforward by observing that $\ker A = \ker A^{1/2}$. The above problem (2.7) can also be equivalently formulated as the following saddle point problem with primal vector x and dual vector z:

$$\min_{x\in\mathbb{R}^n}\max_{z\in\mathbb{R}^n} \frac{1}{2}\|x-u\|^2 + \left\langle A^{1/2}x, z\right\rangle = \max_{z\in\mathbb{R}^n}\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x-u\|^2 + \left\langle A^{1/2}x, z\right\rangle. \tag{2.8}$$

The equivalence between the above reformulation and the original problem (2.1) is trivial: whenever $A^{1/2}x \neq 0$, the maximization problem associated with z will be unbounded.

Consider the equivalently formulated Lagrangian dual problem (2.8). The saddle point (x^*, z^*) of (2.8) satisfies $x^* - u + A^{1/2}z^* = 0$ and $A^{1/2}x^* = 0$. Therefore, to compute an optimal solution, we may simply maintain the relation $x = u - A^{1/2}z$ for any solution pair (x, z). Applying such relation to (2.8) and change the maximization with respect to z to minimization for convenience, we obtain the following dual problem:

$$\min_{z\in\mathbb{R}^n} \frac{1}{2}\|A^{1/2}z - u\|^2 - \frac{1}{2}\|u\|^2. \tag{2.9}$$

Note that as long as we obtain a dual optimal solution z^* for problem (2.9), then $x^* := u - A^{1/2}z^*$ is immediately an optimal solution to the kernel projection problem (2.7).

The dual problem (2.9) can be solved numerically through any gradient-based algorithm, e.g., accelerated gradient descent method [24]. Unfortunately, the gradient of the objective function in problem (2.9) at any z is $Az - A^{1/2}u$, in which the information of $A^{1/2}u$ may not be readily available to us. However, we will show that by implementing a non-Euclidean version of accelerated gradient descent method (see, e.g., [11]), we are able to compute approximate solution $\hat{x}_t$ to the kernel problem (2.7) without requiring any information on $A^{1/2}$. Specifically, introducing a change of variable vector $w := A^{1/2}z$ in problem (2.9), we can reformulate it equivalently as

$$\min_{w \in \mathrm{im}\, A^{1/2}} d(w) := \frac{1}{2}\|w - u\|^2 - \frac{1}{2}\|u\|^2. \tag{2.10}$$

We propose to use accelerated gradient descent method with prox function $V(w_1, w_2) := \frac{1}{2}\|A^{-1/2}(w_1 - w_2)\|^2$. Our algorithm is described in Algorithm 2.2.3.

Algorithm 2.2.3 Accelerated gradient descent method

Require: Initial point $W_0 \in \mathbb{R}^n$, $\gamma \in [0,1]$, $\beta \geq 0$, $q \geq 0$, $\xi \in [0,1]$
  Set $\hat{W}_0 = w_0$.
  for $t = 1, \ldots, N$ **do**
    Compute

$$\begin{aligned}
\underline{w}_t =& (1-\gamma)\hat{w}_{t-1} + \gamma w_{t-1}, \\
w_t =& \operatorname*{argmin}_{w \in \mathbb{R}^n} \left\{ \frac{1}{q}[\langle \nabla d(\underline{w}_t), w\rangle + \beta V(w_t, w)] + V(w_{t-1}, w) \right\}, \\
\hat{w}_t =& (1-\xi)\hat{w}_{t-1} + \xi w_t.
\end{aligned}$$

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

Here our prox-function $V : X^0 \times X \to \mathbb{R}_+$ is defined with a continuously differentiable function v that is strongly convex with respect to Euclidean norm on X^0. For example, if we choose $X = \mathbb{R}^n$, $X^0 = \mathrm{im}A$ and $v(X) = 1/2\|A^{-1/2}X\|^2$, then we derive a prox function $V(x,y) = 1/2\|A^{-1/2}(x-y)\|^2$.

The convergence properties of the accelerated gradient method is well-known in the literature; in this chapter we adopt the description in [11]. Specifically, by Theorem 3.7 in [11], we have the following result:

Proposition 2.2.3. *If function d is strongly convex with respect to the prox function V, and the parameters in Algorithm 2.2.3 are chosen to be $\beta = \sigma$, $\xi = \sqrt{\frac{\sigma}{L}}$, $\gamma = \frac{\xi}{1+\xi}$, $q = \frac{\sigma(1-\xi)}{\xi}$, then the approximate solution $\bar{x}_N$ satisfies*

$$d(\hat{w}_N) - d(w) \leq (1-\xi)^N \left[d(\hat{w}_0) - d(w) + \xi(\sigma + q)V(w_1, w) \right].$$

By definition of $d(w)$ in (2.10), if $\hat{w}_t - w \in \operatorname{im} A$, $d(\hat{w}_t) - d(w) = \frac{1}{2}\|\hat{w}_t - w\|^2$ for any $t = 0, ..., N$. Then for (2.10), Theorem 3.7 in [11] becomes the following proposition.

Proposition 2.2.4. *If $\underline{w}_t$, $\hat{w}_t$, w_t, $t = 0, ..., N$ are iterates of Algorithm 2.2.3, and $\beta = \sigma$, $\xi = \sqrt{\frac{\sigma}{L}}$, $\gamma = \frac{\xi}{1+\xi}$, $q = \frac{\sigma(1-\xi)}{\xi}$, then for any $w \in imA$, the approximate solution $\bar{x}_N$ satisfies*

$$\frac{1}{2}\|\hat{w}_N - w\|^2 + \sigma V(w_{N-1}, w) \leq (1-\xi)^N \left[\frac{1}{2}\|\hat{w}_0 - w\|^2 + \sigma V(w_1, w) \right].$$

Furthermore,

$$\frac{1}{2}\|\hat{w}_N - w\|^2 \leq (1-\xi)^N \left[\frac{1}{2}\|\hat{w}_0 - w\|^2 + \sigma V(w_1, w) \right]. \tag{2.11}$$

To apply the above result, it suffices to demonstrate that d is strongly convex with respect to the prox function V defined in (1.8). To prove this strong convex result, we need to make use of the following lemma.

We can see directly from the definition of $A^{1/2}$ that since $w = A^{1/2}z$, $w \in \operatorname{im} A^{1/2} = \operatorname{im} A$. The strong convexity of d proved in Lemma 2.2.2 follows immediately from Lemma 2.2.1:

Lemma 2.2.1. *For any $x \in \operatorname{im} A$, we have*

$$\sigma \left\| A^{-1/2}x \right\|^2 \leq \|x\|^2 \leq \ell \left\| A^{-1/2}x \right\|^2 \tag{2.12}$$

Proof. Since $A = U\Lambda U^\top$, for any $x \in \operatorname{im} A$, $x \in \operatorname{im} U$, there exists y, z such that $x = Uy$, $x = Az =$

$U\Lambda U^\top z$. Hence $x = UIy = UU^\top Uy = UU^\top x$, and $U^\top x = \Lambda U^\top z \in \text{im}\Lambda$.

$$\begin{aligned}\|x\|^2 =& x^\top UU^\top UU^\top X = x^\top UU^\top x \\ =& x^\top U\Lambda^{-1/2}\Lambda\Lambda^{-1/2}U^\top x + x^\top U\left(I - \Lambda^{-1/2}\Lambda\Lambda^{-1/2}\right)U^\top x \\ =& x^\top U\Lambda^{-1/2}\Lambda\Lambda^{-1/2}U^\top x \\ \in& \left[\sigma\|\Lambda^{-1/2}U^\top x\|^2, \ell\|\Lambda^{-1/2}U^\top x\|^2\right]\end{aligned} \tag{2.13}$$

Here we have the third equality derived because Λ is a diagonal matrix and $I - \Lambda^{-1/2}\Lambda\Lambda^{-1/2} \in (\text{im}\Lambda)^\perp$.

We observe that in (2.13), $\|\Lambda^{-1/2}U^\top x\|^2 = x^\top U\Lambda^{-1/2}U^\top U\Lambda^{-1/2}U^\top x = x^\top A^{-1/2}A^{-1/2}x = \|A^{-1/2}x\|^2$, then (2.12) is obtained. □

Lemma 2.2.2. *The dual problem* (2.9) *is strong convex on imA with prox function* $V(w_1, w_2) := \frac{1}{2}\|A^{-1/2}(w_1 - w_2)\|^2$.

Proof. For any z_1, $z_2 \in \text{im}\, A$, there exist w_1, w_2, and $\sigma > 0$ such that $w_1 = A^{1/2}z_1$, $w_2 = A^{1/2}z_2$. By (2.10) and Lemma 2.2.1, $d(w_1) - d(w_2) - \langle \nabla d(w_2), w_1 - w_2\rangle \ - \sigma V(w_1, w_2) = \frac{1}{2}\|w_1 - w_2\|^2 - \frac{\sigma}{2}\|A^{-1/2}(w_1 - w_2)\|^2 \geq 0$. In other words, the dual problem is strongly convex with respect to the prox function $V(w_1, w_2) := \frac{1}{2}\|A^{-1/2}(w_1 - w_2)\|^2$. □

It should be noted that we choose the prox function V as an operation norm defined by operator $T : \mathbb{R}^n \to \text{im}A$, $T(x) = A^{-1/2}x$ (instead of the trivial Euclidean version under which d is trivially strongly convex) for the sole purpose of avoiding the requirement on knowledge of $A^{1/2}$. Indeed, we can show that the accelerated gradient method in Algorithm 2.2.3 can be described equivalently as in Algorithm 2.2.4. Although they are equivalent, to distinguish their different descriptoins, we denote Algorithm 2.2.3 and 2.2.4 by AGD and AGD2 respectively.

Recall that our purpose is to solve the primal problem of x. Following directly from (2.11), Therom 2.2.2 showed that Algorithm 2.2.4 converge linearly with respect to $\hat{x}_t$.

Theorem 2.2.2. *If we have that* $\beta = \sigma$, $\xi = \sqrt{\frac{\sigma}{L}}$, $\gamma = \frac{\xi}{1+\xi}$ *and* $q = \frac{\sigma(1-\xi)}{\xi}$, *then Algorithm 2.2.4*

Algorithm 2.2.4 An equivalent description of accelerated gradient descent method (AGD2)

Require: Initial point $\hat{x}^0 = u - \underline{w}_0 \in \mathbb{R}_n$, $\hat{w}_0$, w_0, $\underline{w}_0 \in \operatorname{im} A$

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

$$\begin{aligned}
\underline{w}_t &= (1-\gamma)\hat{w}_{t-1} + \gamma w_{t-1}, \\
x_t &= u - \underline{w}_t, \\
w_t &= \frac{1}{q\sigma + q}(q\sigma \underline{w}_t + q w_{t-1} + A x_t), \\
\hat{w}_t &= (1-\xi)\hat{w}_{t-1} + \xi w_t, \\
\hat{x}_t &= u - \hat{w}_t.
\end{aligned}$$

end for

has a $\mathcal{O}\left(\sqrt{\frac{\ell}{\sigma}}\log\frac{1}{\varepsilon}\right)$ *complexity with,*

$$\|\hat{x}_N - x^*\|^2 \leq 2\left(1 - \sqrt{\sigma/\ell}\right)^N \|\hat{x}_0 - x^*\|^2, \tag{2.14}$$

Proof. By definition of $d(w)$ in (2.10), since $\hat{w}_N - w^* \in \operatorname{im} A$, $d(\hat{w}_N) - d(w^*) = \frac{1}{2}\|\hat{w}_N - w^*\|^2$. Let $w = w^*$ in (2.11), we obtain

$$\begin{aligned}
\|\hat{w}_N - w\|^2 \leq & (1-\xi)^N \left[\|\hat{w}_0 - w\|^2 + 2\sigma V(w_1, w)\right]. \\
= & \left(1 - \sqrt{\sigma/\ell}\right)^t \left[\|\hat{w}_0 - w^*\|^2 + \sigma\|A^{-1/2}(w_0 - w^*)\|^2\right] \\
\leq & 2\left(1 - \sqrt{\sigma/\ell}\right)^N \|\hat{w}_0 - w^*\|^2.
\end{aligned}$$

Since $\hat{w}_t = u - \hat{x}_t$, $t = 0, 1, ..., N$ and $w^* = u - x^*$, the above inequality becomes (2.14). Without loss of generality, we can define $w^0 = \underline{w}^0 = w^{-1} = 0$. □

Remark that Theorem 2.2.2 shows an ε-approximate solution is obtained with $\mathcal{O}\left(\sqrt{\ell/\sigma}\log 1/\varepsilon\right)$ matrix-vector multiplication complexity.

2.2.3 The linear algebra perspective

From the optimization perspective described in the previous subsection, we observed that our problem of interest (2.1) has a dual problem (2.9) with respect to dual variable vector z. The

optimality condition of the dual problem is

$$Az = b \quad \text{where } b := A^{1/2}u. \tag{2.15}$$

As mentioned in the previous subsection, the information of b is not readily available. However, if we assume that we know the value of b, it suffices to solve the above linear system to compute an approximate dual optimal solution. Moreover, the iteration complexity theory on solving linear systems has already been well developed thanks to [21]. In this subsection, we describe the results in [21] concerning the complexity for solving the linear system (2.15) by matrix-vector multiplications concerning A.

As described in the introduction, in complexity analysis we are interested in both the upper and lower bounds, where the upper bounds are established by analyzing the convergence properties of numerical methods, and the lower bounds are established by constructing worst-case instances. Concerning the complexity analysis on solving the linear system (2.15), the numerical methods of interest are methods that only utilizes matrix-vector multiplications, and the worst-case instances would be the worst possible choice of matrix A and b in problem (2.15). To better describe the core idea in [21], we deter the technical discussion on general matrix-vector multiplication methods and make a further simplification that the numerical methods of interest are Krylov subspace methods, namely they have iterations in the linear span:

$$z_t \in \text{span}\{b, Ab, \ldots, A^t b\}.$$

Note that the above assumption is equivalent to $z_t = q(A)b$, where q is a polynomial such that $\deg q \leq t$. The design of a Krylov subspace type method is equivalent to find a proper polynomial q. Therefore, the complexity analysis for problem (2.15) can be informally formulated as the following saddle point problem:

$$\inf_{q:\, \deg q \leq t} \sup_{A,b} \|q(A)b - z^*\|.$$

Here z^* is an optimal solution to problem (2.15). The above saddle point problem captures the nature of upper and lower complexity bounds. For the infimum problem, we are interested at designing a numerical method (namely finding q) whose iterate z_t has small error $\|z_t - z^*\|$ in the worst-case.

For the supremum problem, we are interested at designing worst-case problem instances with A and b such that the best possible performance among all numerical methods yield large error $\|z_t - z^*\|$.

Formally, in [21] it is assumed that the problem class for A and b are the following:

$$\mathcal{V}(\Sigma, R) \coloneqq \{(A,b)| \text{ positive spectrum of } A \subset \Sigma,$$
$$Az^*(A,b) = b, z^*(A,b) \in \operatorname{im} A, \|z^*(A,b)\| \leq R.\}$$

The main result of [21] indicates that the best worst-case approximate linear span method, which realizes its worst-case approximation error for some problem instance, will be the best amongst all deterministic methods in terms of the worst-case error, as long as the problem dimension n at least doubles the maximum number of iterations.

Proposition 2.2.5. *Let Σ be a compact set, then for each t,*

$$\inf_{q:\deg q \leq t} \sup_{A,b} \|(q(A)A - I_n)z^*(A,b)\| = R \max_{t\in\Sigma} |p^*(t)|.$$

The best linear span method can be determined by Chebyshev polynomial, when Σ is a closed interval that is strictly positive; see [19] for a reference:

Lemma 2.2.3. *Let $\Sigma = [\sigma, \ell]$, then $p^*(t) = T_{k+1}\big(\frac{\ell+\sigma-2t}{\ell-\sigma}\big)/T_{k+1}\big(\frac{\ell+\sigma}{\ell-\sigma}\big)$, and*

$$q^*(t) = \Big(T_{k+1}\big(\frac{\ell+\sigma-2t}{\ell-\sigma}\big)/T_{k+1}\big(\frac{\ell+\sigma}{\ell-\sigma}\big) - 1\Big)/t.$$

It should be noted that our discussion in this subsection does not apply directly to our problem of interest, since in the linear system (2.15) the value of $b = A^{1/2}u$ is not available. Moreover, there is a drawback in the above results of [21] for practical implementation since the total number of iterations k should be fixed in priori. However, we will show in the next section that we can adopt the technique in [18] to solve our problem of interest by making some modifications. Moreover, for our problem, our analysis does not require knowledge on the total number of iterations.

2.3 Exact complexity under a linear span assumption

In this section, we prove that the lower complexity bound for solving problem 2.1 is able to be obtained by Chebyshev method. Specifically, for any algorithm methods which are in the linear span of (A, u) we design a worst-case problem of problem (2.1) such that the total number of matrix-vector multiplications required by this algorithm when solving problem (2.1) has to be greater than or equal to

$$\log\left((R+\sqrt{R^2-\varepsilon^2})/\varepsilon\right)/\log\left((\sqrt{\ell}+\sqrt{\sigma})/(\sqrt{\ell}-\sqrt{\sigma})\right).$$

Therefore, the lower complexity bound is in the order of $\Omega\left(\sqrt{\ell/\sigma}\log(R/\varepsilon)\right)$. Such complexity is obtained by our proposed Chebyshev method. Moreover, the proposed algorithm achieves not only the optimal complexity with respect to order $\mathcal{O}\left(\sqrt{\ell/\sigma}\log(R/\varepsilon)\right)$, but also the optimal complexity with respect to the constant leading the logarithm.

2.3.1 Lower complexity bound under linear span assumption

In this subsection, we consider the lower oracle complexity bound for the following constrained optimization problem (2.1) where A is a parameter matrix of size $n \times n$ such that $A^\top = A$, $\|A\| = \ell$, and the smallest nonzero eigenvalue of A is $\sigma > 0$; $u \in \mathbb{R}^n$ is a parameter vector. We define such problem as $\mathbb{P}(A, u)$.

In $\mathbb{P}(A, u)$ of problem (2.1), the optimality condition is as follows,

$$\begin{aligned} &x^*(A,u) - u + y^*(A,u) = 0 \\ &y^*(A,u) = Az^*(A,u) \\ &Ax^*(A,u) = 0. \end{aligned} \tag{2.16}$$

Note here that $u = x^*(A, u) + y^*(A, u)$ is the unique decomposition with respect to the direct sum $\mathbb{R}^n = \ker A \oplus \operatorname{im} A$. We define the *accuracy* of x as an approximate solution to problem $\mathbb{P}(A, u)$ by $\varepsilon^*(x, A, u) \coloneqq \|x - x^*(A, u)\|$.

Oracle Assumption. Assume when solving $\mathbb{P}(A, u)$ of problem (2.1), u is explicitly given, but A is not, and at each step we can perform multiplication Av at any $v \in \mathbb{R}^n$ of our choice.

Define the Problem Class of interest

$$\begin{aligned}\mathcal{U}(\Sigma, R) \coloneqq \{\mathbb{P}(A,u) |\ &\text{positive spectrum of } A \subset \Sigma,\\ &y^*(A,u) \in \operatorname{im} A, \|y^*(A,u)\| \le R.\}\end{aligned} \tag{2.17}$$

We fix compact set $\Sigma \subset \mathbb{R}$ which consists only positive values, e.g., $\Sigma = [\sigma,\, \ell]$, and let $t > 0$ be an integer. Define

$$\delta(t, \Sigma) = \min_{q:\, \deg q \le t, q(0)=1} \max_{t \in \Sigma} |q(s)|. \tag{2.18}$$

Recall that in this section, we only consider the methods which are in the linear span of (A, u), i.e., the t-th approximate solution (what the method returns after t multiplications of A and recursively computed vectors) $x_t \in u - \operatorname{span}\{Au, \ldots, A^t u\}$, where t is the iteration counter. In other words, $x_t = q(A)u$ where q is a polynomial such that $\deg q \le t$ and $q(0) = 1$. Let $q_{\Sigma,t}(s)$ be the optimal solution to the above problem $\delta(t, \Sigma)$. Let $\mathbb{M}_\Sigma$ be the method such that, as applied to $\mathbb{P}(A, u)$, it queries at points $u, Au, A^2u \ldots$, recursively computes $Au, A^2u, A^3u, \ldots$, and returns the points

$$\hat{x}_t(A, u) \coloneqq q_{\Sigma,t}(A)u$$

as approximate solutions.

By (2.16), we have that $x_t = x^*(A, u) + q(A)y^*(A, u)$ and $\varepsilon^*(x, A, u) = \|x - x^*(A, u)\| = \|q(A)y^*(A, u)\|$. We formulate the problem of finding the "best" method which gives the "worst-case guarantee" as follows:

$$\begin{aligned}&\inf_{q:\, \deg q \le t,\, q(0)=1} \sup_{\mathbb{P}(A,u) \in \mathcal{U}} \|x - x^*(A,u)\|^2\\ =&\min_{q:\, \deg q \le t,\, q(0)=1} \max_{a \in \Sigma^m,\, b \in \mathbb{R}^m,\, m<n:\, \|b\| \le R} \sum_{i=1}^{m} (q(a_i)b_i)^2\\ =&\min_{q:\, \deg q \le t,\, q(0)=1} \max_{a \in \Sigma^m,\, m<n} \max_{i=1,\ldots,m} R^2 (q(a_i))^2\\ =&R^2 \min_{q:\, \deg q \le t,\, q(0)=1} \max_{s \in \Sigma} (q(s))^2\end{aligned} \tag{2.19}$$

The first equality holds because when $A = U\Lambda U^\top$ is diagonalizable, $\|q(A)y^*(A,u)\| = \|q(\Lambda)(U^\top y^*(A,u))\|$, $\|U^\top y^*(A,u)\| = \|y^*(A,u)\|$, hence it all boils down to the positive eigenvalues of A.

By standard uniform polynomial approximation (see cf. [20], Exercise 12.6.2), we have the

following lemma, which indicates that as long as $n > t+1$, the last equality of the above holds.

Lemma 2.3.1. *A polynomial q is optimal to*

$$\min_{q:\,\deg q\le t,\, q(0)=1} \max_{s\in\Sigma} |q(s)|$$

if and only if there exists $(t+1)$ points $a_0 < \ldots < a_t$ in Σ such that

$$q(a_i) = (-1)^i \max_{t\in\Sigma} |q(t)|, \quad \textit{or} \quad q(a_i) = (-1)^{i+1} \max_{t\in\Sigma} |q(s)|.$$

In the case that $\Sigma = [\sigma, \ell]$, the optimizer polynomial is based on Chebyshev polynomial T_t of degree t:

$$q^*(s) = T_t\big(\frac{\ell+\sigma-2s}{\ell-\sigma}\big)/T_t\big(\frac{\ell+\sigma}{\ell-\sigma}\big),$$

where

$$T_t(s) = \begin{cases} \cos(t\arccos s) & s\in[-1,\,1] \\ \frac{1}{2}[(s-\sqrt{s^2-1})^t + (s+\sqrt{s^2-1})^t] & \text{otherwise} \end{cases} \tag{2.20}$$

Hence the method we propose is

$$\hat{x}_t = \frac{T_t(\frac{\ell+\sigma}{\ell-\sigma} I_n - \frac{2}{\ell-\sigma}A)}{T_t(\frac{\ell+\sigma}{\ell-\sigma})} u$$

Based on the property of Chebyshev polynomials and the chain of equalities (2.19), we note that the problem $\mathbb{P}(A,u)$ which produces the maximum worst-case error, regardless of the method polynomial q, satisfies that

$$\{\lambda > 0 : \lambda \text{ is an eigenvalue of } A\} \supset \{\frac{1}{2}[(L+\sigma) - (L-\sigma)\cos(j\pi/t)] : j = 0, \ldots, t\};$$

note that the above enlisted eigenvalues are all maximizers of $|T_t(\frac{\ell+\sigma-2s}{\ell-\sigma})|$.

Note that Chebyshev polynomial satisfies that $T_0(s) = 1$, $T_1(s) = s$, $T_{t+1}(s) = 2sT_t(s) - T_{t-1}(t)$ for all $t \ge 1$, we have recurrence Algorithm 2.3.5.

Here the sequence $x_t = T_t(\frac{\ell+\sigma}{\ell-\sigma} I_n - \frac{2}{\ell-\sigma}A)u$ is always well defined, because the eigenvalues

Algorithm 2.3.5 Chebyshev method

Require: Initial point $x_0 = u \in \mathbb{R}^n$, $\ell > \sigma > 0$

$x_1 = \dfrac{\ell+\sigma}{\ell-\sigma}u - \dfrac{2}{\ell-\sigma}Au.$

for $t = 1, \ldots, T$ **do**

$$x_{t+1} = 2\frac{\ell+\sigma}{\ell-\sigma}x_t - \frac{4}{\ell-\sigma}Ax_t - x_{t-1},$$
$$\hat{x}_t = \frac{2(\frac{\sqrt{\ell}-\sqrt{\sigma}}{\sqrt{\ell}+\sqrt{\sigma}})^t}{(\frac{\sqrt{\ell}-\sqrt{\sigma}}{\sqrt{\ell}+\sqrt{\sigma}})^{2t}+1}x_t.$$

end for

of $\frac{\ell+\sigma}{\ell-\sigma}I_n - \frac{2}{\ell-\sigma}A$ are all in $[-1, 1]$.

We prove in this subsection that the lower complexity bound for solving problem (2.1) is able to be obtained by Chebyshev method. In the following section, we will explicitly provide the convergence analysis of Chebyshev method.

2.3.2 Chebyshev method

Recall that without loss of generality, we can assume that $\|A\| = \ell$ and the smallest nonzero eigenvalue of A is σ. Also we can perform decomposition of vector u as $u = x^* + y^*$, such that $Ax^* = 0$ and $y^* = Az^*$ for some z^*. In other words, $y^* \in \operatorname{im} A$, $x^* \in \ker A$. Decompose the matrix A as $A = U\Lambda U^\top$ with matrix U orthogonal. Λ is a diagonal matrix and diagonal entries are eigenvalues of matrix A. Since $y^* \in \operatorname{im} A$ and $x^* \in \ker A$, we have that $U^\top y^* \in \operatorname{im}\Lambda$ and $U^\top x^* \in \ker\Lambda$. Assume that $\|y^*\| = \|u - x^*\| \leq R$. We have the optimal Algorithm 2.3.5 solving 2.1 equivalently expressed as Algorithm 2.3.6.

Algorithm 2.3.6 Chebyshev method

Require: Initial point $\hat{x_0} = u \in \mathbb{R}^n$

for $t = 0, 1, \ldots, T$ **do**

$$x_t = u - P(A)Au.$$

end for

The convergence analysis of Algorithm 2.3.6 for solving Problem (2.1) is provided in the following theorem,

Theorem 2.3.1. *Let $\{x_t\}_{t=0}^N$ be iterate of Algorithm 2.3.6 for solving problem* (2.1)*. If the polynomial P is defined as*

$$P(s) = \frac{1}{s}\left[T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right) - T_t\left(\frac{\ell+\sigma-2s}{\ell-\sigma}\right)\right]/T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right), \forall s \in [\sigma, \ell], \tag{2.21}$$

where T_t is the Chebyshev polynomial defines in (2.20)*, then we have*

$$\|x_t - x^*\| \leq R/\left|T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right)\right|.$$

Proof. Combining the above with $x_k - x^* = u - P(A)Ay^* - x^* = y^* - P(A)Ay^*$ and $\|y^*\| \leq R$, we have that

$$\begin{aligned}\|x_t - x^*\| =& \|y^* - P(A)Ay^*\| = \|UU^\top y^* - U\Lambda P(\Lambda)U^\top(x^* + y^*)\| \\ =& \|UU^\top y^* - U\Lambda P(\Lambda)U^\top y^*\| = \|U^\top y^* - \Lambda P(\Lambda)U^\top y^*\| \\ \leq& \|U^\top y^*\| \cdot \|I - \Lambda P(\Lambda)\| \leq R \cdot \|I - \Lambda P(\Lambda)\|.\end{aligned} \tag{2.22}$$

Let A be a matrix with eigenvalues $a^{(1)}$,...,$a^{(m)}$,0,...,0, such that $a^{(i)} \in [\sigma, \ell]$, for $i = 1, ..., m$. Then by (2.22), we have that

$$\|x_t - x^*\|^2 \leq \max_{a\in[\sigma,\ell]^m} \max_{i=1,\ldots,m} R^2(1 - a^{(i)}P(a^{(i)}))^2 \leq \max_{s\in[\sigma,\ell]} R^2(1 - sP(s))^2$$

Thus to find the upper bound of $\|x_t - x^*\|$, it is suffices to solve $\max_{s\in[\sigma,\ell]} R|1 - sP(s)|$. In other words,

$$\|x_t - x^*\| \leq \max_{s\in[\sigma,\ell]} R|1 - sP(s)|. \tag{2.23}$$

By definition of P in (2.21), P satisfies

$$1 - sP(s) = T_t\left(\frac{\ell+\sigma-2s}{\ell-\sigma}\right)/T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right), \ \forall s \in [\sigma, \ell]. \tag{2.24}$$

By the previously aquired inequality (2.24), we have that

$$|1 - sP(s)| \leq \left|T_t\left(\frac{\ell+\sigma-2s}{\ell-\sigma}\right)\right| / \left|T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right)\right|, \ \forall s \in [\sigma, \ell]. \tag{2.25}$$

With $t \leq m+1$, consider a' such that

$$\begin{aligned} a' &= \arg\max_{s\in[\sigma,\ell]} \left|T_t\left(\frac{\ell+\sigma-2s}{\ell-\sigma}\right)\right| / \left|T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right)\right| \\ &= \arg\max_{s\in[\sigma,\ell]} \left|T_t\left(\frac{\ell+\sigma-2s}{\ell-\sigma}\right)\right|. \end{aligned}$$

By the definition of Chebyshev function (2.20), we have that

$$a' = \frac{1}{2}\left[\ell+\sigma-(\ell-\sigma)\cos\left(\frac{j\pi}{t}\right)\right], \quad j = 0, ..., t. \tag{2.26}$$

And when (2.26) satisfies,

$$\left|T_t\left(\frac{\ell+\sigma-2a'}{\ell-\sigma}\right)\right| = 1,$$

and thus by (2.25) for any $s \in [\sigma, \ell]$, we have that

$$|1-sP(s)| \leq \left|T_t\left(\frac{\ell+\sigma-2a'}{\ell-\sigma}\right)\right| / \left|T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right)\right| = 1/\left|T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right)\right|. \tag{2.27}$$

Combining (2.27) with (2.23), we have the upper bound of Algorithm 2.3.6 as

$$\|x_t - x^*\| \leq \max_{s\in[\sigma,\ell]} R|1-sP(s)| \leq R/\left|T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right)\right|.$$

□

Observe from the previous theorem that since $1/T_t\left(\frac{\ell+\sigma}{\ell-\sigma}\right) \in \left[\frac{1}{2}\left(\frac{\sqrt{\ell/\sigma}-1}{\sqrt{\ell/\sigma}+1}\right)^t, \left(\frac{\sqrt{\ell/\sigma}-1}{\sqrt{\ell/\sigma}+1}\right)^t\right]$, we have that the upper bound of $\|x_t - x^*\|$ lies in

$$\left[\frac{R}{2}\left(\frac{\sqrt{\ell/\sigma}-1}{\sqrt{\ell/\sigma}+1}\right)^t, R\left(\frac{\sqrt{\ell/\sigma}-1}{\sqrt{\ell/\sigma}+1}\right)^t\right].$$

Specifically, observe from Theorem 2.3.1 and that to obtain an approximate solution such that

$\|x_t - x^*\| \leq \varepsilon$, the total number of iteration N satisfies

$$\frac{R}{\varepsilon} = \frac{1}{2}\left[\left(\frac{\sqrt{\ell}+\sqrt{\sigma}}{\sqrt{\ell}-\sqrt{\sigma}}\right)^N + \left(\frac{\sqrt{\ell}-\sqrt{\sigma}}{\sqrt{\ell}+\sqrt{\sigma}}\right)^N\right].$$

Noting that the above is a quadratic equation in terms of $\left(\frac{\sqrt{\ell}+\sqrt{\sigma}}{\sqrt{\ell}-\sqrt{\sigma}}\right)^N$ that is greater than 1, we can derive that

$$\left(\frac{\sqrt{\ell}+\sqrt{\sigma}}{\sqrt{\ell}-\sqrt{\sigma}}\right)^N = \frac{R}{\varepsilon} + \sqrt{\frac{R^2}{\varepsilon^2} - 1}.$$

This gives the exact total number of matrix-vector multiplications required by this algorithm when solving $\mathbb{P}(A, u)$ has to be greater than or equal to

$$N = \log\left((R + \sqrt{R^2 - \varepsilon^2})/\varepsilon\right) / \log\left((\sqrt{\ell}+\sqrt{\sigma})/(\sqrt{\ell}-\sqrt{\sigma})\right).$$

Remark that in the previous subsection we proved that the lower matrix-vector multiplications complexity is obtained exactly by Chebyshev method in Algorithm 2.3.6. Thus the upper matrix-vector multiplications complexity bound of the proposed algorithm, in the order of $\mathcal{O}\left(\sqrt{\ell/\sigma}\log(R/\varepsilon)\right)$, matches exactly with our lower complexity bound. More specifically, the proposed algorithm achieves not only the optimal complexity with respect to order, but also the optimal complexity with respect to the constant leading the logarithm.

2.4 Lower complexity bound analysis for general deterministic methods

In this section, we consider more general deterministic methods for problem $\mathbb{P}(A, u)$ of problem (2.1) which are not necessarily in the linear span of (A, u).

For each deterministic method $\mathbb{M}$ solving problems from a collection $\mathcal{U}$ of problems based on the oracle assumption, we let $x_{\mathbb{M}}(t, A, u)$ denote the t-th approximate solution (what the method returns after t multiplications of A and recursively computed vectors), then the worst-case accuracy

measure w.r.t. collection $\mathcal{U}$ is

$$\varepsilon(\mathbb{M}, t, \mathcal{U}) = \sup_{(A,u)\in\mathcal{U}} \varepsilon^*(x_{\mathbb{M}}(t, A, u), A, u),$$

and the best possible worst-case accuracy is

$$e(t, \mathcal{U}) = \inf_{\mathbb{M}} \varepsilon(\mathbb{M}, t, \mathcal{U}).$$

Define the Problem Class of interest

$$\mathcal{U}(\Sigma, R) \coloneqq \{(A, u)| \text{ positive spectrum of } A \subset \Sigma,$$
$$y^*(A, u) \in \operatorname{im} A, \|y^*(A, u)\| \leq R.\}$$

Fix compact set $\Sigma \subset \mathbb{R}$ which consists only positive values, e.g., $\Sigma = [\sigma, \ell]$, and let $t > 0$ be an integer. Define

$$\delta(t, \Sigma) = \min_{q:\, \deg q \leq t,\, q(0)=1} \max_{t\in\Sigma} |q(s)|,$$

where q is a function describing $\mathbb{M}$. Let $q_{\Sigma,t}(s)$ be the optimal solution to the above problem. Let $\mathbb{M}_\Sigma$ be the method such that, as applied to (A, u), it queries at points $u, Au, A^2u \ldots$, recursively computes $Au, A^2u, A^3u, \ldots$, and returns the points

$$\hat{x}_t(A, u) \coloneqq q_{\Sigma,t}(A)u \tag{2.28}$$

as approximate solutions.

Let $H_2 \subset H_1$ such that $H_1 = \operatorname{span}\{x^*\} \oplus H_2$ and that $x^* \perp H_2$. To compute the worst-case accuracy measure, we first introduce the following lemma that shows the approximate solutions are able to be decomposed as a sum of a point on a Krylov space and H_2.

Lemma 2.4.1 (Main Lemma)**.** *Let $\mathbb{M}$ be an arbitrary deterministic method solving problems from $\mathcal{U}_0$, let $2t \leq n-3$. Then there exists an orthogonal matrix $V \in \mathbb{R}^{n\times n}$, $Vu = u$ such that the following holds:*

$\mathcal{R}(t, V)$: the t-th approximate solution x_t found by $\mathbb{M}$ as appied to $(VA_0V^\top, u)$ lies in

$$G_t = E_t(VA_0V^\top, u) + VH_2,$$

where $E_t(B,b) \coloneqq \operatorname{span}\{b, Bb, \ldots, B^t b\}$ *is the* t*-th Krylov space of* $(B,b) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n$.

With the help of Lemma 2.4.1, we are able to show in the following theorem that the accuracy measure is $R\,\delta(t,\Sigma)$.

Theorem 2.4.1. *Let* Σ *be a compact set, then for each* t*,*

$$\varepsilon(\mathbb{M}_\Sigma, t, \mathcal{U}(\Sigma,R)) \le R\,\delta(t,\Sigma), \tag{2.29}$$

and for each $t \le \frac{n-3}{2}$*,*

$$\varepsilon(\mathbb{M}_\Sigma, t, \mathcal{U}(\Sigma,R)) = e(t, \mathcal{U}(\Sigma,R)) \ge R\,\delta(t,\Sigma).$$

Proof. Step (a), consider any problem $(A,u) \in \mathcal{U}(\Sigma,R)$, let $A = U\Lambda U^\top$ be its diagonalization, then

$$\begin{aligned}
\|\varepsilon^*(x_{\mathbb{M}_\Sigma}(t,A,u),A,u)\|^2 &= \|\hat{x}_t(A,u) - x^*(A,u)\|^2 \\
&= \|(q_{\Sigma,t}(0)-1)x^*(A,u) + q_{\Sigma,t}(A)y^*(A,u)\|^2 \\
&= \|q_{\Sigma,t}(A)y^*(A,u)\|^2 \\
&= \|W q_{\Sigma,t}(\Lambda)W^\top y^*(A,u)\|^2 \\
&\le \|q_{\Sigma,t}(\Lambda)\|^2 R^2 \\
&\le R^2\,\delta^2(t,\Sigma).
\end{aligned}$$

Here the first equality is substitution of (2.28); the second equality follows from optimality conditions (2.16); the third equality follows from $q_{\Sigma,t}(0) = 1$; the first inequality follows from (2.17); the second inequality follows from (2.17) and (2.18), the maximum of which is over entire Σ. Since (A,u) is arbitrary, (2.29) is shown.

Step (b), since $\{q : \deg q \le t,\, q(0) = 1\}$ is a vector space of dimension t, then there exists $\Sigma' \subset \Sigma$ containing at most $(t+1)$ points such that

$$\delta(t,\Sigma) = \min_{q:\,\deg q \le t,\, q(0)=1} \max_{s\in\Sigma'} |q(s)|.$$

Let $\Sigma' = \{s_1, \ldots, s_L\}$ be the set of maximizers of the optimal q for some $L \le t+1$. Let $\mu =$

$(\mu_1, \ldots, \mu_L)$ be some probability distribution such that

$$\delta^2(t, \Sigma) = \min_{q:\, \deg q \le t, q(0)=1} \sum_{i=1}^{L} \mu_i |q(s_i)|^2. \tag{2.30}$$

Step (c), let $H_0 = \operatorname{span}\{e_1, \ldots, e_L\}$, $H_1 = H_0^{\perp}$. Define $A_0 \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} A_0 e_i &= s_i e_i \quad 1 \le i \le L, \\ A_0 x &= 0 \qquad x \in H_1. \end{aligned}$$

Let $y^* = R\sum_{i=1}^{L} \mu_i^{1/2} e_i$, and let $x^* \in H_1$ be arbitrary so its magnitude can be arbitrarily large. Further, since $H_2 \subset H_1$, $H_1 = \operatorname{span}\{x^*\} \oplus H_2$ and $x^* \perp H_2$. Define

$$\mathcal{U}_0 \coloneqq \{(A, u) | A = V A_0 V^{\top} \text{ for some orthogonal } V \in \mathbb{R}^{n\times n} \text{ such that } Vu = u.\}$$

It is evident that $\mathcal{U}_0 \subset \mathcal{U}(\Sigma, R)$: firstly, $(A_0, u) \in \mathcal{U}_0$ by definition; for any orthogonal V with $Vu = u$, the optimality condition of problem $(VA_0V^{\top}, u)$ can be written as

$$\begin{aligned} u &= V^{\top} x^*(A, u) + V^{\top} y^*(A, u) \\ V^{\top} y^*(A, u) &= A_0 V^{\top} z^*(A, u) \\ A_0 V^{\top} x^*(A, u) &= 0, \end{aligned}$$

hence, by the unique decomposition of u in terms of $\mathbb{R}^n = \ker A \oplus \operatorname{im} A$, we know $x^*(A, u) = Vx^*$, $y^*(A, u) = Vy^* = RV\sum_{i=1}^{L} \mu_i^{1/2} e_i$.

We utilize Lemma 2.4.1 to complete the proof of theorem. Let $\mathbb{M}$ be a deterministic method, let $(A, u) = (VA_0V^{\top}, u)$ be the corresponding problem as in Lemma 2.4.1, then $\varepsilon^*(x_t, A, u) = \|x_t - x^*(A, u)\| = \|x_t - Vx^*\| = \|V^{\top}x_t - x^*\|$. Since $x_t \in E_t(VA_0V^{\top}, u) + VH_2$, $V^{\top}E_t(VA_0V^{\top}, u) = E_t(A_0, u)$, we know $V^{\top}x_t = w + v$, $w = q(A_0)u$, $v \in H_2$ for some polynomial q such that $\deg q \le t$.

Note that by definition, $\mathbb{R}^n = H_0 \oplus H_2 \oplus \text{span}\{x^*\}$, clearly,

$$\begin{aligned}
\|V^\top x_t - x^*\|^2 &= \|q(A_0)u + v - x^*\|^2 \\
&= \|(q(0)-1)x^* + q(A_0)y^* + v\|^2 \\
&= (q(0)-1)^2\|x^*\|^2 + \|q(A_0)y^*\|^2 + \|v\|^2 \\
&= (q(0)-1)^2\|x^*\|^2 + \sum_{i=1}^{L} R^2 q^2(s_i)\mu_i + \|v\|^2 \\
&\geq \sum_{i=1}^{L} R^2 q^2(s_i)\mu_i \text{ s.t. } \deg q \leq t,\, q(0) = 1 \\
&\geq R^2\, \delta^2(t, \Sigma).
\end{aligned}$$

Here the second equality follows from optimality conditions (2.16); the third equality follows because the three vectors are pairwise orthogonal; the fourth equality follows from (2.30); the first inequality follows from the arbitrariness of $x^* \in H_1$; the second inequality follows from (2.30). The proof is therefore complete. □

Remark 2.4.1. *In fact from the analysis of any Krylov type algorithm which returns $V^\top x_t = q(A_0)u + v = q(0)x^* + q(A_0)y^* + v$, we notice that it will not be optimal if $v \neq 0$, nor in the case that x^* is possibly large, hence the algorithm must have $q(0) = 1$.*

Remark 2.4.2. *It is commonly seen in concensus updates that the mixing matrix W satisfies that $W\mathbf{1} = \mathbf{1}$. In our context where $A = I_n - W$, it means that $\mathbf{1} \in \ker A$. Although $\mathbf{1} \notin \ker A_0 = H_1$, we can still find an orthogonal matrix $V \in \mathbb{R}^{n\times n}$, $Vu = u$ which makes $I_n - A = I_n - VA_0V^\top$ a mixing matrix. Since $\ker VA_0V^\top = VH_1$, it suffices to make sure that $V^\top\mathbf{1} \in H_1$. An orthogonal matrix $V \in \mathbb{R}^{n\times n}$ which satisfies $V^\top u = u$ and $V^\top\mathbf{1} \in H_1$ is not hard to find, and it has $(n-3)$ degree of freedom.*

The proof below of the Main Lemma can be moved to the appendix.

Proof of Lemma 2.4.1. Step (i), prove by induction on $\hat{t}$ that for each $\hat{t} \in \{0, \ldots, t+1\}$, there exists an orthogonal $V_{\hat{t}} \in \mathbb{R}^{n\times n}$, $V_{\hat{t}}u = u$, and an $\hat{t}$-dimensional subspace $H^{\hat{t}} \subset H_2$ such that the following holds:

$\mathcal{R}^*(\hat{t}, V_{\hat{t}}, H^{\hat{t}})$: the points $x(1), \ldots, x(\hat{t})$ at which $\mathbb{M}$ queries during the first $\hat{t}$ iterations as appied to $(V_{\hat{t}} A_0 V_{\hat{t}}^\top, u)$ belongs to subspace

$$F_{\hat{t}} = E_{\hat{t}-1}(V_{\hat{t}} A_0 V_{\hat{t}}^\top, u) + V_{\hat{t}} H^{\hat{t}},$$

here we let $E_{-1}(B, b) \coloneqq \{0\}$.

To ensure $\mathcal{R}^*(0, V_0, H^0)$, as no query/multiplication has been performed yet, we set $V_0 = I_n$, $H^0 = \{0\}$, so $F_0 = \{0\}$.

Assume $\hat{t} \le t$ and $\mathcal{R}^*(\hat{t}, V_{\hat{t}}, H^{\hat{t}})$ holds for certain $V_{\hat{t}}$ and $H^{\hat{t}}$, show $\mathcal{R}^*(\hat{t}+1, V_{\hat{t}+1}, H^{\hat{t}+1})$ with choice of $V_{\hat{t}+1}$ and $H^{\hat{t}+1}$.

Let $x(\hat{t}+1)$ be the $(\hat{t}+1)$-st point at which $\mathbb{M}$ queries as applied to $(A_{\hat{t}}, u) \coloneqq (V_{\hat{t}} A_0 V_{\hat{t}}^\top, u)$, let $x(\hat{t}+1) = w + v$ be an orthogonal decomposition, where $w \in E_{\hat{t}}(A_{\hat{t}}, u)$ and $v \in E_{\hat{t}}(A_{\hat{t}}, u)^\perp$. Note that $\dim(V_{\hat{t}} H_2) = \dim(H_2) = n - L - 1 \ge n - t - 2$, while $\dim(V_{\hat{t}} H^{\hat{t}}) = \hat{t} \le t$. Since $2t \le n - 3$, we know $\dim(V_{\hat{t}} H_2) \ge \dim(V_{\hat{t}} H^{\hat{t}}) + 1$, then there exists a unit vector

$$f = V_{\hat{t}} \phi, \quad \phi \in H_2, \tag{2.31}$$

which is orthogonal to $V_{\hat{t}} H^{\hat{t}}$. Since $V_{\hat{t}} E_{\hat{t}}(A_0, u) \subset V_{\hat{t}}(H_0 + \mathrm{span}\{x^*\})$ and $E_{\hat{t}}(A_{\hat{t}}, u) = E_{\hat{t}}(V_{\hat{t}} A_0 V_{\hat{t}}^\top, u) = V_{\hat{t}} E_{\hat{t}}(A_0, u)$, we know $V_{\hat{t}} H_2 \perp V_{\hat{t}} E_{\hat{t}}(A_0, u) = E_{\hat{t}}(A_{\hat{t}}, u)$, hence $f \perp E_{\hat{t}}(A_{\hat{t}}, u)$.

Consider $P = E_{\hat{t}}(A_{\hat{t}}, u) + V_{\hat{t}} H^{\hat{t}}$, $Q = P + \mathrm{span}\{v\}$, and $S = P + \mathrm{span}\{f\}$. Here $f \perp P$, so there exists an orthogonal $W \in \mathbb{R}^{n \times n}$ such that

$$\begin{aligned} &Wx = x \quad \forall x \in P \\ &WS \supset Q; \end{aligned} \tag{2.32}$$

here W can be found because $\dim(S) \le 2\hat{t} + 2 \le 2t + 2 \le n - 1$.

Since $v \perp E_{\hat{t}}(A_{\hat{t}}, u)$, $W|_{E_{\hat{t}}(A_{\hat{t}}, u)} = I_n|_{E_{\hat{t}}(A_{\hat{t}}, u)}$ and $E_{\hat{t}}(A_{\hat{t}}, u) \perp (V_{\hat{t}} H^{\hat{t}} + \mathrm{span}\{f\})$, we have $W(V_{\hat{t}} H^{\hat{t}} + \mathrm{span}\{f\}) = V_{\hat{t}} H^{\hat{t}} + \mathrm{span}\{v\}$.

Now define

$$H^{\hat{t}+1} = H^{\hat{t}} + \mathrm{span}\{\phi\}, \quad V_{\hat{t}+1} = W V_{\hat{t}}.$$

Now we prove that $\mathcal{R}^*(\hat{t}+1, V_{\hat{t}+1}, H^{\hat{t}+1})$ holds. Clearly, $V_{\hat{t}+1}$ is orthogonal, $\dim(H^{\hat{t}+1}) = \hat{t} + 1$, and

$H^{\hat{t}+1} \subset H_2$. Since $u \in E_{\hat{t}}(A_{\hat{t}}, u) \subset P$, by (2.32), $V_{\hat{t}+1}u = WV_{\hat{t}}u = Wu = u$.

We then show that the first $(\hat{t}+1)$-st points at which $\mathbb{M}$ queries as applied to $(A_{\hat{t}+1}, u) = (V_{\hat{t}+1}A_0V_{\hat{t}+1}^\top, u) = (WA_{\hat{t}}W^\top, u)$ belong to $E_{\hat{t}}(A_{\hat{t}+1}, u) + V_{\hat{t}+1}H^{\hat{t}+1}$. Note that

$$\begin{aligned} E_{\hat{t}}(A_{\hat{t}+1}, u) + V_{\hat{t}+1}H^{\hat{t}+1} &= W[E_{\hat{t}}(A_{\hat{t}}, u) + V_{\hat{t}}(H^{\hat{t}} + \text{span}\{\phi\})] \\ &= W(P + \text{span}\{f\}) \\ &\supset Q. \end{aligned}$$

The first equality follows from the definition of $A_{\hat{t}+1}$, $V_{\hat{t}+1}$, and $H^{\hat{t}+1}$. The second equality follows from the definition of P and (2.31). The last inclusion follows from (2.32). It then suffices to show that the query points all lie in Q, so we prove that,

$$\begin{aligned} &\text{the first } \hat{t} \text{ points at which } \mathbb{M} \text{ queries as applied to problem } (A_{\hat{t}+1}, u) \text{ are} \\ &x(1), \ldots, x(\hat{t}), \text{ the points at which } \mathbb{M} \text{ queries as applied to problem } (A_{\hat{t}}, u). \end{aligned} \tag{2.33}$$

It suffices to show

$$A_{\hat{t}+1}x(i) = A_{\hat{t}}x(i) \text{ for } i = 1, \ldots, \hat{t}. \tag{2.34}$$

By induction assumption $\mathcal{R}^*(\hat{t}, V_{\hat{t}}, H^{\hat{t}})$, for $i \leq \hat{t}$, $x(i) \in E_{\hat{t}-1}(A_{\hat{t}}, u) + V_{\hat{t}}H^{\hat{t}} \subset P$, so, follows from (2.32), $A_{\hat{t}+1}x(i) = WA_{\hat{t}}W^\top x(i) = WA_{\hat{t}}x(i)$. Further, since $A_{\hat{t}}E_{\hat{t}-1}(A_{\hat{t}}, u) \subset E_{\hat{t}}(A_{\hat{t}}, u)$ and $A_{\hat{t}}V_{\hat{t}}H^{\hat{t}} = V_{\hat{t}}A_0H^{\hat{t}} = \{0\} \subset V_{\hat{t}}H^{\hat{t}}$, we have that $A_{\hat{t}}x(i) \in E_{\hat{t}}(A_{\hat{t}}, u) + V_{\hat{t}}H^{\hat{t}} = P$, which together with (2.32) implies that $WA_{\hat{t}}x(i) = A_{\hat{t}}x(i)$, so (2.34) is shown.

Given that method $\mathbb{M}$ is deterministic, by (2.33) and (2.34), the point defined as the $(\hat{t}+1)$-st query at which $\mathbb{M}$ applies to $(A_{\hat{t}}, u)$ is at the same time the $(\hat{t}+1)$-st query at which $\mathbb{M}$ applies to $(A_{\hat{t}+1}, u)$. Hence $x(\hat{t}+1) = u + v \in E_{\hat{t}}(A_{\hat{t}}, u) + \text{span}\{v\} \subset Q$, and for $i \in \{1, \ldots, \hat{t}\}$,

$$x(i) \in E_{\hat{t}-1}(A_{\hat{t}}, u) + V_{\hat{t}}H^{\hat{t}} \subset E_{\hat{t}}(A_{\hat{t}}, u) + V_{\hat{t}}H^{\hat{t}} = P = W[E_{\hat{t}}(A_{\hat{t}}, u) + V_{\hat{t}}H^{\hat{t}}].$$

hence for $i \in \{1, \ldots, \hat{t}+1\}$,

$$\begin{aligned} x(i) &\in W[E_{\hat{t}}(A_{\hat{t}}, u) + V_{\hat{t}} H^{\hat{t}} + \operatorname{span}\{f\}] \\ &= E_{\hat{t}}(A_{\hat{t}+1}, u) + W V_{\hat{t}} H^{\hat{t}+1} \\ &= E_{\hat{t}}(A_{\hat{t}+1}, u) + V_{\hat{t}+1} H^{\hat{t}+1}. \end{aligned}$$

Here the first equality follows because $V_{\hat{t}} H^{\hat{t}} + \operatorname{span}\{f\} = V_{\hat{t}}(H^{\hat{t}} + \operatorname{span}\{\phi\})$ due to (2.31), and $W E_{\hat{t}}(A_{\hat{t}}, u) = W V_{\hat{t}} E_{\hat{t}}(A_0, u) = V_{\hat{t}+1} E_{\hat{t}}(A_0, u) = E_{\hat{t}}(A_{\hat{t}+1}, u)$. Step (i) is hence complete.

Step (ii) we prove the main lemma. Let t (≥ 0) satisfy the premise of the lemma and $\mathbb{M}$ be a method. Without changing any of the first t points at which $\mathbb{M}$ queries as applied to (A, u), assume with out loss of generality that the $(\hat{t}+1)$-st point $x_{A,u}(\hat{t}+1)$ at which $\mathbb{M}$ queries is exactly the t-th approximate solution $x_{\mathbb{M}}(t, A, u)$.

By the proposition in Step (i), we can find orthogonal $V_{t+1} \in \mathbb{R}^{n \times n}$, $V_{t+1} u = u$, a $(t+1)$-dimensional subspace $H^{t+1} \subset H_2$, such that

$$\begin{aligned} x_{V_{t+1} A_0 V_{t+1}^\top, u}(t+1) &= x_{\mathbb{M}}(t, V_{t+1} A_0 V_{t+1}^\top, u) \\ &\in E_t(V_{t+1} A_0 V_{t+1}^\top, u) + V_{t+1} H^{t+1} \\ &\subset E_t(V_{t+1} A_0 V_{t+1}^\top, u) + V_{t+1} H_2, \end{aligned}$$

which shows $\mathcal{R}(t, V)$ for $V = V_{t+1}$. □

2.5 Conclusion

In this chapter we study the problem of computing a matrix kernel projection of a vector, when the matrix is not known, but its matrix-vector oracle is accessible. We studied this problem from three perspectives: the control perspective, the optimization perspective, and the linear algebra perspective. From the control perspective, we find that the best known method by [16] contributes a non-diagonalizable Jordan block for each pair of repeated eigenvalue in their update system, so to obtain their smallest linear convergence rate, an additional logarithmic term would appear in the worst-case; from the optimization perspective, we designed a novel accelerated gradient method, yet its convergence is only optimal in the order of linear convergence; from the linear algebra perspective,

we reviewed that the known study of exact oracle complexity for solving a linear equation [19] provided with the same matrix-vector oracle, yet this result is not readily applicable to our problem of interest. Based on our observations from the three perspectives, under a linear-span assumption, we propose a novel iterative method which attains the exact oracle complexity for our problem of interest. The analysis of our method does not require knowledge on the total number of iterations. In the realm of general methods, we provide an exact lower complexity bound with the assumption that $n - 3 \geq 2t$, where t is the number of matrix-vector oracle calls, and n is the dimension of our problem.

Chapter 3

Gradient norm minimization through a gradient extrapolation method

In this chapter, we propose to study the gradient extrapolation method in [11]. While it is developed for minimizing function value in convex smooth optimization problems, we show that it can be adapted to solve gradient norm minimization problems as well. Moreover, we are able to achieve the optimal gradient evaluation complexity for gradient norm minimization.

3.1 Introduction

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

$$f^* := \min_{x \in \mathbb{R}^n} f(x). \tag{3.1}$$

Here we assume that f is convex and smooth, i.e., $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$. Our goal is to compute an ε-approximate solution $x \in \mathbb{R}^n$ such that $\|\nabla f(x)\|^2 \leq \varepsilon$, where and throughout this chapter $\|\cdot\|$ is the Euclidean norm. Throughout this chapter, we refer to the aforementioned problem as the gradient

norm minimization problem. Complexity analysis of first-order methods for solving such problem is among the traditional research focuses of nonlinear optimization. In fact, checking whether the gradient norm satisfies the accuracy threshold $\|\nabla f(x)\|^2 \leq \varepsilon$ is one of the most widely used stopping criterion in practice. Other commonly used stopping criterion in theoretical studies include function value difference $f(x) - f^* \leq \varepsilon$ and distance to optimal solution $\|x - x^*\|^2 \leq \varepsilon$. However, the latter two criteria are less useful in practice, since accessing the knowledge of f^* and x^* is usually as difficult as solving the original problem itself.

There has been a rich body of literature on the complexity analysis of first-order methods for gradient norm minimization. The lower complexity bound for gradient norm minimization is known to be order $\mathcal{O}\left(1/\varepsilon^{1/4}\right)$ (see [19] and [4]). There have been many attempts designing optimal first-order methods with a match upper complexity bound. For example, in [24], a monotone convergence accelerated gradient method is proposed, which is able to compute an ε-solution with at most $\mathcal{O}(1/\varepsilon^{1/3})$ iterations. By perturbing the objective function f of to $f_\delta(x) := f(x) + (\delta/2)\|x - x_0\|^2$ and minimizing the perturbed function f_δ instead, in [24] one other first-order method is proposed with $\mathcal{O}\left(\ln(1/\varepsilon)/\varepsilon^{1/4}\right)$ complexity, which is sub-optimal with an extra logarithm multiple. In [10] a first-order method called the optimized gradient method for gradient norm minimization (OGM-G) is proposed, which computes an ε-solution with at most $\mathcal{O}(\sqrt{(f(x_0) - f(x^*))/\varepsilon})$ iterations. The convergence analysis of [10] is based on the performance enhancement program introduced in [7], which is a computer-aided proof system for analyzing first-order algorithm complexity. An alternate proof using potential functions is later developed in [6]. Later, in [25], it is pointed out that by a two-phase algorithm design, namely, running $\lceil N/2 \rceil$ steps of any optimal method for minimizing function value difference and $\lceil N/2 \rceil$ steps of OGM-G, where $N = \mathcal{O}(1/\varepsilon^{1/4})$, one actually is able to obtain the optimal $\mathcal{O}(1/\varepsilon^{1/4})$ complexity for gradient norm minimization. Such observation makes OGM-G in [10] the first optimal first-order method for gradient norm minimization. However, in the OGM-G iterations, the total number of iteration N need to be specified in advance, and the knowledge on the exact value of Lipschitz constant L_f is required. Such problems are addressed in [12], in which an optimal and parameter free algorithm with $\mathcal{O}(1/\varepsilon^{1/4})$ complexity is proposed through an accumulative regularization technique. To the best of our knowledge, the aforementioned methods are the only known resolutions for achieving the $\mathcal{O}(1/\varepsilon^{1/4})$ complexity in the literature. To summarize, at present, one should either adopt a two-phase algorithm design structure suggested in [25] or the accumulative regularization technique developed in [12].

However, there are still two remaining issues concerning the two possible resolutions above for achieving the $\mathcal{O}(1/\varepsilon^{1/4})$ complexity. For the OGM-G type algorithms [10, 6, 25], current results in the literature all require the two-phase algorithm design that run two different algorithms consecutively. For the accumulative regularization technique developed in [12], although the proposed algorithm no longer has the two-phase structure, in each iteration it needs to call one other algorithm as subroutine to solve its subproblem. The remaining issues motivates us to study the following research question concerning gradient norm minimization: can we obtain the optimal gradient evaluation complexity for gradient norm minimization by one uniform, single-loop algorithm without any subroutine?

In this chapter, we provide a partial answer to the above research question. Specifically, we show that the gradient extrapolation method (GEM) designed previously for function value difference minimization [15] (see also [11]) can be adapted for solving the gradient norm minimization problem. Specifically, given the maximum number of iterations N, we show that by running $\lceil N/2 \rceil$ iterations of GEM using the parameters described in [15] and $\lceil N/2 \rceil$ iterations of GEM using a novel set of parameters, we are able to achieve the $\mathcal{O}(1/\varepsilon^{1/4})$ gradient evaluation complexity for the gradient norm minimization problem. Although our proposed resolution does not solve the aforementioned research question completely (since we have to use two sets of parameters when applying GEM), it is using one uniform algorithm and does not require any subroutine.

This chapter is organized as follows. In Section 3.2, we describe GEM and its convergence analysis for computing an approximate solution such that $f(x_N) - f(x^*) \leq \varepsilon$. The results have already been developed in [15] and we are including this section for the sake of completeness. In Section 3.3, we prove a new result that GEM can be described alternatively as the linear span of previous gradients under certain appropriate choice of parameters. With the help of such equivalence, we are then able to show in Section 3.4 a novel result that GEM with certain choice of parameters is able to compute an approximate solution to the gradient norm minimization problem with at most $\mathcal{O}(\sqrt{(f(x_0) - f(x^*))/\varepsilon})$ iterations. Consequently, we can apply the observation concerning two-phase algorithm design in [25] to achieve the $\mathcal{O}(1/\varepsilon^{1/4})$ gradient evaluation complexity. Here in the two-phase algorithm design we are calling GEM uniformly, only with two different sets of parameters.

3.2 The gradient extrapolation method

In this section, we state the parameter choice and its convergence analysis of GEM for function value difference minimization. The content of of this section is all based on the description of GEM in Section 5.2.1 of [11]; we only add this section for the sake of completeness. Recall that in Section 1.2.2, accelerated gradient descent method is described with $\mathcal{O}\left(1/\sqrt{\varepsilon}\right)$ complexity for function value difference minimization. However, we are able to observe from Algorithm 1.2.2 that the gradient evaluation and the output approximate solution of accelerated gradient descent method are performed on different points $\underline{x}_t$ and $\overline{x}_t$. Since our goal is gradient norm minimization throughout this chapter, it is imperative that we perform gradient evaluation at the output approximate solution. Consequently, we would like to study optimal first-order methods whose output approximate solution is also the same point for gradient evaluation. GEM is such an optimal method for function value difference minimization.

The GEM algorithm is listed below in Algorithm 3.2.7. With appropriately chosen parameters, we will show that an ε-approximate solution such that $f(x_N) - f(x^*) \leq \varepsilon$ can be obtained within $\mathcal{O}\left(1/\sqrt{\varepsilon}\right)$ gradient evaluations.

Algorithm 3.2.7 The gradient extrapolation method (GEM)

Require: Initial point $u_0 = x_0 \in \mathbb{R}^n$, $g_0 = \nabla f(x_0)$, $g_{-1} = g_0$, $\xi_t,\ \eta_t,\ \tau_t \geq 0$ for $t = 1, 2, \ldots, N$

for $t = 1, 2, \ldots, N$ **do**

$$\hat{g}_t = \xi_t(g_{t-1} - g_{t-2}) + g_{t-1}, \tag{3.2}$$
$$u_t = u_{t-1} - \hat{g}_t/\eta_t, \tag{3.3}$$
$$x_t = (u_t + \tau_t x_{t-1})/(1 + \tau_t), \tag{3.4}$$
$$g_t = \nabla f(x_t). \tag{3.5}$$

end for

Output x_N

In Algorithm 3.2.7, we denote the gradient of each iterate x_N as $g_t = \nabla f(x_t)$ as in (3.5). We first perform a gradient extrapolation step (3.2) to compute $\hat{g}_t$ from gradients g_{t-1} and g_{t-2} of two previous iterations. Next, in (3.3), we perform a gradient descent step along the direction of $\hat{g}_t$. The computed point u_t is then combined with the previous approximate solution x_{t-1} in (3.4); such

convex combination becomes the new approximate solution x_t.

Now we will start to describe the convergence analysis of Algorithm 3.2.7 for solving problem (3.1) to obtain an ε-approximate solution x_N such that $f(x_N) - f^* \leq \varepsilon$. The following proposition shows that when solving (3.1) by GEM, if the parameters $\{\eta_t\}_{t=1}^N$, $\{\tau_t\}_{t=1}^N$ and $\{\xi_t\}_{t=1}^N$ in Algorithm 3.2.7 satisfy conditions described in (3.6) to (3.10), then we are able to estimate an upper bound of the function value difference $f(x_N) - f^*$ at the output approximate solution.

Proposition 3.2.1. *Let f be a function such that $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$, $\{x_t\}_{t=0}^N$ be the iterations of Algorithm 3.2.7 applied for solving* (3.1)*, and θ_t are nonnegative constants for $t = 1, \ldots, N$. Suppose the parameters $\{\eta_t\}_{t=1}^N$, $\{\tau_t\}_{t=1}^N$ and $\{\xi_t\}_{t=1}^N$ in Algorithm 3.2.7 satisfy $\tau_1 = 0$ and*

$$\theta_{t-1} = \xi_t \theta_t, \ t = 2, \ldots, N, \tag{3.6}$$

$$\theta_t \eta_t \leq \theta_{t-1} \eta_{t-1}, \ t = 2, \ldots, N, \tag{3.7}$$

$$\theta_t \tau_t = \theta_{t-1}(1 + \tau_{t-1}), \ t = 2, \ldots, N, \tag{3.8}$$

$$\xi_t L_f \leq \tau_{t-1} \eta_t, \ t = 3, \ldots, N, \tag{3.9}$$

$$2L_f \leq \tau_N \eta_N, \tag{3.10}$$

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

$$\begin{aligned} &\theta_N(1 + \tau_N)(f(x_N) - f(x)) + \frac{\theta_N \eta_N}{4} \|u_N - x\|^2 \\ \leq &\frac{\theta_1 \eta_1}{2} \|x_0 - x\|^2 + \frac{\theta_1}{2} \left(\frac{\xi_2 L_f^2}{\eta_2} - \eta_1 \right) \|u_1 - u_0\|^2. \end{aligned} \tag{3.11}$$

Proof. In Algorithm 3.2.7, since $u_t = u_{t-1} - \hat{g}_t/\eta_t$ and $\hat{g}_t = \xi_t(g_{t-1} - g_{t-2}) + g_{t-1}$, we have

$$\begin{aligned} \langle \xi_t(g_{t-1} - g_{t-2}) + g_{t-1}, u_t - x \rangle = &\eta_t \langle u_{t-1} - u_t, u_t - x \rangle \\ = &\frac{\eta_t}{2} \|x - u_{t-1}\|^2 - \frac{\eta_t}{2} \|x - u_t\|^2 - \frac{\eta_t}{2} \|u_t - u_{t-1}\|^2 \end{aligned} \tag{3.12}$$

for any $x \in \mathbb{R}^n$. Recalling that f is convex smooth, $x_t = (u_t + \tau_t x_{t-1})/(1 + \tau_t)$ and $g_t = \nabla f(x_t)$ in

Algorithm 3.2.7, we are able to derive from (3.12) and Lemma 1.1.4 that

$$\begin{aligned}(1+\tau_t)f(x_t)-f(x) \leq & (1+\tau_t)f(x_t)-(f(x_t)+\langle g_t, x-x_t\rangle)\\ =&\tau_t[f(x_t)-\langle g_t, x_t-x_{t-1}\rangle]-\langle g_t, x-u_t\rangle\\ \leq &-\frac{\tau_t}{2L_f}\|g_t-g_{t-1}\|^2+\tau_t f(x_{t-1})-\langle g_t, x-u_t\rangle,\ \forall x\in\mathbb{R}^n.\end{aligned} \tag{3.13}$$

Combining (3.12) with (3.13), we obtain that

$$\begin{aligned}&(1+\tau_t)f(x_t)-f(x)\\ \leq &-\frac{\tau_t}{2L_f}\|g_t-g_{t-1}\|^2+\tau_t f(x_{t-1})+\langle g_t-g_{t-1}-\xi_t(g_{t-1}-g_{t-2}), u_t-x\rangle\\ &+\frac{\eta_t}{2}\|x-u_{t-1}\|^2-\frac{\eta_t}{2}\|x-u_t\|^2-\frac{\eta_t}{2}\|u_t-u_{t-1}\|^2,\ \forall x\in\mathbb{R}^n.\end{aligned} \tag{3.14}$$

Multiplying both sides of (3.14) with θ_t and summing up from $t=1,\ldots,N$, we have

$$\begin{aligned}\sum_{t=1}^N\theta_t(1+\tau_t)f(x_t)-\sum_{t=1}^N\theta_t f(x)\leq &-\sum_{t=1}^N\frac{\theta_t\tau_t}{2L_f}\|g_t-g_{t-1}\|^2+\sum_{t=1}^N\theta_t\tau_t f(x_{t-1})\\ &+\sum_{t=1}^N\theta_t\langle g_t-g_{t-1}-\xi_t(g_{t-1}-g_{t-2}), u_t-x\rangle\\ &+\sum_{t=1}^N\theta_t\left[\frac{\eta_t}{2}\|x-u_{t-1}\|^2-\frac{\eta_t}{2}\|x-u_t\|^2-\frac{\eta_t}{2}\|u_t-u_{t-1}\|^2\right]\end{aligned} \tag{3.15}$$

for any $x\in\mathbb{R}^n$. We make a few observations in the above relation. First, recalling (3.6) and the fact that $g_{-1}=g_0$, we have that

$$\begin{aligned}&\sum_{t=1}^N\theta_t\langle g_t-g_{t-1}-\xi_t(g_{t-1}-g_{t-2}), u_t-x\rangle\\ =&\theta_N\langle g_N-g_{N-1}, u_N-x\rangle-\sum_{t=2}^N\theta_t\xi_t\langle g_{t-1}-g_{t-2}, u_t-u_{t-1}\rangle,\ \forall x\in\mathbb{R}^n.\end{aligned} \tag{3.16}$$

Second, by inequality (3.7) we have

$$\sum_{t=1}^N\theta_t\left[\frac{\eta_t}{2}\|x-u_{t-1}\|^2-\frac{\eta_t}{2}\|x-u_t\|^2\right]\leq\frac{\theta_1\eta_1}{2}\|x-u_0\|^2-\frac{\theta_N\eta_N}{2}\|x-u_N\|^2,\ \forall x\in\mathbb{R}^n. \tag{3.17}$$

Third, by equality (3.8) we can derive that

$$\sum_{t=1}^{N} \theta_t[(1+\tau_t)f(x_t) - \tau_t f(x_{t-1})] = \theta_N(1+\tau_N)f(x_N) - \theta_1\tau_1 f(x_0). \tag{3.18}$$

Fourth, we can also obtain from (3.8) that

$$\sum_{t=1}^{N} \theta_t = \sum_{t=2}^{N}(\theta_t\tau_t - \theta_{t-1}\tau_{t-1}) + \theta_N = \theta_N(1+\tau_N) - \theta_1\tau_1. \tag{3.19}$$

Thus with $\tau_1 = 0$, applying our observations (3.16), (3.17), (3.18) and (3.19) to (3.15) we have

$$\begin{aligned}
&\theta_N(1+\tau_N)(f(x_N) - f(x)) \\
\leq& -\theta_2\xi_2\langle g_1 - g_0, u_2 - u_1\rangle - \frac{\theta_2\eta_2}{2}\|u_2 - u_1\|^2 - \frac{\theta_1\eta_1}{2}\|u_1 - u_0\|^2 \\
&- \sum_{t=3}^{N}\left[\frac{\theta_{t-1}\tau_{t-1}}{2L_f}\|g_{t-1} - g_{t-2}\|^2 + \theta_t\xi_t\langle g_{t-1} - g_{t-2}, u_t - u_{t-1}\rangle + \frac{\theta_t\eta_t}{2}\|u_t - u_{t-1}\|^2\right] \\
&- \theta_N\left[\frac{\tau_N}{2L_f}\|g_N - g_{N-1}\|^2 - \langle g_N - g_{N-1}, u_N - x\rangle + \frac{\eta_N}{2}\|x - u_N\|^2\right] + \frac{\theta_1\eta_1}{2}\|x - u_0\|^2.
\end{aligned} \tag{3.20}$$

for arbitrary $x \in \mathbb{R}^n$. Observe that for any scalar $a > 0$, $b \in \mathbb{R}$ and vector u, v, we have $b\langle u, v\rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a)$. By this simple relation and (3.6), (3.9) and (3.10), we are able to obtain the following three results:

$$-\theta_2\xi_2\langle g_1 - g_0, u_2 - u_1\rangle - \frac{\theta_2\eta_2}{2}\|u_2 - u_1\|^2 \leq \frac{\theta_2\xi_2^2}{2\eta_2}\|g_1 - g_0\|^2 \leq \frac{\theta_1\xi_2 L_f^2}{2\eta_2}\|u_1 - u_0\|^2, \tag{3.21}$$

$$\begin{aligned}
&-\sum_{t=3}^{N}\left[\frac{\theta_{t-1}\tau_{t-1}}{2L_f}\|g_{t-1} - g_{t-2}\|^2 + \theta_t\xi_t\langle g_{t-1} - g_{t-2}, u_t - u_{t-1}\rangle + \frac{\theta_t\eta_t}{2}\|u_t - u_{t-1}\|^2\right] \\
\leq& \sum_{t=3}^{N}\frac{\theta_t}{2}\left(\frac{\xi_t L_f}{\tau_{t-1}} - \eta_t\right)\|u_t - u_{t-1}\|^2 \leq 0,
\end{aligned} \tag{3.22}$$

$$\begin{aligned}
&-\theta_N\left[\frac{\tau_N}{2L_f}\|g_N - g_{N-1}\|^2 - \langle g_N - g_{N-1}, u_N - x\rangle + \frac{\eta_N}{4}\|x - u_N\|^2\right] \\
\leq& \frac{\theta_N}{2}\left(\frac{L_f}{\tau_N} - \frac{\eta_N}{2}\right)\|x - u_N\|^2 \leq 0, \ \forall x \in \mathbb{R}^n.
\end{aligned} \tag{3.23}$$

The second inequality sign of (3.21) is obtained by the L_f-smoothness of f and the fact that $x_0 = u_0$, $x_1 = u_1$. Applying (3.21), (3.22) and (3.23) to (3.20), we conclude (3.11).

□

Proposition 3.2.1 shows an upper bound estimate of function value difference $f(x_N) - f^*$. The following theorem provides a specific set of parameters such that all conditions in Proposition 3.2.1 are satisfied.

Theorem 3.2.1. *Let f be a function such that $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$ and $\{x_t\}_{t=0}^N$ be the iterations of Algorithm 3.2.7 for solving* (3.1)*. Let x^* be an optimal solution of* (3.1)*, and the parameters $\{\eta_t\}_{t=1}^N$, $\{\tau_t\}_{t=1}^N$ and $\{\xi_t\}_{t=1}^N$ in Algorithm 3.2.7 are set to $\eta_t = 6L_f/t$, $\tau_t = (t-1)/2$ and $\xi_t = (t-1)/t$, for $t = 1, \ldots, N$. Then we have*

$$f(x_N) - f^* \leq \frac{6L_f}{N(N+1)}\|x_0 - x^*\|^2. \tag{3.24}$$

Proof. Defining $\theta_t = t$ for $t = 1, \ldots, N$, (3.6) to (3.10) are satisfied. Thus by Proposition 3.2.1 with the described parameter settings in the theorem applied to (3.11), we conclude (3.24). □

By the above theorem, that to obtain an approximate solution for function value difference minimization such that $f(x_N) - f(x^*) \leq \varepsilon$, the number of iterations required by Algorithm 3.2.7 to solve (2.1) is upper bounded by $\mathcal{O}\left(\sqrt{6L_f\|x_0 - x^*\|^2/\varepsilon}\right)$. As discussed previously in Section 3.1, to obtain an ε-approximate solution that for gradient norm minimization with $\mathcal{O}(1/\varepsilon^{1/4})$ complexity, we may utilize a two-phase algorithm design as suggested in [25]. GEM could serve as an optimal algorithm for function value difference minimization in the first phase. However, to the best of our knowledge, no GEM parameters have been proposed for solving the gradient norm minimization problem. Therefore, it would be interesting if we are able to find a set of parameters of GEM such that $\|\nabla f(x_N)\|^2 \leq \varepsilon$ when the number of iterations N is greater than $\mathcal{O}\left(1/\sqrt{\varepsilon}\right)$. If such set of parameters can be discovered, we have a uniform algorithm (namely GEM) framework for solving the gradient norm minimization problem with optimal complexity. The remainder of this chapter is dedicated to the derivation of such set of parameters. We start in the following section by describing an alternate description of GEM, which is convenient for our analysis on gradient norm minimization.

3.3 GEM as linear span of gradients

In this section, we show that GEM can be visualized as the linear span of gradients. Specifically, we can describe GEM alternatively as in Algorithm 3.3.8. Our description in Algorithm 3.3.8 is not novel; rather, it has been studied previously in the literature (see, e.g., [9, 24, 7, 6, 10]). Indeed, the algorithm structure in Algorithm 3.3.8 is the necessary tool utilized in [10] for discovering the first algorithm with $\mathcal{O}(1/\varepsilon^{1/4})$ complexity for computing an approximate solution x such that $\|\nabla f(x)\|^2 \leq \varepsilon$. We will establish the equivalence between Algorithms 3.2.7 and 3.3.8 for certain parameters. With the help of such equivalence, we are then able to show in the next section that GEM with certain choice of parameters is able to achieve the aforementioned $\mathcal{O}(1/\varepsilon^{1/4})$ upper complexity bound for gradient norm minimization.

Algorithm 3.3.8 A linear span type algorithm description for solving problem (3.1)

Require: Initial point $x_0 \in \mathbb{R}^n$, maximum number of iterations N
  for $t = 1, \ldots, N$ **do**
    Compute

$$g_{t-1} = \nabla f(x_{t-1}),$$
$$x_t = x_{t-1} - \frac{1}{L_f} \sum_{k=0}^{t-1} h_{t,k} g_k.$$

  end for
  Output approximate solution x_N.

From the description of Algorithm 3.3.8, we can observe that the next iterate x_t is evaluated as a point in the set $x_{t-1} + \text{span}\{\nabla f(x_0), \ldots, \nabla f(x_{t-1})\}$. By induction, we may also conclude that x_t is also in the set $x_0 + \text{span}\{\nabla f(x_0), \ldots, \nabla f(x_{t-1})\}$. The goal of algorithm design is to choose appropriate parameters $h_{t,k}$ so that the algorithm achieves desirable convergence properties. In the case when f is a quadratic function (thus the optimization problem is equivalent to solving a linear system), such methods are known as Krylov subspace type methods in numerical linear algebra. Such linear span description is also commonly seen in algorithm design and complexity analysis of first-order methods (see, e.g., [24]). The format in the description of Algorithm 3.3.8 appears the first time in [7] for analyzing first-order methods for minimizing function values of unconstrained convex optimization. In [9], the $h_{t,k}$ parameters of Algorithm 3.3.8 are defined with the structure in (3.25) below. The following proposition shows that if $h_{t,k}$ are defined as in (3.25), then there exists

parameters of GEM that is equivalent to the linear span description in Algorithm 3.3.8.

Proposition 3.3.1. *For any function $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$ and initial point $x_0 \in \mathbb{R}^n$, Algorithms 3.2.7 and Algorithm 3.3.8 are equivalent and produce exactly the same sequence of approximate solutions $\{x_t\}_{t=1}^N$ under the choices of parameters described as follows. In Algorithm 3.3.8, the parameters $\{h_{t,k}\}$ satisfies*

$$h_{tk} = \begin{cases} 1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1}), & t = 1, ..., N, k = t-1 \\ (\alpha_t - \alpha_{t+1})(\gamma_{k+1} - \gamma_k), & t = 2, ..., N, k = 0, ..., t-2 \end{cases} \tag{3.25}$$

where α_t and γ_t are positive constants such that $\gamma_t \geq \gamma_{t-1}$ for all $t = 1, \ldots, N$. In Algorithm 3.2.7, the parameters τ_t, η_t and ξ_t satisfies $(1+\tau_t)(\alpha_t - \alpha_{t+1}) = \tau_{t-1}(\alpha_{t-1} - \alpha_t)$ and

$$\begin{aligned} \xi_t =& \frac{\alpha_t - \alpha_{t+1}}{(\alpha_{t-1} - \alpha_t)(1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1})) - (\alpha_t - \alpha_{t+1})}, \\ \eta_t =& \frac{L}{1+\tau_t} \frac{\alpha_{t-1} - \alpha_t}{(\alpha_t - \alpha_{t+1})} \xi_t. \end{aligned}$$

Proof. Applying (3.25) to iterates $x_t = x_{t-1} - \frac{1}{L}\sum_{k=0}^{t-1} h_{t,k} g_k$ of Algorithm 3.3.8, we have that

$$x_{t-1} - x_t - \frac{1}{L} g_{t-1} = \frac{\alpha_t - \alpha_{t+1}}{L} \sum_{k=0}^{t-1} (\gamma_{k+1} - \gamma_k) g_k, \ \forall t = 1, ..., N. \tag{3.26}$$

Note that (3.26) also implies that for all $t \geq 2$

$$\begin{aligned} & x_{t-1} - x_t - \frac{1}{L} g_{t-1} \\ =& \frac{\alpha_t - \alpha_{t+1}}{L} \left((\gamma_t - \gamma_{t-1}) g_{t-1} + \frac{L}{\alpha_{t-1} - \alpha_t} \left(x_{t-2} - x_{t-1} - \frac{1}{L} g_{t-2} \right) \right). \end{aligned} \tag{3.27}$$

Rearranging terms of (3.27), we obtain that

$$\begin{aligned} x_t =& \left(1 + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t} \right) x_{t-1} - \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t} \left(x_{t-2} - \frac{1}{L} g_{t-2} \right) \\ & - \frac{1}{L} (1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1})) g_{t-1}. \end{aligned} \tag{3.28}$$

Recalling that τ_t's are positive parameters such that $(1 + \tau_t)(\alpha_t - \alpha_{t+1}) = \tau_{t-1}(\alpha_{t-1} - \alpha_t)$, and

letting $\{u_t\}_{t=0}^N$ be vectors such that

$$x_t = \frac{\tau_t}{1+\tau_t} x_{t-1} + \frac{1}{1+\tau_t} u_t, \ \forall t = 1, ..., N, \tag{3.29}$$

Note that if u_t in (3.29) is equivalently defined as in (3.3) of Algorithm 3.2.7, then (3.29) is equivalent to (3.4) in Algorithm 3.2.7 and the equivalence of Algorithm 3.2.7 and Algorithm 3.3.8 is proved. We can observe from (3.29) that

$$\frac{\tau_t}{1+\tau_t} x_{t-1} = \frac{\tau_t}{1+\tau_t} \frac{\tau_{t-1}}{1+\tau_{t-1}} x_{t-2} + \frac{\tau_t}{1+\tau_t} \frac{1}{1+\tau_{t-1}} u_{t-1},$$

which is equivalent to

$$\begin{aligned} & x_t - \left(1 + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t}\right) x_{t-1} + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t} x_{t-2} \\ = & \left[\frac{\tau_t}{1+\tau_t} \frac{\tau_{t-1}}{1+\tau_{t-1}} - \left(1 + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t}\right) \frac{\tau_{t-1}}{1+\tau_{t-1}} + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t}\right] x_{t-2} \\ & + \frac{1}{1+\tau_t} u_t - \frac{1}{1+\tau_{t-1}} \left(1 + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t} - \frac{\tau_t}{1+\tau_t}\right) u_{t-1}. \end{aligned} \tag{3.30}$$

In the above equality, using the relation $(1+\tau_t)(\alpha_t - \alpha_{t+1}) = \tau_{t-1}(\alpha_{t-1} - \alpha_t)$ again, we can observe that

$$\frac{\tau_t}{1+\tau_t} \frac{\tau_{t-1}}{1+\tau_{t-1}} - \left(1 + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t}\right) \frac{\tau_{t-1}}{1+\tau_{t-1}} + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t} = 0,$$

and

$$\frac{1}{1+\tau_{t-1}} \left(1 + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t} - \frac{\tau_t}{1+\tau_t}\right) = \frac{1}{1+\tau_{t-1}} \left(1 + \frac{\tau_{t-1}}{1+\tau_t} - \frac{\tau_t}{1+\tau_t}\right) = \frac{1}{1+\tau_t}.$$

Hence (3.30) can be simplified as

$$x_t - \left(1 + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t}\right) x_{t-1} + \frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t} x_{t-2} = \frac{1}{(1+\tau_t)} (u_t - u_{t-1}). \tag{3.31}$$

Recall from (3.28) that the left-hand side of (3.31) can also be expressed as a linear combination of the previous gradients g_{t-1} and g_{t-2}. Combining (3.28) and (3.31), we may eliminate x_t, x_{t-1}, and

x_{t-2} and obtain that

$$\frac{1}{(1+\tau_t)}(u_t - u_{t-1}) = \frac{1}{L}\left(\frac{\alpha_t - \alpha_{t+1}}{\alpha_{t-1} - \alpha_t} g_{t-2} - (1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1}))g_{t-1}\right). \tag{3.32}$$

We will next show that the right-hand side of (3.32) can be expressed as $\hat{g}_t$ defined in (3.2) multiplied by a scalar. Letting ξ_t in Algorithm 3.2.7 be positive scalars such that

$$\frac{1+\xi_t}{\xi_1} = (1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1}))\frac{\alpha_{t-1} - \alpha_t}{\alpha_t - \alpha_{t+1}},$$

we are able to have ξ_t and $1 + \xi_t$ expressed explicitly as

$$\begin{aligned}
\xi_t =& \frac{\alpha_t - \alpha_{t+1}}{(\alpha_{t-1} - \alpha_t)(1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1})) - (\alpha_t - \alpha_{t+1})}, \\
1 + \xi_t =& \frac{(\alpha_{t-1} - \alpha_t)(1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1}))}{(\alpha_{t-1} - \alpha_t)(1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1})) - (\alpha_t - \alpha_{t+1})}.
\end{aligned}$$

Defining $\hat{g}_t = \xi_t(g_{t-1} - g_{t-2}) + g_{t-1}$ as in Algorithm 3.2.7, we can derive from (3.32) that

$$u_t - u_{t-1} = -\frac{1+\tau_t}{L}\frac{(\alpha_{t-1} - \alpha_t)(1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1})) - (\alpha_t - \alpha_{t+1})}{\alpha_{t-1} - \alpha_t}\hat{g}_t. \tag{3.33}$$

In Algorithm 3.2.7, u_t are defined as $u_t = u_{t-1} - \hat{g}_t/\eta_t$. Observe from (3.33) that if η_t are defined as

$$\begin{aligned}
\eta_t =& \frac{L}{1+\tau_t}\frac{\alpha_{t-1} - \alpha_t}{(\alpha_{t-1} - \alpha_t)(1 + (\alpha_t - \alpha_{t+1})(\gamma_t - \gamma_{t-1})) - (\alpha_t - \alpha_{t+1})} \\
=& \frac{L}{1+\tau_t}\frac{\alpha_{t-1} - \alpha_t}{(\alpha_t - \alpha_{t+1})}\xi_t,
\end{aligned}$$

then (3.33) can be expressed equivalently as in Algorithm 3.2.7. Hence (3.29) is equivalent to (3.4) in Algorithm 3.2.7 and the equivalence of Algorithm 3.2.7 and Algorithm 3.3.8 is proved. □

There has been several linear span type algorithms proposed in the literature for gradient norm minimization; see, e.g., [10, 6, 9]. With the results in the above proposition, we are now able to link such results to GEM parameters. In the following section, we will show that the results in [9] yield a set of GEM parameters for gradient norm minimization.

3.4 GEM for gradient norm minimization

With the help of Proposition 3.3.1, in this section we are able to derive GEM parameters for gradient norm minimization. Our results is based on the following proposition, a results derived in [9].

Proposition 3.4.1. *Let f be a function such that $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$ and $\{x_t\}_{t=0}^N$ be the iterations of Algorithm 3.3.8. Suppose that the parameters $\{h_{t,k}\}$ in Algorithm 3.3.8 are set to*

$$h_{ik} = \begin{cases} 1 + \dfrac{2(N-t+1)(N-t+2)(N-t+3)}{(N-k+1)(N-k+2)(N-k+3)}, \ t = 1, ..., N, k = t-1; \\ \dfrac{2(N-t+1)(N-t+2)(N-t+3)}{(N-k+1)(N-k+2)(N-k+3)}, \ t = 2, ..., N, k = 0, ..., t-2, \end{cases} \tag{3.34}$$

then we have

$$||g_N||^2 \leq \frac{6L_f}{(N+2)(N+3)}(f(x_0) - f(x^*)). \tag{3.35}$$

The above proposition describes a parameter setting of the linear span algorithm described in Algorithm 3.3.8 that solves the gradient norm minimization problem with $\mathcal{O}\left(\sqrt{L_f(f(x_0) - f(x^*))/\varepsilon}\right)$ complexity. Combining the results in the previous two sections, we can already obtain a two-phase algorithm with $\mathcal{O}(\sqrt{L_f\|x_0 - x^*\|}/\varepsilon^{1/4})$ complexity for gradient norm minimization. Specifically, the algorithms used in the first and second phases are GEM with parameters described in Theorem 3.2.1 and the linear span described Algorithm 3.3.8 with parameters in the above proposition. With the help of previously developed Proposition 3.3.1 on the equivalence of algorithms, we will show in Proposition 3.4.2 below that Algorithm 3.3.8 with parameters described in the above proposition is indeed equivalent to GEM. Consequently, running GEM with two sets of parameters will solve the gradient norm minimization problem with the optimal $\mathcal{O}(\sqrt{L_f\|x_0 - x^*\|}/\varepsilon^{1/4})$ complexity.

Proposition 3.4.2. *Let f be a function such that $f \in \mathcal{F}_{L}^{1,1}(\mathbb{R}^n)$ and $\{x_t\}_{t=0}^N$ be the iterations of Algorithm 3.2.7. If x^* is an optimal solution of* (1.3)*, and the parameters $\{\eta_t\}_{t=1}^N$, $\{\tau_t\}_{t=1}^N$ and*

$\{\xi_t\}_{t=1}^N$ *in Algorithm 3.2.7 are set to*

$$\begin{aligned}\xi_t =& \frac{N-t+1}{2N-2t+5},\\ \tau_t =& \frac{(4\tau_0 - N)(N+1)(N+2)(N+3)}{4(N-t+1)(N-t+2)(N-t+3)} + \frac{N-t}{4},\\ \eta_t =& \frac{L}{1+\tau_t}\frac{N-t+4}{2N-2t+5},\end{aligned} \tag{3.36}$$

then Algorithm 3.3.8 is equivalent to Algorithm 3.2.7.

Proof. Applying Proposition 3.3.1 to algorithms with parameters (3.36) and (3.34) respectively, with α_t and γ_k set to

$$\begin{aligned}\alpha_t =& \frac{1}{24}(N-t+1)(N-t+2)(N-t+3)(N-t+4),\\ \gamma_k =& \frac{6}{(N-k+2)(N-k+3)},\end{aligned}$$

we conclude the equivalence of Algorithm 3.3.8 and Algorithm 3.2.7. □

With the help of Theorem 3.2.1 and Proposition 3.4.2, we are now ready to prove that running GEM with two sets of parameters will yield the optimal $\mathcal{O}(\sqrt{L_f\|x_0 - x^*\|}/\varepsilon^{1/4})$ complexity for gradient norm minimization.

Theorem 3.4.1. *Suppose that the maximum number of iterations N is pre-specified. Let f be a convex smooth function such that $f \in \mathcal{F}_{L_f}^{1,1}(\mathbb{R}^n)$ and $\{x_t\}_{t=1}^N$ be iterates of Algorithm 3.2.7. When running GEM described in Algorithm 3.2.7, if we choose the parameters as in Theorem 3.2.1 during the first $\lceil N/2 \rceil$ steps, followed by the parameters described in Proposition 3.4.2 for the next $\lceil N/2 \rceil$ steps, then we have*

$$\|g_N\|^2 \leq \frac{576L_f^2\|x_0 - x^*\|^2}{(N+1)(N+3)(N+5)(N+7)}. \tag{3.37}$$

Proof. For the first $\lceil N/2 \rceil$ steps of GEM with parameters in Theorem 3.2.1, by the complexity bound (3.24) we have

$$f(x_{\lceil N/2 \rceil}) - f(x^*) \leq \frac{24L_f\|x_0 - x^*\|^2}{(N+1)(N+3)}. \tag{3.38}$$

Next, starting from $x_{N/2}$ and running $\lceil N/2 \rceil$ steps of GEM with parameters defined in Proposition 3.2.1, by the complexity bound (3.35) we have

$$||g_N||^2 \leq \frac{24L_f}{(N+5)(N+7)}(f(x_{\lceil N/2 \rceil}) - f(x^*)). \tag{3.39}$$

Combining (3.38) and (3.39), we conclude (3.37). □

According to the result (3.37) in the above theorem, in order to make sure that the output approximate solution x_N of GEM is an ε-approximate solution with $\|\nabla f(x_N)\|^2 \leq \varepsilon$, it suffices to set the total number of iterations N to

$$N \geq \mathcal{O}\left(\frac{\sqrt{L_f\|x_0 - x^*\|}}{\varepsilon^{1/4}}\right),$$

which has been proven to be optimal (see [19] and [4]) for gradient norm minimization. Remark that although we need to utilize two different sets of parameters in the first $\lceil N/2 \rceil$ iterations and the last $\lceil N/2 \rceil$ iterations, the optimal complexity bound is obtained by one uniform algorithm (GEM).

3.5 Conclusion

In this chapter, we show that the gradient extrapolation method (GEM), which was previously developed for function value minimization, can also be adapted to solve the gradient norm minimization problem. Specifically, we show that by running GEM with two sets of parameters, we are able to compute an ε-approximate solution such that $\|\nabla f(x_N)\|^2 \leq \varepsilon$ with at most $\mathcal{O}\left(\sqrt{L_f\|x_0 - x^*\|}/\varepsilon^{1/4}\right)$ gradient evaluations. Such complexity matches the lower complexity bound for gradient norm minimization. Although such optimal complexity can also be obtained by other first-order methods (e.g., [10, 6, 9, 12]), we provide in this chapter the first framework with one uniform algorithm (GEM) and no subroutine. One drawback of our proposed framework is that we need to apply two different sets of parameters in order to obtain the optimal complexity for gradient norm minimization. An interesting future research topic would be the design of a method that could achieve the optimal complexity through only one set of parameters and no subroutines.

Chapter 4

Conditional gradient methods for smooth functional constrained optimization

In this chapter we focus on projection-free algorithms for solving functional constrained smooth nonlinear optimization problems. Methods for solving convex and nonconvex functional constrained optimization problems are proposed respectively. We describe the problem of interest in detail and introduce the essential definitions in the following section.

4.1 Introduction

The problem of interest in this chapter is the following functional constrained smooth nonlinear optimization problem:

$$f^* := \min_{x \in X} f(x), \text{ s.t. } g_i(x) \leq 0, \ i = 1, \ldots, m. \tag{4.1}$$

Here $X \in \mathbb{R}^n$ is a compact set. We assume that an optimal solution x^* exists, the objective function f is L_f-smooth with respect to norm $\|\cdot\|$, and the constraint functions g_i are all L_g-smooth with respect to the same norm. Our goal is to design projection-free first-order methods to

compute numerical solutions with specified accuracy thresholds. We use the term "projection-free" to emphasize that our methods of interest should avoid computing projections onto the compact set X. We consider several different possible definitions of numerical solutions in terms of their associated accuracy threshold, which are listed below. In our definitions, for any real number u we denote $[u]_+ := \max\{u, 0\}$.

Definition 4.1.1. *We say that $\hat{x} \in X$ is an $(\varepsilon_f, \varepsilon_g)$-approximate solution to problem (4.1) if it satisfies $f(\hat{x}) - f^* \leq \varepsilon_f$ and $[g_i(\hat{x})]_+ \leq \varepsilon_g$ for all $i = 1, \ldots, m$.*

The $(\varepsilon_f, \varepsilon_g)$-approximate solution defined in Definition 4.1.1 ensures that the objective function value difference $f(\hat{x}) - f^*$ and the feasibility violation $[g_i(\hat{x})]_+$ are within the specified tolerances ε_f and ε_g, respectively. Clearly, when $\varepsilon_f = \varepsilon_g = 0$, $\hat{x}$ becomes an optimal solution to problem (4.1). We will study the complexity for computing an $(\varepsilon_f, \varepsilon_g)$-approximate solution to (4.1) when it is a convex optimization problem. When it is nonconvex, we will study the complexity of computing two different definitions of numerical solutions, as described below.

Definition 4.1.2. *We say that $\hat{x} \in X$ is an $(\varepsilon_f, \varepsilon_g)$-stationary point to problem (4.1) if it satisfies $[g_i(\hat{x})]_+ \leq \varepsilon_g$ for all $i = 1, \ldots, m$ and*

$$\langle \nabla f(\hat{x}), \hat{x} - x \rangle \leq \varepsilon_f, \ \forall x \in X \text{ s.t. } g_i(x) \leq 0 \text{ for all } i = 1, \ldots, m.$$

Definition 4.1.3. *We say that $\hat{x} \in X$ is an $(\varepsilon_f, \varepsilon_g)$-Fritz-John (FJ) point to problem (4.1) if it satisfies $[g_i(\hat{x})]_+ \leq \varepsilon_g$ for all $i = 1, \ldots, m$ and there exists nonnegative multipliers $\lambda_0, \ldots, \lambda_m$ such that $\sum_{i=0}^m \lambda_i = 1$ and*

$$\langle \lambda_0 \nabla f(\hat{x}) + \sum_{i=1}^m \lambda_i \nabla g_i(\hat{x}), \hat{x} - x \rangle \leq \varepsilon_f, \ \forall x \in X. \tag{4.2}$$

The $(\varepsilon_f, \varepsilon_g)$-stationary point defined in Definition 4.1.2 ensures that the Wolfe gap $\langle \nabla f(\hat{x}), \hat{x} - x \rangle$ and the feasibility violation $[g_i(\hat{x})]_+$ are within the specified tolerances ε_f and ε_g, respectively. When $\varepsilon_f = \varepsilon_g = 0$, $\hat{x}$ satisfies a first-order necessary optimality condition for problem (4.1). The $(\varepsilon_f, \varepsilon_g)$-FJ point defined in Definition 4.1.3 ensures that the Fritz-John condition violation

$\langle \lambda_0 \nabla f(\hat{x}) + \sum_{i=1}^m \lambda_i \nabla g_i(\hat{x}), \hat{x} - x \rangle$ and the feasibility violation $[g_i(\hat{x})]_+$ are within the specified tolerances ε_f and ε_g, respectively. When $\varepsilon_f = \varepsilon_g = 0$, $\hat{x}$ satisfies the first-order Fritz-John necessary optimality condition for problem (4.1). The FJ condition is weaker than the Karush-Kuhn-Tucker (KKT) necessary condition, which requires that $\lambda_0 \neq 0$ in (4.2). However, for an optimal solution to satisfy the first-order KKT necessary condition, the KKT theorem requires additional constraint qualification. On the other hand, the FJ theorem does not require any additional condition for an optimal solution to satisfy the FJ condition. It should also be pointed out that the FJ condition does not require that $\sum_{i=0}^m \lambda_i = 1$ as stated in Definition 4.1.3; rather, it only requires that $\lambda_0, \ldots, \lambda_m$ are not all zero. Consequently, if $\hat{x}$ satisfies the FJ condition with multipliers $\lambda_0, \ldots, \lambda_m$, then it also satisfies the FJ condition with multiplier $c\lambda_0, \ldots, c\lambda_m$ for any $c > 0$. However, for the definition of approximate FJ points, we need to regularize the multiplies $\lambda_0, \ldots, \lambda_m$ so that the tolerance ε_f can be properly enforced. Without this regularization, the multipliers could be scaled arbitrarily, making it impossible to evaluate the complexity of algorithms for obtaining the specified tolerance levels.

There has been previous literature on algorithm design and complexity analysis of first-order methods for solving problem (4.1). We briefly describe the existing results and related works below. For the simplicity of description, we denote $\varepsilon := \min\{\varepsilon_f, \varepsilon_g\}$ and state the complexity results only in terms of their order of dependence on ε.

We first focus on algorithms solving convex constrained optimization problems. Since The lower complexity bound of projection-free method for solving constrained optimization problems is currently unknown, to conjecture the ideal complexity, we can consider the projection-based method as well as unconstrained or affinely constrained problems. For convex optimization problem $\min_{x \in X} f(x)$ with no constraints, the lower gradient evaluations complexity bound for any projection-based methods in order to compute an ε-solution is $\mathcal{O}(1/\sqrt{\varepsilon})$ (see [19]). In [8], the lower complexity bound for solving linear objective optimization subproblems for projection-free methods is $\mathcal{O}(1/\varepsilon)$. Also, [26] shows that the lower complexity bound of first-order methods for solving an affinely constrained convex problems is of order $\mathcal{O}(1/\varepsilon)$. As we mentioned before, for functional constrained problems, the lower gradient evaluation complexity has not yet been studied in the literature. But we can conjecture from the above three lower complexity bound that our desired upper complexity bound is similar. In [28], a projection-based method, Accelerated Constrained Gradient Descent (ACGD) with sliding is proposed with $\mathcal{O}(1/\sqrt{\varepsilon})$ complexity for solving convex functional

constrained optimization problems. Observe that such lower complexity bound is identical to the lower complexity bound of projection-based methods for solving unconstrained convex optimization problem, which is inspiring for designing projection-free methods for solving functional constrained methods. We conjecture that the lower complexity bound for solving unconstrained convex optimization problem might be able to achieved also by projection-free methods for solving constrained convex optimization problems. Going back to our focus on designing projection-free methods, there are some existing results of projection-free methods for solving convex functional constrained problems. A level conditional gradient method is proposed in [5] for solving convex functional constrained optimization problems with a $\mathcal{O}(\log(1/\varepsilon)/\varepsilon^2)$ complexity. In [13], a constraint extrapolated condition gradient (Co-exCG) method is proposed for solving both smooth and structured nonsmooth function constrained convex optimization with a $\mathcal{O}(1/\varepsilon^2)$ complexity. As we have conjectured, we speculate that such complexity results can be further improved.

Next, we move on to the case when the objective function is nonconvex. We will discuss when the constraint functions are convex and nonconvex respectively. Since the lower complexity bound of projection-free methods for solving such problems are not yet studied, we conjecture from the complexity property of projection-based method for solving unconstrained problems or functional constrained problems. The description of [2] and [3] gives the lower complexity bound for solving a nonconvex optimization problem without constraints as $\mathcal{O}(1/\varepsilon^2)$. However, the lower complexity bound for functional constrained nonconvex optimization problems has not yet been studied, no matter the functional constraints are convex or nonconvex. A related result is established when the constraints are affine. In [17], the lower complexity complexity bound of a projection-based first-order method for solving affinely constrained nonconvex problems are proved to be $\mathcal{O}(1/\varepsilon^2)$. We can conjecture from the above lower complexity bound that the $\mathcal{O}(1/\varepsilon^2)$ complexity can also be achieved by projection-free method for solving nonconvex constrained problems. It is described in [5] and [1], the study on algorithms for solving nonconvex functional constrained problems is scarce. In [5], the Level Inexact Proximal Point (IPP-LCG) method and the Direct Nonconvex Conditional Gradient (DNCG) method are introduced for solving nonconvex constrained optimization problems. The IPP-LCG convert the nonconvex problem into a series of convex subproblems and obtain an $\mathcal{O}(\log(1/\varepsilon)/\varepsilon^3)$ complexity. The DNCG is a single-loop projection-free method with $\mathcal{O}(1/\varepsilon^4)$ complexity. As we have conjectured, we speculate that such complexity results can be further improved.

Now with the related studies are described, the structure of this chapters is as follows.

In Section 4.2, we focus on projection-free algorithms for solving convex optimization problems with convex constrained. A constrained conditional gradient (CCG) method is introduced such that the gradient evaluation complexity and the linear objective optimization complexity are both of order $\mathcal{O}(1/\varepsilon)$. Then motivated by the projection-based accelerated constrained gradient descent method with sliding (ACGD-S) proposed in [28], we propose an CCG with sliding (CCG-S) method for solving convex functional constrained problems with an $\mathcal{O}(1/\sqrt{\varepsilon})$ gradient evaluation complexity and $\mathcal{O}(1/\varepsilon)$ linear objective optimization complexity.

In Section 4.3, we develop algorithms solving nonconvex optimization problems with convex and nonconvex constraints. Although the CCG-S method has lower gradient evaluation complexity, the convergence performance has not yet been studied for nonconvex constrained problem. Our focus is on projection-free conditional gradient (CG) type algorithms. We propose CCG with line search to obtain an approximate solution with in $\mathcal{O}(1/\varepsilon^2)$ iterations.

In next section, we will first consider when the functional constrained optimization problem is convex.

4.2 Convex smooth functional constrained optimization

In this chapter, we study first-order projection-free methods for solving convex smooth functional constrained optimization problems, i.e., when the objective function f and constraints g_i, $i = 1, \ldots, m$ in problem (4.1) are all convex functions. We start by introducing a straightforward adaptation of CG method and extend to scenarios with functional constraints. Specifically, in subsection 4.2.1, we propose a constrained conditional gradient (CCG) method and analyze its convergence properties for solving problem (4.1). We show that CCG is a parameter-free algorithm that can compute an $(\varepsilon_f, \varepsilon_g)$-approximate solution in at most $\mathcal{O}\left(\max\{L_f D_X^2/\varepsilon_f, L_g D_X^2/\varepsilon_g\}\right)$ gradient evaluations. Then in subsection 4.2.2, we show that the gradient evaluation complexity can be improved to $\mathcal{O}\left(\max\{\sqrt{L_f D_X^2/\varepsilon_f}, \sqrt{L_g D_X^2/\varepsilon_g}\}\right)$.

4.2.1 Constrained conditional gradient method

We begin with the straightforward adaptation of the CG method for functional constrained optimization and propose a constrained conditional gradient (CCG) method. The proposed CCG method for solving problem (4.1) is described in Algorithm 4.2.9. The idea of the algorithm is the

same to vanilla CG: We first perform a conditional gradient step (4.3) to obtain the moving direction $s^t - x^{t-1}$, then obtain a new iterate x^t from the previous iterate x^{t-1} by moving along such direction with step size γ^t.

Algorithm 4.2.9 Constrained conditional gradient method (CCG) for solving (4.1)

Require: $x^0 \in X$ for all $i = 1, \ldots, m$.
for $t = 1, \ldots, N$ **do**
Compute

$$s^t \in \operatorname*{argmin}_{x \in X} \langle \nabla f(x^{t-1}), x \rangle \tag{4.3}$$
$$\text{s.t. } g_i(x^{t-1}) + \langle \nabla g_i(x^{t-1}), x - x^{t-1} \rangle \leq 0, \ i = 1, \ldots, m,$$
$$x^t = (1 - \gamma^t) x^{t-1} + \gamma^t s^t. \tag{4.4}$$

end for
Output x^N.

A few remarks concerning Algorithm 4.2.9 are in place. First, the optimization problem in (4.3) is always feasible. This is since $x^* \in X$ always satisfies $g_i(x^{t-1}) + \langle \nabla g_i(x^{t-1}), x^* - x^{t-1} \rangle \leq g_i(x^*) \leq 0$ due to the convexity of g_i's for all $i = 1, \ldots, N$. Second, since we assume that X is a compact set, the optimization problem in (4.3) is bounded. Finally, by (4.4) we have $x^t \in X$ for all t, although they are not necessarily feasible for the original problem (4.1).

Our analysis of CCG is a straightforward adaptation of that of the CG method without functional constraints. We start with two technical lemmas.

Lemma 4.2.1. *Suppose that $h : \mathbb{R}^n \to \mathbb{R}$ is a L-smooth function (with respect to norm $\|\cdot\|$). For any $x^t, x^{t-1}, s^t \in \mathbb{R}^n$ and $\gamma^t \in [0, 1]$ that satisfies (4.4), we have*

$$h(x^t) \leq (1 - \gamma^t) h(x^{t-1}) + \gamma^t (h(x^{t-1}) + \langle \nabla h(x^{t-1}), s^t - x^{t-1} \rangle) + \frac{L (\gamma^t)^2}{2} \|s^t - x^{t-1}\|^2. \tag{4.5}$$

Proof. Recalling that the L-smoothness of h implies that

$$h(x^t) \leq h(x^{t-1}) + \langle \nabla h(x^{t-1}), x^t - x^{t-1} \rangle + \frac{L}{2} \|x^t - x^{t-1}\|^2,$$

applying (4.4) to the above relation, we conclude (4.5). □

While we are studying convex optimization problems in this section, note that the only requirement of the function h in Lemma 4.2.1 is smoothness; the convexity is not necessarily needed.

The inequality (4.5) shows that for any $x^t, x^{t-1}, s^t \in \mathbb{R}^n$ and $\gamma^t \in [0,1]$ that satisfies a convex combination relation in (4.4), $h(x^t)$ can be upper bounded by the sum of a linear approximation of $h(x^t)$ and the norm of $s^t - x^{t-1}$.

The following lemma of a simple algebraic relationship is also needed in the convergence analysis.

Lemma 4.2.2. *Suppose that $\{\omega^t\}_{t=1}^N$, $\{a^t\}_{t=0}^N$ are nonnegative real-valued sequences and $\{\gamma^t\}_{t=1}^N$, $\{b^t\}_{t=1}^N$ are real-valued sequences. If they satisfy $\gamma^1 = 1$, $\gamma^t \leq 1$ and $\omega^{t-1} \geq \omega^t(1-\gamma^t)$ for all $t \geq 2$, and*

$$a^t \leq (1-\gamma^t)a^{t-1} + b^t, \ \forall t = 1, \ldots, N, \tag{4.6}$$

then we have

$$\omega^t a^t \leq \sum_{k=1}^{t} \omega^k b^k, \ \forall t = 1, \ldots, N.$$

Proof. The result is immediate by multiplying (4.6) by ω^t and summing from $k = 1 \ldots, t$. □

With the help of the above Lemmas 4.2.1 and 4.2.2, we are able to analyze the convergence properties of CCG in Algorithm 4.2.9 for solving convex smooth functional constrained optimization problems.

Theorem 4.2.1. *If the parameters in Algorithm 4.2.9 are set to $\gamma^t = 2/(t+1)$, the objective function f is L_f-smooth with respect to norm $\|\cdot\|$, and the constraint functions g_i are all L_g-smooth with respect to norm $\|\cdot\|$, then we have*

$$f(x^t) - f(x^*) \leq \frac{2L_f D_X^2}{t+1} \ \text{and} \tag{4.7}$$

$$[g_i(x^t)]_+ \leq \frac{2L_g D_X^2}{t+1}, \ \forall i = 1, \ldots, m, \tag{4.8}$$

where $D_X := \min_{x,y \in X} \|x - y\|$ is the diameter of X with respect to norm $\|\cdot\|$.

Proof. Applying Lemma 4.2.1 to the objective function f and constraint functions g_i's we have

$$f(x^t) \leq (1-\gamma^t) f(x^{t-1}) + \gamma^t (f(x^{t-1}) + \langle \nabla f(x^{t-1}), s^t - x^{t-1} \rangle) + \frac{L_f (\gamma^t)^2}{2} \|s^t - x^{t-1}\|^2, \quad (4.9)$$

$$g_i(x^t) \leq (1-\gamma^t) g_i(x^{t-1}) + \gamma^t (g_i(x^{t-1}) + \langle \nabla g_i(x^{t-1}), s^t - x^{t-1} \rangle) + \frac{L_g (\gamma^t)^2}{2} \|s^t - x^{t-1}\|^2. \quad (4.10)$$

Here applying the optimality condition of s^t in (4.3) and the convexity of f to (4.9) we have

$$\begin{aligned} f(x^t) \leq & (1-\gamma^t) f(x^{t-1}) + \gamma^t (f(x^{t-1}) + \langle \nabla f(x^{t-1}), x^* - x^{t-1} \rangle) + \frac{L_f (\gamma^t)^2}{2} \|s^t - x^{t-1}\|^2 \\ \leq & (1-\gamma^t) f(x^{t-1}) + \gamma^t f(x^*) + \frac{L_f (\gamma^t)^2}{2} \|s^t - x^{t-1}\|^2. \end{aligned}$$

Rearranging terms in the above relation and noting that $s^t, x^{t-1} \in X$ we have

$$f(x^t) - f(x^*) \leq (1-\gamma^t)(f(x^{t-1}) - f(x^*)) + \frac{L_f (\gamma^t)^2}{2} D_X^2.$$

Define a sequence $\{\omega^t\}_{t=1}^N$ such that $\omega^t = t(t+1)/2$, then ω^t and $\gamma^t = 2/(t+1)$ satisfy the assumptions in Lemma 4.2.2, applying the lemma to the above relation we obtain

$$\omega^t (f(x^t) - f(x^*)) \leq \frac{L_f D_X^2}{2} \sum_{k=1}^t \omega^k \left(\gamma^k\right)^2 = \frac{L_f D_X^2}{2} \sum_{k=1}^t \frac{2k}{k+1} \leq t L_f D_X^2,$$

which yields (4.7). Also, applying the feasibility condition of s^t in (4.3) to (4.10) and noting that $s^t, x^{t-1} \in X$ and $\gamma^t \in [0,1]$ we have

$$g_i(x^t) \leq (1-\gamma^t) g_i(x^{t-1}) + \frac{L_g (\gamma^t)^2}{2} D_X^2 \leq (1-\gamma^t)[g_i(x^{t-1})]_+ + \frac{L_g (\gamma^t)^2}{2} D_X^2.$$

Since the right-hand side of 4.2.1 is non-negative, the above relation implies that

$$[g_i(x^t)]_+ \leq (1-\gamma^t)[g_i(x^{t-1})]_+ + \frac{L_g (\gamma^t)^2}{2} D_X^2.$$

Recalling that $\omega^t = t(t+1)/2$ and $\gamma^t = 2/(t+1)$ and applying Lemma 4.2.2 to the above relation we conclude (4.8).

□

A few remarks are in place for the above theorem. First, in order to compute an $(\varepsilon_f, \varepsilon_g)$-approximate solution to the original problem (4.1), the number of iterations required by Algorithm 4.2.9 is bounded by $\mathcal{O}\left(\max\{L_f D_X^2/\varepsilon_f, L_g D_X^2/\varepsilon_g\}\right)$. Second, note that the above analysis applies to any Lipschitz smoothness conditions of f and g_i's with respect to any norm $\|\cdot\|$. Therefore, the convergence result applies to the best possible geometric structure of the problem (in terms of Lipschitz smoothness constants L_f, L_g, the associated norm $\|\cdot\|$, and diameter D_X). This is indeed a well-known feature of CG-type algorithms when solving problems with no functional constraint. Our result above shows that such feature persists in our proposed Algorithm 4.2.9 for functional constrained problems. In the following subsection, we will propose another projection-free method for solving convex constrained optimization problems.

4.2.2 Conditional gradient sliding for convex constrained optimization

As mentioned in the remark after Theorem 4.2.1, denote $\varepsilon = \min\{\varepsilon_f, \varepsilon_g\}$ for simplicity, the complexity for the number of gradient evaluations of ∇f, ∇g_i's, and the linear objective optimization subproblems are all in the order $\mathcal{O}(1/\varepsilon)$. Among the complexity results, it is known that the lower complexity bound for the number of linear objective optimization subproblems for projection-free methods is $\mathcal{O}(1/\varepsilon)$ (see, [8]). However, for functional constrained problems, the lower complexity bounds for the number of gradient evaluations of ∇f and ∇g_i's have not yet been studied in the literature. At present, the only related lower complexity bound known in the literature is the number of gradient evaluations of ∇f in the objective function. Specifically, for problem of form $\min_{x\in X} f(x)$, the number of gradient evaluations of ∇f of any projection-based methods in order to compute an ε-solution is lower bounded by $\mathcal{O}(1/\sqrt{\varepsilon})$. The lower complexity results involving gradient evaluations of g_i's is still open for general convex smooth constraint functions; there are only some results for the special case when g_i's are affine functions (see, [26]).

Based on the above summary concerning lower complexity bounds, it is an interesting research question to study whether it is possible to develop projection-free methods that has better complexity results than that of Algorithm 4.2.9 in terms of gradient evaluations of ∇f and ∇g_i's. Indeed, there has been existing methods in the literature that motivates us to develop better projection-free algorithms. Specifically, for problem of form $\min_{x\in X} f(x)$, it has been shown in [14]

that the upper complexity bound for gradient evaluations of ∇f and linear objective subproblems are $\mathcal{O}(1/\sqrt{\varepsilon})$ and $\mathcal{O}(1/\varepsilon)$ respectively. It has also been shown in ([28]) that the upper complexity bounds on the number of gradient evaluations of ∇f and ∇g_i's are both of order $\mathcal{O}(1/\sqrt{\varepsilon})$. Therefore, we might expect that the upper complexity bounds of gradient evaluations of both the objective and constraint functions described in Theorem 4.2.1 can be improved. In this section, motivated by the projection-based accelerated constrained gradient descent method with sliding (ACGD-S) proposed in [28], we propose CCG with sliding for solving convex functional constrained problems.

Algorithm 4.2.10 The CCG with sliding (CCG-S) outer loop for solving (4.1)

Require: $x^0 = \bar{x}^0 \in X$ for all $i = 1, \ldots, m$.
for $t = 1, \ldots, N$ **do**
Compute

$$\underline{x}^t = (1 - \beta^t)\bar{x}^{t-1} + \beta^t x^{t-1}. \tag{4.11}$$

Construct the quadratic program

$$\begin{aligned} \operatorname*{argmin}_{x \in X} \; & l_f(\underline{x}^t; x) + \frac{\eta^t}{2}\|x - x^{t-1}\|^2 \\ \text{s.t.} \; & l_{g_i}(\underline{x}^t; x) \leq 0, \; \forall i = 1, \ldots, m. \end{aligned}$$

Run the CG sliding subroutine described in Algorithm 4.2.11 with $y^0 = x^{t-1}$ and accuracy requirement δ^t to obtain an inexact solution x^t.
Set $\bar{x}^t$ to

$$\bar{x}^t = (1 - \beta^t)\bar{x}^{t-1} + \beta^t x^t, \tag{4.12}$$

end for
Output $\bar{x}^N$.

To begin with, if functions h are smooth, we define the linear approximation function of h at y as

$$l_h(y; x) = h(y) + \langle x - y, \nabla h(y) \rangle.$$

Then we have the CCG-S algorithm defined in Algorithm 4.2.10. In each iteration of the outer loop step in Algorithm 4.2.10, we first update $\underline{x}^t$ for gradient evaluation with a convex combination $\bar{x}^{t-1}$ and x^{t-1} of previous iteration. Then a CG sliding subroutine Algorithm (4.2.11) is performed to find the direction $x^t - \bar{x}^{t-1}$. Finally in (4.12), we update our iterate $\bar{x}^t$ by moving from $\bar{x}^{t-1}$ to the direction $x^t - \bar{x}^{t-1}$ with step size parameter β^t. In each CG sliding subroutine in Algorithm (4.2.11),

Algorithm 4.2.11 The CG sliding subroutine of Algorithm 4.2.10 for solving (4.1)

Require: $y^0 = x^{t-1}$ and target accuracy δ
for $s = 0, 1, 2, 3, \ldots$ **do**
Compute

$$d^{t,s} = \underset{y \in X}{\operatorname{argmin}}\ l_f(\underline{x}^t; y) + \eta\langle y, y^{t,s} - x^{t-1}\rangle \quad \text{s. t. } l_g(\underline{x}^t; y) \le 0,\ \forall i = 1, \ldots, m. \tag{4.13}$$

End loop if

$$\mathcal{G}(y^{t,s}) := \langle y^{t,s} - d^{t,s}, \nabla f(\underline{x}^t) + \eta(y^{t,s} - x^{t-1})\rangle \le \delta.$$

Set y^{s+1} to

$$y^{t,s+1} = \alpha^{t,s+1} d^{t,s} + (1 - \alpha^{t,s+1}) y^{t,s}. \tag{4.14}$$

end for
Output $y^{t,s}$.

a conditional gradient problem is solve to update the direction $y^{t,s} - \bar{x}^{t-1} = x^t - \bar{x}^{t-1}$. Remark that since there is only one gradient evaluation in each loop of the subroutine, the gradient evaluation complexity of the out loop stays the same. The convergence analysis of each subroutine is provided in in Lemma 4.2.3. With help of Lemma 4.2.3, we will then give the convergence analysis of the outer loop in Proposition 4.2.1.

Lemma 4.2.3. *Consider the iterates $\{y^{t,s}\}$ generated by the CG sliding subroutine in Algorithm 4.2.10 with $\alpha^{t,1} = 1$*

$$\alpha^{t,s} = \min\{1, \frac{\langle y^{t,s-1} - d^{t,s-1}, \nabla f(\underline{x}^t) + \eta(y^{t,s-1} - x^{t-1})\rangle}{\eta\|y^{t,s-1} - x^{t-1}\|^2}\},\ s = 2, \ldots, N. \tag{4.15}$$

Let S denote the number of inner iterations when the solution $x^t := y^{t,S}$ is output by the method, the following relation is valid

$$l_f(\underline{x}^t; x^t) - l_f(\underline{x}^t; x^*) + \frac{\eta}{2}[\|x^t - x^*\|^2 + \|x^t - x^{t-1}\|^2] \le \frac{\eta}{2}\|x^{t-1} - x^*\|^2 + \delta \tag{4.16}$$

$$\lambda l_g(\underline{x}^t; x^t) \le 0,\ \forall \lambda. \tag{4.17}$$

And we have $S \le c\eta D_X^2/\delta$, where η is some universal constant.

Proof. Let $X(t)$ denote the feasible region $\{x \in X : l_g(x; \nabla g(\underline{x}^t)) \le 0\}$. Note that $X(t)$ is convex

for all $t = 1, \ldots, N$. With $x^t = y^{t,S}$, the termination condition implies that $\mathcal{G}(x^t) \leq \delta$. Since $d^{t,S}$ is the minimizer of (4.13) and the optimal solution of (4.1) satisfy $x^* \in X(t)$, we have

$$\langle x^t - x^*, \nabla f(\underline{x}^t) + \eta(x^t - x^{t-1})\rangle \leq \langle x^t - d^{t,S}, \nabla f(\underline{x}^t) + \eta(x^t - x^{t-1})\rangle = \mathcal{G}(x^t) \leq \delta. \quad (4.18)$$

The following qualities are derived from simple algebraic.

$$\langle x^t - x^*, \eta(x^t - x^{t-1})\rangle = \frac{\eta}{2}[\|x^t - x^*\|^2 + \|x^t - x^{t-1}\|^2 - \|x^{t-1} - x^*\|^2], \quad (4.19)$$

$$\langle x^t - x^*, \nabla f(\underline{x}^t)\rangle = l_f(\underline{x}^t; x^t) - l_f(\underline{x}^t; x^*). \quad (4.20)$$

Combining (4.18), (4.19) and (4.20), we are able to obtain (4.16). Moreover, since $X(t)$ is convex and fixes throughout the sliding updates, and with $\alpha^{t,1} = 1$, the iterates x^t are convex combinations of points in $X(t)$, we have that $x^t \in X(t)$ and (4.17) is satisfied for any $\lambda \in \mathbb{R}^m_+$.

We now derive the number of iterations required in the sliding subroutine in Algorithm (4.2.11) to generate the output x^t.

Notice our objective function for the sliding subroutine is given by f_t below

$$f_t(y) := l_f(\underline{x}^t; y) + \frac{\eta}{2}\|y - x^{t-1}\|^2.$$

Since f is a convex smooth function, we are able to conclude that f_t is also convex and smooth with Lipschitz constant η. Applying Lemma 4.2.1 and the convexity of f_t, we obtain the following inequality for any $\gamma^{t,s} \in [0, 1]$ and $\tilde{y}^{t,s} := \gamma^{t,s} d^{t,s} + (1 - \gamma^{t,s}) y^{t,s-1}$

$$f_t(\tilde{y}^{t,s}) - f_t^* \leq (1 - \gamma^{t,s})(f_t(y^{t,s-1}) - f_t^*) + (\gamma^{t,s})^2 \frac{\eta D_X^2}{2}. \quad (4.21)$$

According to (4.14), $y^{t,s}$ lies on the line interval $[y^{t,s-1}, y^{t,s-1} + d^{t,s-1}]$ and is determined by the step size $\alpha_{t,s}$. In particular, we choose step size $\alpha_s \in [0, 1]$ such that the quadratic function f_t is minimized.

Since when $\alpha^s = \langle y^{t,s-1} - d^{t,s-1}, \nabla f(\underline{x}^t) + \eta(y^{t,s-1} - x^{t-1})\rangle / \eta \|y^{t,s-1} - x^{t-1}\|^2$ we have that

$$\frac{\partial f_t}{\partial \alpha_s} = \langle d^{t,s-1} - y^{t,s-1}, \nabla f(\underline{x}^t) + \eta(y^{t,s-1} - x^{t-1})\rangle + \alpha^s \eta \|y^{t,s-1} - x^{t-1}\|^2 = 0,$$

and (4.15) gives the appropriate step size α_t. With such step size α_t, $y^{t,s}$ is the minimizer of f_t and $f_t(y^{t,s}) - f_t^* \leq f_t(\tilde{y}^{t,s}) - f_t^*$. Thus we derive from (4.21) that

$$f_t(y^{t,s}) - f_t^* \leq (1-\gamma^{t,s})(f_t(y^{t,s-1}) - f_t^*) + (\gamma^{t,s})^2 \frac{\eta D_X^2}{2}.$$

In particular, by Lemma 4.2.2, we can select $\gamma^{t,s} = 2/(s+1)$ and $\omega^s = s(s+1)$ to obtain

$$f_t(y^{t,s}) - f_t^* \leq \frac{2\eta D_X^2}{s+1}, \ \forall s \geq 1.$$

By the smoothness of function f_t, we derive from $f_t(y^{t,s}) - f_t^* \leq f_t(\tilde{y}^{t,s}) - f_t^*$ that

$$\begin{aligned} \gamma^{t,s}\mathcal{G}(y^{t,s}) \leq & (f_t(y^{t,s}) - f_t^*) - (f_t(\tilde{y}^{s+1}) - f_t^*) + (\gamma^{t,s})^2 \frac{\eta}{2}\|d^{t,s} - y^{t,s}\|^2 \\ \leq & (f_t(y^{t,s}) - f_t^*) - (f_t(y^{s+1}) - f_t^*) + (\gamma^{t,s})^2 \frac{\eta D_X^2}{2}, \end{aligned} \tag{4.22}$$

where $\mathcal{G}(y^{t,s}) = \langle y^{t,s} - d^{t,s}, \nabla f(\underline{x}^t) + \eta(y^{t,s} - x^{t-1})\rangle$. Apply the weight $\omega_s = s(s+1)$ on (4.22) and sum from $s=1$ to $s=S$, we are able to obtain

$$\begin{aligned} & \left(\sum_{s=1}^{S} \omega^s \gamma^{t,s}\right) \min_{s=1,\ldots,S} \mathcal{G}(y^{t,s}) \\ \leq & \sum_{s=2}^{S} (\omega^s - \omega^{s-1})(f_t(y^{t,s}) - f_t^*) + \omega^1(f_t(y^{t,1}) - f_t^*) - \omega^S(f_t(y^{S+1}) - f_t^*) + \sum_{s=1}^{S} \omega^s (\gamma^{t,s})^2 \frac{\eta D_X^2}{2} \\ \leq & \sum_{s=2}^{S} (\omega^s - \omega^{s-1})(f_t(y^{t,s}) - f_t^*) + \sum_{s=1}^{S} \omega^s (\gamma^{t,s})^2 \frac{\eta D_X^2}{2}, \end{aligned}$$

where the second inequality is derived from the fact that $\omega^0 = 0$ and $f_t(y^{t,S+1}) - f_t^* \geq 0$. With $\omega_s = s(s+1)$ and $\gamma^{t,s} = 1/(s+1)$, we have

$$\min_{s=1,\ldots,S} \mathcal{G}(y^{t,s}) \leq \frac{1}{S(S+1)} \left(\sum_{s=2}^{S} \frac{s}{s+1} 4\eta D_X^2 + 2S\eta D_X^2 \right) \leq \frac{6\eta D_X^2}{S+1},$$

and $S \leq 6\eta D_X^2/\delta$ implies that the end loop condition $\mathcal{G}(y^{t,s}) \leq \delta$ is satisfied. □

Remark that Lemma 4.2.3 implied that the iteration complexity of each subroutine is bounded by an universal constant, which will result that the total complexity of CCG-S method is determined by the complexity of the outer loop. The next proposition establishes an important

recursive relation for the CCG-S outer loop.

Proposition 4.2.1. *Consider x^t generated by the CCG-S Algorithm in* (4.2.10) *with the proximal parameter chosen to satisfy $\eta^t \geq L(\Lambda)\beta^t$, where $L(\Lambda) := L_f + L_g(\Lambda)$ with $L_g(\Lambda)$ representing an upper bound Lipschitz smoothness constant of $\sum_{i-1}^m \lambda_i g_i(x)$ for any $\lambda \in \Lambda$, then we have*

$$\begin{aligned} &f(\bar{x}^t) - f^* + \lambda g(\bar{x}^t) + \frac{\beta^t \eta^t}{2}\|x^t - x^*\|^2 \\ \leq &(1-\beta^t)(f(\bar{x}^{t-1}) - f^* + \lambda g(\bar{x}^{t-1})) + \frac{\beta^t \eta^t}{2}\|x^{t-1} - x^*\|^2 + \beta^t \delta^t, \ \forall \lambda \in \Lambda. \end{aligned} \tag{4.23}$$

Moreover, if there exists some weight ω^t satisfying $\omega^t \beta^t \eta^t \geq \omega^{t-1}\beta^{t-1}\eta^{t-1}$, and $\omega^{t-1} \geq \omega^t(1-\beta^t)$, then we have

$$\begin{aligned} &f(\bar{x}^N) - f^* + \lambda g(\bar{x}^N) \\ \leq &\frac{1}{\omega^N}\left(\omega^1(1-\beta^1)(f(x^0) - f^* + \lambda g(\bar{x}^0)) + \frac{\omega^1\beta^1\eta^1}{2}D_X^2 + \sum_{i=1}^m \omega^t\beta^t\delta^t\right), \ \forall \lambda \in \Lambda. \end{aligned} \tag{4.24}$$

In particular, choosing $\beta^t = 3/(t+2)$, $\eta^t = 3L(\Lambda)/(t+2)$, $\delta^t = L(\Lambda)D_X^2/t(t+1)$, $\omega^t = t(t+1)(t+2)$ would lead to

$$f(\bar{x}^N) - f^* + \lambda g(\bar{x}^N) \leq \frac{3L(\Lambda)D_X^2}{2N(N+1)}, \ \forall \lambda \in \Lambda.$$

Proof. Since $f + \lambda g$ is convex smooth, denote the Lipschitz continuous constant as $L(\Lambda) \leq \beta^t\eta^t$, by (4.11) and (4.12) in Algorithm (4.2.10), we have that

$$\begin{aligned} &f(\bar{x}^t) + \lambda g(\bar{x}^t) \\ \leq &l_f(\underline{x}^t; \bar{x}^t) + \lambda l_g(\underline{x}^t; \bar{x}^t) + \frac{L(\Lambda)}{2}\|\bar{x}^t - \underline{x}\|^2 \\ \leq &(1-\beta^t)[l_f(\underline{x}^t; \bar{x}^{t-1}) + \lambda l_g(\underline{x}^t; \bar{x}^{t-1})] + \beta^t[l_f(\underline{x}^t; x^t) + \lambda l_g(\underline{x}^t; x^t)] + \frac{\beta^t\eta^t}{2}\|x^t - x^{t-1}\|^2 \\ \leq &(1-\beta^t)[f(\bar{x}^{t-1}) + \lambda g(\bar{x}^{t-1})] + \beta^t[l_f(\underline{x}^t; x^t) + \lambda l_g(\underline{x}^t; x^t)] + \frac{\beta^t\eta^t}{2}\|x^t - x^{t-1}\|^2, \ \forall \lambda \in \Lambda. \end{aligned} \tag{4.25}$$

where the last inequality follows from convexity. By Lemma 4.2.3, we have that

$$l_f(\underline{x}^t; x^t) - l_f(\underline{x}^t; x^*) + \frac{\eta}{2}[\|x^t - x^*\|^2 + \|x^t - x^{t-1}\|^2] \leq \frac{\eta}{2}\|x^{t-1} - x^*\|^2 + \delta \tag{4.26}$$

$$\lambda l_g(\underline{x}^t; x^t) \leq 0, \ \forall \lambda \in \Lambda. \tag{4.27}$$

Combining (4.25) with (4.26) and (4.27), we have

$$\begin{aligned} f(\bar{x}^t) + \lambda g(\bar{x}^t) \leq & (1-\beta^t)[f(\bar{x}^{t-1}) + \lambda g(\bar{x}^{t-1})] + \beta^t l_f(\underline{x}^t; x^*) \\ & + \frac{\beta^t \eta^t}{2}(\|x^{t-1} - x^*\|^2 - \|x^t - x^*\|^2) + \beta^t \delta^t \\ \leq & (1-\beta^t)[f(\bar{x}^{t-1}) + \lambda g(\bar{x}^{t-1})] + \beta^t f(x^*) \\ & + \frac{\beta^t \eta^t}{2}(\|x^{t-1} - x^*\|^2 - \|x^t - x^*\|^2) + \beta^t \delta^t, \ \forall \lambda \in \Lambda. \end{aligned} \tag{4.28}$$

where the last inequality follows from the convexity of f and g. Rearrange (4.28), (4.23) is derived.

If ω^t satisfies $\omega^t \beta^t \eta^t \geq \omega^{t-1}\beta^{t-1}\eta^{t-1}$, and $\omega^{t-1} \geq \omega^t(1-\beta^t)$, then by multiplying ω^t to (4.23) and summing up for $t = 1, \ldots, N$, we have

$$\begin{aligned} & \omega^N(f(\bar{x}^N) - f^* + \lambda g(\bar{x}^N)) \\ \leq & \left(\omega^1(1-\beta^1)(f(x^0) - f^* + \lambda g(\bar{x}^0)) + \frac{\omega^1\beta^1\eta^1}{2}\|x^0 - x^*\|^2 - \frac{\omega^t\beta^t\eta^t}{2}\|x^t - x^*\|^2 + \sum_{i=1}^m \omega^t\beta^t\delta^t\right) \\ \leq & \left(\omega^1(1-\beta^1)(f(x^0) - f^* + \lambda g(\bar{x}^0)) + \frac{\omega^1\beta^1\eta^1}{2}D_X^2 + \sum_{i=1}^m \omega^t\beta^t\delta^t\right), \ \forall \lambda \in \Lambda, \end{aligned}$$

and (4.24) in derived. In particular, choosing $\beta^t = 3/(t+2)$, $\eta^t = 3L(\Lambda)/(t+2)$, $\delta^t = L(\Lambda)D_X^2/t(t+1)$, $\omega^t = t(t+1)(t+2)$ would lead to

$$f(\bar{x}^N) - f^* + \lambda g(\bar{x}^N) \leq \frac{1}{N(N+1)(N+2)}\left(3L(\Lambda)D_X^2 + \frac{3NL(\Lambda)D_X^2}{2}\right) = \frac{3L(\Lambda)D_X^2}{2N(N+1)},$$

where λ is arbitrary in Λ. □

Remark that the above proposition implies $f(\bar{x}^N) - f^* + \lambda g(\bar{x}^N) \leq \varepsilon$ if N is greater than $\mathcal{O}(\sqrt{L(\Lambda)D_X^2/\varepsilon})$ for any accuracy threshold ε. The following theorem further proved that if (4.24) in Proposition 4.2.1 is satisfied, x_N is an $(\varepsilon_f, \varepsilon_g)$-approximate solution such that $f(\bar{x}^N) - f^* \leq \varepsilon_f$

and $[g(\bar{x}^N)]_+ \leq \varepsilon_g$.

Theorem 4.2.2. *Consider x^t generated by the ACGD Algorithm in* (4.2.10) *with the proximal parameter chosen to satisfy $\eta^t \geq L(\Lambda)\beta^t$, where $L(\Lambda) := L_f + L_g(\Lambda)$ with $L_g(\Lambda)$ representing an upper bound Lipschitz smoothness constant of $\sum_{i-1}^m \lambda_i g_i(x)$ for any $\lambda \in \Lambda$, and there exists some weight ω^t satisfying $\omega^t\beta^t\eta^t \geq \omega^{t-1}\beta^{t-1}\eta^{t-1}$, and $\omega^{t-1} \geq \omega^t(1-\beta^t)$. If we choose $\Lambda = \lambda^* + \|\lambda^*\|B(0,1)$, then $\bar{x}_N$ is an $(\varepsilon_f, \varepsilon_g)$-approximate solution such that $f(\bar{x}_N) - f^* \leq \varepsilon_f$ and $[g(\bar{x}^N)]_+ \leq \varepsilon_g$ with $N = \mathcal{O}(\max\{\sqrt{L(\Lambda)D_X^2/\varepsilon_f}, \sqrt{L(\Lambda)D_X^2/\varepsilon_g}\})$.*

Proof. By Proposition 4.2.1, with such choice of parameters $\{\eta^t\}_{t=1}^N$ and $\{\beta^t\}_{t=1}^N$, if the total number of iterations N is greater than $N = \mathcal{O}(\max\{\sqrt{L(\Lambda)D_X^2/\varepsilon_f}, \sqrt{L(\Lambda)D_X^2/\varepsilon_g}\})$, then $f(\bar{x}^N) - f^* + \lambda g(\bar{x}^N) \leq \min\{\varepsilon_f, \varepsilon_g\}$ for any $\lambda \in \Lambda$. Since λ is arbitrary, if we choose $\lambda = 0$, then $f(\bar{x}^N) - f^* \leq \varepsilon_f$ is derived. Denote $(\bar{x}^*, \lambda^*)$ as the saddle point of $f(x) + \lambda g(x)$. Without loss of generality we assume $\|\lambda^*\| = 1$. Then we have that

$$0 \leq f(\bar{x}^N) - f^* + \lambda^* g(\bar{x}^N) - 0^\top g(\bar{x}^*) = f(\bar{x}^N) - f^* + \lambda^* g(\bar{x}^N). \tag{4.29}$$

Choose $\lambda = \lambda^* + [g(\bar{x}^N)]_+/\|[g(\bar{x}^N)]_+\|$, then we are able to derive from (4.29) that

$$[g(\bar{x}^N)]_+ \leq \|[g(\bar{x}^N)]_+\| \leq f(\bar{x}^N) - f^* + \lambda g(\bar{x}^N) \leq \varepsilon_g/\|\lambda^*\| = \varepsilon_g.$$

□

Remark from Theorem 4.2.2 that since there is only one gradient evaluation of functions f and g in each subroutine, the total iteration number required to obtain an $(\varepsilon_f, \varepsilon_g)$-approximate solution is greater than $N = \mathcal{O}(\max\{\sqrt{L(\Lambda)D_X^2/\varepsilon_f}, \sqrt{L(\Lambda)D_X^2/\varepsilon_g}\})$. This is a better complexity result for solving convex constrained optimization problem (4.1) than the CCG method. However, the complexity property of CCG-S has for solving nonconvex optimization problems has not yet been studied. In next section, the constrained conditional gradient method with line search is proposed for solving nonconvex problem (4.1).

4.3 Nonconvex smooth functional constrained optimization

In this section, we continue to examine the nonconvex optimization scenario where the convexity assumption for f is relaxed. However, we continue to uphold the assumption that the feasible set X remains convex. In Section 4.3.1, the constraints g_i, $i = 1, \ldots, m$ are convex, and the corresponding convergence analysis are given. In Section 4.3.2, g_i, $i = 1, \ldots, m$ are also nonconvex with the convergence analysis given.

For nonconvex smooth optimization, two optimality conditions are commonly known in the literature: the Fritz-John (FJ) and the Karush–Kuhn–Tucker (KKT) conditions. Specifically, the FJ necessary condition of problem (4.1) at a feasible point $\hat{x}$ is that there exists nonnegative multipliers $\lambda_0, \ldots, \lambda_m$ that are not all zero such that

$$\langle \lambda_0 \nabla f(\hat{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\hat{x}), \hat{x} - x \rangle \leq 0, \ \forall x \in X.$$

Without loss of generality, we assume that $\sum_{i=1}^{m} \lambda_i = 1$. The KKT necessary condition is a special case of FJ in which $\lambda_0 = 1$. Our goal is to obtain am approximate FJ point.

We propose the CCG with line search algorithm in Algorithm 4.3.12 solving nonconvex optimization problems. The structure is based on the original CCG algorithm in Algorithm 4.2.9. To accelerate the convergence of CCG, we add a line search step (4.31) for a best step size γ^t.

Algorithm 4.3.12 Conditional gradient method with line search (CCG-L) for solving (4.1)

Require: $x^0 \in X$ for all $i = 1, \ldots, m$.
for $t = 1, \ldots, N$ **do**
Compute

$$\begin{aligned} s^t \in \underset{x \in X}{\operatorname{argmin}} \ & \langle \nabla f(x^{t-1}), x \rangle \\ \text{s.t.} \ & g_i(x^{t-1}) + \langle \nabla g_i(x^{t-1}), x - x^{t-1} \rangle \leq 0, \ \forall i = 1, \ldots, m. \end{aligned} \tag{4.30}$$

Set x^t to

$$x^t = (1 - \gamma^t) x^{t-1} + \gamma^t s^t,$$

where γ^t is chosen such that

$$\gamma^t \in \underset{\gamma \geq 0}{\operatorname{argmin}} f((1-\gamma)x^{t-1} + \gamma s^t) \text{ s.t. } g_i((1-\gamma)x^{t-1} + \gamma s^t) \leq \varepsilon_g, \forall i = 1, \ldots, m. \tag{4.31}$$

end for
Output x^N.

Before we discuss the convergence analysis of CCG-L when the constraints are convex and nonconvex respectively, remark that the feasibility of the the convergence analysis of CCG in Section 4.2 is derived by convexity. Without the convexity assumption, in Algorithm 4.3.12 the optimization problem in (4.30) is not necessarily feasible. The following Proposition show that if an iterate x^{t-1} is assumed to be ε_g-feasible, even if the optimization problem in (4.3) is not feasible, x^{t-1} is an $(\varepsilon_f, \varepsilon_g)$-approximate FJ point as defined in Definition 4.1.3.

Proposition 4.3.1. *If x^{t-1} in Algorithm 4.2.9 is ε_g-feasible but the optimization problem in* (4.3) *is infeasible, then x^{t-1} is an $(\varepsilon_f, \varepsilon_g)$-approximate FJ point.*

Proof. To construct an $(\varepsilon_f, \varepsilon_g)$-approximate FJ point, we first consider the feasibility problem

$$\begin{aligned} &\min_{x \in X} 0 \\ &\text{s.t. } \; g_i(x^{t-1}) + \langle \nabla g_i(x^{t-1}), x - x^{t-1} \rangle \leq 0, \; i = 1, \ldots, m \end{aligned} \tag{4.32}$$

associated with problem (4.3), whose dual problem is

$$\max_{\lambda_1, \ldots, \lambda_m \geq 0} \min_{x \in X} \sum_{i=1}^{m} \lambda_i (g_i(x^{t-1}) + \langle \nabla g_i(x^{t-1}), x - x^{t-1} \rangle).$$

Note that the dual problem is always feasible (with $\lambda_1 = \cdots = \lambda_m = 0$). Therefore, problem (4.32) is infeasible if and only if there exists $\lambda_1, \ldots, \lambda_m \geq 0$ such that $\min_{x \in X} \sum_{i=1}^{m} \lambda_i (g_i(x^{t-1}) + \langle \nabla g_i(x^{t-1}), x - x^{t-1} \rangle) > 0$, i.e.,

$$\sum_{i=1}^{m} \lambda_i \langle \nabla g_i(x^{t-1}), x^{t-1} - x \rangle < \sum_{i=1}^{m} \lambda_i g_i(x^{t-1}), \; \forall x \in X.$$

Clearly, we have $\sum_{i=1}^{m} \lambda_i > 0$; otherwise the above inequality will not hold. Without loss of generality, we assume that $\sum_{i=1}^{m} \lambda_i = 1$. Recalling the assumption that x^{t-1} is ε_g-feasible, with $\lambda_0 = 0$, we have

$$\left\langle 0 \cdot \nabla f(x^{t-1}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^{t-1}), x^{t-1} - x \right\rangle < \varepsilon_g \leq \varepsilon_f.$$

Therefore, x^{t-1} is an approximate FJ point. □

As shown in the previous proposition, we would like to keep the ε_g-feasibility of iterates. The following lemma shows that if an iterate x^{t-1} is ε_g-feasible, then with appropriate parameters γ^t chosen, the iterates will remain ε_g-feasible.

Lemma 4.3.1. *Suppose that x^{t-1} is ε_g-feasible in Algorithm 4.2.9. Then x^t is also ε_g-feasible if $\gamma^t \leq \varepsilon_g / L_g D_X^2 \leq 1$.*

Proof. Applying Lemma 4.2.1, denote $\Delta_t := \gamma^t[\gamma^t L_g D_X^2/2 - g(x^t)]$, we have that $g(x^{t+1}) \leq g(x^t) + \Delta_t$. Since $\gamma^t \leq \varepsilon_g / L_g D_X^2$, we have that $\gamma^t L_g D_X^2/2 \leq \varepsilon_g/2$. If $\varepsilon_g/2 < g_i(x^t) < \varepsilon_g$, then $\Delta_t < 0$, and hence $g_i(x^{t+1}) < g_i(x^t) < \varepsilon_g$. Otherwise, if $g_i(x^t) \leq \varepsilon_g/2$, then $\Delta_t \leq \gamma^t \varepsilon_g/2 \leq \varepsilon_g/2$ with $\gamma^t \leq 1$. Thus $g_i(x^{t+1}) \leq g_i(x^t) + \Delta_t \leq \varepsilon_g/2 + \varepsilon_g/2 \leq \varepsilon_g$. In conclusion, $\max_i g_i(x^{t+1}) \leq \varepsilon_g$ and x^{t+1} is ε_g-feasible. □

Lemma 4.3.1 implies that with an ε_g-feasible starting point x_0, all the iterates x_t are ε_g-feasible. Since we add a line search step for the best step size parameter γ_t, we are not able to choose a constant γ^t as in the convergence analysis of CCG when the problem is convex. however, note that Lemma 4.3.1 implies that there always exists an $\gamma^t \leq \varepsilon_g / L_g D_X^2 \leq 1$ such that (4.31) is feasible with an ε_g-feasible starting point x_0. The convergence analysis of the line search algorithm Algorithm 4.3.12 is provided in the following sections for solving the nonconvex optimization problem (4.1) with both convex constraints and nonconvex constraints. We will first provide the convergence of the convex constrained problem.

4.3.1 Convergence analysis for solving convex constrained nonconvex optimization problem

In this subsection, we first consider the nonconvex constrained optimization where the constraints g_i, $i = 1, \ldots, m$ are convex. The following Theorem shows that with appropriate parameters chosen, we are able to obtain an $(\varepsilon_f, \varepsilon_g)$-stationary point solution.

Theorem 4.3.1. *Consider* (4.1) *with convex constraints g_i, $i = 1, \ldots, m$. In Algorithm 4.2.9, if we start from an ε_g-feasible starting point x_0, then denote $\Delta_f := f(x_0) - f(x^*)$, there exists*

$t \in \{1, 2, \ldots, N\}$ *such that*

$$\begin{aligned}\langle \nabla f(x^{t-1}), x^{t-1} - x\rangle \leq & \frac{\Delta_f D_X^2}{N \min\{\frac{\varepsilon_f}{L_f}, \frac{\varepsilon_g}{L_g}\}} + \frac{\varepsilon_f}{2}, \forall x \in X \text{ s.t. } g_i(x) \leq 0,\ i = 1, \ldots, m \text{ and } \quad (4.33)\\ g_i(x^{t-1}) \leq & \varepsilon_g,\ \forall i = 1, \ldots, m,\ t = 1, \ldots, N.\end{aligned}$$

Proof. Since we start from an ε_g-feasible solution x_0, the iterates always remain ε_g-feasible. Thus it suffices to only consider the convergence of the Wolfe gap $\max_{g(x)\leq 0, x\in X}\langle x^t - x, \nabla f(x^t)\rangle$. Denote $X^0 := \{x \in X, g(x) \leq 0\}$, $X^1 := \{x \in X, \langle \nabla g(x^{t-1}), x - x^{t-1}t\rangle + g(x^{t-1}) \leq 0\}$. Since g_i, $i = 1, \ldots, m$ are convex and $X^0 \subseteq X^1$,

$$\max_{x\in X^0}\langle \nabla f(x^{t-1}), x^{t-1} - x\rangle \leq \max_{s\in X^1}\langle \nabla f(x^{t-1}), x^{t-1} - x\rangle = \langle \nabla f(x^{t-1}), x^{t-1} - s^t\rangle. \quad (4.34)$$

The last equality is derived from (4.30). Define $\underline{x}^t = (1-\gamma)x^{t-1} + \gamma s^t$, where $\gamma = \min\{\varepsilon_f/L_f D^2, \varepsilon_g/L_g D^2\}$, then derive from Lemma (4.2.1) and the smoothness of the objective function f that

$$\begin{aligned}\gamma\langle \nabla f(x^{t-1}), x^{t-1} - s^t\rangle \leq & f(x^{t-1}) - f(\underline{x}^t) + \gamma^2 \frac{L_f}{2}\|s^t - x^{t-1}\|_2^2 \quad (4.35)\\ \leq & f(x^{t-1}) - f(x^t) + \gamma^2 \frac{L_f}{2} D_X^2.\end{aligned}$$

where the second inequality is obtained by the fact that $f(x^t) \leq f(\underline{x}^t)$ and $s^t, x^{t-1} \in X$. Summing up the above inequality (4.35) for $t = 1, \ldots, N$, since $f(x^0) - f(x^N) \leq f(x^0) - f(x^*)$, we have

$$\sum_{t=1}^N \gamma\langle \nabla f(x^{t-1}), x^{t-1} - s^t\rangle \leq (f(x^0) - f(x^*)) + \sum_{t=1}^N \frac{\gamma^2 L_f D_X^2}{2}. \quad (4.36)$$

Derive from (4.34) that $\langle \nabla f(x^{t-1}), x^{t-1} - x\rangle \leq \langle \nabla f(x^{t-1}), x^{t-1} - s^t\rangle$, $\forall x \in X_0$, $t \in \{1, \ldots, N\}$, (4.36) implies that there exists an t such that (4.33) is derived. □

Remark from the above theorem that the number of iterations required by Algorithm 4.3.12 is bounded by $\mathcal{O}\left(\max\{\Delta_f L_f D_X^2/\varepsilon_f^2, \Delta_f L_g D_X^2/\varepsilon_f\varepsilon_g\}\right)$ to obtain an $(\varepsilon_f, \varepsilon_g)$-stationary point solution to the original problem (4.1). In the following subsection, we will continue to the general nonconvex constrained optimization problems.

4.3.2 Convergence analysis for solving nonconvex constrained nonconvex optimization problem

In this subsection, we perform the convergence analysis of Algorithm 4.3.12 for solving (4.1) with nonconvex objective and constraint functions. The following Theorem provides complexity analysis for computing an $(\varepsilon_f, \varepsilon_g)$-approximate FJ point.

Theorem 4.3.2. *In Algorithm 4.2.9, if we start from an ε_g-feasible starting point x_0, and the total iteration number $N \geq 2\max\{\Delta_f L_f D_X^2/\varepsilon_f^2, \Delta_f L_g D_X^2/\varepsilon_f\varepsilon_g\}$, then denote $\Delta_f := f(x_0) - f(x^*)$, there exists $\lambda_i \geq 0$, $i = 0, \ldots, m$ and $k \in \{1, \ldots, N\}$ such that*

$$\begin{aligned} \langle \lambda_0 \nabla f(x^{t-1}) + \sum_{i=1}^{m} \hat{\lambda}_i \nabla g_i(x^{t-1}), x^{t-1} - x \rangle \leq \varepsilon_f, \ \forall x \in X. \\ g_i(x^{t-1}) \leq \varepsilon_g, \ \forall i = 1, \ldots, m, \ t = 1, \ldots, N. \end{aligned} \tag{4.37}$$

Proof. Since we start from an ε_g-feasible solution x_0, the iterates always remain ε_g-feasible. By Proposition 4.3.1, if the optimization problem in (4.30) is infeasible, then x^{t-1} is an $(\varepsilon_f, \varepsilon_g)$-approximate FJ point. Thus we only consider when (4.30) is feasible.

Define $\underline{x}^t = (1-\gamma)x^{t-1} + \gamma s^t$, where $\gamma := \min\{\varepsilon_f/L_f D_X^2, \varepsilon_g/L_g D_X^2\}$. From Lemma (4.2.1) and the smoothness of the objective function f, we have

$$\begin{aligned} \gamma \langle \nabla f(x^{t-1}), x^{t-1} - s^t \rangle \leq & f(x^{t-1}) - f(\underline{x}^t) + \gamma^2 \frac{L_f}{2} \|s^t - x^{t-1}\|_2^2 \\ \leq & f(x^{t-1}) - f(x^t) + \gamma^2 \frac{L_f}{2} D_X^2. \end{aligned}$$

Therefore, summing from $t = 1, \ldots, N$ and recalling that $\gamma := \min\{\varepsilon_f/L_f D^2, \varepsilon_g/L_g D^2\}$ we have

$$\begin{aligned} \min_{t=1,\ldots,N} \langle \nabla f(x^{t-1}), x^{t-1} - s^t \rangle \leq & \frac{1}{N} \sum_{t=1}^{N} \langle \nabla f(x^{t-1}), x^{t-1} - s^t \rangle \leq \frac{\Delta_f}{\gamma N} + \frac{\gamma L_f D_X^2}{2} \\ \leq & \frac{\Delta_f D_X^2}{N \min\{\frac{\varepsilon_f}{L_f}, \frac{\varepsilon_g}{L_g}\}} + \frac{\varepsilon_f}{2}. \end{aligned} \tag{4.38}$$

Let us use k to denote the index such that

$$\langle \nabla f(x^{k-1}), x^{k-1} - s^k \rangle = \min_{t=1,\ldots,N} \langle \nabla f(x^{t-1}), x^{t-1} - s^t \rangle$$

and study the optimization problem

$$\begin{aligned}&\min_{x\in X}\langle\nabla f(x^{k-1}),x-x^{k-1}\rangle\\&\text{s.t. } g_i(x^{k-1})+\langle\nabla g_i(x^{k-1}),x-x^{k-1}\rangle\le 0,\ \forall i=1,\dots,m\end{aligned}$$

in which one optimal solution is $x=s^k$. The saddle point form of the above problem is

$$\min_{x\in X}\max_{\lambda\ge 0}\langle\nabla f(x^{k-1}),x-x^{k-1}\rangle+\sum_{i=1}^{m}\lambda_i\left(g_i(x^{k-1})+\langle\nabla g_i(x^{k-1}),x-x^{k-1}\rangle\right).$$

From the optimality condition and complementary slackness, there exists optimal multipliers $\hat{\lambda}\ge 0$ such that

$$\begin{aligned}&\langle\nabla f(x^{k-1}),s^k-x^{k-1}\rangle\\=&\langle\nabla f(x^{k-1}),s^k-x^{k-1}\rangle+\sum_{i=1}^{m}\hat{\lambda}_i\left(g_i(x^{k-1})+\langle\nabla g_i(x^{k-1}),s^k-x^{k-1}\rangle\right)\\\le&\langle\nabla f(x^{k-1}),x-x^{k-1}\rangle+\sum_{i=1}^{m}\hat{\lambda}_i\left(g_i(x^{k-1})+\langle\nabla g_i(x^{k-1}),x-x^{k-1}\rangle\right).\end{aligned}$$

From the above relation and (4.38) we have

$$\begin{aligned}&\langle\nabla f(x^{k-1}),x^{k-1}-x\rangle+\sum_{i=1}^{m}\hat{\lambda}_i\langle\nabla g_i(x^{k-1}),x^{k-1}-x)\rangle\\\le&\langle\nabla f(x^{k-1}),x^{k-1}-s^k\rangle+\sum_{i=1}^{m}\hat{\lambda}_i g_i(x^{k-1}),\ \forall x\in X\\\le&\frac{\Delta_f D_X^2}{N\min\{\frac{\varepsilon_f}{L_f},\frac{\varepsilon_g}{L_g}\}}+\frac{\varepsilon_f}{2}+\sum_{i=1}^{m}\hat{\lambda}_i\varepsilon_g.\end{aligned}$$

Denote $\lambda_0=(1+\sum_{i=1}^{m}\hat{\lambda}_i)^{-1}$ and $\lambda_i=\hat{\lambda}_0\hat{\lambda}_i$ for $i=1,2,\dots,m$.

With $N\ge 2\max\{\Delta_f L_f D_X^2/\varepsilon_f^2,\Delta_f L_g D_X^2/\varepsilon_f\varepsilon_g\}$, derive from (4.39) that there exists $k\in\{1,2,\dots,N\}$ such that

$$\langle\lambda_0\nabla f(x^{k-1})+\sum_{i=1}^{m}\lambda_i\nabla g_i(x^{k-1}),x^{k-1}-x\rangle\le\lambda_0\left(\varepsilon_f+\sum_{i=1}^{m}\hat{\lambda}_i\varepsilon_f\right)\le\varepsilon_f,\ \forall x\in X.\tag{4.39}$$

thus (4.37) is derived. □

Remark from (4.37) that if the number of iterations N required by Algorithm 4.3.12 satisfies $N \geq \mathcal{O}\left(\max\{\Delta_f L_f D_X^2/\varepsilon_f^2, \Delta_f L_g D_X^2/\varepsilon_f \varepsilon_g\}\right)$, then $z^t \leq \varepsilon_f$, and iterate x^t is an $(\varepsilon_f, \varepsilon_g)$-approximate FJ point of Algorithm 4.2.9 for solving (4.1).

4.4 Conclusion

We now have a brief summary of this chapter. The research goal is to design projection-free algorithms for solving functional constrained optimization problem in this chapter. We first describe the problem of interest and essential definitions of different types of approximate solutions in Section 4.1. Secondly, in Section 4.2, we provide two projection-free methods for solving convex constrained optimization problems. The constrained conditional gradient (CCG) method is proposed such that the gradient evaluation complexity and the linear objective optimization complexity are both of order $\mathcal{O}(1/\varepsilon)$. Then CCG with sliding (CCG-S) method is described with an $\mathcal{O}(1/\sqrt{\varepsilon})$ gradient evaluation complexity and $\mathcal{O}(1/\varepsilon)$ linear objective optimization complexity. Finally, in Section 4.3, we develop projection-free conditional gradient (CG) type algorithms for solving nonconvex constrained optimization problems. We propose CCG with line search (CCG-L) to obtain an approximate solution with in $\mathcal{O}(1/\varepsilon^2)$ iterations no matter the constraint functions are convex or nonconvex.

Bibliography

[1] Digvijay Boob, Qi Deng, and Guanghui Lan. Stochastic first-order methods for convex and nonconvex functional constrained optimization. *Mathematical Programming*, 197(1):215–279, 2023.

[2] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *arXiv preprint arXiv:1710.11606*, 2017.

[3] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: First-order methods. *arXiv preprint arXiv:1711.00841*, 2017.

[4] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020.

[5] Yi Cheng, Guanghui Lan, and H Edwin Romeijn. Functional constrained optimization for risk aversion and sparsity control. *arXiv preprint arXiv:2210.05108*, 2022.

[6] Jelena Diakonikolas and Puqian Wang. Potential function-based framework for minimizing gradients in convex and min-max optimization. *SIAM Journal on Optimization*, 32(3):1668–1697, 2022.

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

[8] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 427–435, 2013.

[9] Yunheng Jiang. Optimal first order methods for reducing gradient norm in unconstrained convex smooth optimization. Master's thesis, Clemson University, South Carolina, USA, 2022.

[10] 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.

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

[12] Guanghui Lan, Yuyuan Ouyang, and Zhe Zhang. Optimal and parameter-free gradient minimization methods for convex and nonconvex optimization. *arXiv e-prints*, pages arXiv–2310, 2023.

[13] Guanghui Lan, Edwin Romeijn, and Zhiqiang Zhou. Conditional gradient methods for convex optimization with general affine and nonlinear constraints. *SIAM Journal on Optimization*, 31(3):2307–2339, 2021.

[14] Guanghui Lan and Yi Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

[15] Guanghui Lan and Yi Zhou. Random gradient extrapolation for distributed and stochastic optimization. *SIAM Journal on Optimization*, 28(4):2753–2782, 2018.

[16] Ji Liu and A Stephen Morse. Accelerated linear iterations for distributed averaging. *Annual Reviews in Control*, 35(2):160–165, 2011.

[17] Wei Liu, Qihang Lin, and Yangyang Xu. First-order methods for affinely constrained composite non-convex non-smooth problems: Lower complexity bound and near-optimal methods. *arXiv preprint arXiv:2307.07605*, 2023.

[18] A. Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley, XV, 1983.

[19] A. S. Nemirovski. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992.

[20] Arkadi Nemirovski. Information-based complexity of convex programming. *Lecture notes*, 834, 1995.

[21] AS Nemirovsky. On optimality of krylov's information when solving linear operator equations. *Journal of Complexity*, 7(2):121–130, 1991.

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

[23] 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.

[24] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[25] 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.

[26] Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, 185(1):1–35, 2021.

[27] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[28] Zhe Zhang and Guanghui Lan. Solving convex smooth function constrained optimization is almost as easy as unconstrained optimization. *arXiv preprint arXiv:2210.05807*, 2022.